Chaudhary et al. BMC Bioinformatics 2012, 13(Suppl 10):S11
http://www.biomedcentral.com/1471-2105-13-S10-S11

BMC Bioinformatics

PROCEEDINGS Open Access

Efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence

Ruchi Chaudhary[1], J Gordon Burleigh[2], Oliver Eulenstein[1*]

From 7th International Symposium on Bioinformatics Research and Applications (ISBRA'11)
Changsha, China. 27-29 May 2011

Abstract

Background: Gene tree - species tree reconciliation problems infer the patterns and processes of gene evolution within a species tree. Gene tree parsimony approaches seek the evolutionary scenario that implies the fewest gene duplications, duplications and losses, or deep coalescence (incomplete lineage sorting) events needed to reconcile a gene tree and a species tree. While a gene tree parsimony approach can be informative about genome evolution and phylogenetics, error in gene trees can profoundly bias the results.

Results: We introduce efficient algorithms that rapidly search local Subtree Prune and Regraft (SPR) or Tree Bisection and Reconnection (TBR) neighborhoods of a given gene tree to identify a topology that implies the fewest duplications, duplication and losses, or deep coalescence events. These algorithms improve on the current solutions by a factor of n for searching SPR neighborhoods and n^2 for searching TBR neighborhoods, where n is the number of taxa in the given gene tree. They provide a fast error correction protocol for ameliorating the effects of gene tree error by allowing small rearrangements in the topology to improve the reconciliation cost. We also demonstrate a simple protocol to use the gene rearrangement algorithm to improve gene tree parsimony phylogenetic analyses.

Conclusions: The new gene tree rearrangement algorithms provide a fast method to address gene tree error. They do not make assumptions about the underlying processes of genome evolution, and they are amenable to analyses of large-scale genomic data sets. These algorithms are also easily incorporated into gene tree parsimony phylogenetic analyses, potentially producing more credible estimates of reconciliation cost.

Introduction

The availability of large-scale genomic data from a wide variety of taxa has revealed much incongruence between gene trees and the phylogeny of the species in which the genes evolve. This incongruence may be caused by evolutionary processes such as gene duplication and loss, deep coalescence, or lateral gene transfer. The variation in gene tree topologies can be used to infer the processes of genome evolution. Gene tree - species tree (GT-ST) reconciliation methods seek to map the history of gene trees into the context of species evolution and thus potentially link processes of gene evolution to phenotypic changes and diversification. Yet these methods can be confounded by error in the gene trees, which also may cause incongruence between the gene and species topologies. We introduce efficient algorithms to correct gene tree topologies based on the gene duplication, duplication and loss, or deep coalescence cost models. The algorithms work by identifying the small rearrangements in the gene trees that reduce the reconciliation cost. They are extremely fast and thus amenable to analyses of enormous genomic data sets.

Perhaps the most commonly used and computationally feasible approach to GT-ST reconciliation is gene tree

* Correspondence: oeulenst@cs.iastate.edu
[1]Department of Computer Science, Iowa State University, Ames, IA 50011, USA
Full list of author information is available at the end of the article

parsimony, which seeks to infer the fewest evolutionary events (e.g., duplication, loss, coalescence, or lateral gene transfer) needed to reconcile a gene tree and species tree topology [1]. This approach also can be extended to infer species phylogenies, finding the species tree that implies the fewest evolutionary events implied by the gene trees (e.g., [2-4]). However, the gene trees often are estimated using heuristic methods from short sequence alignments, and consequently, there is often much error in the estimated gene tree topologies. Error in the gene trees creates more GT-ST incongruence and can radically affect GT-ST reconciliation analyses, implying far more duplications, duplications and losses, or deep coalescence events than actually exist. For example, Rasmussen and Kellis [5] estimated that error in gene tree reconstruction can lead to 2-3 fold overestimates of gene duplications and losses. Gene tree error also can erroneously imply large numbers of duplications near the root of the species tree [6,7], and it can mislead gene tree parsimony phylogenetic analyses (e.g., [8-10]).

Several approaches have been proposed to address gene tree error in GT-ST reconciliation. First, questionable nodes in a gene tree or nodes with low support may be collapsed prior to gene tree reconciliation, and the resulting non-binary gene trees may be reconciled with species trees [11-13]. Similarly, GT-ST reconciliations can use a distribution of gene tree topologies, such as bootstrap gene trees, rather than a single gene tree estimate [6,14,15]. Both of these approaches may help account for stochastic error and uncertainty in gene tree topologies, but they do not explicitly confront gene tree error. Methods also exist to simultaneously infer the gene tree topology and the gene tree reconciliation with a known species tree [5,16]. While these sophisticated statistical approaches appear very promising, they are computationally intensive, and it is unclear if they will be tractable for large-scale analyses. Another, perhaps a more computationally feasible, approach is to allow a limited number of local rearrangements in the gene tree topology if they reduced the reconciliation cost [17,18].

