

A Hybrid Micro-Macroevo­lutionary Approach to Gene Tree Reconstruction

Dannie Durand¹, Bjarni V. Halldórsson², and Benjamin Vernot³

¹ Departments of Biological Sciences and Computer Science,
Carnegie Mellon University
durand@cmu.edu

² Department of Mathematical Sciences, Carnegie Mellon University,
Current address: deCode Genetics, Sturlugata 8, 101 Reykjavik, Iceland
bjarni.halldorsson@decode.is

³ Department of Biological Sciences, Carnegie Mellon University
bvernot@cs.cmu.edu

Abstract. Gene family evolution is determined by microevolutionary processes (e.g., point mutations) and macroevolutionary processes (e.g., gene duplication and loss), yet macroevolutionary considerations are rarely incorporated into gene phylogeny reconstruction methods. We present a dynamic program to find the most parsimonious gene family tree with respect to a macroevolutionary optimization criterion, the weighted sum of the number of gene duplications and losses. The existence of a polynomial time algorithm for duplication/loss phylogeny reconstruction stands in contrast to most formulations of phylogeny reconstruction, which are NP-complete.

We next extend this result to obtain a two-phase method for gene tree reconstruction that takes both micro- and macroevolution into account. In the first phase, a gene tree is constructed from sequence data, using any of the previously known algorithms for gene phylogeny construction. In the second phase, the tree is refined by rearranging regions of the tree that do not have strong support in the sequence data to minimize the duplication/lost cost. Components of the tree with strong support are left intact. This hybrid approach incorporates both micro- and macroevolutionary considerations, yet its computational requirements are modest in practice because the two phase approach constrains the search space.

We have implemented these algorithms in a software tool, NOTUNG 2.0, that can be used as a unified framework for gene tree reconstruction or as an exploratory analysis tool that can be applied *post hoc* to any rooted tree with bootstrap values. NOTUNG 2.0 also has a new graphical user interface and can be used to visualize alternate duplication/loss histories, root trees according to duplication and loss parsimony, manipulate and annotate gene trees and estimate gene duplication times.

1 Introduction

The evolutionary history of a gene family is determined by a combination of microevolutionary events (sequence evolution) and macroevolutionary events. The

macroevolutionary events considered here are speciation, gene duplication and gene loss. Gene tree reconstruction should be based on a model that incorporates both micro- and macroevolutionary events [12], yet few phylogeny reconstruction tools based on such a unified model are available. Furthermore, reconciliation of a gene tree and a species tree can be used to investigate a variety of questions related to macroevolution, such as inferring gene duplications and losses, estimating upper and lower bounds on times these events occurred and determining whether a given pair of homologs is orthologous or paralogous. Algorithmic and software support for both these tasks is needed.

In the current work, we present a dynamic programming algorithm to find all most parsimonious phylogenies with respect to a macroevolutionary model of gene duplication and loss. Given the number of gene family members found in each species as input, our algorithm will construct a tree with the fewest duplications and losses required to explain the data. We show that this problem can be solved in polynomial time, in contrast to most phylogeny reconstruction problems, which are NP-complete [4, 6, 7].

Using this result, we develop a two-phase approach to gene tree reconstruction that incorporates sequence evolution, gene duplication and gene loss in the evaluation of alternate phylogenies. In phase one, a tree is constructed based on a microevolutionary model only. In phase two, regions of the tree that are not strongly supported by the sequence data are refined with respect to a macroevolutionary parsimony model, while regions with strong support are left intact. By reserving consideration of macroevolutionary events until phase two and focusing only on those areas where the sequence data cannot resolve the topology, this hybrid approach reduces the search space, leading to a method that incorporates both types of events, yet has modest computational requirements.

We have implemented these algorithms in a software tool called NOTUNG 2.0, which can be used as a unified framework for gene tree reconstruction or as an exploratory analysis tool that can be applied *post hoc* to any rooted tree with bootstrap values. This extension of an earlier program [3] has the following new features:

- a polynomial time algorithm for finding *all* most parsimonious trees with respect to a weighted gene duplication and loss cost.
- a new graphical interface for tree manipulation and visualization, with features especially designed for analysis of gene duplications, including identification of duplicated nodes, lost leaves and/or subtrees, annotation of subfamilies and color highlighting of rootable edges.

In addition, NOTUNG 2.0 retains all features of the original program including high throughput identification of gene duplications with time estimates and rooting unrooted trees based on duplication/loss score. NOTUNG 2.0, described on the supplementary web site (<http://www.cs.cmu.edu/~durand/Lab/Notung/>), is available for free, public distribution.

