

Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution

Lars Arvestad
Stockholm Bioinformatics
Center and Dept. of Numerical
Analysis and Computing
Science
Royal Institute of Technology
Stockholm, Sweden
arve@nada.kth.se

Ann-Charlotte Berglund
Stockholm Bioinformatics
Center and Dept. of
Biochemistry and Biophysics
Stockholm University
Stockholm, Sweden
lottab@sbc.su.se

Jens Lagergren
Stockholm Bioinformatics
Center and Dept. of Numerical
Analysis and Computing
Science
Royal Institute of Technology
Stockholm, Sweden
jensl@nada.kth.se

Bengt Sennblad
Stockholm Bioinformatics
Center and Center for
Genomics and Bioinformatics
Karolinska Institutet
Stockholm, Sweden
bens@sbc.su.se

ABSTRACT

Gene tree and species tree reconstruction, orthology analysis and reconciliation, are problems important in multigenome-based comparative genomics and biology in general. In the present paper, we advance the frontier of these areas in several respects and provide important computational tools. First, exact algorithms are given for several probabilistic reconciliation problems with respect to the probabilistic gene evolution model, previously developed by the authors. Until now, those problems were solved by MCMC estimation algorithms. Second, we extend the gene evolution model to the *gene sequence evolution* model, by including sequence evolution. Third, we develop MCMC algorithms for the gene sequence evolution model that, given gene sequence data allows: (1) orthology analysis, reconciliation analysis, and gene tree reconstruction, w.r.t. a species tree, that balances a likely/unlikely reconciliation and a likely/unlikely gene tree and (2) species tree reconstruction that balance a likely/unlikely reconciliation and a likely/unlikely gene trees. These MCMC algorithms take advantage of the exact algorithms for the gene evolution model. We have successfully tested our dynamical programming algorithms on real data for a biogeography problem. The MCMC algorithms

perform very well both on synthetic and biological data.

Categories and Subject Descriptors

G.3 [Probability and Statistics]: Probabilistic algorithms (including Monte Carlo); J.3 [Life and Medical Sciences]: Biology and genetics

General Terms

Algorithms

Keywords

Algorithms, bayesian analysis, gene tree, orthology, reconciliation

1. INTRODUCTION

Phylogenetic analysis, with applications to orthology analysis, is of fundamental importance to comparative genomics. Gene tree reconstruction and reconciliation of gene and species trees are powerful tools for the translation of genomic information between organisms and the study of gene families. Species tree reconstruction is ubiquitous in biology. We give algorithms and develop computational tools for central problems in these areas.

Previously, orthology analysis has been performed in a two-step process, by first constructing a gene tree and then reconciling it with the appropriate species tree. Consequently, in the extreme cases, the reconciliation have included far too many duplications, or in other ways contradicted known biological facts, and thereby revealed the incorrectness of the gene tree. Although, recent advanced phylogeny programs [7] can propose alternative gene trees, there

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RECOMB'04, March 27–31, 2004, San Diego, California, USA.
Copyright 2004 ACM 1-58113-755-9/04/0003 ...\$5.00.

is a more fundamental problem: a reconciliation describes how a gene tree has evolved w.r.t. a species tree and any reconciliation implies constraints on the times of the edges in the gene tree and hence also on the sequence evolution. These constraints may directly contradict the times used in the reconstruction of the gene tree. There is also a trade-off between a likely/unlikely gene tree and a likely/unlikely reconciliation and this trade-off should be handled in a probabilistically sound way. The natural conclusion is that the reconstruction of the gene tree and the reconciliation should be integrated. In fact, such an integration gives a gene tree reconstruction procedure, which takes a known species tree into account. Today, the species tree is completely disregarded in gene tree reconstruction. Of course, sometimes gene trees are constructed precisely for the purpose of estimating the corresponding species tree. However, in the presence of duplications it is necessary to use gene trees from several gene families. The integration then gives a biologically sound supertree construction method that in the presence of duplications yields a species tree given gene trees.

In [2], a gene evolution model was introduced. In the model, a gene tree evolves “inside” a given species tree; this does not only produce a gene tree but it also induces a reconciliation of the gene tree and the species tree. We now extend this to the *gene sequence evolution* model by including sequence evolution. That is, from the creation of a new gene lineage, by a duplication or a speciation, a sequence evolves according to some standard sequence evolution model, see e.g. [3], until the lineage is split into two, again by a duplication or a speciation, which causes the sequence to split into two initially identical but henceforth independently evolving sequences. The gene sequence evolution model is hierarchical, in the sense that, evolving the gene tree with duplication times on the internal vertices using the gene evolution model and then letting a standard sequence evolution model act on the branches of the gene tree gives the same result.

In [2], MCMC algorithms were described that for a given gene tree and a species tree S , estimates the posterior distribution of reconciliations, $Pr[\gamma | G]$, and estimates the posterior probability of two genes, a and b , being orthologs, $Pr[a \text{ and } b \text{ orthologs} | G]$. Here and in the rest of the introduction we omit, for simplicity, the parameters of the birth-death process. In each sampling step of the MCMC algorithm, a likelihood computation is performed for a gene tree G and a reconciliation γ , i.e. $Pr[G, \gamma]$.

In the present paper, the main algorithmic contribution is the formulation of dynamic programming algorithms for computing the sum of the likelihoods of all reconciliations, i.e. $\sum_{\gamma} Pr[G, \gamma]$, and for computing the maximum likelihood over all reconciliations, i.e. $\max_{\gamma} Pr[G, \gamma]$. Moreover, a procedure is given that samples from the posterior distribution of reconciliations; that is, we show how to sample from the distribution $Pr[\gamma | G]$, whilst the MCMC algorithm of [2] merely facilitates sampling from an estimation of this distribution. The fundamental problem in formulating such algorithms is that the number of reconciliations between a species tree and a gene tree, as well as the number of subproblems from any obvious decomposition based on the reconciliation itself (i.e., the sliced subtrees of [2]), is exponential. The algorithms developed in this paper use mathematically non-trivial techniques for managing the subproblems, thus providing important improvements to the efficacy of analysis.

By taking advantage of these exact algorithms, we are able

to give MCMC algorithms that estimate posterior distributions w.r.t. the gene sequence evolution model. We give an algorithm that estimates the posterior distribution on gene trees, $Pr[G | F]$, where F is a set of gene sequence data, as well as an algorithm that estimates the posterior probability that a pair of genes, a and b , are orthologs given a set of sequence data. Finally, we show how this approach can be taken one step further so that the posterior distribution for species trees, w.r.t. a number of observed gene families, are obtained.

We have successfully tested our dynamical programming algorithms on real data for a biogeography problem. Likewise, the MCMC algorithms were successfully tested on synthetic and biological data.

The rest of the paper is organized as follows. First, some (mostly standard) definitions are introduced, followed by a review of our probabilistic gene evolution model and extensions to the gene sequence evolution model. The general MCMC framework is then described briefly together with a description of how it is applied in the present case. Then follows a formal definition of reconciliations. This forms the base for the ensuing description of exact algorithms for computing the reconciliation that maximizes the likelihood, the sum of likelihoods over all reconciliation and sampling reconciliations from the posterior distribution. Finally, experimental results are presented.