Previously [17,18] described a method to allow NNI-branch swaps on selected branches of a gene tree to reduce the reconciliation cost. Following [17,18], we address gene tree error in the reconciliation process by assuming that the correct gene tree can be found in a particular neighborhood of the given gene tree. We describe this approach for the gene duplication, duplication and loss, and deep coalescence models, which identify the fewest respective events implied from a given gene tree and given species tree. This neighborhood consists of all trees that are within one edit operation of the gene tree. While [17,18] use Nearest Neighbor Interchange (NNI) edit operations to define the neighborhood, we use the standard tree edit operations SPR [19,20] and TBR [19], which significantly extend upon the search space of the NNI neighborhood. The SPR and TBR local search problems find a tree in the SPR and TBR neighborhood of a given gene tree, respectively, that has the smallest reconciliation cost when reconciled with a given species tree. Using the algorithm from Zhang [21] the best known (naïve) runtimes are $O(n^3)$ for the SPR local search problem and $O(n^4)$ for the TBR local search problem, where n is the number of taxa in the given gene tree. These runtimes typically are prohibitively long for the computation of larger GT-ST reconciliations. We improve on these solutions by a factor of n for the SPR local search problem and a factor of n^2 for the TBR local search problem. This makes the local search under the TBR edit operation as efficient as under the SPR edit operation, and it provides a high-speed gene tree error-correction protocol that is computationally feasible for large-scale genomic data sets.

We also evaluated the performance of our algorithms using the implementation of SPR based local search algorithms. Note, that the SPR neighborhood is properly contained in the TBR neighborhood for any given tree. Thus the performance of the SPR based program provides a conservative estimate of the performance of the TBR based program. We test our programs on a collection of 106 yeast gene trees, some of which contain hundreds of leaves, and we demonstrate how it can be easily incorporated into large-scale gene tree parsimony phylogenetic analyses.

Basic notation and preliminaries

Throughout this paper, the term tree refers to a rooted full binary (phylogenetic) tree.

Let T be a tree. The leaf set of T is denoted by $Le(T)$. The set of all vertices of T is denoted by $V(T)$ and the set of all edges by $E(T)$. The *root* of T is denoted by $Ro(T)$. The set of internal vertices of T is $I(T) := V(T) \backslash Le(T)$.

Given a vertex $v \in V(T)$, we denote the *parent* of v by $Pa_T(v)$. Let $u := Pa_T(v)$. The edge that connects v with u is written as (u, v). The first element in the pair is always the parent of the second element. The set of all children of v is denoted by $Ch_T(v)$ and the children are called *siblings*. For $w \in Ch_T(v)$, the sibling of w is denoted by $Sb_T(w)$.

We define $\leq_T$ to be the *partial order* on $V(T)$ where $x \leq_T y$ if y is a vertex on the path between $Ro(T)$ to x, and write $x <_T y$ if $x \leq_T y$ and $x \neq y$. The *least common ancestor* of a non-empty subset $L \subseteq V(T)$, denoted as $LCA_T(L)$, is the unique smallest upper bound of L under $\leq_T$. Given $x, y \in V(T)$, $d_T(x, y)$ denotes the number of edges on the unique path between x and y in T.

Given $U \subseteq V(T)$, we denote by $T(U)$ the unique rooted subtree of T that spans U with the minimum number of vertices. Furthermore, the *restriction* of T to U, denoted by $T_{|U}$, is the rooted tree that is obtained from $T(U)$ by suppressing all non-root vertices of degree two. The

subtree of T rooted at $u \in V(T)$, denoted by T_u, is defined to be $T_{|U}$, for $U := \{v \in Le(T): v \leq_T u\}$. Two trees T_1 and T_2 are called *isomorphic* if there exists a bijection between the vertex sets of T_1 and T_2 which maps a vertex u_1 of T_1 to vertex u_2 of T_2 if the subtree rooted at u_1 in T_1 has the same leaf set as the subtree rooted at u_2 in T_2. If an isomorphism exists between T_1 and T_2, we write $T_1 \simeq T_2$.

Given function $f: A \to B$, we denote by $f(A')$ where $A' \subseteq A$ a set of images of each element in A' under f.

The reconciliation cost models

A *species tree* is a tree that depicts the branching pattern representing the divergence of a set of species, whereas a *gene tree* is a tree that depicts the evolutionary history among the sequences encoding one gene (or gene family) for a given set of species. We assume that each leaf of the gene tree is labeled with the species from which that gene was sampled. Let G be a gene tree and S a species tree. In order to compare G with S, we require a mapping from each gene $g \in V(G)$ to the most recent species in S that could have contained g.

Definition 1 (Mapping). *The* leaf-mapping $\mathcal{L}_{G,S}: Le(G) \to Le(S)$ *is a surjection that maps each leaf* $g \in Le(G)$ *to that unique leaf* $s \in Le(S)$ *which has the same label as g. The extension* $\mathcal{M}_{G,S}: V(G) \to V(S)$ *is the mapping defined by* $\mathcal{M}_{G,S}(g) := LCA(\mathcal{L}_{G,S}(Le(G_g)))$. *For convenience, we write* $\mathcal{M}(g)$ *instead of* $\mathcal{M}_{G,S}(g)$ *when G and S are clear from the context.*

Definition 2 (Comparability). *Given trees G and S, we say that G is* comparable *to S if a leaf-mapping* $\mathcal{L}_{G,S}$ *(g) is well defined.*

Throughout this paper we use the following terminology: G is a gene tree that is comparable to the species tree S through a leaf-mapping $\mathcal{L}_{G,S}$, and n is the number of taxa present in both input trees.

Now we define different reconciliation costs from G to S for a given mapping $\mathcal{M}_{G,S}$ that extends $\mathcal{L}_{G,S}$. The reconciliation cost are based on the models of gene duplication [22,23], duplication-loss [21], and deep coalescence [21].

