Pacific Symposium on Biocomputing 15:31-42(2010)

A PRACTICAL ALGORITHM FOR ESTIMATION OF THE MAXIMUM LIKELIHOOD ANCESTRAL RECONSTRUCTION ERROR

GLENN HICKEY AND MATHIEU BLANCHETTE

McGill Centre for Bioinformatics and School of Computer Science, McGill University, 3480 University St., Montréal, Québec, H3A 2B4, Canada.

The ancestral sequence reconstruction problem asks to predict the DNA or protein sequence of an ancestral species, given the sequences of extant species. Such reconstructions are fundamental to comparative genomics, as they provide information about extant genomes and the process of evolution that gave rise to them. Arguably the best method for ancestral reconstruction is maximum likelihood estimation. Many effective algorithms for accurately computing the most likely ancestral sequence have been proposed. We consider the less-studied problem of computing the expected reconstruction error of a maximum likelihood reconstruction, given the phylogenetic tree and model of evolution, but not the extant sequences. This situation can arise, for example, when deciding which genomes to sequence for a reconstruction project given a gene-tree phylogeny (The Taxon Selection Problem). In most applications, the reconstruction error is necessarily very small, making Monte Carlo simulations very inefficient for accurate estimation. We present the first practical algorithm for this problem and demonstrate how it can be used to quickly and accurately estimate the reconstruction accuracy. We then use our method as a kernel in a heuristic algorithm for the taxon selection problem. The implementation is available at http://www.mcb.mcgill.ca/ blanchem/mlerror

Keywords: Ancestral Reconstruction Error; Maximum Likelihood; Ancestral Genomes; Error estimation; Sequence Evolution

1. Introduction

The rapid increase in DNA sequencing throughput over the last few years has greatly increased the number of species whose genome is completely or partially sequenced. This represents an extraordinary opportunity for comparative genomics and genome evolution studies. The availability of a few dozen mammalian genomes paves the way for efforts toward the computational inference of ancestral genomes, based on those of extant species. Ancestral sequence reconstructions have been undertaken at several levels. Ancestral protein sequences have been inferred and their function tested[1–3] . Blanchette et al.[4] have reconstructed large genomic regions of ancestral mammals and Paten et al.[5] have proposed reconstructions for whole ancestral mammalian genomes. Ancestral sequence reconstructions are key to understanding how genomes evolve and how sequences adapt. They further provide useful information toward the functional annotation of extant genomes, through the identification of evolutionary signatures.[6–8]

Much of the work in the ancestral sequence reconstruction community has focused on designing algorithms to infer ancestral states as accurately as possible. The process starts with the multiple alignment of orthologous sequences[9–12] and a model of sequence evolution, from which a phylogenetic tree is derived. Given this data, one seeks assignments of sequences to ancestral nodes that produce the most parsimonious or most likely evolutionary scenario. If only substitutions are allowed and sites evolve independently from each other, ancestral states can be inferred separately for each site, using the Fitch algorithm for parsimony,[13] Sankoff's algorithm for weighted parsimony,[14] or Felsenstein's algorithm for maximum likelihood,[15] all of which run in time $O(n \cdot L)$, where n is the number of extant species and L is the number of sites. When insertions and deletions are considered, finding the maximum parsimony solution becomes NP-hard[16] because sites cannot be treated independently. However, good heuristics have recently been developed.[7,17] The presence of genome rearrangements and duplications complicate ancestral inference further.[18–21]

While good algorithms or heuristics exist for most versions of the ancestral sequence reconstruction problem, the problem of assessing the accuracy with which the sequence at given ancestral node can be reconstructed has received less attention. Specifically, we are interested in the following problem:

Pacific Symposium on Biocomputing 15:31-42(2010)

ANCESTRAL RECONSTRUCTION EXPECTED ACCURACY PROBLEM
Given:

- A phylogenetic tree $\mathcal{T}$
- A stochastic model of sequence evolution along each branch of $\mathcal{T}$
- An ancestral sequence reconstruction algorithm M

Find: The expected accuracy with which algorithm M can reconstruct the sequence at each ancestral node, where the expectation is taken over all possible realizations (set of sequences at the leaves of $\mathcal{T}$) of the evolutionary process. Note that no sequences are part of the input to the AREA problem. Instead, sequences are random variables generated by the stochastic model of evolution: a random sequence is generated at the root of tree $\mathcal{T}$ and evolves randomly, according to the stochastic model provided for each branch, until leaf sequences are obtained. Let X_i be the random sequence generated at node i. An ancestral sequence reconstruction algorithm M computes a deterministic function of the random sequences generated at the leaves, $X_{l_1}, X_{l_2}, ..., X_{l_n}$ to predict an ancestral sequence $\hat{X}_{root}$ at the root of $\mathcal{T}$. Our goal is to estimate the expected value of the difference $d(X_{root}, \hat{X}_{root})$ between the generated and predicted ancestral sequence, for some appropriate edit-distance measure $d(\cdot)$.

Perhaps the simplest approach to assess the expected reconstruction accuracy of a given algorithm on a given tree is through simulations: Repeatedly generate sequences on T, apply the algorithm to the set of sequences generated at the leaves, and measure the distance between the generated and inferred sequences at the root of T, to eventually obtain an unbiased estimate of the expected error. Blanchette et al.[22] used this approach to show that, given the genomes of 20 well-chosen mammalian species, the genome of the Boreoeutherian ancestor (ancestor of all eutherian mammals except Xenarthrans (e.g. armadillo) and Afrotherians (e.g. elephant)) could be reconstructed with only approximately 1% error. However, simulations are very computationally expensive and provide little understanding of the fundamentals of the problem. A more efficient approach is sought because AREA lends itself to use as a kernel for more general algorithms. For instance, even a very basic heuristic for the taxon selection problem (defined below) can require up to $O(n!)$ reconstruction error estimates to be performed. Another example would be in cases where an aggregate of different site-by-site reconstruction errors across an entire genome must be computed. This can happen in the presence of a varying mutation rate or gaps, and may require millions of instances of AREA to be solved.