In Section 2, we discuss previous work and review reconciliation of a gene tree with a species tree. A taxonomy of macro- and microevolutionary models is presented in Section 3. We state our gene tree reconstruction problems formally

in Section 4 and present algorithms to solve these problems in Section 5. In Section 6, we summarize the capabilities of our software tool, NOTUNG 2.0, and discuss the application of NOTUNG 2.0 to two large data sets to investigate the role of gene duplication in genetic adaptation to environmental change.

2 Previous Work on Reconciliation

The problem of disagreement between gene trees and species trees was first raised in the context of inferring a species tree from a gene tree that may contain paralogies [12]. These concepts were further developed and formalized in [13, 14, 16, 17, 20, 21, 28, 29]. Formally, given a set of rooted gene trees, the problem is to find the species tree that optimizes an evaluation criterion. Several optimality criteria have been proposed (see [9, 10] for a comparative survey), all of which attempt to capture the notion that gene duplication and subsequent loss are rare events. The problem of finding an optimal species tree is NP-hard [16] for the optimality criteria considered so far.

A related body of work deals with algorithms and software tools to analyze the history of duplications and losses in the evolution of a gene family. Page and Charleston [19, 22] have developed two software packages, COMPONENT and GENETREE, that will compute and display duplication histories for rooted gene trees, as well as inferring species trees. Zmasek and Eddy’s Forester package performs some reconciliation functions and provides a tree visualization tool [30, 31].

All of these approaches are based on reconciliation of a gene tree with a species tree, a procedure that can be used to infer the number of duplications and losses and estimate the times at which they occurred. We review reconciliation here. Let T_G be a gene tree and let T_S be a species tree of taxa from which the gene sequences were sampled. The identification of duplication nodes requires constructing a mapping from every node in T_G to a target node in T_S . Each leaf node in T_G is mapped to the node in T_S representing the species from which the sequence was obtained. (Leaf nodes in T_G represent sequences, whereas leaf nodes in T_S represent species.) Each internal node in T_G is mapped to the least common ancestor (*lca*) of the target nodes of its children. In Fig. 1(a), for example, the leaf nodes in the rightmost subtree are mapped to MOUSE and HUMAN. The root of this subtree is mapped to *eut* (eutherian), since the *lca* of MOUSE and HUMAN in the species tree is *eut*.

The number of duplications can be determined by observing that under the mapping, a node in T_G is a *speciation* node if its children are mapped to independent lineages in T_S . If the children of v are mapped to the same lineage (either they are mapped to the same species or one’s mapping is an ancestor of the other’s), then v is a *duplication* node. In the gene tree in Fig. 1(a), both nodes labeled *eut* are speciation nodes since rodents and primates are separate lineages in T_S . The root of the tree is a duplication node because it has the same label as both of its children. The number of losses can also be determined from the mapping. Details of both the duplication and loss count computations are given in [3].

evolution. While it is hard to imagine a case in which optimizing duplication and loss alone would produce a biologically meaningful tree, models based solely on duplication and loss can be important intermediate steps toward a comprehensive model. In the current work, we present a dynamic programming algorithm for finding the most parsimonious duplication and loss tree, which is a useful subroutine in a procedure that does produce biologically important results. In addition, the fact that this problem admits a polynomial time solution is of theoretical interest, since most formulations of phylogeny reconstruction are NP-complete [4, 6, 7]. In a Bayesian context, such models have also been used to investigate related questions such as estimating rates of duplication and loss, determining the posterior probability of a tree with respect to a reconciliation and probabilistic ortholog identification [1, 11].

Unified model: *Find the tree that best explains the data with respect to a model that takes both sequence evolution and duplication and loss into account.* In a 1979 landmark paper, Goodman and colleagues [12] pointed out the importance of using a model of both micro- and macroevolutionary events for constructing gene trees and introduced the term “reconciliation” to describe fitting a gene tree to a species tree. They implemented a heuristic search procedure for obtaining parsimony trees that optimize a cost function based on nucleotide replacement and gene duplication and loss. However, it is difficult to determine how to incorporate events occurring on very different spatial and temporal scales in a single model, and most gene tree reconstruction in the intervening 25 years has been based on sequence evolution alone. Very recently, some Bayesian approaches to a unified model have appeared. Arvestad *et al.* [2] have presented a maximum likelihood method that evaluates a set of sequences with respect to a model that includes the gene tree and the reconciliation, parameterized by birth, death and substitution rates.

The Bayesian framework permits a unified model of macro- and microevolution facilitating an approach that uses both types of information in the reconstruction process. It also has the advantage that it allows us to test evolutionary models and infer parameters of those models. On the down side, Bayesian approaches are notoriously computationally intensive and require sufficient data to obtain reasonable estimates of the parameters. Furthermore, a unified, Bayesian model is a strength when both sequence evolution and gene duplication and loss can be modeled by a neutral, stochastic process, but less natural for data sets under strong selective pressure. Evolutionary change in this latter regime is not stochastic and parsimony may be a more appropriate model. With these considerations in mind, we propose the following approach:

Hybrid model: *Obtain an initial tree using a microevolutionary model and refine it with respect to macroevolutionary considerations.* More specifically, we construct an initial tree based on sequence data alone and assess the support for each edge using bootstrapping [8] (or some other method of significance estimation). Subsequently, we rearrange the resulting tree around edges with weak support in the sequence data (e.g., bootstrap values below some

threshold) to optimize the number of duplications and losses needed to explain the tree, while preserving the structure of the tree at edges with strong support.