Definition 3 (Duplication cost).

- *The* duplication cost *from* $g \in V(G)$ *to S,*
$$\mathcal{C}_D(G,S,g) := \begin{cases} 1, & \text{if } \mathcal{M}(g) \in \mathcal{M}(Ch(g)); \\ 0, & \text{otherwise.} \end{cases}$$
- *The* duplication cost *from G to S,*
$$\mathcal{C}_D(G,S) := \sum_{g \in I(G)} \mathcal{C}_D(G,S,g).$$

Definition 4 (Duplication-loss cost).

- *The* loss cost *from* $g \in V(G)$ *to S,*
$$\mathcal{C}_L(G,S,g) := \begin{cases} 0, & \text{if } \forall h \in Ch(g): \mathcal{M}(g) = \mathcal{M}(h); \\ \sum_{h \in Ch(g)} |d_S(\mathcal{M}(g), \mathcal{M}(h)) - 1|, & \text{otherwise.} \end{cases}$$
- The duplication-loss cost *from G to S,*
$$\mathcal{C}_{DL}(G,S) := \sum_{g \in I(G)} \left(\mathcal{C}_D(G,S,g) + \mathcal{C}_L(G,S,g)\right).$$

Definition 5 (Deep coalescence cost).

- *The* number of lineages from $g \in V(G)$ *to* $h \in Ch(g)$ *in S,*
$$\mathcal{C}_{DC}(G,S,g) := \sum_{h \in Ch(g)} d_S(\mathcal{M}(g), \mathcal{M}(h)).$$
- *The* deep coalescence cost *from G to S,*
$$\mathcal{C}_{DC}(G,S) := \sum_{g \in I(G)} \mathcal{C}_{DC}(G,S,g) - |E(S)|.$$

The error-correction problems

Here we give definitions for tree rearrangement operations TBR [19] and SPR [19,20], and then formulate the Error-Correction problems that were motivated in the introduction.

Definition 6 (Tree Bisection and Reconnection (TBR)). *Let T be a tree. For this definition, we regard the* planted tree $Pl(T)$ *as the tree obtained from adding the* root edge$\{r, Ro(T)\}$ *to* $E(T)$, *where* $r \notin V(T)$.

Let $e := (u, v) \in E(T)$, *and X and Y be the connected components that are obtained by removing edge e from T such that* $v \in X$ and $u \in Y$. *We define* $TBR_T(v, x, y)$ *for* $x \lfloor X$ *and* $y \in Y$ *to be the tree that is obtained from* $Pl(T)$ *by first deleting edge e, and then adjoining a new edge f between X and Y as follows*:

1. *If* $x \neq Ro(X)$ *then suppress* $Ro(X)$ *and create a new root by subdividing edge* $(Pa(x), x)$.
2. *Subdivide edges* $(Pa(y), y)$ *by introducing a new vertex y'.*
3. *Re-connect components X and Y by adding edge* $f = (y', Ro(x))$.
4. *Suppress the vertex u, and rename vertex y' as u.*
5. *Contract the root edge.*

We say that, the tree $TBR_T(v, x, y)$ *is obtained from T by a* tree bisection and reconnection (TBR) *operation that bisects the tree T into the components X and Y , and reconnects them above the nodes x and y. (See* Figure 1.*) We define the following neighborhoods for the TBR operation*:

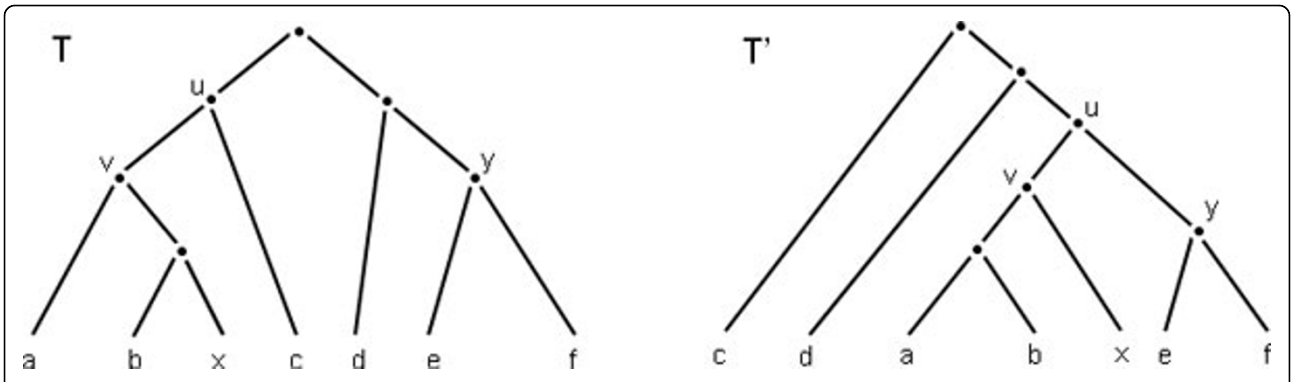

Figure 1 An TBR operation. Tree $T' = TBR_T(v, x, y)$ results from T after performing single TBR operation.