In this paper, we seek a more efficient approach to estimate the expected reconstruction error for maximum likelihood inference for substitutions. We start by reviewing related work, giving basic definitions and notation, introducing a straightforward random sampling algorithm. We then describe a faster, heuristic sampling approach and prove that it can provide an upper bound on the error. Finally, results on biological and simulation data are presented in Section 6, demonstrating that our approach is applicable to a wide range of trees. Equipped with efficient algorithm for reconstruction error estimation, in Section 7 we consider the problem of species selection: Given a large phylogenetic tree with a particular internal node N of interest, which subset of k leaves yields the maximum information about ancestor N. The ability of fast and accurate error estimation can thus open the door to a number of applications.

2. Previous work

An exact method for computing the accuracy of ancestral reconstruction using parsimony, given a tree topology and stochastic model of evolution, was presented by Maddison.[23] Although efficient, this dynamic programming algorithm cannot be extended to more general models of parsimony where, for example, different transitions can be associated with different scores. Parsimony also suffers the drawback that, depending on the model of evolution used, the reconstruction accuracy when using parsimony may actually decrease as more taxa are added to the tree.[24] Maximum Likelihood (ML) provides a statistically robust framework for performing ancestral reconstructions under general models of evolution. Yang et al.[25] adapted Felsenstein's pruning algorithm[15] to efficiently estimate the most likely ancestral states of protein sequences with the

Pacific Symposium on Biocomputing 15:31-42(2010)

greatest marginal maximum likelihood. Others have published similar results.[26–28] In most of these studies, the accuracy of the method is indirectly measured by the relative contribution of the reconstructed ancestor to the overall likelihood of the tree, including the observed character states at the leaves. Some work has also been done to determine how topology relates to reconstruction error.[24,29–31] However, there is no known analog for Maddison's algorithm in the ML setting that exactly computes the reconstruction error given only the tree topology and model, and we conjecture that the problem is NP-Hard. Ma and Zhang[32] have recently shown that the problem does admit a fully polynomial time approximation scheme (FPTAS), but the complexity of their algorithm is $O(\frac{n^{17}}{\epsilon^8})$ for a DNA alphabet, making their result primarily of theoretical, rather than practical, use.

3. Definitions

Let $\mathcal{T} = (V, E)$ be a rooted phylogenetic tree with n leaves. The length of edge e is denoted $\ell(e)$. Throughout this paper we use the Jukes Cantor model of evolution[33] on the alphabet $A = \{a_1, a_2, ..., a_{|A|}\}$. Under this model, all ancestral states have equal prior probabilities ($\pi = \frac{1}{|A|}$) and the probability of a mutation occurring on edge e is $p_e = \frac{|A|-1}{|A|}(1 - e^{\ell(e)})$. Let set $D = \{d_1, d_2, ..., d_{|A|^n}\}$ represent the set of all possible assignments of character states to the leaves of $\mathcal{T}$. $\Pr[d_i|\mathcal{T}]$ is the likelihood of the tree for the site given leaf configuration d_i, and can be computed in $O(n \cdot |A|^2)$ time using Felsenstein's dynamic programming algorithm.[15] The marginal maximum likelihood ancestral reconstruction for a site given a leaf configuration $d \in D$ is

$$R(\mathcal{T}, d) = \underset{a \in A}{\operatorname{argmax}} \Pr[d|r = a, \mathcal{T}] \tag{1}$$

where r is the ancestral state at the root of $\mathcal{T}$. Given that the prior probabilities are all equal, we assume throughout that the true ancestral state is a_1, and it follows that the reconstruction error can be expressed as

$$RE(\mathcal{T}) = \sum_{d \in D} \Pr[d|r = a_1] \cdot \begin{cases} 0 & \text{if } \mathrm{R}(\mathcal{T}, d) = \{a_1\} \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Note that in this study, we consider the unambiguous reconstruction error, and a reconstruction is only considered correct of a_1 is the unique solution returned by $R(\mathcal{T}, d)$. It is clear that a naive implementation of Equation 2 would require time $O(|A|^{n+2} \cdot n)$, which is impractical for even moderately large trees. In this paper, we develop heuristics to efficiently estimate this summation, focusing on a small number of terms that contribute most to the total reconstruction error.

4. Monte Carlo Simulation

An estimate of the reconstruction error can be obtained by a simulation, as mentioned above, which runs as follows. The true ancestral state is selected using the prior (equilibrium) distribution. The desired substitution model is then used to simulate random substitutions downwards along the branches until a configuration of states at the leaves is obtained. This configuration of leaf states is then used to predict the ancestral state. If it does so incorrectly, the trial is counted as an error event. Let K be the random variable denoting the total number of errors encountered after N trials. Because the outcomes of each trial are independent, $K \sim \mathrm{Bin}(N, p)$. The reconstruction error can be estimated as $\hat{p} = \frac{K}{N}$ and the normal approximation can be used to estimate the binomial confidence interval:

$$p \in \left[\hat{p} - z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{N}}, \hat{p} + z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \right] \tag{3}$$

where $z_{1-\alpha/2}$ is the critical value of the two two-tailed normal distribution at level α (e.g. $z_{1-\alpha/2} = 1.96$ for 95% confidence). This approach is powerful in the sense that it can be used to accurately estimate the error with very few assumptions about the underlying model. However, as the true reconstruction

Pacific Symposium on Biocomputing 15:31-42(2010)

error becomes very small, it is possible that an intractable number of samples will be required to obtain a reasonable estimate. This is because the size of the confidence interval decreases proportional to $\sqrt{N}$. Furthermore, if $K = 0$ (as expected if $p < \frac{1}{N}$), then the confidence interval is undefined.

5. Prioritized Enumeration Algorithm

In this section, we describe an enumeration approach that will accurately estimate the reconstruction error in much fewer than $\frac{1}{RE(\mathcal{T})}$ trials. This method is based on the observation that a relatively small subset of leaf configurations often account for nearly all of the reconstruction error.