2. DEFINITIONS AND NOTATION

A *directed tree* T consists of a set of *vertices* $V(T)$ and a set of *arcs* $A(T)$. The set of *leaves* of T is denoted $L(T)$. The *subforest* of a directed tree T , induced by a subset U of $V(T)$ is the forest $T \setminus (V(T) \setminus U)$. In a *rooted tree* T all arcs are directed away from the root and the root is denoted $r(T)$. Such a tree is *binary* if each non-leaf has out degree two. For a directed rooted tree T and $u \in V(T)$, the subtree rooted at u , denoted T_u , is the subtree of T induced by all vertices reachable by directed paths from u in T . For a binary vertex u in a tree T , $c_1(u)$ and $c_2(u)$ denotes different children of u in T , when T is clear from the context. Moreover, the *arc subtree* of T for $\langle u, v \rangle \in A(T)$, denoted $T^{u,v}$, is $T_v \cup \{\langle u, v \rangle\}$. A *species tree* S is a rooted directed arc-weighted binary tree S with weight function $w_S : A(S) \rightarrow R^+$. A *gene tree* is a rooted directed binary tree given together with a leaf labeling function $\sigma : L(G) \rightarrow L(S)$, where S is the species tree associated with G . Intuitively, leaves in a gene tree G represent genes, leaves in a species tree S represent species, and the gene $l \in L(G)$ belongs to the genome of the species $\sigma(l)$.

3. PROBABILISTIC MODELS

In this section, we review the probabilistic gene evolution model, which was introduced and more thoroughly described in [2]. Thereafter, we extend it to the more general gene sequence evolution model. A reconciliation describes how a gene tree has evolved w.r.t. a species tree; this concept will be formally defined. In the gene evolution model, a gene tree evolves “inside” a given species tree S ; this does not only yield a gene tree but also induces a reconciliation.

3.1 Gene evolution model

The *gene evolution model* describes how a gene tree G evolves over time by speciation, duplication and loss events. Speciations and duplications introduce vertices in G , while

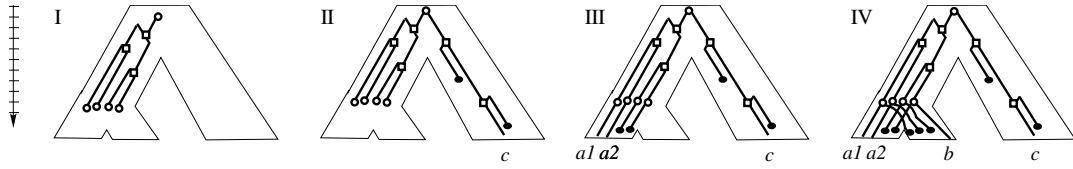


Figure 1: Example of how a gene tree evolves inside a species tree.

losses removes edges and vertices from the gene tree. In this paper we consider, except for Section 5.3, S as fixed, thus constraining the evolution of G . The gene tree can be viewed as evolving “inside” the species tree. Over arcs of the species tree, the gene tree is modeled by a standard birth-death process [9], where birth and death events correspond to duplications and losses, with birth rate λ and death rate μ . When the process reaches the end of an arc, i.e., a species tree vertex, it is split into two identical and independent copies. One of the processes evolves down the left outgoing arc of and the other evolves down the right outgoing arc; moreover, this evolution continues recursively downward to the leaves of the species tree where it stops. After the recursive process has stopped, the gene tree is pruned by removing vertices that have no descendants in the leaves of the species tree and finally by short-cutting vertices of degree two (i.e., removing the vertex and connecting its neighbors in the natural way). A leaf of the resulting gene tree is labeled, in the natural way, with the unique leaf of the species tree to which the gene tree leaf reached in the evolution process. This ends the description of how the gene evolution model generates gene trees. In Figure 1, an example is provided of how a gene tree evolves inside a species tree. We will often view the gene evolution model as a process generating a gene tree G as well as a labeling $\tau : V(G) \setminus L(G) \rightarrow R^+$ of its internal vertices with duplication times.

3.2 Gene sequence evolution model

We now extend the gene evolution model to the gene sequence evolution model by including sequence evolution. That is, from the creation of a new gene lineage, by a duplication or a speciation, a sequence evolves according to some standard sequence evolution model, see e.g. [3], until the lineage is split into two, again by a duplication or a speciation, which cause the sequence to split into two initially identical but henceforth independently evolving sequences. In the results presented in this paper, we have, for simplicity, assumed the Jukes-Cantor model of sequence evolution [8] as well as a molecular clock model for substitution rate variation among arcs of the gene tree. More general models, e.g. GTR [10, 12] or evolving rates [15], can easily be incorporated in our method; the model parameters would then be estimated in the MCMC-analysis. The gene sequences evolution model is hierarchical, in the sense that evolving the gene tree using the gene evolution model and then letting a standard sequence evolution model act on the gene tree gives the same result.

4. RECONCILIATIONS

In this section, we proceed to define a *reconciliation*. The definition in [2] was incomplete, below follows the correct one. Before we set out some additional definitions are re-

quired. Let T be a rooted directed tree. For $u \in V(T)$, the *descendants* of u in T are the vertices of T_u . This means that u is a descendant of u . That v is a descendant of u in T is denoted $v \leq_T u$. Let $u, v \in V(T)$, then u and v are incomparable in T if $u \not\leq_T v$ and $v \not\leq_T u$. A set $\mathcal{A} \subseteq V(T)$ is a \leq_T -*antichain* if for each pair $u, v \in \mathcal{A}$ it holds that u and v are incomparable in T . A set $U \subset V(T)$ *separates* $u \in V(T)$ and $v \in V(T)$ if there is no path between u and v in $T \setminus U$. A tree T' is a *subdivision* of a tree T if removing each vertex of degree at most two in T' , and connecting its neighbors in the natural way, yields T .

Formally, a *reconciliation* of a species tree S and a gene tree G , with leaf labeling σ , is a pair, (γ, G') , where G' is a subdivision of G and γ is a function $\gamma : V(S) \rightarrow 2^{V(G')}$ such that:

1. $\gamma(r(S)) \neq \emptyset$
2. For any $l \in L(G')$, $l \in \gamma(\sigma(l))$
3. For any $x, y \in V(S)$, $\gamma(x) \cap \gamma(y) \neq \emptyset \Rightarrow x = y$
4. For any $x \in V(S)$, $\gamma(x)$ is a $\leq_{G'}$ -antichain
5. For any $y, z \in V(S)$ such that $v \in \gamma(y)$ and $w \in \gamma(z)$, if y, z are incomparable in S and v, w are incomparable in G' , then $\gamma(\text{LCA}(y, z))$ separates v and w
6. For any $x, y, z \in V(S)$, such that $u \in \gamma(x)$, $w \in \gamma(z)$, $z \leq_S y \leq_S x$ and $w \leq_{G'} u$, there is a $v \in \gamma(y)$ for which $w \leq_{G'} v \leq_{G'} u$
7. For any u with child v in G' , if $u \in \gamma(x)$, then there is a child y of x in S such that $V(G'_v) \cap \gamma(V(S_y)) \subseteq \gamma(V(S_y))$
8. For any $u \in \gamma(V(G'))$ with two children v and w , $V(G'_v) \cap \gamma(x) \neq \emptyset \Rightarrow V(G'_w) \cap \gamma(x) = \emptyset$ and $V(G'_w) \cap \gamma(x) \neq \emptyset \Rightarrow V(G'_v) \cap \gamma(x) = \emptyset$

We will, in the subsequent text, let G' be implicitly given by G and, hence, use γ alone to indicate a reconciliation.