1. $TBR_G(v, x) := \cup_{y \not\perp Y} TBR_G(v, x, y)$
2. $TBR_G(v) := \cup_{x \not\perp X} TBR_G(v, x)$
3. $TBR_G := \cup_{(u,\ v) \in E(G)} TBR_G(v)$

Definition 7 (Subtree Prune and Regrafting (SPR)). *The SPR operation is defined as a special case of the TBR operation. Let* $e := (u, v) \in E(T)$, *and X and Y be the connected components that are obtained by removing edge e from T such that* $v \in X$ and $u \in Y$. *We define* $SPR_T(v, y)$ *for* $y \in Y$ *to be* $TBR_T(v, v, y)$. *We say that the tree* $SPR_T(v, y)$ *is obtained from T by performing subtree prune and regraft (SPR) operation that prunes subtree* T_v *and regrafts it above y. (See* Figure 2 (a).)

We define the following neighborhoods *for the SPR operation*:

1. $SPR_G(v) := \cup_{y \not\perp Y} SPR_G(v, y)$
2. $SPR_G := \cup_{(u,\ v) \in E(G)} SPR_G(v)$

We now state the SPR based error-correction problems for duplication (D), duplication-loss (DL), and deep coalescence (DC). Let $\Gamma \in$ {D, DL, DC}.

Problem 1 (SPR based error-correction for Γ (SEC-Γ))
Instance: *A gene tree G and a species tree S.*
Find: A gene tree $G^* \in SPR_G$ *such that* $\mathcal{C}_\Gamma(G^*, S) = \min_{G' \in SPR_G} \mathcal{C}_\Gamma(G', S)$.

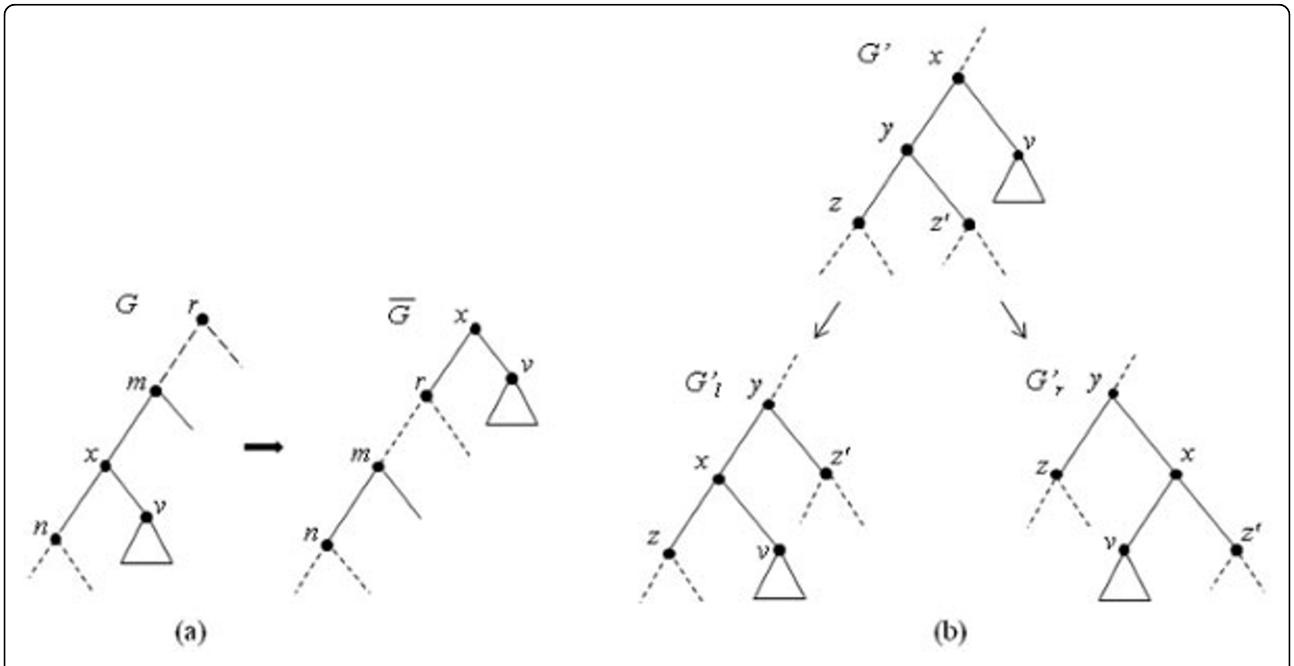

Figure 2 The NNI adjacency graph. (a) The tree $\overline{G}$ is obtained from G by pruning and regrafting subtree G_v to the root of G. The vertex $x \in V(G)$ is suppressed, and the new vertex above root in $\overline{G}$ is named x. (b) Two NNI operations $NNI_G(z')$ and $NNI_G(z)$ produce left-child G'_l and right-child G'_r of G' in $\mathcal{X}$.

The TBR based error-correction for Γ (TEC-Γ) problems are defined analogously to the SPR based error-correction for Γ (SEC-Γ) problems.

Solving the SEC-Γ problems

In this section we study the SPR based error-correction problems, for duplication (D), duplication-loss (DL), and deep coalescence (DC), in more detail. Our efficient solution for these problems are based on solving restricted versions of these problems efficiently. For each $\Gamma \in \{D, DL, DC\}$ we first define a restricted version of the SEC-Γ problem, which we call the restricted SPR based error-correction for the Γ (R-SEC-Γ) problem.

Problem 2 (Restricted SPR based error-correction for (R-SEC-Γ))

Instance: A gene tree G, a species tree S, and $v \in V(G)$.