5.1. *Mutation Scenarios*

We analyze leaf configurations in relation to the mutation scenarios that can give rise to them. Define a *mutation scenario* $m \subseteq E$ as a set of edges of $\mathcal{T}$ where mutations occur. The reconstruction error can be rewritten in terms of the error of all $|\mathcal{P}E| = 2^{|E|}$ possible scenarios, where $\mathcal{P}E$ is the power set of E. We then have

$$RE(\mathcal{T}) = \sum_{m \in \mathcal{P}E} \Pr[m] \cdot RE(\mathcal{T}|m), \tag{4}$$

where

$$\Pr[m] = \prod_{e \in m} p_e \prod_{e \in E-m} (1 - p_e),$$

and the reconstruction error for an individual scenario can in principle be computed by analyzing all possible $(|A| - 1)^{|m|}$ leaf state configurations that it can give rise to:

$$RE(\mathcal{T}|m) = \sum_{d \in D} \Pr[d|m] \cdot \begin{cases} 0 & \text{if } R(\mathcal{T}, d) = a_1 \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

$\Pr[d|m]$ is the probability of a particular leaf configuration occurring on the tree given that mutations only occur on the edges of m. If d assigns a mutated state to a leaf that does not lie below m, then $\Pr[d|m] = 0$.

There are an exponential number of possible mutation scenarios, and the cost of computing the reconstruction error of a single scenario is exponential in the size of that scenario. As such, using (4) to exhaustively compute the error is no more efficient than the original definition (2). However, we hypothesize that only a small fraction of scenarios contribute significantly to the reconstruction error. Furthermore, we expect that the RE of these scenarios, $RE(\mathcal{T}|m)$ will tend to be relatively large, and therefore faster to estimate accurately using sampling. Intuitively, these significant scenarios will often contain mutations located closer to the root of the tree, where a smaller number of mutations can affect more leaves, causing a greater loss of signal. The two primary computational challenges of this approach are to rapidly determine the most relevant scenarios, and quickly estimate the reconstruction error of each. These tasks are equivalent to identifying largest terms in Equation (4) and computing the summation in Equation (5), respectively, and are explained in the following two subsections.

5.2. *Prioritization Strategy*

The ideal order in which to explore mutation scenarios is decreasing on $\Pr[m] \cdot RE(\mathcal{T}|m)$. This way, if only a k-subset of all scenarios are evaluated, the accuracy of the estimated error will be maximal. It is unknown how to exactly compute the k scenarios that contribute the most to the reconstruction error, but we use the next two lemmas below in our algorithm to efficiently estimate them. We say that a leaf x_i lies *below* a scenario m, which we denote $x_i \prec m$, if there exists an edge $e \in m$ on the path from x_i to the root of $\mathcal{T}$. We also use a similar notation to compare two scenarios: $m \preceq m'$ if, for all $e \in m$, there exists an edge $e' \in m'$ such that e lies on the path between e' and the root (including the possibility that $e = e'$).

Pacific Symposium on Biocomputing 15:31-42(2010)

Lemma 5.1. *Let m be a mutation scenario where no two mutations are on the same root-to-leaf path, and m' be any scenario obtainable by moving mutations in m away from the root in $\mathcal{T}$, then*

$$RE(\mathcal{T}|m') \leq RE(\mathcal{T}|m).$$

Proof. Let m' be obtained by moving edge $e_v \in m$ to one of its children, e_u. For any leaf configuration d such that $R(\mathcal{T}, d) = a_1$ and $\Pr[d|m] > 0$, there exists a unique configuration d' obtainable from d by assigning all leaves below e_v but not below e_u to state a_1. Furthermore, $\Pr[d'|m'] = \Pr[d|m]$ because $|m| = |m'|$ and no mutations occur on the same root-to-leaf path. From Lemma A.1 (see Appendix), $R(\mathcal{T}, d) = a_1 \rightarrow R(\mathcal{T}, d') = a_1$. $RE(\mathcal{T}|m') \leq RE(\mathcal{T}|m)$ follows directly from Equation (5), and the proof can be completed by induction on the number of edges moved. □

Lemma 5.2. *Let m and m' be two mutation scenarios such that $|m| \leq |m'|$ and let ℓ_{max} be the length of the longest edge in $\mathcal{T}$. Then*

$$\frac{\Pr[m']}{\Pr[m]} < \tau^{|m'|-|m|},$$

where $\tau = \frac{(|A|-1)(1-e^{-\ell_{max}})}{1+e^{-\ell_{max}}}$ and $\tau < 1$ whenever $\ell_{max} < -\ln \frac{|A|-2}{2(|A|-1)}$. (Proof follows directly from the definition of the Jukes Cantor model.)

For example, for $|A| = 4$, $\tau < 1$ when $\ell_{max} < 1.09$, which corresponds to an extremely long branch, never observed in trees that lend themselves to ancestral reconstruction. In the case of the mammalian tree used in Section 6, $\ell_{max} = 0.18$ and $\tau = 0.27$, which means that the probability of a scenario will tend to drop quickly with its size (but note also that the *number* of scenarios grows exponentially with their size).

Our algorithm enumerates mutation scenarios in increasing order of their size. Scenarios of the same size are evaluated in lexicographic order, where edges are ranked based on their depth in the tree (i.e. the edges closest to the root come first). Lemmas 5.1 and 5.2 show that this is generally a good approximation to a lexicographic ordering on $(\Pr[m], RE(\mathcal{T}|m))$, which we find to be an effective estimate of the ranking based on the true objective, $\Pr[m]RE(\mathcal{T}|m)$. Figure 1 provides an example of the enumeration procedure. The edges of the phylogenetic tree are first ranked (Figure 1(a)), then the scenarios are visited in the order of a breadth-first search of the tree shown in Figure 1 (b). It is important to note that this tree traversal visits each scenario once, and that all scenarios below m in the search tree are supersets of m. Also, as shown in the following section, mutation scenarios with multiple mutations on a single root-to-leaf path are never explicitly evaluated. Therefore the fact that Lemma 5.2 does not apply in these cases does not impact its applicability here.

