A Hybrid Micro-Macroevolutionary Approach to Gene Tree Reconstruction

Dannie Durand,

Departments of Biological Sciences and Computer Science,

Carnegie Mellon University, Pittsburgh, PA, USA. durand@cmu.edu

Bjarni V. Halldórsson

deCODE Genetics, Reykjavik, Iceland, bjarni.halldorsson@decode.is

Benjamin Vernot*

Department of Biological Sciences , Carnegie Mellon University,

Pittsburgh, PA, 15213, USA. bvernot@cs.cmu.edu

November 18, 2005

*phone: (412) 268-3199, fax: (412) 268-7129

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 delay 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. Our hybrid algorithm can also be used to resolve non-binary nodes in a multifurcating gene tree.

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. The NOTUNG 2.0 graphical user interface 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. It also offers a command line option that enables high-throughput analysis of a large number of trees.

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 (Goodman et al., 1979), 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 a species tree and 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. An extension of this algorithm can also be applied to non-binary species trees. In contrast to most phylogeny reconstruction problems, which are NP-complete (Chor and Tuller, 2005; Day et al., 1986; Day, 1987), our results show that macroevolutionary parsimony can be solved in polynomial time per output tree.

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. This hybrid approach can also be used to resolve non-binary nodes in a multifurcating tree.

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. NOTUNG 2.0 has the following features:

- a polynomial delay algorithm for generating all most parsimonious trees with respect to a weighted gene duplication and loss cost.

- a graphical interface, with features especially designed for analysis of gene duplications, including visualization of duplicated nodes, lost leaves and/or subtrees, the ability to browse the space of all optimal hybrid trees, annotation of subfamilies, color highlighting of rootable edges and generating of Portable Network Graphics image files for publication purposes.

- a command line interface for high-throughput analysis of large numbers of trees.

NOTUNG 2.0 I/O uses Newick Taxonomy tree format. The command line version will also generate text summaries of duplication histories, which can then be parsed by scripts. The NOTUNG 2.0 executable is available for free, public distribution (*http://www.cs.cmu.edu/∼durand/Notung/*).

The rest of this paper is organized as follows: In Section 2, we discuss related 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 (Goodman et al., 1979). This concept was further developed and formalized by Guigo et al. (1996); Hallett and Lagergren (2000); Ma et al. (2000); Mirkin et al. (1995); Page (1994); Page and Charleston (1996); Stege (1999); Zhang (1997). Formally, given a set of rooted gene trees, the problem is to find the species tree that optimizes an evaluation criterion. Several parsimony-based optimality criteria have been proposed (see Eulenstein et al. (1996, 1998) for a comparative survey). The problem of finding an optimal species tree is NP-hard (Ma et al., 2000) for the optimality criteria considered so far. Several authors have pointed out that it is difficult to distinguish true gene loss from genes that have not yet been sequenced and discuss approaches to distinguishing true losses from apparent losses in the cost function (Goodman et al., 1979; Mirkin et al., 1995; Dufayard et al., 2005). More recently, reconciliation algorithms that also take horizontal gene transfer into account have been presented (Gorecki, 2004; Hallett and Lagergren, 2001; Hallett et al., 2004).

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. Component and GeneTree are software packages that will compute and display duplication histories for rooted gene trees and infer species trees (Page and Charleston, 1997; Page, 1998). The Forester package (Zmasek and Eddy, 2001b,a) performs some reconciliation functions and provides a tree visualization tool. FamFetch (Dufayard et al., 2005) provides a graphical interface for specifying gene tree templates which can be used to search a database of reconciled trees.

All of these approaches are based on reconciliation of a gene tree with a species tree, a procedure that can be used to infer duplications and losses and estimate the times at which they occurred. We review reconciliation here. Let T_{G} be a gene tree and T_{S} be a species tree of taxa from which the gene sequences were sampled. (If T_{S} contains additional species, these must be pruned from the species tree before proceeding.) The identification of duplication nodes requires constructing a mapping, M, from every node, v, in T_{G} to a target node, $M(v)$, 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; that is, $M(v) = lca(M(l(v)), M(r(v)))$. 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.

Figure 1

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 $M(r(v)) = M(l(v))$ or $M(r(v))$ is an ancestor of $M(l(v))$ or vice versa), 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 duplications is obtained by simply counting the duplication nodes determined in the above procedure. The total number of losses, can be computed by summing the gene losses over all edges in T_G. The loss associated with the edge $e = (p(v), v)$ is given by $(\delta(v) - \delta(p(v)) - 1) + \text{IsDup}(p(v))$, where $p(v)$ is the parent of v, $\delta(v)$ returns the depth of the target node, $M(v)$, in the pruned species tree and $\text{IsDup}(v) = 1$, if v is a duplication node, and zero, otherwise. The first term in this equation counts gene losses associated with speciation nodes. The labels of a speciation node and its child should differ by one if no loss has occurred and will increase by one with each gene loss. In contrast, if no gene loss has occurred on the edge immediately below a duplication node, the labels of a duplication node and its child should be identical. Thus, if $p(v)$ is a duplication node, the first term will be off by one. The second term corrects for this case. Note that if a set of lost genes forms a monophyletic clade, we assume that a single loss occurred in their common ancestor and rather than many losses at the leaves. For example, Fig. 2(a) shows one loss, in eut, rather than two losses, one in MOUSE and one in HUMAN.

The cost of duplications and losses can be expressed by the following optimization function:

Definition 2.1. *The **D/L Score** of a gene tree is $c_\lambda L + c_\delta D$, the weighted sum of the number of duplications, D, and the number of losses, L, in the tree.*

3 A Taxonomy of Models

The goal of phylogeny reconstruction is to determine the hypothesis, expressed as an evolutionary tree, that best explains the data with respect to a model of evolutionary change. Given a set of sequences from a gene family, which may include both paralogous sequences from the same species and orthologous sequences from different species, a gene tree can be reconstructed according to various evolutionary models.