This hybrid approach incorporates both micro- and macroevolutionary considerations in phylogeny reconstruction. Although in pathological cases an exponential number of trees must be considered, in practice its computational and data requirements are modest because the two-phase approach constrains the search space. Furthermore, it makes it possible to decouple the micro- and macroevolutionary models. Since duplication and loss occur rarely relative to sequence mutation, parsimony may be a more appropriate macroevolutionary model than maximum likelihood for many data sets. In this paper, we present a particular implementation of this approach where the initial sequence-based tree is constructed using any standard phylogeny reconstruction method and, hence, any microevolutionary model. The refinement step is based on a parsimony model of duplication and loss.

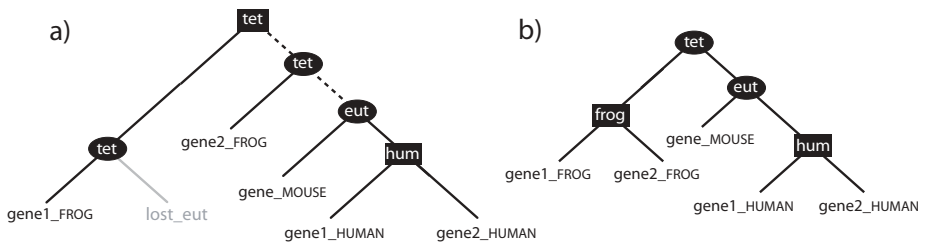


Fig. 2. (a) A gene tree with two duplication nodes and one lost gene. Weak edges are shown as dashed lines. The edge between the *eut* and *human* nodes is robust. The edges adjacent to leaf nodes are neither weak nor robust, since these edges cannot be evaluated using bootstrapping. (b) An alternate topology for the same gene family, with two duplications and no losses

This hybrid approach is illustrated by the hypothetical gene tree in Fig. 2(a), which shows a gene family with two members in frog, two in human and one in mouse. The topology of the tree indicates that two duplications occurred in this gene family. The first occurred in the common ancestor of all three species. One copy was retained in all three species, while the second was retained in frog and lost in mouse and human. We represent this as a single loss in their common ancestor (*eut*), since this is the most parsimonious explanation for the two missing genes. A second duplication occurred within the human lineage. The total score for this tree is two duplications and one loss. However, the edge grouping *gene2_FROG* with the mouse and human genes is weak, suggesting that the sequence evidence does not, in fact, strongly support this topology. Rearranging the tree around the weak edge to place *gene2_FROG* in the left subtree with *gene1_FROG* (Fig. 2(b)) results in a tree that requires two duplications and no losses to explain.

4 Formal Macroevolutionary Reconstruction Problems

The problem of finding all most parsimonious trees with respect to a duplication/loss cost function alone, ignoring sequence information, can be formally stated as follows:

Macroevolutionary Phylogeny

Input: A rooted species tree, T_S with s leaves; a list of multiplicities $m_1 \dots m_s$, where m_l is the number of gene family members found in species l ; weights c_λ and c_δ .

Output: The set of all rooted gene trees $\{T_G\}$ with $\sum_{l=1}^s m_l$ leaves such that the **D/L Score** of T_G is minimal.

We provide a polynomial time algorithm to solve this problem, showing that, unlike most formulations of phylogeny reconstruction, which are NP-complete [4, 6, 7], an optimal solution to the **Macrophylogeny** problem can be obtained in polynomial time.

Second, we extend this result to obtain an algorithm to refine a tree built from sequence data. Note that the removal of any edge, e , in a tree bipartitions the set of leaf nodes. If the bootstrap value of e is low, it suggests that the evidence in the data for that bipartition is weak. It does not reflect on the certainty of the structure of any other part of the tree. We exploit this observation to obtain our refinement strategy. Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be trees with the same leaf set. We say T_1 and T_2 agree at edges $e_1 \in E_1$ and $e_2 \in E_2$ if the removal of e_1 from T_1 and e_2 from T_2 results in identical bipartitions of the leaves. Further, T agrees with T_1 at e_1 , if there is some edge in T that agrees with e_1 .