The evolution of the gene tree inside the species tree also induces a reconciliation of the two trees as follows: $u \in \gamma(x)$

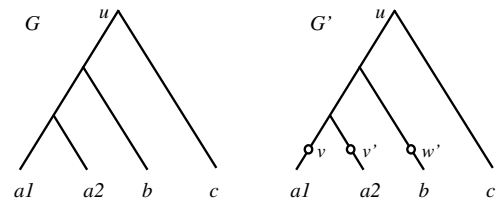


Figure 2: G' is a subdivision of G .

if and only if u is a “leaf” of the birth-death process when it reaches x , and u at the end of the process has a descendant below x .

5. MCMC-ESTIMATION OF THE POSTERIOR DISTRIBUTION OF GENE TREES

This section contains a brief introduction to the MCMC framework as well as a brief description of how this framework is applied in the present case. For a more thorough account of MCMC and standard MCMC terminology, we refer to [5].

MCMC is a technique that facilitates estimation of the stationary distribution of a Markov Chain. It provides a uniform framework to design transition probabilities of a Markov Chain so that a sought stationary probability distribution is obtained. A random walk is performed in the Markov Chain according to the transition probabilities. In the present case, a state in the Markov chain is a triple (G, λ, μ) where G is a gene tree, λ a birth rate, and μ is a death rate. In order to simplify the description, we describe the Markov chain as if λ and μ were discrete parameters. Let $F = \{q_1, \dots, q_n\}$ be the set of observed gene sequences. The sought stationary distribution is the *a posteriori* distribution of gene trees, i.e. posterior to the observations of F ,

$$Pr[G, \lambda, \mu | F] = \frac{Pr[F, G | \lambda, \mu] Pr[\lambda, \mu]}{Pr[F]}.$$

We use a uniform *prior* for λ , and μ . The probability $Pr[F, G | \lambda, \mu]$ is called the likelihood and we will in Section 5.1 show how it can be approximated with small error. The posterior distribution thus assigns to a state (G, λ, μ) the probability that, in the gene evolution process: G was the gene tree, λ was the birth rate, and μ was the death rate, conditioned by the observed gene sequences F . In the limit, the fraction of visits to a state during the simulation, in relation to the total number of visits, is the stationary probability. In practice, frequencies are collected after a period of *burn in*, i.e., the time it takes for the chain to “forget” its starting state, and up to an estimated *stopping time*, sufficiently late to make the estimation of the stationary distribution reliable, see [5].

In each iteration of an MCMC simulation a new state is proposed according to a specific proposal distribution. The proposal distribution that we use is based on, so called, nearest neighbor interchange branch swapping, [14], The new state is accepted, i.e. becomes the current state, or rejected, in which case no change is made of the current state, according to an acceptance distribution. In the present case, a symmetric proposal distribution is used, which means that the algorithm proposed here is an instance of the Metropolis method [5]. Consequently, when the present state is (G, λ, μ) , the acceptance probability for a proposed state (G', λ', μ') is $\alpha_{ij} = \frac{Pr[F, G' | \lambda', \mu']}{Pr[F, G | \lambda, \mu]}$.

Notice that the MCMC framework together with the capacity to compute the likelihood, $Pr[F, G | \lambda, \mu]$, and to sample from the proposal distribution gives us the MCMC algorithm that estimates the *a posteriori* distribution of reconciliations.

5.1 Approximating the likelihood well

The likelihood $Pr[F, G | \lambda, \mu]$ can be expressed as follows

$$\sum_{\gamma, \tau} Pr[F | \tau, \gamma, G, \lambda, \mu] Pr[\tau | \gamma, G, \lambda, \mu] Pr[\gamma, G | \lambda, \mu],$$

where τ is the duplication time.

Sampling τ from $Pr[\tau | \gamma, G, \lambda, \mu]$ is fairly uncomplicated and the description of it is omitted. Sampling γ from $Pr[\gamma, G | \lambda, \mu]$, which in contrast is complicated, can be done using our dynamic programming approach. Hence, to estimate the likelihood we can sample τ, γ from their joint distribution m times and sum the values of $Pr[F | \tau, \gamma, G, \lambda, \mu]$. Since the substitution process depends only on G and τ this probability simplifies to $Pr[F | \tau, G]$, which is standard to compute [14]. Hoeffding’s bound [6] implies that after m samples our sum differs from the likelihood by at most ϵ with probability at most $1 - 2e^{-2m\epsilon^2}$.

5.2 Orthology analysis

We will now describe how to estimate the posterior probability that a given pair of genes a and b are orthologs, i.e. $Pr[a \text{ and } b \text{ orthologs} | F]$. Since this probability can be expressed as:

$$\sum_{G, \lambda, \mu} \frac{Pr[a \text{ and } b \text{ orthologs}, F, G | \lambda, \mu]}{Pr[F, G | \lambda, \mu]} Pr[G, \lambda, \mu | F]$$

and we already know how to estimate $Pr[G, \lambda, \mu | F]$, it suffices to show how to obtain the probabilities $Pr[a \text{ and } b \text{ orthologs}, F, G | \lambda, \mu]$ and $Pr[F, G | \lambda, \mu]$. The gene tree vertices associated with species tree vertices by a reconciliation are speciations; all other gene tree vertices are duplications. According to the original definition by Fitch [4], two genes are orthologs if and only if their least common ancestor in the gene tree is a speciation. Thus, by adapting the sampling approach from the previous section, summing the probability $\sum_{\gamma, \tau} Pr[F | \tau, \gamma, G, \lambda, \mu]$ separately for the reconciliations where the gene pairs, a and b , are orthologs and where they are paralogs, we can estimate the probability $Pr[a \text{ and } b \text{ orthologs}, F, G | \lambda, \mu]$.