Microevolutionary model: *Find the tree that best explains the data with respect to a model of sequence evolution only.* In practice, most gene trees are constructed based on a model of sequence evolution alone, as this approach requires a less complex model, is less computationally intensive, and may be achieved with one of the many tools currently available for sequence-based phylogeny reconstruction. However, information about macroevolutionary events is not incorporated in this approach. In many cases, workers subsequently infer the macroevolutionary history implied by the sequence-based tree, either by inspection or by using a software tool for exploratory analysis (e.g. Chen et al., 2000; Page and Charleston, 1998; Zmasek and Eddy, 2001b,a), thereby treating the gene tree as though it were data rather than a hypothesis. If bootstrap values (Efron and Gong, 1983) indicate weak support for some edges in the tree, some practitioners discuss alternate hypotheses motivated by *post hoc* macroevolutionary considerations, partially mitigating this problem.

Macroevolutionary model: *Find the tree that best explains the data with respect to a model of duplications and losses with no consideration of sequence 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 can be usefully applied in the solution of several other, more biologically important problems. In addition, the fact that this problem admits a polynomial delay solution is of theoretical interest, since most formulations of phylogeny reconstruction are NP-complete (Chor and Tuller, 2005; Day, 1987; Day et al., 1986). 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 (Arvestad et al., 2003; Felsenstein, 2003, p. 514).

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 et al. (1979) 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 both nucleotide replacement and gene duplication and loss[1]. 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. (2004) 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, an initial tree is constructed based on sequence data alone, with edge weights representing the support for each taxon bipartition in the tree. Subsequently, edges with weak support in the sequence data (i.e., with edge weights below a specified threshold) are rearranged to optimize the number of duplications and losses needed to explain the tree, while the structure of the tree at edges with strong support is preserved. Note that the removal of any edge, e, in a tree bipartitions the set of leaf nodes. If

[1] Goodman et al. (1979) actually refer to gene duplication and gene expression events. In 1979, they were working with amino acid sequences and had to consider the possibility that missing sequences might be encoded by genes that were present in the genome but not expressed.

the support for 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. Quantitative measures for assessing support for taxon bipartitions include bootstrap values (Efron and Gong, 1983), posterior probabilities (e.g., Ronquist and Huelsenbeck, 2003) or edge lengths.

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, the hybrid approach 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.

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 ($\underline{\text{eut}}$), 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. Figure 2

4 Formal Macroevolutionary Reconstruction Problems

The problem of finding all most parsimonious trees with respect to duplications and losses alone, ignoring sequence information, can be formally stated as follows:

Macroevolutionary phylogeny

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

Output: The set of all rooted, binary gene trees $\{T_{\mathrm{G}}\}$ with m_s leaves drawn from each species s and such that the **D/L Score** of T_{G} is minimal.

In Section 5.1 we provide a dynamic program to solve this problem, showing that, unlike most formulations of phylogeny reconstruction, which are NP-complete (Chor and Tuller, 2005; Day, 1987; Day et al., 1986), an optimal solution to the **Macrophylogeny** problem can be obtained in polynomial time per output tree. An extension to this algorithm for non-binary species trees is given in Section 5.2.

Next, we extend this result to obtain an algorithm to refine a tree built from sequence data. Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be trees with the same leaf set; that is, $L(T_1) = L(T_2)$, where $L(T)$ is the leaf set of T. We say T_2 agrees with T_1 at e_1, if there is some edge, $e_2 \in E_2$, that generates the same taxon bipartition as e_1.

Definition 4.1. *Let $T_G = (V, E)$ be a rooted tree and let $R \subseteq E$ be a set of robust edges. A tree T is a rearrangement tree of T_G if $L(T) = L(T_G)$ and T agrees with T_G at every edge in R. Further, we define $\mathcal{T}_{G,R}$ to be the set of all rooted, binary rearrangement trees of T_G and denote by $\mathcal{T}^*_{G,R}$ the subset of$\mathcal{T}_{G,R}$ with minimum **D/L Score**.*

We now state the reconstruction problem for the more general, hybrid parsimony model.

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

Note that in the case where $R = \emptyset$, $\mathcal{T}_{G,R}$ is simply the set of all trees with $|L(T_{\mathrm{G}})|$ leaves, and the **Hybrid Micro-Macrophylogeny** problem reduces to the **Macrophylogeny** problem.

An algorithm to solve the **Hybrid Micro-Macrophylogeny** problem can be used to incorporate macroevolutionary information in phylogeny reconstruction when the sequence data does not strongly support a single topology. In this case, R is the set of edges with weights above some threshold, θ. The same approach can also be used to resolve a multifurcating gene tree, when the sequence data is not sufficient to obtain a complete binary resolution of the historical relationships between all taxa. This is achieved by replacing every k-ary node with an embedded binary tree with k leaves, such that the **D/L Score** of the embedded subtree is minimal with respect to a given species tree.

Let $T_M = (V_M, E_M)$ and $T_B = (V_B, E_B)$ be trees, where T_M is multifurcating, T_B is binary and $L(T_M) = L(T_B)$. We say T_B agrees with T_M if every taxon bipartition in T_M is also found in T_B. This relationship is not symmetric since T_M must have fewer edges than T_B. There will always be at least one bipartition in T_B that is not present in T_M. Let $T_{\mathrm{G}} = (V, E)$ be a multifurcating gene tree in which all edges are robust ($R = E$). Then, $\mathcal{T}^*_{G,R}$, defined above is exactly the set of binary trees that optimally resolve T_{G}.

In both binary and multifurcating tree refinement, there are potentially a large number of equally parsimonious trees in $\mathcal{T}^*_{G,R}$ due to several types of degeneracy.

Definition 4.2. *An* <u>*event history*</u> *is 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 than one event 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 event 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 event history. Multiple trees for the same event 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. Finally, when $c_\lambda = 0$, additional degeneracy can arise because multiple optimal event histories may correspond to the same tree.