5.3. *Scenario Evaluation and Bounding the Search*

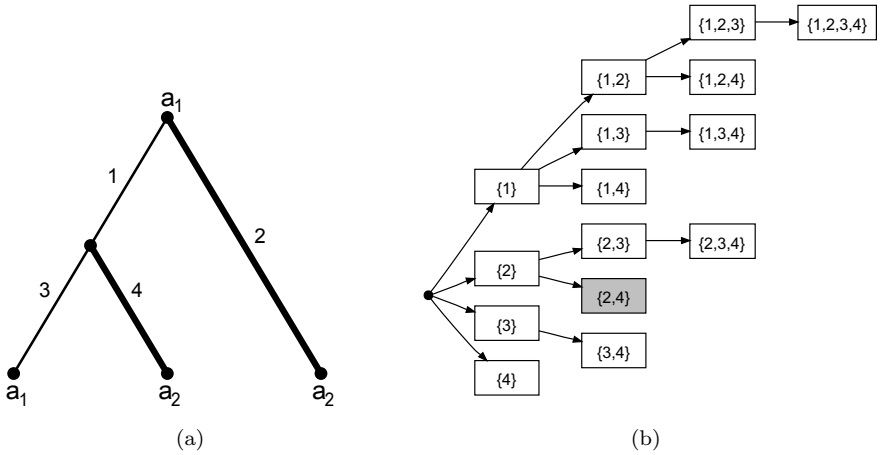
The algorithm must be able to quickly estimate $RE(\mathcal{T}|m)$ to effectively explore the search space. The cost of exactly computing this value, as described in Equation (5), is $O((|A|-1)^{|m|} \cdot n \cdot |A|^2)$, which is impractical for all but the smallest scenarios. We therefore propose a procedure to efficiently estimate $RE(\mathcal{T}|m)$ by considering many scenarios at once. For a given scenario m, consider the following "pessimistic" leaf configuration $pess(m) = x_1, x_2, ..., x_n$, where

$$x_i = \begin{cases} a_1 \text{ if } m \text{ contains no edge on the path from } i \text{ to the root} \\ a_2 \text{ if } m \text{ contains at least one edge the path from } i \text{ to the root} \end{cases}$$

The leaf configuration $pess(m)$ is pessimistic because all mutations lead to the same character a_2, which causes the maximal probability of reconstruction error. Note that it is possible that $\Pr[pess(m)|m] = 0$ (when there are two mutations in the same lineage). $RE^{pess}(\mathcal{T}|m)$ is used to denote the pessimistic estimate of $RE(\mathcal{T}|m)$ and is computed in $O(n \cdot |A|^2)$ as follows:

$$RE^{pess}(\mathcal{T}|m) = \begin{cases} 0 & \text{if } R(\mathcal{T}, pess(m)) = a_1 \\ 1 & \text{otherwise.} \end{cases}$$

Pacific Symposium on Biocomputing 15:31-42(2010)



(a) (b)

Fig. 1. (a) Phylogenetic tree where a mutation scenario $\{2, 4\}$ is highlighted. The ancestral state (a_1) mutates to a_2 along branches 2 and 4. (b) Corresponding search tree, traversed breadth first by the heuristic sampling algorithm. Scenario $\{2, 4\}$ is explored on the second level.

We now present a fast algorithm for estimating $RE^{pess}(\mathcal{T}|m)$, before relaxing it produce a more accurate estimator. Let $ext(m) = \{m' : m \subseteq m'\}$ be the set of scenarios that extend m and let $below(m) = \{m' : m' \preceq m\}$ be the set of scenarios below m. The following two properties, which directly follow from Lemma A.1, enable a branch and bound approach to further speed up the search.

Lemma 5.3. *If* $RE^{pess}(\mathcal{T}|m) = 1$*, then* $RE^{pess}(\mathcal{T}|m') = 1, \forall m' \in ext(m)$

Lemma 5.4. *If* $RE^{pess}(\mathcal{T}|m) = 0$*, then* $RE^{pess}(\mathcal{T}|m') = 0, \forall m' \in below(m)$

Lemma 5.3 allows us to prune the search tree at any scenario that does not yield a correct reconstruction under the pessimistic evaluation, since all scenarios below it in the search tree are supersets and will necessarily fail the pessimistic evaluation. In the example search shown in Figure 1(b), if $RE(\{2\}) = 1$, then the three nodes below $\{2\}$ in the search tree will also have $RE^{pess} = 1$ and need not be visited. The total probability of $ext(m)$ can be computed in linear time:

$$\Pr[m \cap ext(m)] = \prod_{e \in m} p_e \prod_{e \notin (m \cup ext(m))} (1 - p_e).$$

$\Pr[m \cap below(m)]$ can be computed in a similar manner. Lemma 5.4 allows for a similar optimization, except only extensions consisting of edges below m in the tree can be pruned. In the example, if $RE(\mathcal{T}|\{1\}) = 1$ then the search tree can be pruned at $\{1, 3\}$ and $\{1, 4\}$ but not $\{1, 2\}$.

Let $succ(m)$ be the set of all scenarios below m on the *search* tree. Pruning the search tree as described above does not take full advantage of Lemmas 5.3 and 5.4, because $ext(m)$ and $below(m)$ may contain scenarios that are not in $succ(m)$. These sets grow exponentially with $|m|$, however, so storing in memory all scenarios that have and have not been visited quickly becomes infeasible. Instead, a small cache is used to store the outcome of the most recently visited scenarios in the breadth-first search. As each new scenario is visited, the cache is checked before the scenario is evaluated. If it extends or lies below an appropriate scenario in the cache, its $RE(m|\mathcal{T})$ can be retrieved without an explicit evaluation. Let $RE^{pess}(\mathcal{T}) = \sum_{m \in \mathcal{P}_E} \Pr[m] \cdot RE^{pess}(\mathcal{T}|m)$. Lemmas 5.3 and 5.4 guarantee that when using the pessimistic evaluation of scenarios as described, the algorithm will exactly compute $RE^{pess}(\mathcal{T})$. Furthermore, his value is an upper bound for $RE(\mathcal{T})$ (Theorem A.1, see Appendix).

Assuming all mutations mutate to the same state is simplistic, as it becomes more likely that independent mutations give rise to two or more different states as the size of m increases. This situation is accounted for by estimating $RE(\mathcal{T}|m)$ using a small number of random samplings, whereby scenario m is used to generate a set

Pacific Symposium on Biocomputing 15:31-42(2010)

of r random leaf configurations. The error is estimated as $\frac{k}{r}$ where k is the number of failed reconstructions out of r random trials. Finally, Lemmas 5.3 and 5.4 are relaxed to apply when $RE(\mathcal{T}|m) \geq 1-\alpha$ and $RE(\mathcal{T}|m) < 1-\alpha$ respectively, where α is a chosen constant. Pseudocode for both the pessimist and relaxed sampling versions is provided in Algorithm 5.1.