5.3 Posterior distribution of species trees

In this section, we show that the likelihood computations described in this paper are sufficient to obtain an MCMC algorithm that estimates the posterior distribution of a species tree w.r.t. a number of gene families $\mathcal{F} = \{F_1, \dots, F_k\}$, where each F_i is a set of sequence data for a gene family and each gene sequence is labeled with the species in which genome it can be found. Thus, the species tree is not fixed in this section. We will now describe an MCMC algorithm that estimates the posterior probability

$$Pr[S, G_1, \dots, G_k, \lambda, \mu | \mathcal{F}], \quad (1)$$

from which it is easy to obtain the wanted distribution $Pr[S | \mathcal{F}]$. The posterior (1) equals

$$\frac{Pr[\mathcal{F}, G_1, \dots, G_k | S, \lambda, \mu] Pr[S, \lambda, \mu]}{Pr[\mathcal{F}]}.$$

Using uniform priors on species trees and rates, only the likelihood needs to be handled and it can be expressed as follows,

$$Pr[\mathcal{F}, G_1, \dots, G_k | S, \lambda, \mu] = \prod_{1 \leq i \leq k} Pr[F_i, G_i | S, \lambda, \mu],$$

where each factor $Pr[F_i, G_i | S, \lambda, \mu]$ can be handled as in Section 5.1.

6. EXACT ALGORITHMS FOR MAXIMUM, SUM, AND SAMPLING OF LIKELIHOODS

In [2] a dynamic programming algorithm was described for computing the likelihood of a reconciliation and a gene tree, given a species tree and model parameters. This algorithm was then used to develop a MCMC-based framework that allowed estimation of the posterior distribution of reconciliations.

A decomposition of the gene tree is central to the dynamic programming algorithm for the likelihood in [2]. The gene tree is decomposed into, so called, sliced subtrees and rooted subtrees. A sliced subtree of the gene tree is only defined w.r.t. to a reconciliation; it consists of the part of a gene tree that according to the reconciliation occurred inside a specific edge of the species tree. The rooted subtrees are the subtrees of the gene tree rooted at the leaves of the sliced subtree, and hence also those are defined with respect to a reconciliation.

The main technical obstacle to the design of the sum, the maximum, and the sampling algorithm is that there is an exponential number of reconciliations, in terms of the size of the gene tree, and that they also give rise to an exponential number of sliced trees. This obstacle is circumvented by devising dynamic programming algorithms that do not need to be presented with explicit sliced trees, for which instead it suffices to base the computation on small local configurations of the sliced tree, consisting of a parent and its two children, and a small number of stored values.

We will now review the recursive expression for the likelihood, on which the dynamic programming algorithm was based. For simplicity, we will assume that our reconciliation γ satisfy $\gamma(r(S)) = \{r(G)\}$.

6.1 Decomposition and recursive expression for a specific reconciliation

This subsection describes how to decompose the gene tree so that a recursive expression for the likelihood can be obtained. If $u \in V(T)$ and $U \subseteq V(T)$, then $T_{u,U}$ denotes the subtree of T induced by $\{v : \exists u' \in U, u' \leq_T v \leq_T u\}$. For $\langle x, y \rangle \in A(S)$ and $u \in \gamma(x)$, we call $G_{u,\gamma(y)}$ a *sliced subtree*. The tree $G_u^{x,y}$ refers to $G_{u,U}$ with $U = V(G_u) \cap_{z \in V(S^{x,y})} \gamma(z)$, that is, the part of G_u that has evolved in the arc subtree of S for $\langle x, y \rangle$. These definitions are illustrated by an example appearing in Figure 3. Let $\gamma_u : V(S) \rightarrow 2^{V(G_u)}$ be defined by $\gamma_u(z) = \gamma(z) \cap V(G_u)$, for any $z \in V(S)$. Similarly, let $\gamma_u^{x,y} : V(S^{x,y}) \rightarrow 2^{V(G_u^{x,y})}$ be defined by $\gamma_u^{x,y}(z) = \gamma(z) \cap V(G_u^{x,y})$.

Define $e_V(\gamma_u, x, u)$ as the probability that G_u and γ_u have evolved from u starting at x in S_x . Similarly, define $e_A(\gamma, y, u)$ as the probability that $G_u^{x,y}$ and $\gamma_u^{x,y}$ has evolved from u starting at x in $S^{x,y}$. Assume that x has children y and z in S . In [2], the dynamic programming for computing the likelihood of a specific reconciliation was based on the mutually recursive equations (2)-(4) below. Due to independence,

$$e_V(\gamma, x, u) = \begin{cases} 1, & x \in L(S), u \in L(G) \\ e_A(\gamma_u^{x,y}, y, u) e_A(\gamma_u^{x,z}, z, u), & \text{otherwise} \end{cases} \quad (2)$$

There are two cases for e_A : first, for $\gamma_u(y) \neq \emptyset$,

$$e_A(\gamma, y, u) = p_y(|\gamma(y)|) h(\gamma, y, u) \phi(\gamma, y, u) \prod_{v \in \gamma_u(y)} e_V(\gamma_v, y, v), \quad (3)$$

second, for $\gamma_u(y) = \emptyset$,

$$e_A(\gamma, y, u) = p_y(0). \quad (4)$$

In the next subsections, we will define and explain h and ϕ . The factor $p_y(l)$ is up to a normalizing factor the probability that the tree generated by the birth-death process over the edge $\langle x, y \rangle$ has l leaves; how to compute it follows from [2, 9].

6.2 Probability for sliced tree

To express the probability of a sliced subtree $G_{u,\gamma(y)}$, we view the birth-death process over an edge $\langle x, y \rangle \in A(S)$ as generating trees in two steps. In the first step, labeled trees are generated in such a way that all labeled trees with l leaves are equiprobable. In the second step, the labels are simply removed and the result is an unlabeled tree.

The probability of an unlabeled tree T with l leaves to be generated can, hence, be expressed as a product $a(T)h(T)$, where $a(T)$ is the probability that the labeled tree has l leaves and $h(T)$ is the fraction all of labeled trees with l leaves that has T as its underlying unlabeled tree.

A *labeling* of a tree T is a vertex labeling together with an arc labeling of T . A *labeled tree* is a tree T given together with a labeling. An *arc labeling* of a tree T is a function $\chi_A : A(T) \rightarrow \{L, R\}$, where a label specify if an arc is a left (L) or a right (R) outgoing arc. A *vertex labeling* of a tree T is a function $\chi_V : V(T) \setminus L(T) \rightarrow [1, |L(T)| - 1]$ such that $u \leq_T v$ implies $\chi(u) \leq \chi(v)$. The vertex labels specify the birth order of vertices. To derive the the following lemma, we will count the number of underlying labellings of a tree T by first counting the number of arc labellings of T and then counting the number of vertex labellings of T together with any of these arc labellings.

LEMMA 1.

$$h(\gamma, y, u) = \begin{cases} 1 & \text{if } u \in \gamma(y) \\ \kappa h(\gamma, y, c_1(u)) h(\gamma, y, c_2(u)) & \text{otherwise,} \end{cases}$$

$$\text{where } \kappa = \frac{2^{\delta(G_u, \gamma(y))}}{|L(G_u, \gamma(y))| - 1}.$$

6.3 The number of ways to generate the rest of the tree

It is important for the understanding of this subsection to realize that the gene tree G and the corresponding reconciliation γ actually represent all pairs of gene trees and reconciliations such that the gene trees are isomorphic with respect to the reconciliation. An isomorphism f between gene trees G and G' is said to respect the corresponding reconciliations γ and γ' if and only if, for any $u \in V(G)$ and $x \in V(S)$, it holds that $u \in \gamma(x) \Leftrightarrow f(u) \in \gamma'(x)$. We denote the set of such ordered pairs $\langle G, \gamma \rangle$ by $[G, \gamma]_{\cong}$, i.e. $[G_i, \gamma_i]_{\cong}$ is an isomorphism class. Thus the names of vertices of G are irrelevant.