Definition 2. Let $T_G = (V, E)$ be a rooted tree and let $R \subseteq E$ be a set of robust edges (e.g., edges with bootstrap values above some threshold, θ). We define $\mathcal{T}_{G,R}$ to be the set of all rooted trees that agree with T_G at every edge in R and call $T \in \mathcal{T}_{G,R}$ a rearrangement tree of T_G . $\mathcal{T}_{G,R}^* \subset \mathcal{T}_{G,R}$ denotes the subset of trees with minimum **D/L Score**.

We now state the reconstruction problem for the more general, hybrid parsimony model. Note that in the case where $R = \emptyset$, $\mathcal{T}_{G,R}$ is simply the set of all trees with $|L(T_G)|$ leaves, where $L(T_G)$ is the leaf set of T_G , and the **Hybrid Micro-Macrophylogeny** problem reduces to the **Macrophylogeny** problem.

Hybrid Micro-Macrophylogeny

Input: A rooted gene tree, T_G with robust edges $R \subseteq E$; a rooted species tree, T_S ; weights c_λ and c_δ .

Output: $\mathcal{T}_{G,R}^*$

Due to several types of degeneracy there are potentially a large number of equally parsimonious trees in $\mathcal{T}_{G,R}^*$. We define a duplication/loss history to be a set of event, edge pairs, where an event is a duplication or a loss and the associated edge in T_S specifies when the event occurred. There may be more

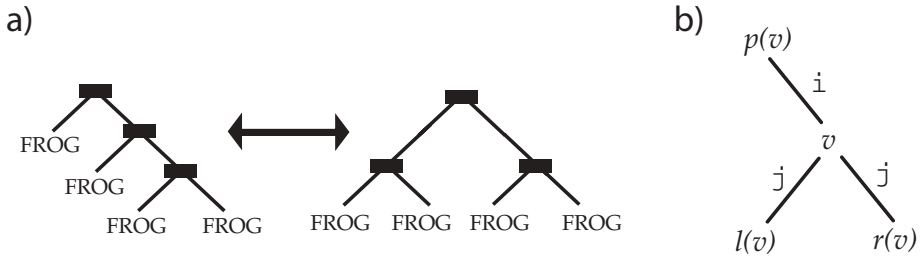


Fig. 3. (a) Two equally parsimonious arrangements of four FROG genes, each with three duplications. (b) Schematic showing the meaning of variables in the algorithm in Fig. 4

than one history with the same **D/L Score**. For example, if $c_\lambda = c_\delta = 1$, then the gene trees in Fig. 1(a) and Fig. 2(b) both have a **D/L Score** of two, although they have different histories (one duplication and one loss versus two duplications).

Additional degeneracy arises because the same history may correspond to more than one tree. This occurs when labels in the gene tree can be permuted without changing the score. For example, in Fig. 1(a), exchanging *gene1_HUMAN* and *gene2_HUMAN* results in a different tree for the same history. Multiple trees for the same history can also occur when subtrees with different topologies within a single species have the same number of duplication nodes. This is illustrated in Fig. 3(a).

Our algorithm generates one tree for each distinct history. We present all other trees for each history to the user through NOTUNG 2.0’s graphical user interface. Nodes that may be swapped without changing the **D/L Score** are highlighted, allowing the user to generate alternate minimum cost permutations using a point and click interface.

5 Reconstruction Algorithms

In this section, we first present a dynamic program for reconstructing a gene tree based on macroevolutionary considerations only. Since sequence evolution is not taken into account, the only information about the gene family required is the number of family members observed in each species in the species tree. Given the species tree, these multiplicities and positive weights c_λ and c_δ as inputs, our algorithm determines the minimum **D/L Score** and generates all most parsimonious histories.

In Section 5.2, we discuss how to incorporate these results in an algorithm for optimizing the reconciliation of a gene tree with weak edges. Each connected component of weak edges (*CCW*) in this tree is a binary, rooted tree whose leaves are strong subtrees in the original tree. An extension of the dynamic program presented in 5.1 is used to obtain all minimum cost rearrangements of the embedded rooted tree associated with each *CCW*. We then reinsert the

rearranged components in the tree using a theorem of [3] that proves that for cost functions with certain properties, including **D/L Score** used here, each *CCW* may be optimized independently.

5.1 Macrophylogeny

We now describe our algorithm for reconstructing all most parsimonious histories, where only macroevolutionary events are considered. Each history can be represented as a species tree, where each node is annotated with its multiplicity; that is, the number of copies extant in the species associated with that node. The multiplicity of the root must be one, while the multiplicities of the leaves, denoted $m_1 \dots m_s$, are specified in the input. An example of this is shown in Fig. 1(b).