Find: A gene tree $G^* \in SPR_G(v)$ *such that* $\mathcal{C}_\Gamma(G^*, S) = \min_{G' \in SPR_G(v)} \mathcal{C}_\Gamma(G', S)$.

Observation 1. Let $\Gamma \in \{D, DL, DC\}$. *Given a gene tree G and a species tree S*, the *SEC-Γ problem can be solved as follows: (i) solve the R-SEC-Γ problem for every* $v \in V(G)$ *where* $v \neq Ro(G)$, *(ii) under all solutions found return a minimum scoring gene tree G*.*

Naïvely, the R-SEC-Γ problem can be solved in $\Theta(n^2)$ time by computing the cost $\mathcal{C}_\Gamma(G', S)$ for each $G' \in SPR_G(v)$. The cost for a given gene and species tree can be computed in $\Theta(n)$ time [21]. We introduce a novel algorithm for the R-SEC-Γ problem that improves by a factor of n on the naïve solution. This speedup is achieved by semi-ordering the trees in $SPR_G(v)$, for each $v \in V(G)$, such that the score-difference of any two consecutive trees in this order can be computed in constant time.

Ordering the trees in $SPR_G(v)$

Consider a graph on trees in $SPR_G(v)$, in which every two adjacent trees are one NNI [19] operation apart. We show that such a graph is a rooted full binary tree, after providing necessary definitions.

Definition 8 (Nearest Neighbor Interchange (NNI)). *We define the NNI operation as a special case of the SPR operation. Let* $e \in E(T)$ *where* $e := (u, v)$, *and X and Y be the connected components that are obtained by removing edge e from T such that* $v \in X$ *and* $u \in Y$. *We define* $NNI_T(v)$ *to be* $SPR_T(v, y)$ *for* $y := Pa(u)$, *and say that* $NNI_T(v)$ *is obtained from T by performing nearest neighbor interchange (NNI) operation that prunes subtree* T_v *and regrafts it above the parent of v's parent. (See* Figure 2(b).)

Definition 9 (NNI distance). *Let the* NNI-distance, *denoted as* $d_{NNI}(T_1, T_2)$, *between two trees* T_1 *and* T_2 *over n taxa be the minimum number of NNI operations required to transform* T_1 *into* T_2.

Definition 10 (NNI-adjacency graph). *The* NNI-adjacency graph, *denoted as* $\mathcal{X} = (V, E)$, *is the graph where* $V = SPR_G(v)$ *and* $\{T_1, T_2\} \in E$ *if* $d_{NNI}(T_1, T_2) = 1$.

Lemma 1. $\mathcal{X}$ *is a tree.*

Proof. We prove it by showing that there exists a unique path between every two vertices in $\mathcal{X}$. Let G', $G'' \in V(\mathcal{X})$, thus G', $G'' \in SPR_G(v)$. Let $G' := SPR_G(v, x_1)$ and $G'' := SPR_G(v, x_2)$. We use induction on $d_G(x_1, x_2)$. Let $d_G(x_1, x_2) = 1$ and assume without loss of generality that $x_2 = Pa_G(x_1)$. Thus, $G'' = NNI_{G''}(Sb(x_1))$. So the hypothesis holds for $d_G(x_1, x_2) = 1$. Assume now that the hypothesis is true for $d_G(x_1, x_2) \leq k$ and suppose $d_G(x_1, x_2) = k + 1$. Since G is a tree, there must be a unique path between x_1 and x_2; let y be a vertex on this path. Let $d_G(y, x_1) = 1$, and $G^n := SPR_G(v, y)$. If $y = Pa_G(x_1)$, then $G^n = NNI_{G'}(v)$; otherwise $G^n = NNI_{G'}(Sb(y))$. Since $d_G(y, x_2) = k$, thus (by induction hypothesis) the hypothesis is valid for $d_G(x_1, x_2) = k + 1$. □

Theorem 1. $\mathcal{X}$ *is a rooted full binary tree.*

Proof. In view of Lemma 1, it suffices to show that except a unique vertex of degree 2 all other vertices in $\mathcal{X}$ are of degree 1 or 3. Let $G' \in V(\mathcal{X})$, thus $G' = SPR_G(v, y)$ for some $y \in V(G)$. There are three cases:

Case 1: y **is a root**. Let $y_1 \in Ch_G(y)$. Let $G^1 := SPR_G(v, y_1)$, thus $G' = NNI_{G^1}(v)$. Hence $\{G^1, G'\} \in E(\mathcal{X})$. Since $|Ch_G(y)| = 2$, G' must be a degree 2 vertex in $\mathcal{X}$.

Case 2: y **is a leaf**. Let $y_1 = Pa_G(y)$. Let $G^1 := SPR_G(v, y_1)$, thus $G^1 = NNI_{G'}(v)$. Hence $\{G^1, G'\} \in E(\mathcal{X})$, and consequently, G' is a degree 1 vertex in $\mathcal{X}$.

Case 3: y **is an internal vertex**. Let $y_1 = Pa_G(y)$ and $y_2 \in Ch_G(y)$. Let $G^1 := SPR_G(v, y_1)$, thus $G^1 = NNI_{G'}(v)$. Let $G^2 := SPR_G(v, y_2)$, thus $G' = NNI_{G^2}(v)$. Since y has one parent and two children in G, thus G' is a degree 3 vertex in $\mathcal{X}$.