The expression $\phi(\gamma, y, u)$ in (3) is defined to be

$$\phi'(G_{u,\gamma(y)}, [G_{v_1}, \gamma_{v_1}]_{\cong}, \dots, [G_{v_l}, \gamma_{v_l}]_{\cong}),$$

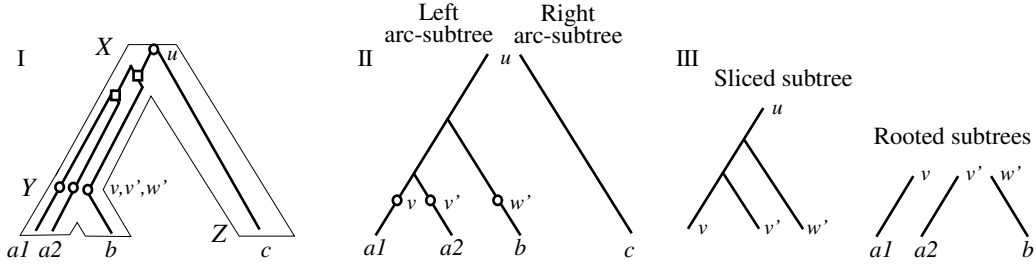


Figure 3: (I) A reconciliation γ of G and S . (II) The trees $G_u^{x,y}$ and $G_u^{x,z}$ with respect to G, S and γ in (I). (III) The sliced subtree $G_{u,\gamma(y)}$ and the rooted subtrees $G_v, G_{v'}$ and $G_{w'}$ for the same gene tree, species tree and reconciliation as above.

where v_1, \dots, v_l are the vertices of $\gamma_u^{x,y}(y)$ and the yet undefined parts of the expression are defined below. For a tree T with leaves $[l]$ and labels m_1, \dots, m_l , the factor $\phi'(T, m_1, \dots, m_l)$ is the number of different leaf labellings $\mathcal{L} : [l] \rightarrow \{m_1, \dots, m_l\}$ such that there is an automorphism $f : V(T) \rightarrow V(T)$ (i.e. an isomorphism from T to itself) satisfying $m_i = \mathcal{L}(f(i))$.

Let $\delta_\gamma(T)$ be defined by

$$\delta_\gamma(T) = \begin{cases} 0 & \text{if } T_{c_1(u)} \not\cong T_{c_2(u)} \\ 0 & \text{if } [T_{c_1(u)}\gamma_{c_1(u)}] \cong [T_{c_2(u)}, \gamma_{c_2(u)}] \\ 1 & \text{otherwise.} \end{cases}$$

LEMMA 2.

$$\phi(\gamma, y, u) = \begin{cases} 1, & \text{if } u \in \gamma(y) \\ \eta \phi(\gamma, y, c_1(u)) \phi(\gamma, y, c_2(u)), & \text{otherwise,} \end{cases}$$

where $\eta = 2^{\delta_\gamma(G_{u,\gamma(y)})}$.

6.4 Dynamic programming for max

In this subsection, we describe the recursions on which the dynamic programming for the maximum likelihood reconciliation is based. We then briefly describe how the recursions are turned into a dynamic programming algorithm.

In the dynamic programming algorithm for maximum, we will for each pair of vertices $y \in V(S)$ and $u \in V(G)$ store the values of $e_A(\gamma, y, u)$ and $e_V(\gamma, y, u)$ for the three different reconciliations that maximize those values. The final answer, i.e. the likelihood for the most likely reconciliation, is the maximum of $e_V(\gamma, r(S), r(G))$. To be able to compute the values of these variables in the dynamic programming, we would like to show recursions for the variables, i.e. show that they can be expressed using variables for (1) $y' <_S y$ and $u' \leq_G u$ or (2) $y' \leq_S y$ and $u' <_G u$. For $e_V(\gamma, y, u)$, this is shown in (2). To handle $e_A(\gamma, y, u)$, we introduce an additional function

$$f_A(\gamma, y, u) = h(\gamma, y, u) \phi(\gamma, y, u) \prod_{v \in \gamma_u(y)} e_V(\gamma_v, y, v).$$

Clearly

$$e_A(\gamma, y, u) = p_y(|\gamma(y)|) f_A(\gamma, y, u) \quad (5)$$

which gives the recursion for e_A . Using the recursions for h and ϕ from the previous sections, we can show that the

following recursion holds for f_A .

$$f_A(\gamma, y, u) = \begin{cases} 1, & \text{if } G_u^{x,y} \text{ is empty} \\ e_V(\gamma, y, u), & \text{if } u \in \gamma(y) \\ \xi f_A(\gamma_{c_1(u)}, y, c_1(u)), & \text{otherwise,} \\ f_A(\gamma_{c_2(u)}, y, c_2(u)) \end{cases} \quad (6)$$

where $\xi = \frac{2^{\delta(G_u) + \delta_\gamma(G_u)}}{|\gamma(y)| - 1}$. Notice that the sum $\delta(G_u) + \delta_\gamma(G_u)$ is equal to,

$$\begin{cases} 0, & \text{if } [G_{c_1(u)}, \gamma_{c_1(u)}] \cong [G_{c_2(u)}, \gamma_{c_2(u)}] \\ 1, & \text{otherwise.} \end{cases}$$

Naturally, in the recursions (2), (4), (5), and (6) the variable for a reconciliation γ is expressed in terms of variables for the same reconciliation γ or in terms of reconciliations which are "subreconciliations" of γ , i.e. restrictions of γ to subtrees of G . The dynamic programming, however, will as usual be performed bottom up in the trees, from the leaves to the roots. The recursions (4) and (5), where the right-hand sides only contains the same reconciliation as the left-hand sides or no reconciliation are unproblematic; whilst the recursions (2) and (6) require a few observations. We now make the key observations needed to convert the recursions to a dynamic programming algorithm for the maximum. Let $1 \leq i \leq 3$, let x be a vertex with children z and x in S , and let u be a vertex with children v and w in G .

We first make the observation concerning (2). If γ is the reconciliation giving the i :th greatest value of $e_V(\gamma, x, u)$, then $\gamma_u^{x,y}$ gives the j :th greatest value for $e_A(\gamma_u^{x,y}, y, u)$ where $1 \leq j \leq 3$ and $\gamma_u^{x,z}$ gives the k :th greatest value for $e_A(\gamma_u^{x,z}, z, u)$ where $1 \leq k \leq 3$.