Figure 3

NOTUNG 2.0 generates one tree for each distinct event history. We present all other trees for each event 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 binary 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 gene family members observed in each species. Given a binary species tree, species 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. We extend this algorithm to handle non-binary species trees in Section 5.2.

In Section 5.3, we discuss how to incorporate these results in an algorithm for optimizing the reconciliation of a binary 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 Section 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 Chen et al. (2000) 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 event histories, where only macroevolutionary events are considered. Each event history can be represented as a species tree, where each node is annotated with the number of paralogs found at that node in the history under consideration. This number must be one at the root. The number of gene family members in each leaf species is specified in the input. We refer to these as multiplicities. Fig. 1(b) shows a species tree with leaves annotated with multiplicities.

Our dynamic program considers all possible histories by enumerating all possible assignments of gene copies to internal nodes in the species tree. For a given node v, the variable i is the number of paralogous copies extant in its parent, $p(v)$, and j is the number of copies in v. If $i \neq j$, then one or more duplications or losses occured on the edge from $p(v)$ to v. We also define $l(v)$ and $r(v)$ as the two children of v. These relationships are shown in Fig. 4. For each node v, the dynamic program computes the minimum **D/L Score** of the subtree rooted at v for all possible values of i and j. It is sufficient to consider only values of i and j in the range from one to $\hat{m} \leftarrow \max_{\forall l \in L(T_S)}\{m_l\}$[2]. These values are stored in the cost table, $cost_v[i,j]$. The array $cost_v^{min}[i] \leftarrow \min_{\forall j}\{cost_v[i,j]\}$ is also stored to enable quick lookup of the minimum cost given that v inherits i genes from its parent.

Figure 4

Pseudocode for this algorithm is shown in Algorithm 1. RECONSTRUCT, the main loop, calls the procedure

[2] To see this, note that if the number of gene copies in any vertex, v, of T_S is larger than $\hat{m}$, then additional losses will be required in the subtree rooted at v in order to reduce the gene count to the multiplicities on the leaves of T_S. As we require c_δ and c_λ to be 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.

ASCEND, which annotates the minimum cost tables for all nodes of T_S. To generate all alternate histories from these tables, RECONSTRUCT repeatedly calls DESCEND followed by CONSTRUCT. In each iteration, DESCEND selects 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 $false$ to indicate that all histories have been generated.

To generate all optimal histories, each node must keep track of those lowest cost entries in its cost table that have already been selected for alternate histories by DESCEND. This is done by considering all possible values of i at each node that could result in an optimal score. Each node maintains 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. Each call to DESCEND attempts to set these state variables to obtain a new optimal history, and returns $true$ if it was succesfull. This success is also recorded in the state variable $v.changed$. Additional bookkeeping information is stored on each node for use by the NOTUNG 2.0 user interface, to enable the user to generate alternate minimum cost gene trees, if they exist, for each history displayed.

Algorithm 1.

```
RECONSTRUCT[$T_S$, $\{m_1 \ldots m_s\}$]
    ASCEND[$root(T_S)$];
    $reset \leftarrow true$;
    while ($true$) {
        $root(T_S).changed \leftarrow$ DESCEND[$root(T_S)$, $reset$, 1];
        if( $!root(T_S).changed$ )
            break;
        $T_G \leftarrow$ CONSTRUCT[$root(T_S)$];
        output($T_G$);
        $reset \leftarrow false$;
        ** reset construction counters **
    }
```

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

```
DESCEND[v, $reset$, i]
if $v$ is a leaf:
    $v.out \leftarrow 0$;
    $v.losses \leftarrow \max((i - m_v), 0)$;
    $v.dups \leftarrow \max((m_v - i), 0)$;
    return $reset$;
if (!$reset$)
    ** check each child - if either has another solution, return true **
    $l(v).changed \leftarrow$ DESCEND$[l(v), false, v.out]$
    if ( $l(v).changed$ )
        return $true$;
    $r(v).changed \leftarrow$ DESCEND$[r(v), false, v.out]$
    if ( $r(v).changed$ )
        $l(v).changed \leftarrow$ DESCEND$[l(v), true, v.out]$
        return $true$;
    ** neither child has another solution, go on to next optimal value of $v.out$ **
    repeat { $v.out++$ } until ( $cost_v[i, v.out] == cost_v^{min}[i]$ OR $v.out > \hat{m}$ );
    if( $v.out > \hat{m}$ )
        return $false$;
```

```
else ** reset to first optimal value of $v.out$ **
    $v.out \leftarrow 0$;
    repeat { $v.out++$; } until ( $cost_v[i, v.out] == cost_v^{min}[i]$ );
** we are computing values for a new $v.out$ **
$l(v).changed \leftarrow$ DESCEND$[l(v), true, v.out]$;
$r(v).changed \leftarrow$ DESCEND$[r(v), true, v.out]$;
$v.losses \leftarrow \max((i - v.out), 0)$;
$v.dups \leftarrow \max((v.out - i), 0)$;
return $true$;
```

```
CONSTRUCT[s]
$g \leftarrow newgenenode$; $g.species \leftarrow s$;
if ($s.currDup < s.dups$)
    $s.currDup++$;
    $l(g) \leftarrow$ CONSTRUCT[$s$]; $r(g) \leftarrow$ CONSTRUCT[$s$];
    ** mark $g$ as a duplication node **
else if ($s.currLoss < s.losses$)
    $s.currLoss++$;
    ** mark $g$ as a loss node **
else if ($s.currSpec < s.out$)
    $s.currSpec++$;
    $l(g) \leftarrow$ CONSTRUCT[$l(s)$]; $r(g) \leftarrow$ CONSTRUCT[$r(s)$];
    ** mark $g$ as a speciation node **
return $g$;
```

Lemma 5.1. *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 paralogous copies 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 $O(n\hat{m})$.