Our dynamic program considers all possible multiplicities for each internal node. The variable i is the number of gene copies a node inherits from its parent, $p(v)$, and j is the number of gene copies a node sends to its children, $l(v)$ and $r(v)$, as shown in Fig. 3(b). The values of i and j range from one to $\hat{m} \leftarrow \max_{v \in L(T_S)} \{m_v\}$, where $L(T_S)$ is the set of leaves in T_S ¹. For each node, v , the dynamic program calculates the minimum **D/L Score** of the subtree rooted at v for all possible values of i and j . These values are stored in the minimum cost table, $cost_v^{min}[i] \leftarrow \min_{v,j} \{cost_v[i,j]\}$. The array $cost_v^{min}[i]$ is kept to enable quick lookup of the minimum cost given that v inherits i genes from its parent.

Pseudocode for this algorithm is shown in Fig. 4.

RECONSTRUCT, the main loop, calls the procedure ASCEND, which annotates the minimum cost tables for all nodes of T_S . To enumerate all alternate histories from these tables, RECONSTRUCT repeatedly calls DESCEND followed by CONSTRUCT. In each iteration, DESCEND extracts a new annotation that corresponds to a distinct optimal history. CONSTRUCT then builds a gene tree to represent the history marked by DESCEND. This gene tree is output and DESCEND is called again to find the next history. The procedure terminates when DESCEND returns *true* to indicate that all histories have been enumerated.

The enumeration scheme requires that each node keep track of those lowest cost entries in its cost table that have already been selected for alternate histories by DESCEND. Each node maintains persistent state variables including $v.dups$, the optimal number of duplicated genes in this species; $v.losses$, the optimal number of lost genes in this species; and $v.out$, the optimal number of genes for v to pass to its children. DESCEND also uses the variable $v.done$, which is *true* if v has no more histories to return for its current value of i . This value is reinitialized by calling DESCEND with a new i and the status variable *reset* set to *true*. Additional bookkeeping information is stored on each node for use by the NOTUNG user interface, to enable the user to generate alternate minimum cost permutations, if they exist, for each history displayed.

¹ To see this, note that if the number of gene copies in any vertex, v , of T_S is larger than \hat{m} there will have to be losses to reduce the gene count below v in T_S . As long as c_δ and c_λ are positive this will increase the **D/L Score** of the resulting gene tree. Therefore, it is never necessary to consider more than \hat{m} gene copies in any vertex of T_S .

```

RECONSTRUCT[ $T_S$ ,  $\{m_1 \dots m_s\}$ ]
  ASCEND[ $root(T_S)$ ];  $root(T_S).done \leftarrow true$ ;
  repeat {
     $root(T_S).done \leftarrow DESCEND[root(T_S), root(T_S).done, 1]$ ;
    ** reset construction counters **
    if (  $!root(T_S).done$  )  $T_G \leftarrow CONSTRUCT[root(T_S)]$ ; output( $T_G$ );
  } until(  $root(T_S).done$  )

```

```

ASCEND[ $v$ ]
if  $v$  is not a leaf: ASCEND[ $l(v)$ ]; ASCEND[ $r(v)$ ];
 $\forall i, j$  s.t.  $1 \leq i \leq \hat{m}, 0 \leq j \leq \hat{m}$ 
  if  $v$  is a leaf:  $cost_v^{min}[i] \leftarrow c_\delta * \max(m_v - i, 0) + c_\lambda * \max(i - m_v, 0)$ ;
  if  $v$  is not a leaf:
     $cost_v[i, j] \leftarrow c_\delta * \max(j - i, 0) + c_\lambda * \max(i - j, 0) + cost_{l(v)}^{min}[j] + cost_{r(v)}^{min}[j]$ ;
 $\forall i$   $cost_v^{min}[i] \leftarrow \min_{\forall j} \{cost_v[i, j]\}$ ;

```

```

DESCEND[ $v$ , reset,  $i$ ]
if  $v$  is a leaf:
   $v.losses \leftarrow \max((i - m_v), 0)$ ;  $v.dups \leftarrow \max((m_v - i), 0)$ ;
   $v.out \leftarrow 0$ ; return false;
if (!reset)
  if (  $v.done$  ) return true ** no more histories. **
  if (  $!(l(v).done \leftarrow DESCEND[l(v), false, v.out])$  ) return false;
  if (  $!(r(v).done \leftarrow DESCEND[r(v), false, v.out])$  ) return false;
  ** go on to next optimal out degree **
  repeat {  $v.out++$  } until (  $cost_v[i, v.out] == cost_v^{min}[i]$  OR  $v.out > \hat{m}$  );
  if(  $v.out > \hat{m}$  ) return  $v.done \leftarrow true$ ;
else ** reset == true **
   $v.out \leftarrow 0$ ; repeat {  $v.out++$ ; } until (  $cost_v[i, v.out] == cost_v^{min}[i]$  );
 $l(v).done \leftarrow DESCEND[l(v), true, v.out]$ ;  $r(v).done \leftarrow DESCEND[r(v), true, v.out]$ ;
 $v.losses \leftarrow \max((i - v.out), 0)$ ;  $v.dups \leftarrow \max((v.out - i), 0)$ ;
return false;