We now make the observation concerning (6). If γ is the reconciliation giving the i :th greatest value of $f_A(\gamma, y, u)$, then $\gamma_{c_1(u)}$ gives the j :th greatest value for $f_A(\gamma_{c_1(u)}, y, c_1(u))$ where $1 \leq j \leq 3$ and $\gamma_{c_2(u)}$ gives the k :th greatest value for $e_A(\gamma_{c_2(u)}, y, c_2(u))$ where $1 \leq k \leq 3$. Moreover, $\delta(u) + \delta_\gamma(u) = 0$ if and only if $G_{c_1(u)} \cong G_{c_2(u)}$ and $j = k$. Using latter observation $\delta(u) + \delta_\gamma(u)$ can be computed without knowing γ .

6.5 The probability of a gene tree

We now turn to the problem of computing $\Pr(G|S)$ by summing over all reconciliations:

$$\Pr(G|S) = \sum_{\gamma \in \Gamma} \Pr(G, \gamma|S), \quad (7)$$

$$\begin{aligned}
A(y, u) &= \begin{cases} p_y(0), & \text{if } c(y, u) \text{ does not exist} \\ A(y, c(y, u)) & \text{if } u \in \gamma^*(x), \text{ where } x \text{ is the parent of } y \\ \sum_{k=\mathcal{L}(y,u)}^{\mathcal{U}(u)} p_y(k) B(y, u, k), & \text{otherwise} \end{cases} \\
B(y, u, k) &= \begin{cases} 1, & u \in L(G), y \in L(S) \\ A(c_1(y), u) A(c_2(y), u), & k = 1 \\ \frac{2^{\delta(\tau_u)}}{k-1} \sum_{(k_1, k_2) \in \mathcal{K}(y, u, k)} B(y, c_1(u), k_1) B(y, c_2(u), k_2), & \mathcal{L}(y, u) \leq k \leq \mathcal{U}(u) \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}$$

Figure 4: Recursion for computing the sum (12).

where Γ is the set of all reconciliations.

The primary means of computation will be expressing $\Pr(G_u^{x,y} | S^{x,y})$, i.e., the probability of a subtree of G being reconciled to an arc subtree of S . This which will facilitate a recursion over G and S .

For the rest of this subsection, let γ^* denote the reconciliation with the least number of postulated duplications and losses.

As $\Pr(G_u^{x,y} | S^{x,y})$ is a different computational objective, we will rephrase the functions we are interested in slightly. Assume that $x \in V(S)$ has children y and z in $V(S)$. Define

$$\begin{aligned}
e_B(\gamma, y, u) &= p_y(|\gamma(y)|) h(G_{u, \gamma(y)}) \\
&\prod_{v \in \gamma_u(y)} e_W(\gamma_v, y, v) \quad (8)
\end{aligned}$$

and

$$\begin{aligned}
e_W(\gamma, x, u) &= \\
&\begin{cases} 1, & x \in L(S), u \in L(G) \\ e_B(\gamma_u^{x,y}, y, u) e_B(\gamma_u^{x,z}, z, u), & \text{otherwise} \end{cases} \quad (9)
\end{aligned}$$

as analogs to e_A and e_V , in expression (3), (4) and (2) correspondingly. The difference is that the ϕ factor, which is the number of equivalent reconciliations, is removed from e_A . It is possible to express e_A in terms of e_B by

$$e_A(\gamma, y, u) = \sum_{\gamma' \in I(\gamma)} e_B(\gamma', y, u), \quad (10)$$

where $I(\gamma) = \{\gamma' : [G, \gamma] \cong [G, \gamma']\}$.

A slice $G_{u, \gamma(y)}$ is bounded in size due to constraints on reconciliations. No slice can be larger than the number of leaves in the subtree rooted at u , so there is an upper bound $\mathcal{U}(u) = |L(G_u)|$. There are also lower bounds for slice sizes, since some duplications by necessity must occur before certain speciations.

The lower bound is expressed as: Let $u \in \gamma^*(y')$, $y' \in V(S)$ then

$$\mathcal{L}(y, u) = \begin{cases} 1 & \text{if } y' \leq_S y \\ \mathcal{L}(y, c_1(u)) + \mathcal{L}(y, c_2(u)) & \text{otherwise.} \end{cases} \quad (11)$$

Define $\Gamma_u^{x,y}$ as the set of reconciliations from $V(S^{x,y})$ to $2^{V(G_u^{x,y})}$. We can now state our computational goal:

$$\Pr(G_u^{x,y} | S^{x,y}) = \sum_{\gamma \in \Gamma_u^{x,y}} e_B(\gamma, y, u). \quad (12)$$

The following definitions are needed for constructing a recursion for computing the sum (12).

Let $y' \in V(S)$ such that $y' \leq_S y$. If y' is associated to $c_i(u)$, $i = 1, 2$, or its descendants, then $c(u, y) = c_i(u)$. Otherwise $c(u, y)$ does not exist.

$$\begin{aligned}
\mathcal{K}(y, u, k) &= \{(k_1, k_2) : k_1 + k_2 = k, \\
&\mathcal{L}(c_1(y), c_1(u)) \leq k_1 \leq \mathcal{U}(c_1(u)), \\
&\mathcal{L}(c_2(y), c_2(u)) \leq k_2 \leq \mathcal{U}(c_2(u))\},
\end{aligned}$$

which describes the possible slice sizes.

A recursion for computing the sum (12) is given in Figure 4 using two mutually recursive functions, A and B . The former is used to start a slice, thus expressing the probability of reconciling a subtree of G with a subtree of S , while the latter is used to assemble partial results from the lower vertices of a gene tree slice to the slice's root vertex.

The recursion for computing the sum as stated in Figure 4 is inefficient to compute. By doing a depth-first search in the gene tree and computing values for A and B in a carefully chosen order, a time complexity of $O(|V(G)| |V(S)|)$ is achieved.

The recursions A and B in Figure 4 can be utilized for constructing a probability distribution for gene-tree slice sizes. Sampling a reconciliation amounts to create slices while recursing down the gene tree. When a new slice $G_{u, \gamma(y)}$ is to be created, its size $k = |\gamma_u(y)|$ is chosen randomly according to the distribution $\frac{p_y(k) B(y, u, k)}{A(y, u)}$. Next, the number of leaves k_1 in $G_{c_1(u), \gamma(y)}$ is chosen with probability

$$\frac{B(y, c_1(u), k_1) B(y, c_2(u), k - k_1)}{\sum_{(k_1, k_2) \in \mathcal{K}(y, u, k)} B(y, c_1(u), k_1) B(y, c_2(u), k_2)}.$$

Thus, the structure of the two subtrees have been constrained, and we recurse down to determine their actual structure. If $k = 1$, then it is time to start a new slice.