Algorithm 5.1 Estimate $RE(\mathcal{T})$

```
TotalError ← 0;
ScenarioQueue q ← {};
while |q| > 0 do
  m ← q.pop()
  error ← { RE^pess(T|m)    if pessimist mode
          { RE^sample(T|m)  if sample mode
  if error ≥ 1 − α then
    TotalError ← TotalError + error · Pr[m ∪ succ(m)]
  else
    TotalError ← TotalError + error · Pr[m ∪ succ(m) ∩ below(m)]
    for e ∈ succ(m) − below(m) do
      q.push(m ∪ e)
return TotalError
```

6. Results

The prioritized enumeration algorithm was implemented in C++. Since we are unable to exactly compute the true error for non-trivial trees, the random Monte Carlo sampling approach was also implemented to use as a baseline for comparison. We carried out various experiments to determine the accuracy of the upper bound and estimates proposed here.

We first estimated the reconstruction error for the Boreoeutherian ancestor based on an actual mammalian phylogeny made of 32 species.[34,35] Figure 2 shows the average estimates obtained from 100 separate Monte Carlo simulations, as a function of the number of trials, N. This yields an unbiased estimator of the true reconstruction error, together with a mean confidence interval for that value. The mean estimates obtained from the prioritized enumeration, either using the pessimistic upper bound or the random sampling version [a] are plotted as well. The standard deviation of the estimates obtained with the sampling-based prioritize enumeration approach is too small to plot ($\sigma = 0.00043$). We observe that, as expected, the pessimistic version of the algorithm overestimates the reconstruction error. However, the sampling approach quickly converges to the correct value and, moreover, it does so with much less variance than the Monte Carlo simulations. In both cases, a fairly accurate estimate of the RE is obtained by our algorithm after fewer than 50 scenarios.

We then estimated the accuracy of our reconstruction errors on random trees. A set of 50 trees of size n and expected total number of substitutions μ were randomly generated using the Yule model of speciation.[36] Branch lengths were assigned using a uniform random distribution and scaled to total μ. For each tree, the 99% confidence interval $I_{99\%}$ for the RE was first estimated based on 5000 Monte Carlo simulations. Reconstruction errors were then estimated, evaluating a number k of leaf configuration ranging from 5 to 5000. Our first set of random trees have 50 leaves, with $\mu = 1$, and with an average 99% reconstruction error confidence interval of 0.007 ± 0.002. Figure 3(a) shows, for each value of k, the fraction of the trees for which the estimate obtained after k configurations lies within $I_{99\%}$. The second set of trees (Figure 3(b))

[a] $r = 2$ and relaxation parameter $\alpha = 0$ consistently gave the best performance and were used for all experiments.

Pacific Symposium on Biocomputing 15:31-42(2010)

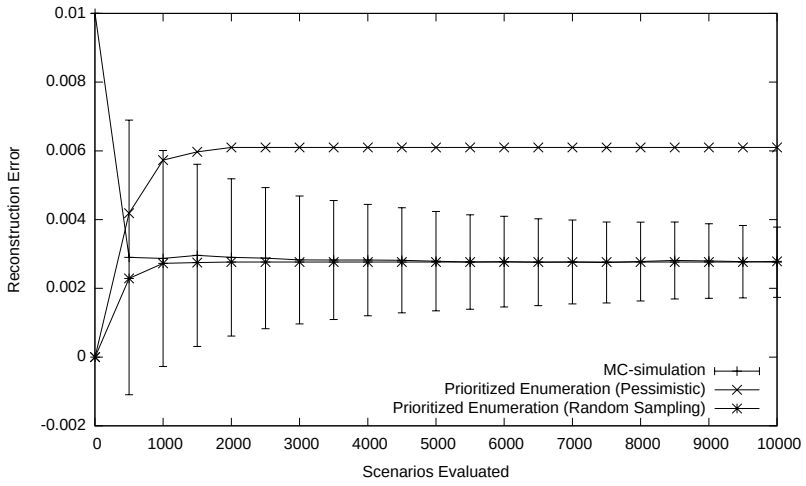


Fig. 2. Reconstruction Error of the Boreoeutheirian common ancestor. 99% confidence interval is shown for the Monte Carlo simulation.

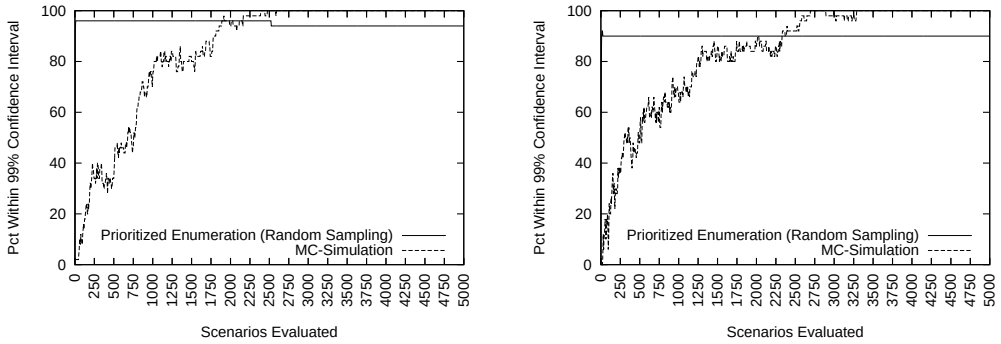
was generated with $n = 100, \mu = 5$ and had an average 99% confidence interval for the reconstruction error of 0.020 ± 0.004. In terms of running time, the cost of evaluating each scenario is effectively identical to that of a single simulation trial. The graphs therefore show that the heuristic algorithm was able to accurately estimate 98% and 82% of the trees in the batches, respectively, an order of magnitude faster than running the full simulation. The same procedure was also applied to trees randomly selected from PANDIT 17.0,[37] a database of protein phylogenies. Two groups of 50 trees were created: the first with between 25 and 50 taxa (mean 99% C.I. $= 0.007 \pm 0.002$; Figure 3(c)) and the second with between 50 and 100 taxa (mean 99% C.I. $= 0.003 \pm 0.001$, Figure 3(d)). Clearly, the smaller the RE to be estimated, the more accurate and faster our approach is, as fewer scenarios need to be evaluated. The algorithm performed worse overall on the PANDIT trees, indicating that balance and branch-length distribution are factors as well.