This completes the proof.

The score difference of consecutive trees in $\mathcal{X}$

To solve the R-SEC-Γ problems we traverse tree $\mathcal{X}$. Two adjacent trees in $V(\mathcal{X})$ are one NNI operation apart. We show that $\mathcal{C}_\Gamma$ score of a tree can be computed in constant time from the LCA computation of its adjacent tree.

Let $e := (G', G'')$ be an edge in $\mathcal{X}$. Let $x := Pa(v)$, $y := Sb(v)$, and $z, z' \in Ch(y)$ in G' (see Figure 2(b)). Without loss of generality, let $G'' := NNI_{G'}(z)$. (Observe, G'' is similar to G'_r of Figure 2(b).)

Lemma 2. $\mathcal{M}_{G'',S}(y) = \mathcal{M}_{G',S}(x)$.

Proof. From NNI operation, $v, z' \in Ch_{G''}(x)$ and $z, x \in Ch_{G''}(y)$. Also, $G'_z \simeq G''_z$, $G'_{z'} \simeq G''_{z'}$, $G'_v \simeq G''_v$, so $Le(G''_y) = Le(G'_x)$. Thus, $\mathcal{M}_{G',S}(x) = \mathrm{LCA}(\mathcal{L}_{G',S}(Le(G'_x))) = \mathrm{LCA}(\mathcal{L}_{G'',S}(Le(G''_y))) = \mathcal{M}_{G'',S}(y)$. □

Lemma 3. $\mathcal{M}_{G'',S}(w) = \mathcal{M}_{G',S}(w)$, *for all* $w \in V(G')\backslash\{x, y\}$.

Proof. For $g \in V(G'_v) \cup (G'_z) \cup (G'_{z'})$, since $G'_g \simeq G''_g$, therefore $\mathcal{M}_{G',S}(g) = \mathcal{M}_{G'',S}(g)$. Also, except for subtree G'_x, the rest of the tree remains the same in G''_x. Thus by Lemma 2, $\mathcal{M}_{G',S}(Pa_{G'}(x)) = \mathcal{M}_{G'',S}(Pa_{G''}(y))$. Inductively, $\mathcal{M}_{G',S}(g) = \mathcal{M}_{G'',S}(g)$, for all $g \in V(G')\backslash V(G'_x)$. □

Lemma 4. $\mathcal{M}_{G'',S}(x) = LCA(\mathcal{M}_{G',S}(v), \mathcal{M}_{G',S}(z'))$.

Proof. From Lemma 3, $\mathcal{M}_{G'',S}(v) = \mathcal{M}_{G',S}(v)$ and $\mathcal{M}_{G'',S}(z') = \mathcal{M}_{G',S}(z')$. Thus, $\mathcal{M}_{G'',S}(x) = \text{LCA}(\mathcal{M}_{G'',S}(v), \mathcal{M}_{G'',S}(z')) = \text{LCA}(\mathcal{M}_{G',S}(v), \mathcal{M}_{G',S}(z'))$. □

Lemma 5. $\mathcal{C}_\Gamma(G'', S, g) = \mathcal{C}_\Gamma(G', S, g)$, *for all* $g \in V(G'')\backslash\{x, y\}$ *and* $\Gamma \in \{D, DL, DC\}$.

Proof. The gene duplication and loss status of a vertex, and the number of lineages from a vertex to its children in G' can change in G'' if its mapping or mapping of any of its children changes in $\mathcal{M}_{G'',S}$. From Lemma 3, and also, since $\mathcal{M}_{G'',S}(w) = \mathcal{M}_{G',S}(w)$, for $w \in Ch(Pa_{G'}(x))$, must have $\mathcal{C}_\Gamma(G'', S, Pa_{G'}(x)) = \mathcal{C}_\Gamma(G', S, Pa_{G'}(x))$. Thus the Lemma follows. □

Let $e := (G', G'') \in E(\mathcal{X})$ and $\Gamma \in \{\text{D, DL, DC}\}$. We define $\Gamma e := \mathcal{C}_\Gamma(G'', S) - \mathcal{C}_\Gamma(G', S)$ with respect to the given species tree S. Observe that this score can be negative too. We study how Γ_e can be computed efficiently for each edge e in $\mathcal{X}$.

Theorem 2. $\Gamma_e = \sum_{g \in \{x,y\}} \left(\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g)\right)$.

Proof. $\Gamma_e = \mathcal{C}_\Gamma(G'', S) - \mathcal{C}_\Gamma(G', S) = \sum_{g \in V(G'')} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g)) = \sum_{g \in V(G'')\backslash\{x,y\}} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g)) + \sum_{g \in \{x,y\}} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g)) = \sum_{g \in \{x,y\}} (\mathcal{C}_\Gamma(G'', S, g) - \mathcal{C}_\Gamma(G', S, g))$. □

Definition 11. *Let* $\overline{G} := SPR_G(v, Ro(G))$, *and let* $P_{G'}$ *be a path from* $\overline{G}$ *to* G' *in* $\mathcal{X}$. *For* G', *we define the* score-difference $\Gamma_{\overline{G},G'}$ *as* $\Gamma_{\overline{G},G'} := \sum_{e \in E(P_{G'})} \Gamma_e$.