7. EXPERIMENTAL RESULTS

7.1 Application of the dynamic programming algorithm to biogeography

As a test case of our dynamic programming algorithm, we considered the problem of the breakup of the Gondwana supercontinent considered in [13]: Given a reliable phylogeny for 24 southern beeches (*Nothofagus*) taxa (Figure 5), which one out of three hypotheses for the breakups of Eastern Gondwana (Figure 6), formulated *a priori* from geological evidence, is most likely? Notice that in this biogeography context, a tree describing a continental breakup is taking the place of a species tree. Vertices in the *Nothofagus* phylogeny

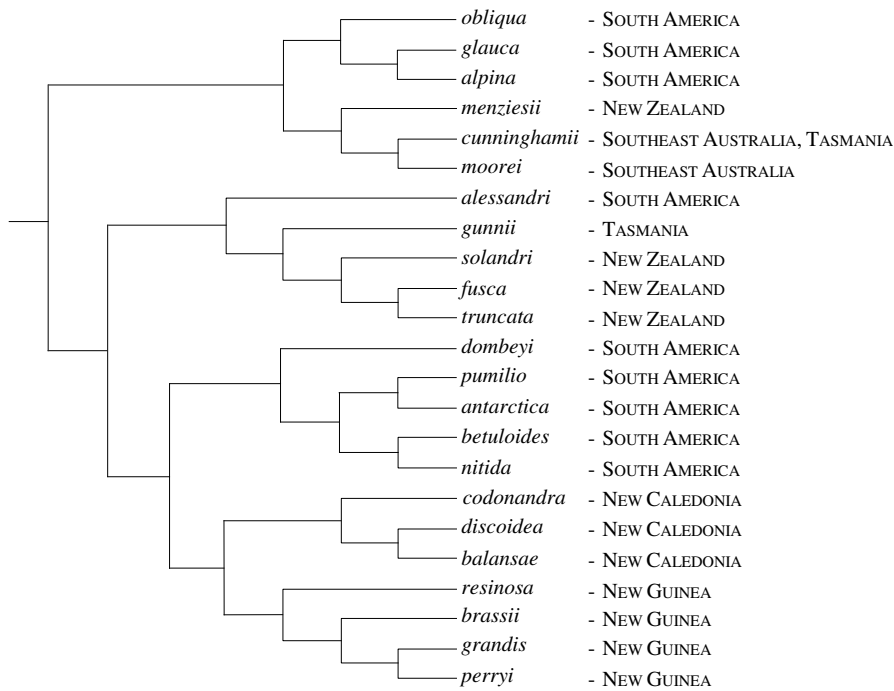


Figure 5: The *Nothofagus* phylogeny (redrawn from [13]).

Table 1: Probabilities, speciation (λ) and extinction (μ) rates, and minimum number of speciations for the three hypotheses of Gondwanan breakup.

Hypoth.	P	λ	μ	Min spec.
A	8.4e-33	0.0615	0.0694	17
B	4.2e-29	0.0745	0.0863	19
C	6.2e-31	0.0715	0.0753	18

are either from continental separation (allopatric speciation) or (sympatric) speciations within the continent.

Using the standard parsimony reasoning, hypothesis A in Figure 6 could be favored over hypothesis B and C since it postulates only 17 speciations while the others requires as many as 19 and 18 respectively. However, using an MCMC approach to integrate over the birth and death parameters in the gene evolution model, we can estimate the probability of the *Nothofagus* phylogeny to evolve within the alternative breakups. As can be seen in Table 1, hypothesis B is the most likely alternative, despite the extra speciations. In all three scenarios, the estimated (sympatric) speciation and extinction rates are quite similar, thus vouching for some stability in the result.

7.2 MCMC on synthetic data

In [2] “The 90%-test” was introduced, i.e., statistics concerning the fraction of the total number of experiments in which the generated gene tree was among best 90% of the gene trees in the *a posteriori* distribution. Ties were broken using a probabilistically sound method. For a sufficiently large number of experiments, the better the *a posteriori* distribution has been estimated the closer the *expected value* of this fraction will be to 0.9. Thus, this test can be used as a measure of the efficiency of the *a posteriori* estimation.

The 90%-test was performed on the methods described in this paper, using MCMC to integrate over gene trees. We generated gene trees and gene sequence data using the gene sequence evolution model on a species tree with 3 taxa and various birth and death rates, which gave up to approximately 12 genes, each with a DNA sequence of 100 positions. One hundred thousand MCMC-iterations were performed on the generated data, sampling every tenth iteration. We present here an experiment result with $\lambda = 2.2$ and $\mu = 2.0$, which yielded the value 0.908 for the 90%-test.

7.3 The major histocompatibility complex (MHC) multigene family

In [2], an MCMC-implementation of gene evolution model was applied to an orthology problem from the major histocompatibility complex family. The phylogenetic tree of MHC class I genes from Orangutan, Gorilla, Tamarin and Cat was used. This is a subtree of the gene tree described in [11]. To evaluate sensitivity to genome sampling, the Human MHC class I genes were removed from this subtree, simulating that the human genome had not been sampled. When included, the human genes reveal that two groups of genes that in the subtree can be paralogs or orthologs to each other, are in fact paralogs (cf. Figure 7). It was shown that while parsimony, falsely, predicted these groups to be orthologs, the gene evolution model allowed a significant probability, 0.11, for the correct answer.

We here apply the gene sequence evolution model, as described in this paper, to the same problem. As in [2], we used a species tree with divergence time estimates from [1], but here the analysis was not restricted to the single gene tree from [11]; instead we investigated all alternative gene trees for the given sequences.

An initial MCMC analysis without sequence evolution,

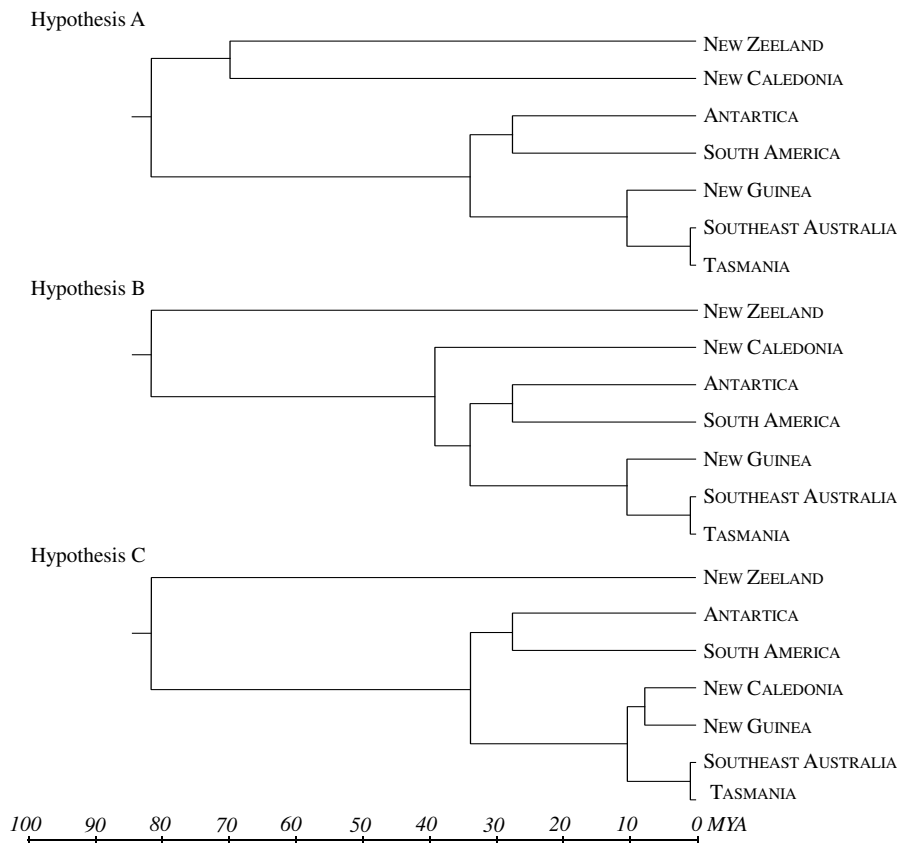


Figure 6: The three geological hypotheses A, B and C for the Gondwanan breakup (redrawn from [13]).