Figure 4 reports the estimated reconstruction error obtained at each ancestral node of the mammalian phylogeny. Of course, these numbers do not reflect the accuracy of actual reconstructed ancestral sequences (as reported in Blanchette et al.[22]), because they only model substitutions and ignore alignment issues. Nonetheless, the estimated REs are informative about the ancestral nodes that can best be reconstructed. We first note that reconstruction errors vary significantly across the tree (from 0.11% to 3.5%). Excluding recent primate ancestors, the nodes that can best be reconstructed correspond to early ancestors that lived during the mammalian radiation. Indeed, those are the ones for which the largest number of nearly independent noisy copies exist. Interestingly, the RE of the ancestral primate is nearly two times higher than that of the Boreoeutherian ancestor, which predates it by approximately 25 Million years. However, all human ancestors, except the eutherian mammals ancestor, can be reconstructed with less than 0.7% error. On the other hand, nodes adjacent to three long branches (mouse-rat ancestor, rabbit-pika ancestor, hedgehog-shrew ancestor) have much higher reconstruction errors.

7. The Taxon Selection Problem

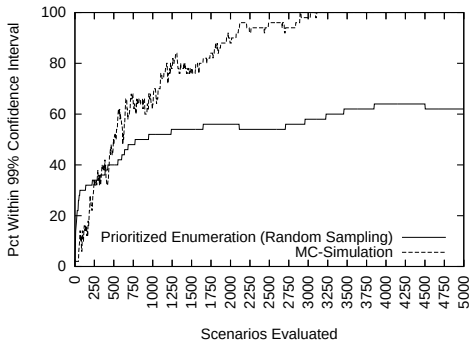
A natural application for an efficient algorithm to compute RE is the taxon selection for the ancestral reconstruction problem, introduced by Li et al.[39] This problem asks to select among the leaves of a large phylogenetic tree, a set of k taxa that will allow reconstructing a given ancestral node with minimum error, provided the tree and a model of evolution. Using the Maddison algorithm as a kernel, Li et al. showed

Pacific Symposium on Biocomputing 15:31-42(2010)

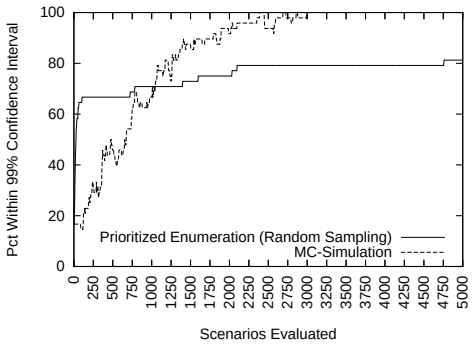


(a) Random Simulation ($n = 50$, $\mu = 1$)

(b) Random Simulation ($n = 100$, $\mu = 5$)



(c) PANDIT ($25 \leq n < 50$)



(d) PANDIT ($50 \leq n < 100$)

Fig. 3. Fraction of errors estimated within the 99% confidence interval by number of mutation scenarios evaluated for batches of simulated and real trees.

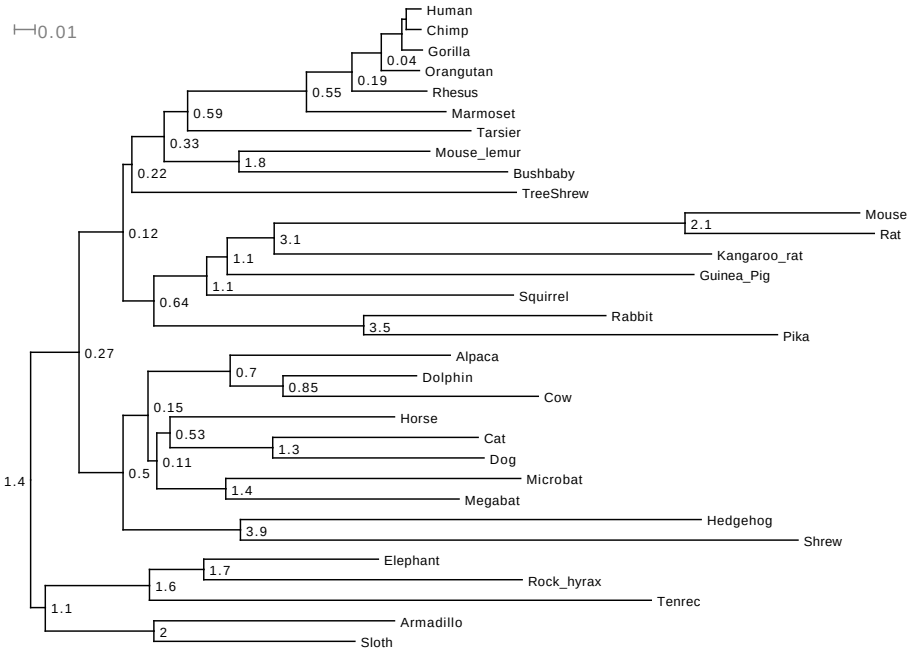


Fig. 4. Estimated reconstruction error (in percent) for all internal nodes of the eutherian mammals phylogeny (Human/Chimp and Human/Chimp/Gorilla ancestors both have RE=0.013%).

Pacific Symposium on Biocomputing 15:31-42(2010)

that a backward selection algorithm is an effective heuristic for genome selection under parsimony.[39] The algorithm starts from the complete set of species and greedily prunes the least valuable leaves from the tree until k remain.

We have investigated a similar algorithm to select an ideal set of species for ML-based ancestral reconstruction. The algorithm is similar to Li et al.'s. The only major difference is that when multiple species candidates for removal yield RE that cannot be distinguished based on our estimation algorithms, the species that is the furthest away from the target ancestral node is the one selected for pruning. Figure 5 reports the order in which species are removed, and the resulting RE for the Boreoeutherian ancestor. We note that a RE of less than 1% can be achieved by selecting only seven (armadillo through orangutan) slow-evolving species that sample the outgroup (Xenarthra, Afrotheria), and the main descendant phyla (primates, laurasia, etc.). This algorithm evaluates $O(n!)$ different topologies requiring an efficient reconstruction error estimate. The reduced variance of our approach, relative to MC simulation, is also desirable for this application: In a comparable amount of time, it will produce a much more stable solution.