In order to report k optimal histories, ASCEND must be called once, while DESCEND and CONSTRUCT are called k times each. Therefore, the time complexity for reporting k optimal histories is $O(n\hat{m}(k + \hat{m}))$. □

Lemma 5.2. RECONSTRUCT *finds all histories with minimum* ***D/L Score****.*

Proof. RECONSTRUCT repeatedly calls DESCEND to get all optimal histories with minimal **D/L Score**. Remember that for a given node v, the value i is the number of copies passed to v from its parent, and j is the number of copies present in v after duplication or loss events at v have occurred. The cost tables generated by ASCEND are used in DESCEND, and guarantee that only optimal values of i and j are considered. DESCEND is first called in RECONSTRUCT with $i = 1$. We prove by induction on the structure of T_{S} that repeated calls to DESCEND with v and i will generate all optimal histories rooted at v, given that v receives i paralogous copies from its parent. Subsequent calls to DESCEND will report that no more histories exist. We show that this is true for the base case, a leaf node of T_{S}. We then show that if this is true for the left and right children of an internal node v, it is also true for v.

Base case: For a leaf node v, there is a single optimal history for every value of i. The first call to DESCEND will return that optimal history. Subsequent calls to DESCEND will report that no more histories exist.

Inductive step: Let v be an internal node. In repeated calls to DESCEND with v and i, DESCEND examines all values of j that will result in optimal histories. For each value of j, it repeatedly calls DESCEND with $l(v)$ and j and DESCEND with $r(v)$ and j, using an enumeration scheme that ensures all combinations of optimal histories rooted at $r(v)$ and $l(v)$ will be generated. When all optimal values of j have been exhausted, and all optimal histories have been generated, DESCEND reports that it is done for (v, i).

□

5.2 The Macro Algorithm for Non-Binary Species Trees

The Macrophylogeny reconstruction algorithm described above assumes a binary species tree and generates one or more binary gene trees. There are a number of reasons, however, why a species tree might not be binary. There may not be enough data to determine a unique binary history for a set of species, or a non-binary speciation event may have actually occurred. Algorithm 1 can be modified to handle non-binary species trees, where any non-binary nodes in the species tree are considered to be actual non-binary speciation events. This new algorithm will generate non-binary gene trees.

Let $n(v)$ be the number of children of v, and $c_k(v)$ be the kth child of v. In order to accommodate non-binary species trees, we must modify ASCEND to visit all $n(v)$ children of v, and change the computation of the **D/L Score** to include the score of all subtrees of v. In addition, for each node v, DESCEND must search through all $n(v)$ children for a new solution, and reset all previous children when a new solution is found. Algorithm 2 shows the modifications to ASCEND and DESCEND. The RECONSTRUCT routine remains unchanged. CONSTRUCT is modified to visit all children by replacing Lines 4 and 11 with:

$$\forall k = 1 \ldots n(v) \;\; c_k(g) \leftarrow \text{CONSTRUCT}[c_k(s)]$$

Algorithm 2.

```
ASCEND[$v$]
    if $v$ is a leaf
        $\forall i \;\; s.t. \;\; 1 \leq i \leq \hat{m}$
            $cost_v^{min}[i] \leftarrow c_\delta * \max(m_v - i, 0) + c_\lambda * \max(i - m_v, 0)$;
    if $v$ is not a leaf
        $\forall k = 1 \ldots n(v) \;\; \text{ASCEND}[c_k(v)]$
        $\forall i, j \;\; s.t. \;\; 1 \leq i \leq \hat{m}, \;\; 1 \leq j \leq \hat{m}$
            $cost_v[i,j] \leftarrow c_\delta * \max(j - i, 0) + c_\lambda * \max(i - j, 0) + \sum_{k=1}^{n(v)} cost_{c_k(v)}^{min}[j]$
        $\forall i \;\; s.t. \;\; 1 \leq i \leq \hat{m}$
            $cost_v^{min}[i] \leftarrow \min_{\forall j}\{cost_v[i,j]\}$;

DESCEND[v, $reset$, i]
    if $v$ is a leaf:
        $v.out \leftarrow 0$;
        $v.losses \leftarrow \max((i - m_v), 0)$;
        $v.dups \leftarrow \max((m_v - i), 0)$;
        return $reset$;
    if (!$reset$)
```

```
        ** if any child has another solution, reset previous children and return true **
        $\forall k = 1 \ldots n(v)$
            $c_k(v).changed \leftarrow$ $\textsc{Descend}[c_k(v), false, v.out]$
            if ( $c_k(v).changed$ && $k > 1$ )
                $\forall j = 1 \ldots k - 1$
                    $c_j(v).changed \leftarrow$ $\textsc{Descend}[c_j(v), true, v.out]$
                return $true$;
        ** no child has another solution, go on to next optimal value of $v.out$ **
        repeat $\{$ $v.out++$ $\}$ until ( $cost_v[i, v.out] == cost_v^{min}[i]$ OR $v.out > \hat{m}$ );
        if( $v.out > \hat{m}$ )
            return $false$;
    else ** reset to first optimal value of $v.out$ **
        $v.out \leftarrow 0$;
        repeat $\{$ $v.out++$; $\}$ until ( $cost_v[i, v.out] == cost_v^{min}[i]$ );
    ** we are computing values for a new $v.out$ **
        $\forall k = 1 \ldots n(v)$ $l(v).changed \leftarrow \textsc{Descend}[c_k(v), true, v.out]$
    $v.losses \leftarrow \max((i - v.out), 0)$;
    $v.dups \leftarrow \max((v.out - i), 0)$;
    return $true$;
```

5.3 Hybrid Micro-Macrophylogeny

The **Macrophylogeny** algorithm presented in Section 5.1 can be used with minor modifications as a subroutine in a solution for the **Hybrid Micro-Macrophylogeny** problem set forth in Section 4. The **Hybrid Micro-Macrophylogeny** algorithm starts by applying a wrapper function, REARRANGE, to the root of the input tree, T_{G}. REARRANGE descends T_{G}visiting each CCW in preorder, rearranging it to minimize the **D/L Score** before reinserting it into T_{G}. REARRANGE can be found in Algorithm 3.