Table 2: Orthology and paralogy probabilities for the MHC-data using different methods of orthology analysis. (Abbreviations used: GEM = Gene Evolution Model, GSEM = Gene Sequence Evolution Model)

Hypothesis	Parsimony	GEM	GSEM
Orthology	Yes	0.89	0.75
Paralogy	No	0.11	0.25

i.e., using the gene evolution model, and with the gene tree from [11] fixed, was used to identify reasonable values for birth- and death parameters, $\lambda = 0.024$, $\mu = 0.013$. The parameters were fixed for these values in the subsequent full gene sequence evolution MCMC performed over 1000 iterations to integrate over the gene trees. The Markov chain converged after approximately 200 iterations, and the preceding samples were discarded. The results are shown in Table 2. The paralogy probability for the pairs of genes is now increased to 0.25, thus demonstrating a further improvement in the robustness to incomplete genome sampling.

8. ACKNOWLEDGMENTS

The authors wish to thank Isaac Elias and four anonymous reviewers for comments on a previous draft of the manuscript. Ulf Swenson kindly shared his knowledge on Pacific biogeography. Funding for this project was in part from the Swedish Foundation for Strategic Research, the Swedish Research Council and the Carl Trygger Foundation.

9. REFERENCES

- [1] U. Arnason and A. Janke. Mitogenomic analyses of eutherian relationships. *Cytogenetic and Genome Research*, 96(1-4):20–32, 2002.
- [2] L. Arvestad, A.-C. Berglund, J. Lagergren, and B. Sennblad. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics/ISMB'03*, 19:i7–i15, 2003.
- [3] J. Felsenstein. Evolutionary trees from DNA-sequences - a maximum-likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981.
- [4] W. M. Fitch. Distinguishing homologous from analogous proteins. *Systematic Zoology*, 19(2):99–113., 1970.
- [5] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.
- [6] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, Mar. 1963.
- [7] J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, Aug 2001.
- [8] T. Jukes and C. Cantor. Evolution of protein molecules. In H. Munro, editor, *Mammalian protein metabolism*, pages 21–132. Academic Press, New York, 1969.
- [9] D. G. Kendall. On the generalized "birth-and-death" process. *Annals of mathematical statistics*, 19:1–15, 1948.

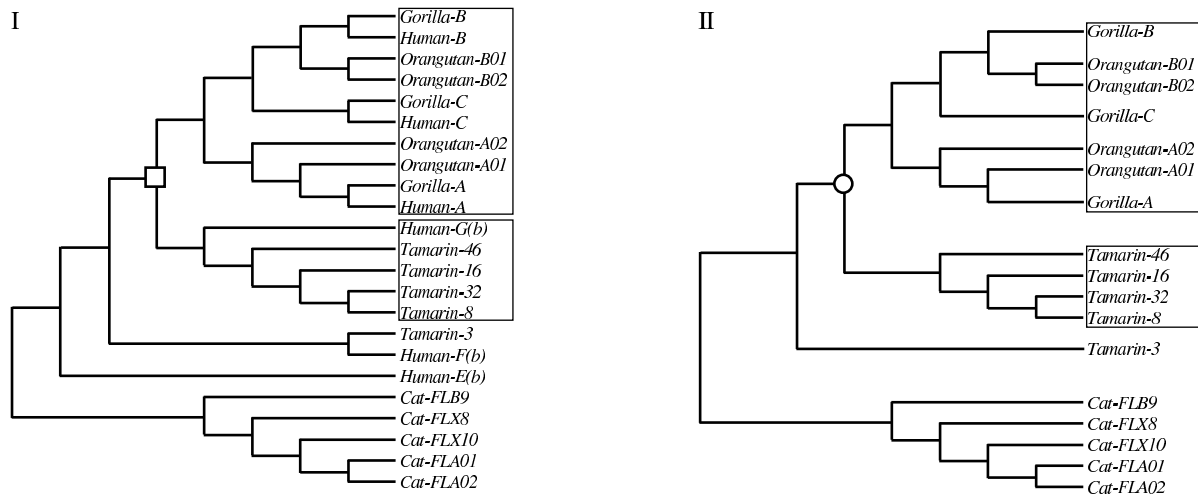


Figure 7: The MHC class I gene trees for primate sequences discussed in the text; the MHC class I genes for cat is included as an outgroup. The two homolog groups of interest are boxed and the status of the least common ancestor, *v*, of these two groups as interpreted by parsimony reconciliation is indicated. (I) The gene tree including including human sequences. Parsimony reconciliation correctly identifies *v* as a duplication (indicated by a square). (II) The tree from (I), but with all human sequences removed, simulating that the human genome was not sampled. Parsimony reconciliation now erroneously identifies *v* as a speciation (indicated by a circle).

- [10] C. Lanave, G. Preparata, and C. Saccone. Mammalian genes as molecular clocks? *Journal of Molecular Evolution*, 21(4):346–50, 1984.
- [11] M. Nei, X. Gu, and T. Sitnikova. Evolution by the birth-and-death process in multigene families of vertebrate immune system. *Proceedings of the National Academy of Science, USA*, 94(15):7799–7806, July 1997.
- [12] F. Rodriguez, J. L. Oliver, A. Marin, and J. R. Medina. The general stochastic model of nucleotide substitution. *Journal of Theoretical Biology*, 142(4):485–501, 1990.
- [13] U. Swenson, A. Backlund, S. McLoughlin, and R. S. Hill. *Nothofagus* biogeography revisited with special emphasis on the enigmatic distribution of subgenus *Brassospora* in New Caledonia. *Cladistics*, 17(1):28–47, 2001.
- [14] D. L. Swofford, G. J. Olsen, P. J. Waddell, and D. M. Hillis. Phylogenetic inference. In D. M. Hillis, C. Moritz, and B. K. Mable, editors, *Molecular systematics*, pages 407–514. Sinauer Associates Inc., Sunderland, MA, 2nd edition, 1996.
- [15] J. L. Thorne, H. Kishino, and I. S. Painter. Estimating the rate of evolution of the rate of molecular evolution. *Molecular Biology and Evolution*, 15(12):1647–1657, 1998.