```

```

CONSTRUCT[ $s$ ]
 $g \leftarrow$  new gene node;  $g.species \leftarrow s$ 
if (  $s.currDup < s.dups$  )
   $s.currDup++$ ;  $l(g) \leftarrow CONSTRUCT[s]$ ;  $r(g) \leftarrow CONSTRUCT[s]$ ;
else if (  $s.currLoss < s.losses$  )
   $s.currLoss++$ ;
else if (  $s.currSpec < s.out$  )
   $s.currSpec++$ ;
  if  $s$  is not a leaf:  $l(g) \leftarrow CONSTRUCT[l(s)]$ ;  $r(g) \leftarrow CONSTRUCT[r(s)]$ ;
return  $g$ ;

```

Fig. 4. The Macrophylogeny Reconstruction Algorithm

Lemma 1. *The time required for RECONSTRUCT to find a single optimal history is $O(n\hat{m}^2)$, where n is the number of nodes in the species tree and \hat{m} is the maximum number of genes drawn from any species. The time complexity for reporting k optimal histories is $O(n\hat{m}(k + \hat{m}))$.*

Proof. RECONSTRUCT calls ASCEND once. The leaves of the species tree can be annotated with multiplicities in $O(n)$ time. The main computational cost in ASCEND is calculating the cost matrix in the internal species nodes, and the minimum cost vector in the leaf nodes. Each cost matrix is of size $\hat{m}(\hat{m} + 1)$, and each minimum cost vector is of size \hat{m} . Thus, the time complexity of ASCEND is $O(n\hat{m}^2)$.

DESCEND and CONSTRUCT are called once per optimal history. The time required for DESCEND to find the value of j that minimizes the cost at given a node for a given value of i takes time $O(\hat{m})$. Each node in the species tree will be visited at most two times, so the total complexity for DESCEND is $O(n\hat{m})$. CONSTRUCT inserts duplication and loss nodes in the new tree, which can number in total no more than \hat{m} per node in T_S . Hence, the total complexity for CONSTRUCT is $O(n\hat{m})$. \square

Lemma 2. RECONSTRUCT finds all histories with minimum **D/L Score**.

Proof. We first show that ASCEND correctly calculates $cost_v^{min}[i]$, the minimum cost, for all nodes v and all possible number of inherited genes, i . We prove this by induction on the root of every subtree of T_S . Note that the algorithm is trivially correct for all trees containing only one node, independent of i . In a species tree with more than one node, either a duplication or a loss can occur at the same node in the species tree, but never both, as such a history will never be optimal. For a given i and j , the duplication/loss cost for a species node, v , is the optimal cost of the left and right subtrees, given that they inherit j gene copies from v , plus the cost of losses ($c_\lambda \cdot \max(i - j, 0)$) or duplications ($c_\delta \cdot \max(j - i, 0)$), whichever event occurs at this node. The minimum cost is then the lowest value, taken over all j from 1 to \hat{m} . Then, by the induction hypothesis, ASCEND calculates the minimum **D/L Score** for each of subtree, for all i .

We next show by induction that a call to DESCEND will then always result in an optimal history, given that ASCEND has correctly annotated the species tree. For a single node, DESCEND will correctly select each optimal history calculated by ASCEND. For each internal node, v , in a non-trivial tree, for a given i , and for every j which minimizes the cost of the subtree, DESCEND requests optimal histories from each child of v until they both report done. Thus, DESCEND considers and reports every combination of minimum cost histories from every node in a tree. \square

5.2 Hybrid Micro-Macrophylogeny

In order to incorporate the algorithm in Fig. 4 in a framework for optimizing a gene tree T_G with both strong and weak edges, we must make a few

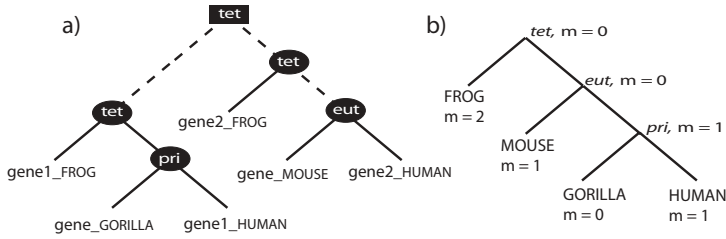


Fig. 5. (a) A gene tree with a weakly connected component, shown here by dotted edges. (b) Annotated tree for the species in (a). Note that the internal species node *pri* (primate) has a multiplicity, while GORILLA does not

modifications. A preprocessing step is needed to extract the rooted tree corresponding to each *CCW* and record its multiplicities in the species tree. Note that the leaves of a *CCW* may be strong subtrees in T_G . Thus, unlike the algorithm in Fig. 4, both the internal nodes and leaves of the species tree are annotated with multiplicities. An example of a gene tree with a strong subtree and the corresponding annotated species tree can be seen in Fig. 5. Additional bookkeeping is required in order to reconnect the reconstructed *CCW*s to strong parts of the tree. The complete code for this algorithm is available at (<http://www.cs.cmu.edu/~durand/Lab/Notung/>).