Algorithm 3.

```
REARRANGE[$g$]
    if ($robust$(g) AND ($weak(l$(g)) OR $weak(r$(g))))
```

```
        PREPROCESS[$g, M(g)$];
        $g \leftarrow$ RECONSTRUCT$CCW$[$M(g)$];
    $l(g) \leftarrow$ REARRANGE[$l(g)$];
    $r(g) \leftarrow$ REARRANGE[$r(g)$];
```

The function PREPROCESS, which is not shown, extracts the rooted tree corresponding to each CCW and records 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 Algorithm 1, 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.

The RECONSTRUCT routine from Algorithm 1 is modified to save a list of reconstructed optimal event histories for each CCW, along with additional bookkeeping required to generate all permutations of optimal histories from this list. These modifications are also not shown.

Figure 5

The routines ASCEND, DESCEND and CONSTRUCT given in Algorithm 1 must be modified in order to account for multiplicities on internal nodes of the species tree. Line 7 in ASCEND must be replaced with

$$cost_v[i,j] \leftarrow c_\delta * \max\left(j - i + m_v, 0\right) + c_\lambda * \max\left(i - j - m_v, 0\right) + cost^{min}_{l(v)}[j] + cost^{min}_{r(v)}[j];$$

Additionally, Lines 25 & 26 in DESCEND must be replaced with

```
    $v.losses \leftarrow \max\left((i - v.out - m_v), 0\right)$;
    $v.dups \leftarrow \max\left((v.out - i + m_v), 0\right)$;
```

These changes address the need to count losses and duplications differently when internal nodes in the species tree are annotated with multiplicities. Finally, the CONSTRUCT routine must be modified to reinsert the optimized CCW into the original gene tree. To allow CONSTRUCT to reconnect strong subtrees to the reconstructed CCW, the following lines are added between Lines 8 & 9:

```
    else if ($s.copiesHere < m_v$)
        $s.copiesHere++$;
        $g \leftarrow$ a strong subtree with lca $s$
```

Algorithm 3 can also be used to refine multifurcating gene trees. In this case, all edges in the multifurcating tree are marked as robust edges. The non-binary nodes in the gene tree must then be expanded to an arbitrary binary subtree, and any edges added in that process marked as weak edges. The **Hybrid Micro-Macrophylogeny** algorithm can then be run on this gene tree to produce a binary gene tree refined with respect to duplication and loss.

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, an edge weight threshold, θ, and positive costs c_δ and c_λ, NOTUNG 2.0 computes all optimal rearrangement histories using the **Hybrid Micro-Macrophylogeny** algorithm and presents one tree for each optimal history. All other trees for the same history can be generated using a point and click interface that allows the user to swap nodes within the same species; i.e., that can be interchanged without changing the **D/L Score**. The NOTUNG 2.0 graphical user interface was constructed, in part, using the tree visualization library provided by ATV (version 1.92) (Zmasek and Eddy, 2001a). NOTUNG 2.0 can also be executed from the command line, allowing for automated analysis of a large number of trees.

In a previous study (Chen et al., 2000), we developed a test set of thirteen trees discussed in three recent articles on large scale duplication (Hughes, 1998; Pebusque et al., 1998; Ruvinsky and Silver, 1997). 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 trees. Phylogenetic analysis suggests that gene duplication followed by functional differentiation is an adaptive response to environmental change in both of these families. The first data set includes ATP-binding Cassette (ABC) transporter sequences from *Dictyostelium discoideum*, *Arabidopsis thaliana* and *Saccharomyces cerevisiae* and a number of *Plasmodia* species, provided by Roxana Cintron, Dr. Adelfa Serrano and colleagues (University of Puerto Rico). Several ABC transporter subfamilies are associated with malarial drug resistance in *Plasmodia* and pesticide resistance in *A. thaliana*. The second tree, provided by Dr. Hugh Nicholas (Pittsburgh Supercomputing Center) is derived from the Glutathione-S Transferases (Sheehan et al., 2001), a superfamily of detoxification enzymes.

Sequence trees for both data sets were constructed using Neighbor Joining (Saitou and Nei, 1987). Bootstrap replicates were obtained using the SEQBOOT program from Felsenstein's Phylip package (v. 3.6.1) available at the Pittsburgh Supercomputing Center. Details are given in the appendix. The divergence in both families was substantial, resulting in a large number of edges with low bootstrap values. NOTUNG 2.0 was used to determine 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. In addition, macroevolutionary rearrangement reduces the number of hypotheses that must be considered to a dramatic extent. When $c_\delta = 1.5$, $c_\delta = 1$ and $\theta = 50\%$, there is only one most parsimonious history for each test tree. A cutoff of 90% for the GST tree resulted in only two most parsimonious histories. Table 1

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. Visualization of a tree before and after rearrangement reveals the temporal organization of the duplications in the history of the family and their distribution within lineages and subfamilies. For example, analysis and visualization of the ABC tree (Cintron et al., 2004) 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. When combined with ecological and biochemical data, this type of 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.

Acknowledgments

We thank R. Cintron, D. Danicic, M. Farach-Colton, A. Goldman, 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. Serrano for providing the ABC transporter and GST data sets, and A. McLeod for everything. 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.