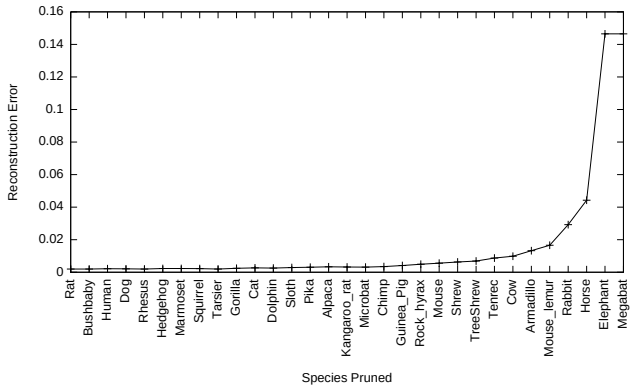


Fig. 5. Greedy species selection algorithm. The order in which species are pruned is presented along with the reconstruction error at each step. The final species is Sloth

8. Conclusion

We have presented a novel algorithm for estimating the error of maximum likelihood ancestral reconstruction. It is based upon quickly enumerating and evaluating the mutation scenarios that contribute the most to $RE(\mathcal{T})$ using a branch-and-bound search strategy. The assumption of a pessimistic leaf configuration is used to guarantee a non-trivial upper bound on the error. This assumption is then relaxed by performing a restricted sampling for each scenario, in order to more closely approximate the true error. We used our method to compute the RE of the Boreoeutherian mammal ancestor, the subject of several important ancestral reconstruction studies, and it quickly converged on an extremely accurate estimate. Benchmarks run on batches of simulated trees and those drawn at random from an online database showed that estimated reconstruction errors were within the 99% confidence interval of the unbiased estimator in the majority of cases. The best performance was seen in cases where the error is very small, a situation that we foresee in most real-life reconstruction applications.

The high speed and low variance of the prioritized enumeration algorithm make it a useful building block for solutions to complex problems. In this study, it was used to quickly determine all the errors of all internal nodes of the mammalian tree, and as an objective function in a greedy heuristic solution to the taxon selection problem. We envision more applications, such as large genomes that require multi-site analysis due

Pacific Symposium on Biocomputing 15:31-42(2010)

to hypothesized gaps or changes in substitution rate parameters, that could require efficient estimation of small errors. One avenue of future work that could lead to an increase of applicability of the algorithm is by exploring more complex models of DNA substitution. While not all properties stated in this paper hold for more general models, we conjecture that the overall strategy will be effective for any model where the probability of mutation is relatively low. Another area in which our approach can potentially be optimized is in the scenario enumeration. The current approach has a sound basis, but it may be possible to more directly sample the relevant mutation scenarios, using more information about the topology and branch lengths.

Acknowledgements

M.B. and G.H. are funded by NSERC. We thank Leonid Chindelevitch for initial discussions on this problem.

Appendix A. Upper Bound

Lemma A.1. *If $R(\mathcal{T}, d) = a_s$ then $R(\mathcal{T}, d') = a_s$ where d' is obtained from d by reassigning to a_s the state of any subset of leaves.*

Proof. We begin by showing that $\Pr[v = a_s|d'] \geq \Pr[v = a_s|d]$ for any node v on the tree. If v is a leaf, $\Pr[v = a_s|d'] = 1$ and $\Pr[v = a_s|d] = 0$ if v is reassigned and $\Pr[v = a_s|d'] = \Pr[v = a_s|d]$ otherwise. If v is an internal node, we make the inductive hypothesis that the inequality holds for v's children. That is, $\Pr[u = a_s|d'] = \Pr[v = a_s|d] + \delta_u$ where $\delta_u \geq 0$. To simplify the notation, we use p and q to respectively denote $\Pr[u = a_s|v = a_s]$ and $\Pr[u = a_s|v = a_t]$ for $a_t \neq a_s$. $\Pr[v = a_s|d]$ can be computed from its children as follows:[15]

$$\Pr[v = a_s|d] = \prod_{u \in \text{child}(v)} \left(p \cdot \Pr[u = a_s|d] + q \cdot \sum_{i \neq s} \Pr[u = a_i|d] \right)$$

By the inductive hypothesis,

$$\begin{aligned}
\Pr[v = a_s|d'] &= \prod_{u \in \text{child}(v)} \left(p \cdot (\Pr[u = a_s|d] + \delta_u) + q \cdot \left(\left(\sum_{i \neq s} \Pr[u = a_i|d] \right) - \delta_u \right) \right) \\
&= \prod_{u \in \text{child}(v)} \left(p \cdot \Pr[u = a_s|d'] + q \cdot \sum_{i \neq s} \Pr[u = a_i|d'] + \delta_u(p - q) \right) \\
&\geq \Pr[v = a_s|d]
\end{aligned}$$

since, under the Jukes Cantor model, $p \geq q$. The same procedure can be used to show that that $\Pr[v = a_t|d'] <= \Pr[v = a_t|d]$ for $t \neq s$. As before, the leaf case is trivial. If v is an internal node,

$$\Pr[v = a_t|d] = \prod_{u \in \text{child}(v)} \left(p \cdot \Pr[u = a_t|d] + q \cdot \sum_{i \neq t} \Pr[u = a_i|d] \right).$$

From the induction hypothesis that $\Pr[u = a_t|d'] = \Pr[u = a_t|d]\delta_u$ for $u \in child(v)$,

$$\begin{aligned}
&= \prod_{u \in \text{child}(v)} \left(p \cdot (\Pr[u = a_t|d] - \delta_u) + q \cdot \left(\left(\sum_{i \neq t} \Pr[u = a_i|d] \right) + \delta_u \right) \right) \\
&= \prod_{u \in \text{child}(v)} \left(p \cdot \Pr[u = a_t|d'] + q \cdot \sum_{i \neq t} \Pr[u = a_i|d'] + \delta_u(q - p) \right) \leq \Pr[v = a_s|d].
\end{aligned}$$

Since $\Pr[v = a_s|d'] \geq \Pr[v = a_s|d]$ and $\Pr[v = a_t|d'] \leq \Pr[v = a_t|d]$ for any $t \neq s$, then $\text{argmax}_{i \in A} \Pr[v = a_i|d] = a_s \rightarrow \text{argmax}_{i \in A} \Pr[v = a_i|d'] = a_s$. □