6 Experimental Results

The algorithms described in the previous sections have been implemented in a Java program called NOTUNG 2.0. Given a rooted gene family tree, a rooted species tree, a bootstrap threshold, θ , and costs c_δ and c_λ , NOTUNG 2.0 computes all optimal rearrangement histories and presents one tree for each optimal history. The user can generate all other trees for the same history using a point and click interface. Input trees are represented in Newick format [18] and must be binary. The graphical user interface was constructed using the tree visualization library provided by ATV (version 1.92) [30].

In a previous study [3], we developed a test set of thirteen trees discussed in three recent articles on large scale duplication [15,24,25]. For each rooted tree in the test set, we compared the results automatically generated by NOTUNG 2.0 with those of the original authors. The duplication histories generated were consistent with the analyses of the authors of the original papers for all trees considered.

In addition, we used NOTUNG 2.0 to analyze two large data sets, summarized in Table 1. The first data set [5] includes ATP-binding cassette (ABC) transporter sequences from *D. discoideum*, *A. thaliana* and *S. cerevisiae* and a number of *Plasmodia* species, provided by Roxana Cintron, Dr. Adelfa Serrano and colleagues (University of Puerto Rico). The second, provided by Dr. Hugh Nicholas (Pittsburgh Supercomputing Center) is derived from the Glutathione-

Table 1. Duplications and losses in the ABC and GST trees before and after rearrangement by NOTUNG 2.0, with bootstrap thresholds of 50% and 90% (GST tree only) and weights of $c_\delta = 1.5$ and $c_\delta = 1$. Note that two distinct histories were obtained for $\theta = 90\%$ for the GST tree

	leaves	D	L	θ	D	L
ABC	350	304	165	50%	271	61
GST	121	67	250	90%	51	58
GST				90%	49	61
GST				50%	55	164

S Transferases [27], a superfamily of detoxification enzymes. Initial sequence trees for both data sets were constructed using Neighbor Joining [26]. Bootstrap replicates were obtained using the SEQBOOT program from Felsenstein’s Phylip package (v. 3.6.1) available at the Pittsburgh Supercomputing Center. The divergence in both families was substantial resulting in a large number of edges with low bootstrap values. NOTUNG 2.0 was used to count the number of duplications and losses in the resulting sequence trees and then to rearrange the trees. The results given in Table 1 show that macroevolutionary rearrangement substantially reduces the number of duplications and losses needed to explain the data.

The trees analyzed here provide a concrete example of Notung’s utility in functional as well as evolutionary applications. Gene duplication followed by functional differentiation is a mode of adaptation to environmental change. Identification of recent duplications in specific lineages indicates genes that are sites of rapid change and, when combined with ecological and biochemical data, the environmental forces to which they are responding. This information can be used to plan additional experimental studies, suggest strategies for circumventing drug and pesticide resistance in parasites, identify potential detoxification enzymes for use in bioremediation and design breeding programs to enhance pest resistance in cash crops.

Drug and toxin resistance in the ATP-binding Cassette (ABC) transporter family in *Arabidopsis thaliana* provides a concrete example of this process. Analysis and visualization of ABC tree [5] with NOTUNG 2.0 revealed a large number of recently duplicated genes in the Multi-Drug Resistance subfamily, suggesting a pattern of recent, lineage-specific adaptation, possibly in response to new pesticides. These genes are potential industrial targets for bioremediation.

Acknowledgments

We thank R. Cintron, M. Farach-Colton, R. Hoberman, D. Miranker, H. B. Nicholas, Jr. and R. Ravi for helpful discussions, R. Ravi for his contributions to algorithm development and R. Cintron, H. B. Nicholas, Jr. and A. E. Serano for providing the ABC transporter and GST data sets. D.D. and B.V were supported by NIH grant 1 K22 HG 02451-01 and a David and Lucille Packard

Foundation fellowship. B.V.H. was supported by a Merck Computational Biology and Chemistry Program Graduate Fellowship from the Merck Company Foundation.