A Data set preparation

ATP-binding cassette transporters: Multiple alignments of 350 ABC transporter sequences were generated with CLUSTALW (Thompson et al., 1994) and TCOFFEE (Notredame et al., 2000) and manually edited using the GeneDoc (version 2.6.002) Multiple Sequence Alignment Editor and Shading Utility (K.B. Nicholas and Deerfield, 1997). A phylogenetic tree was built using the Phylip package from Felsenstein (v. 3.6.1) available at the Pittsburgh Supercomputing Center, which contains several programs to construct a bootstrap phylogeny (Efron and Gong, 1983). Briefly, the SEQBOOT program was run and the output file used as input to run the PROTDIST program in order to calculate the distance matrices from each of the replicate data sets. SEQBOOT generated 432 replicate data sets. PROTDIST produced 500 distance matrices for the phylogeny using the PAM model for amino acid substitutions. The output of the distance program was used as input for NEIGHBOR in order to create phylogenetic trees using the Neighbor-joining method.

Glutathione S-Transferases: An initial set of 153 GST sequences were aligned using the T-COFFEE program (Notredame et al., 2000). The same sequences were analyzed by the MEME program (Bailey and Elkan, 1994) using the ZOOPS (Zero Or One Per Sequence) which returned 20 motifs found in the sequences. The GeneDoc program (K.B. Nicholas and Deerfield, 1997) was used to highlight the MEME motifs within the global multiple sequence alignment produced by T-COFFEE. These motifs were used to guide manual editing and refinement of the alignment. This refined alignment was used to create a position specific scoring matrix and used to search the NBRF/PIR sequence database for additional GST sequences using the PSSMSearch program (Ropelewski et al., 2000). The additional sequences were aligned to the initial refined alignment using the profile alignment routine in ClustalW (Thompson et al., 1994). Duplicate and incomplete sequences were removed and the final set of 247 complete sequences were submitted for analysis by MEME, again using the ZOOPS model. This final set of 20 motifs was used to guide additional refinement of the complete alignment in GeneDoc. From this alignment, a subset of 131 best annotated sequences was selected for this study. A Neighbor Joining tree was built using PROTDIST (Phylip) with pam distances. Five hundred bootstrap replicates were obtained using Phylip's SEQBOOT program.

References

Arvestad, L., Berglund, A.-C., Lagergren, J., and Sennblad, B., 2003. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. Bioinformatics 19 Suppl 1, i7–15.

Arvestad, L., Berglund, A.-C., Lagergren, J., and Sennblad, B., 2004. 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, 326–335. ACM Press.

Bailey, T. and Elkan, C., 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proc Int Conf Intell Syst Mol Biol 2, 28–36.

Chen, K., Durand, D., and Farach-Colton, M., 2000. Notung: A program for dating gene duplications and optimizing gene family trees. Journal of Computational Biology 7, 429–447.

Chor, B. and Tuller, T., 2005. Maximum likelihood of evolutionary trees is hard. In Miyano, S., ed., Proceedings of the Ninth Annual International Conference on Computational Molecular Biology (RECOMB 2005), Lecture Notes in Bioinformatics. Springer Verlag.

Cintron, R., Nicholas Jr., H. B., Ferrer, I., Gonzalez, R., Vernot, B., Durand, D., and Serrano, A. E., 2004. The ABC superfamily: Evolutionary implications for drug resistance in *Plasmodia*. In Functional Genomics and Bioinformatics Approaches to Infectious Disease Research. American Society for Microbiology. Abstract book.

Day, W. H., 1987. Computational complexity of inferring phylogenies from dissimilarity matrices. Bull Math Biol 49, 461–7.

Day, W. H. E., Johnson, D. S., and Sankoff, D., 1986. The computational complexity of inferring rooted phylogenies by parsimony. Math Biosci 81, 33–42.

Dufayard, J.-F., Duret, L., Penel, S., Gouy, M., Rechenmann, F., and Perrire, G., 2005. Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases. Bioinformatics 21, 2596–603. URL http://dx.doi.org/10.1093/bioinformatics/bti325.

Efron, B. and Gong, G., 1983. A leisurely look at the bootstrap, jackknife, and cross-validation. The American Statistician 37, 36–48.

Eulenstein, O., Mirkin, B., and Vingron, M., 1996. 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.

Eulenstein, O., Mirkin, B., and Vingron, M., 1998. Duplication-based measures of difference between gene and species trees. Journal of Computational Biology 5, 135–148.

Felsenstein, J., 2003. Inferring Phylogenies. Sinauer.

Goodman, M., Czelusniak, J., Moore, G. W., Romero-Herrera, A. E., and Matsuda, G., 1979. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. Syst Zool 28, 132–163.

Gorecki, P., 2004. Reconciliation problems for duplication, loss and horizontal gene transfer. In Proceedings of the eighth annual international conference on Computational molecular biology, 316–325. ACM Press.

Guigo, R., Muchnik, I., and Smith, T., 1996. Reconstruction of ancient phylogenies. Molecular Phylogenetics and Evolution 6, 189–213.

Hallett, M., Lagergren, J., and Tofigh, A., 2004. Simultaneous identification of duplications and lateral transfers. In Proceedings of the eighth annual international conference on Computational molecular biology, 347–356. ACM Press.

Hallett, M. T. and Lagergren, J., 2000. New algorithms for the duplication-loss model. In RECOMB2000, Fourth Annual International Conference on Computational Molecular Biology.

Hallett, M. T. and Lagergren, J., 2001. Efficient algorithms for lateral gene transfer problems. In Proceedings of the fifth annual international conference on Computational biology, 149–156. ACM Press. URL `http://doi.acm.org/10.1145/369133.369188`.

Hughes, A. L., 1998. Phylogenetic tests of the hypothesis of block duplication of homologous genes on human chromosomes 6, 9, and 1. Mol Biol Evol 15, 854–70.

K.B. Nicholas, H. N. and Deerfield, D., 1997. Genedoc: Analysis and visualization of genetic variation. EMBNEW.NEWS 4, 14. URL `http://www.psc.edu/biomed/genedoc/`.

Ma, B., Li, M., and Zhang, L., 2000. From gene trees to species trees. SIAM J. on Comput. .