Theorem 3. *For given S, G, and* $v \in V(G)$, *the tree* $G' \in V(\mathcal{X})$ *is the output of a R-SEC-*Γ *problem iff* $\Gamma_{\overline{G},G'} = min_{G'' \in V(\mathcal{X})} \Gamma_{\overline{G},G''}$.

Proof. Let $\Gamma_{\overline{G},G'} = min_{G'' \in V(\mathcal{X})} \Gamma_{\overline{G},G''}$. We prove that G' is the output of R-SEC-Γ problem. Since $\Gamma_{\bar{G},G'} = \sum_{e \in E(P_{G'})} \Gamma_e = \Gamma(G', S) - \Gamma(\bar{G}, S)$, thus G' gives the minimum normalized $\mathcal{C}_\Gamma$ score over all trees in $V(\mathcal{X})$. Hence, G' must be the output of the R-SEC-Γ problem. The other direction follows similarly. □

The algorithm

We describe a general algorithm, called Algo-R-SEC-Γ, to solve the R-SEC-Γ problem for each $\Gamma \in \{D, DL, DC\}$. Initially Algo-R-SEC-Γ computes (i) the root vertex of the NNI-adjacency graph $\mathcal{X}$, which we call $\overline{G}$, by regrafting the subtree G_v above the root of G, (ii) the LCA mapping from $\overline{G}$ to S, and (iii) the Γ score from $\overline{G}$ to S. Then recursively Algo-R-SEC-Γ computes the LCA mapping and Γ score for every vertex G' in $\mathcal{X}$ when the LCA mapping and Γ score of its parent vertex in $\mathcal{X}$ is known. Algorithm 1 details Algo-R-SEC-Γ.

Algorithm 1 - Algo-R-SEC-Γ

Input: *A gene tree G, a species tree S, and* $v \in V(G)$

Output: *A tree* $G^* \in SPR_G(v)$ *such that* $\mathcal{C}_\Gamma(G^*, S) = \min_{G' \in SPR_G(v)} \mathcal{C}_\Gamma(G', S)$

01. Compute $\overline{G}$ *by pruning* G_v *and regrafting at Ro(G)*
02. Compute LCA mapping $\mathcal{M}_{\bar{G},S}$
03. Call $\mathcal{C}_G(\overline{G}, S) = Algo - Comp - Score(\overline{G}, S, \mathcal{M}_{\overline{G},S})$
04. Set $BestTree = \overline{G}$, $BestScore = 0$
05. Set $G' = \bar{G}$, $\mathcal{M}_{G',S} = \mathcal{M}_{\bar{G},S}$, $\mathcal{C}_\Gamma(G', S) = \mathcal{C}_\Gamma(\bar{G}, S)$, $\Gamma_{\bar{G},G'} = 0$
06. ***For*** *each* $k \neq Ro(\bar{G}_{Sb(v)})$ *in preorder traversal of* $\bar{G}_{Sb(v)}$, ***do***
07. ***If*** *not backtracking,* ***then***
08. *Set* $x = Pa_{G'}(v)$, $y = Sb_{G'}(v)$
09. *Set* $G'' = NNI_{G'}(Sb_{G'}(k))$
10. *Set* $\mathcal{M}_{G'',S} = \mathcal{M}_{G',S}$, $\mathcal{M}G'',_S(y) = \mathcal{M}_{G',S}(x)$
11. $\mathcal{M}_{G'',S}(x) = LCA(\mathcal{M}_{G',S}(k), \mathcal{M}_{G',S}(v))$
12. *Call* $\Gamma_{(G,G')} = \sum_{he\{x,y\}} Algo - G - Score(G'', S, \mathcal{M}_{G'',S}, h) - Algo - G - Score(G', S, \mathcal{M}_{G',S}, h)$
13. $\Gamma_{\bar{G},G''} = \Gamma_{\bar{G},G'} + \Gamma_{\{G',G''\}}$
14. ***If*** $\Gamma_{\bar{G},G''} < BestScore$, ***then***
15. *Set* $BestTree = G''$, $BestScore = \Gamma_{\bar{G},G''}$
16. ***Else,***
17. *Set* $x = Pa_{G'}(v)$, $y = Pa_{G'}(x)$
18. *Set* $G'' = NNI_{G'}(v)$
19. *Set* $\mathcal{M}_{G'',S} = \mathcal{M}_{G',S}$, $\mathcal{M}_{G'',S}(x) = \mathcal{M}_{G',S}(y)$
20. *Set* $\mathcal{M}_{G'',S}(y) = LCA(\mathcal{M}_{G',S}(Sb_{G'}(x)), \mathcal{M}_{G',S}(k))$
21. *Call* $\Gamma_{(G',G')} = \sum_{he\{x,y\}} Algo - G - Score(G', S, \mathcal{M}_{G',S}, h) - Algo - G - Score(G'', S, \mathcal{M}_{G',S}, h)$
22. *Set* $\Gamma_{\bar{G},G''} = \Gamma_{\bar{G},G'} - \Gamma_{(G'',G')}$
23. *Set* $G' = G''$, $\mathcal{M}_{G',S} = \mathcal{M}_{G'',S}$, $\Gamma_{\bar{G},G'} = \Gamma_{\bar{G},G''}$
24. ***Return*** *BestTree*