Pacific Symposium on Biocomputing 15:31-42(2010)

Theorem A.1. *Let $RE^{pess}(\mathcal{T})$ be the reconstruction error obtained by pessimistic scenario evaluation. Then $RE(\mathcal{T}) \leq RE^{pess}(\mathcal{T})$.*

Proof. It is sufficient to show that for any mutation scenario m, $RE(\mathcal{T}|m) \leq RE^{pess}(\mathcal{T}|m)$. The inequality trivially holds when $RE(\mathcal{T}|m) = 0$, so we need only consider the case where m is a scenario such that $RE(\mathcal{T}|m) > 0$. By definition, there exists a leaf configuration $d = \{d_1, d_2, ..., d_n\}$ such that $R(\mathcal{T}|d) = a_s \neq a_1$. Now consider the scenario $pess(m)$, where every leaf below a mutation has state a_s and all other leaves have the true ancestral state a_1. Both d and $pess(m)$ only contain non-a_1 states at leaves which are below mutations. It follows that $pess(m)$ can be constructed from d by changing the states of a subset of leaves to a_s. From Lemma A.1, $R(\mathcal{T}, pess(m)) = a_s$, giving $RE^{pess}(\mathcal{T}|m) = 1$, which implies that $RE(\mathcal{T}|m) \leq RE^{pess}(\mathcal{T}|m)$. □

References

1. B. Chang, K. Jonsson, M. Kazmi, M. Donoghue and T. Sakmar, *Molecular biology and evolution* **19**, 1483 (2002).
2. D. Kuang, Y. Yao, D. MacLean, M. Wang, D. Hampson and B. Chang, *Proceedings of the National Academy of Sciences* **103**, p. 14050 (2006).
3. J. Thornton, E. Need and D. Crews, *Science* **301**, 1714 (2003).
4. M. Blanchette, E. Green, W. Miller and D. Haussler, *Genome Res.* **14**, 2412 (2004).
5. B. Paten, J. Herrero, S. Fitzgerald, K. Beal, P. Flicek, I. Holmes and E. Birney, *Genome Res.* **18**, p. 1829 (2008).
6. J. Taylor, S. Tyekucheva, D. King, R. Hardison, W. Miller and F. Chiaromonte, *Genome Res.* **16**, p. 1596 (2006).
7. J. Kim and S. Sinha, *Bioinformatics* **23**, p. 289 (2007).
8. J. Kim, X. He and S. Sinha, *PLoS Genetics* **5** (2009).
9. M. Blanchette, W. Kent, C. Riemer, L. Elnitski, A. Smit, K. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E. Green *et al.*, *Genome research* **14**, 708 (2004).
10. B. Paten, J. Herrero, K. Beal, S. Fitzgerald and E. Birney, *Genome Research* **18**, p. 1814 (2008).
11. M. Brudno, C. Do, G. Cooper, M. Kim, E. Davydov, N. Program, E. Green, A. Sidow and S. Batzoglou, *Genome research* **13**, 721 (2003).
12. N. Bray and L. Pachter, *Genome Research* **14**, 693 (2004).
13. W. Fitch, *Systematic zoology* **20**, 406 (1971).
14. D. Sankoff, *SIAM Journal on Applied Mathematics* , 35 (1975).
15. J. Felsenstein, *Journal of molecular evolution* **17**, 368 (1981).
16. L. Chindelevitch, Z. Li, E. Blais and M. Blanchette, *J. of Bioinformatics and Comp. Biol.* **4**, 721 (2006).
17. A. Diallo, V. Makarenkov and M. Blanchette, *Journal of Computational Biology* **14**, 446 (2007).
18. J. Ma, L. Zhang, B. Suh, B. Raney, R. Burhans, W. Kent, M. Blanchette, D. Haussler and W. Miller, *Genome Res.* **16**, p. 1557 (2006).
19. G. Bourque, P. Pevzner and G. Tesler, *Genome Res.* **14**, 507 (2004).
20. D. Sankoff and M. Blanchette, *Journal of Computational Biology* **5**, 555 (1998).
21. J. Ma, A. Ratan, B. Raney, B. Suh, W. Miller and D. Haussler, *PNAS* **105**, p. 14254 (2008).
22. M. Blanchette, E. Green, W. Miller and D. Haussler, *Genome Res.* **14**, 2412 (2004).
23. W. Maddison, *Systematic Biology* **44**, p. 474 (1995).
24. G. Li, M. Steel and L. Zhang, *Systematic Biology* **57**, 647 (2008).
25. Z. Yang, S. Kumar and M. Nei, *Genetics* **141**, 1641 (1995).
26. J. Koshi and R. Goldstein, *Journal of molecular evolution* **42**, 313 (1996).
27. M. Pagel, *Syst. Biol* **48**, 612 (1999).
28. D. Schluter, T. Price, A. Mooers and D. Ludwig, *Evolution* , 1699 (1997).
29. B. Lucena and D. Haussler, *Systematic biology* **54**, 693 (2005).
30. A. Mooers, *Systematic biology* **53**, 809 (2004).
31. J. Zhang and M. Nei, *Journal of molecular evolution* **44**, p. 139 (1997).
32. B. Ma and L. Zhang, *Journal of Combinatorial Optimization, IN PRESS* (2009).
33. T. Jukes and C. Cantor, *Mammalian protein metabolism* **3**, 21 (1969).
34. W. Murphy, E. Eizirik, S. O'Brien, O. Madsen *et al.*, *Science* **294**, 2348 (2001).
35. D. Karolchik, R. Kuhn, R. Baertsch, G. Barber *et al.*, *Nucleic Acids Research* **36**, p. D773 (2008).
36. G. Yule, *Phil. Trans. Royal Soc. of London. Series B, Containing Papers of a Biological Character* , 21 (1925).
37. S. Whelan, P. de Bakker and N. Goldman, *Bioinformatics* **19**, 1556 (2003).
38. D. Huson, D. Richter, C. Rausch, T. Dezulian, M. Franz and R. Rupp, *BMC bioinformatics* **8**, p. 460 (2007).
39. G. Li, J. Ma and L. Zhang, p. 110 (2007).