Mirkin, B., Muchnik, I., and Smith, T., 1995. A biologically consistent model for comparing molecular phylogenies. Journal of Computational Biology 2, 493–507.

Newick Taxonomy, 1986. The Newick tree format. Extithttp://evolution.genetics. washington.edu/phylip/newicktree.html.

Notredame, C., Higgins, D., and Heringa, J., 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol 302, 205–17. URL `http://dx.doi.org/10.1006/jmbi.2000.4042`.

Page, R., 1994. Maps between trees and cladistic analysis of historical associations among genes, organisms and areas. Syst Zool 43, 58–77.

Page, R., 1998. GeneTree: comparing gene and species phylogenies using reconciled trees. Bioinformatics 14, 819–20.

Page, R. and Charleston, M., 1996. Reconciled trees and incongruent gene and species trees. Mathematical Heirarchies and Biology, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 37, 57–70.

Page, R. and Charleston, M., 1997. From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem. Molecular Phylogenetics and Evolution 7, 231–240.

Page, R. and Charleston, M., 1998. Trees within trees: phylogeny and historical associations. Trends in Ecology and Evolution 13, 356–359.

Pebusque, M.-J., Coulier, F., Birnbaum, D., and Pontarotti, P., 1998. Ancient large-scale genome duplications: phylogenetic and linkage analyses shed light on chordate genome evolution. Mol Biol Evol 15, 1145–59.

Ronquist, F. and Huelsenbeck, J. P., 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19, 1572–4.

Ropelewski, A.J., Jr., N., H.B., II, D., and D., 2000. Selective and sensitive comparison of genetic sequence data on high performance computers. In Koniges, A., ed., Parallel Applications Technology Program, 453 – 479. Morgan Kaufmann.

Ruvinsky, I. and Silver, L. M., 1997. Newly indentified paralogous groups on mouse chromosomes 5 and 11 reveal the age of a T-box cluster duplication. Genomics 40, 262–266.

Saitou, N. and Nei, M., 1987. The neighbor-joining method: A new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4, 406–425.

Sheehan, D., Meade, G., Foley, V., and Dowd, C., 2001. Structure, function and evolution of glutathione transferases: implications for classification of non-mammalian members of an ancient enzyme superfamily. Biochem J 360, 1–16.

Stege, U., 1999. 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).

Thompson, J., Higgins, D., and Gibson, T., 1994. Improved sensitivity of profile searches through the use of sequence weights and gap excision. Comput Appl Biosci 10, 19–29.

Zhang, L., 1997. On a Mirkin–Muchnik-Smith conjecture for comparing molecular phylogenies. Journal of Computational Biology 4, 177–188.

Zmasek, C. M. and Eddy, S. R., 2001a. ATV: display and manipulation of annotated phylogenetic trees. Bioinformatics 17, 383–4. URL http://bioinformatics.oupjournals.org/cgi/reprint/17/4/383.

Zmasek, C. M. and Eddy, S. R., 2001b. A simple algorithm to infer gene duplication and speciation events on a gene tree. Bioinformatics 17, 821–8. URL http://bioinformatics.oupjournals.org/cgi/reprint/17/9/821.

List of Figures

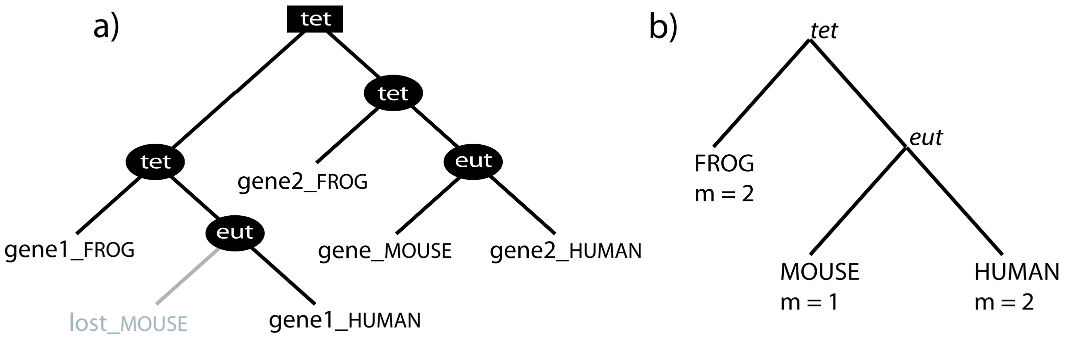

Figure 1: (a) A gene tree with one duplication and one loss. Internal nodes represent duplication (boxes) or speciation (ovals). Lost genes are shown in gray. (b) A species tree for frog, human and mouse. Leaves of the species tree are labeled with the number of gene family members found in each species. Internal nodes in the species tree are labeled eut (eutherian) and tet (tetrapod).

⇑

Dannie Durand,
Departments of Biological Sciences and Computer Science,
Carnegie Mellon University, Pittsburgh, PA, USA. durand@cmu.edu, Bjarni
V. Halldórsson
deCODE Genetics, Reykjavik, Iceland, bjarni.halldorsson@decode.is,
Benjamin Vernot
Department of Biological Sciences , Carnegie Mellon University,
Pittsburgh, PA, 15213, USA. bvernot@cs.cmu.edu

Figure 1 (of 5)

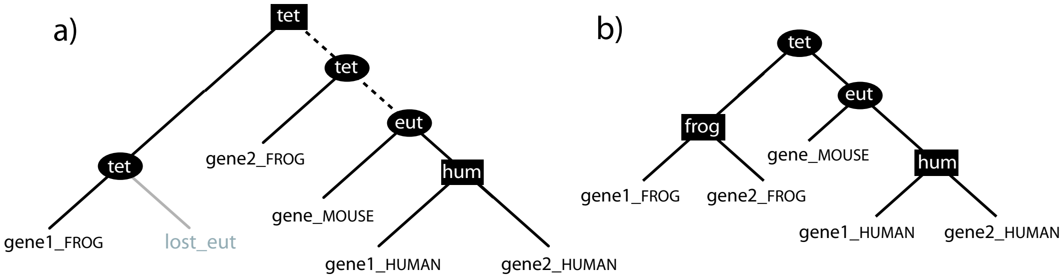

Figure 2: (a) A gene tree with two duplication nodes and one loss. Weak edges are shown as dashed lines. The edge between the eut and human nodes is robust. The edges adjacent to leaf nodes are neither weak nor robust. (b) An alternate topology for the same gene family, with two duplications and no losses.