References

1. Lars Arvestad, Ann-Charlotte Berglund, Jens Lagergren, and Bengt Sennblad. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, 19 Suppl 1:i7–15, 2003.
2. Lars Arvestad, Ann-Charlotte Berglund, Jens Lagergren, and Bengt Sennblad. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology*, pages 326–335. ACM Press, 2004.
3. K. Chen, D. Durand, and M. Farach-Colton. Notung: A program for dating gene duplications and optimizing gene family trees. *Journal of Computational Biology*, 7(3/4):429–447, 2000.
4. B. Chor and T. Tuller. Maximum likelihood of evolutionary trees is hard. In Satoru Miyano, editor, *Proceedings of the Ninth Annual International Conference on Computational Molecular Biology (RECOMB 2005)*, Lecture Notes in Bioinformatics. Springer Verlag, 2005.
5. R. Cintron, H. B. Nicholas Jr., I. Ferrer, R. Gonzalez, B. Vernot, D. Durand, and A. E. Serrano. The ABC superfamily: Evolutionary implications for drug resistance in *exititPlasmodia*. In *Functional Genomics and Bioinformatics Approaches to Infectious Disease Research*. American Society for Microbiology, 2004. Abstract book.
6. W. H. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bull Math Biol*, 49(4):461–7, 1987.
7. W. H. E. Day, D. S. Johnson, and D. Sankoff. The computational complexity of inferring rooted phylogenies by parsimony. *Math Biosci*, 81(33):33–42, 1986.
8. Bradley Efron and Gail Gong. A leisurely look at the bootstrap, jackknife, and cross-validation. *The American Statistician*, 37(1):36–48, 1983.
9. O. Eulenstein, B. Mirkin, and M. Vingron. Comparison of a annotating duplication, tree mapping, and copying as methods to compare gene trees with species trees. *Mathematical Hierarchies and Biology, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 37:71–93, 1996.
10. O. Eulenstein, B. Mirkin, and M. Vingron. Duplication-based measures of difference between gene and species trees. *Journal of Computational Biology*, 5:135–148, 1998.
11. Joseph Felsenstein. *Inferring Phylogenies*, chapter 29, page 514. Sinauer, 2003.
12. M. Goodman, J. Czelusniak, G.W. Moore, A.E. Romero-Herrera, and G Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst Zool*, 28:132–163, 1979.
13. R. Guigo, I. Muchnik, and T.F. Smith. Reconstruction of ancient phylogenies. *Molecular Phylogenetics and Evolution*, 6:189–213, 1996.
14. M. T. Hallett and J. Lagergren. New algorithms for the duplication-loss model. In *RECOMB2000, Fourth Annual International Conference on Computational Molecular Biology*, 2000.

15. A. L. Hughes. Phylogenetic tests of the hypothesis of block duplication of homologous genes on human chromosomes 6, 9, and 1. *MBE*, 15(7):854–70, 1998.
16. B. Ma, M. Li, and L. Zhang. From gene trees to species trees. *SIAM J. on Comput.*, 2000.
17. B. Mirkin, I. Muchnik, and T.F. Smith. A biologically consistent model for comparing molecular phylogenies. *Journal of Computational Biology*, 2:493–507, 1995.
18. The Newick tree format, 1986. <http://evolution.genetics.washington.edu/phylip/newicktree.html>
19. R. D. Page. GeneTree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9):819–20, 1998.
20. R. D. M. Page. Maps between trees and cladistic analysis of historical associations among genes, organisms and areas. *Syst Zool*, 43:58–77, 1994.
21. R. D. M. Page and M. A. Charleston. Reconciled trees and incongruent gene and species trees. *Mathematical Heirarchies and Biology, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 37:57–70, 1996.
22. R. D. M. Page and M. A. Charleston. From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem. *Molecular Phylogenetics and Evolution*, 7:231–240, 1997.
23. R. D. M. Page and M. A. Charleston. Trees within trees: phylogeny and historical associations. *Trends in Ecology and Evolution*, 13(9):356–359, 1998.
24. M.-J. Pebusque, F. Coulier, D. Birnbaum, and P. Pontarotti. Ancient large-scale genome duplications: phylogenetic and linkage analyses shed light on chordate genome evolution. *MBE*, 15(9):1145–59, 1998.
25. I. Ruvinsky and L. M. Silver. Newly identified paralogous groups on mouse chromosomes 5 and 11 reveal the age of a t-box cluster duplication. *Genomics*, 40:262–266, 1997.
26. N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.
27. D. Sheehan, G. Meade, V. M. Foley, and C. A. Dowd. Structure, function and evolution of glutathione transferases: implications for classification of non-mammalian members of an ancient enzyme superfamily. *Biochem J*, 360(Pt 1):1–16, Nov 2001.
28. U. Stege. Gene trees and species trees: The gene-duplication problem is fixed-parameter tractable. In *Proceedings of the 6th International Workshop on Algorithms and Data Structures (WADS'99)*, 1999.
29. L. Zhang. On a Mirkin–Muchnik–Smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4:177–188, 1997.
30. C. M. Zmasek and S. R. Eddy. ATV: display and manipulation of annotated phylogenetic trees. *Bioinformatics*, 17(4):383–4, Apr 2001.
31. C. M. Zmasek and S. R. Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17(9):821–8, Sep 2001.