⇑

Dannie Durand,

Departments of Biological Sciences and Computer Science,

Carnegie Mellon University, Pittsburgh, PA, USA. durand@cmu.edu, Bjarni

V. Halldórsson

deCODE Genetics, Reykjavik, Iceland, bjarni.halldorsson@decode.is,

Benjamin Vernot

Department of Biological Sciences , Carnegie Mellon University,

Pittsburgh, PA, 15213, USA. bvernot@cs.cmu.edu

Figure 2 (of 5)

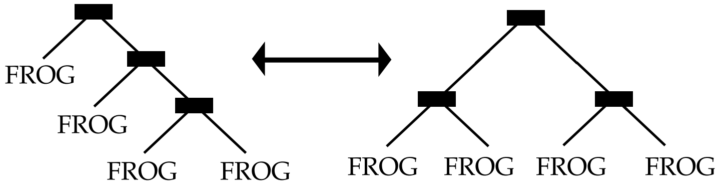

Figure 3: Two equally parsimonious arrangements of four FROG genes, each with three duplications.

⇑

Dannie Durand,

Departments of Biological Sciences and Computer Science,

Carnegie Mellon University, Pittsburgh, PA, USA. durand@cmu.edu, Bjarni V. Halldórsson

deCODE Genetics, Reykjavik, Iceland, bjarni.halldorsson@decode.is,

Benjamin Vernot

Department of Biological Sciences , Carnegie Mellon University,

Pittsburgh, PA, 15213, USA. bvernot@cs.cmu.edu

Figure 3 (of 5)

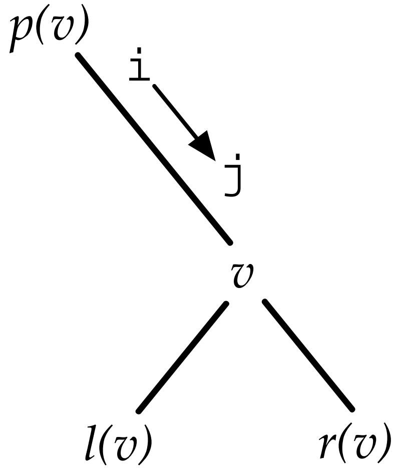

Figure 4: Schematic showing the meaning of variables in Algorithm 1.

⇑

Dannie Durand,

Departments of Biological Sciences and Computer Science,

Carnegie Mellon University, Pittsburgh, PA, USA. durand@cmu.edu, Bjarni V. Halldórsson

deCODE Genetics, Reykjavik, Iceland, bjarni.halldorsson@decode.is,

Benjamin Vernot

Department of Biological Sciences , Carnegie Mellon University,

Pittsburgh, PA, 15213, USA. bvernot@cs.cmu.edu

Figure 4 (of 5)

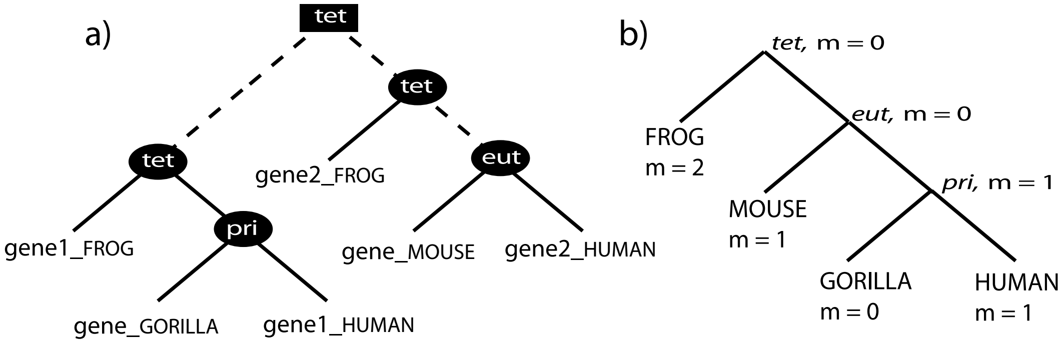

Figure 5: (a) A gene tree with a weakly connected component, shown here by dotted edges. (b) Annotated tree for the species in (a). Note that the internal species node pri (primate) has a multiplicity, while GORILLA does not.

⇑

Dannie Durand,

Departments of Biological Sciences and Computer Science,

Carnegie Mellon University, Pittsburgh, PA, USA. durand@cmu.edu, Bjarni V. Halldórsson

deCODE Genetics, Reykjavik, Iceland, bjarni.halldorsson@decode.is,

Benjamin Vernot

Department of Biological Sciences , Carnegie Mellon University,

Pittsburgh, PA, 15213, USA. bvernot@cs.cmu.edu

Figure 5 (of 5)

List of Tables

	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

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.

⇑

Dannie Durand,

Departments of Biological Sciences and Computer Science,

Carnegie Mellon University, Pittsburgh, PA, USA. durand@cmu.edu, Bjarni V. Halldórsson

deCODE Genetics, Reykjavik, Iceland, bjarni.halldorsson@decode.is,

Benjamin Vernot

Department of Biological Sciences , Carnegie Mellon University,

Pittsburgh, PA, 15213, USA. bvernot@cs.cmu.edu

Table 1 (of 1)