Algorithm 2 - Algo-Comp-Score

Input: *A gene tree G, a species tree S, and LCA mapping* $\mathcal{M}_{G,S}$

Output: $\mathcal{C}_\Gamma(G,S)$
01. score = 0
02. ***For*** *each* $g \in I(G)$ *in preorder traversal of G,* ***do***
03. *Call score = score +* $Algo - G - Score(G, S, \mathcal{M}_{G,S}, g)$
04. ***If*** Γ *is DC,* ***then***
05. *score = score -* $|I(S)|$
06. ***Return*** *score*

Algorithm 3 - Algo-G-Score

Input: A gene tree G, a species tree S, LCA mapping $\mathcal{M}_{G,S}$, *and* $g \in I(G)$

Output: $\mathcal{C}_\Gamma(G, S, g)$
01. ***If*** Γ *is D,* ***then***

02. **If** $\mathcal{M}(g) \in \mathcal{M}(Ch(g))$, **then**
03. **Return** *1*
04. **ElseIf** Γ*is DL,* **then**
05 $ls = \sum_{h \in Ch(g)} |dp(\mathcal{M}(h)) - dp(\mathcal{M}(g)) - 1|$
06. **If** $\mathcal{M}(g) \in \mathcal{M}(Ch(g))$, **then**
07. **Return** *ls + 1*
08. **Else**
09. **Return** *ls*
10. **Else** //Γ *is DC*
11 **Return** $\sum_{h \in Ch(g)} |dp(\mathcal{M}(h)) - dp(\mathcal{M}(g))|$

Lemma 6. *The R-SEC-Γ problem is correctly solved by Algo-R-SEC-Γ.*

Proof. Lemma 1-5 and Theorem 1-3 directly imply that in order to prove the correctness of algorithm Algo-R-SEC-Γ, it is sufficient to prove that it correctly returns G' of minimum $\Gamma_{\bar{G},G'}$ among all $G' \in V(\mathcal{X})$. We will show that algorithm Algo-R-SEC- accounts each $G' \in V(\mathcal{X})$, correctly computes $\Gamma_{\bar{G},G'}$ for $\Gamma \in \{D, DL, DC\}$, and returns the right G' as output.

From Definition 10, $V(\mathcal{X}) = \text{SPR}_G(v)$. In Algo-R-SEC-Γ, step 1 prunes subtree G_v and regrafts it above the root of G to create $\overline{G}$. Step 5 sets G' to $\overline{G}$. The for-loop in step 6 traverses subtree $\bar{G}_{Sb(v)}$ in preorder. For each traversed vertex $k \neq Ro(\bar{G}_{Sb(v)})$, step 9 builds the tree $G'' := \text{SPR}_G(v, k)$ by applying NNI operation on the last build G''. Each for-loop iteration sets G' to the last build G'' in step 23. $\overline{G}$ and G''s constitute all the trees in $SPR_G(v)$.

For $\overline{G}$, step 2 computes the LCA mapping and step 5 sets $\Gamma_{\bar{G},G'}$ to zero. Following Lemma 2-4 and Theorem 2, step 10 and 11 update the LCA of G'' and step 12 computes $\Gamma_{\{G',G''\}}$ by calling algorithm Algo-G-Score. Depending on $\Gamma \in \{D, DL, DC\}$, there are three cases:

Case 1: Γ **is D**. Algo-G-Score returns 1, if the vertex $g \in V(G'')$ maps to the same vertex in S as any of its children maps to, otherwise 0.

Case 2: Γ **is DL**. Algo-G-Score computes losses by applying the formula of Definition 4. Further, it adds 1 if there is a duplication.

Case 3: Γ **is DC**. Algo-G-Score, returns the number of lineages from g to each of its children $h \in Ch(g)$ in S. For each $h \in Ch(g)$, depth of $\mathcal{M}(g)$ is subtracted from depth of $\mathcal{M}(h)$ to count number of edges between $\mathcal{M}(g)$ and $\mathcal{M}(h)$.

In Algo-R-SEC-Γ, step 13 computes $\Gamma_{\overline{G},G''}$ by adding $\Gamma_{\overline{G},G'}$ and $\Gamma_{\{G',G''\}}$. When backtracking, steps 17-22 are executed to restore the right G' to compute the next unique $G'' \in Ch_{\mathcal{X}}(G)$. This ensures that the correct $\Gamma_{\bar{G},G'}$ is computed for each $G' \in V(\mathcal{X})$.

In Algo-R-SEC-Γ, step 4 sets $\overline{G}$ as the BestTree and $\Gamma_{\bar{G},\overline{G}} = 0$ as BestScore. Every time a new $G'' \in Ch_{\mathcal{X}}(G)$ is encountered, step 14 compares $\Gamma_{\bar{G},G''}$ with BestScore, and updates BestTree with G'' of the minimum $\Gamma_{\bar{G},G''}$. After the for-loop, step 24 returns the BestTree. □

Lemma 7. *The R-SEC-Γ and SEC-Γ problems can be solved in* $\Theta(n)$ *and* $\Theta(n^2)$ *time, respectively.*

Proof. We will prove that the algorithm Algo-R-SEC-Γsolves the restricted SPR based error-correction problems for each $\Gamma \in \{D, DL, DC\}$ in $\Theta(n)$ time. In Algo-R-SEC-Γ, step 1 takes constant time. Step 2 precomputes LCA values for species tree in O(n) time [24], and so, finds LCA mapping from $\overline{G}$ to S in $O(n)$ time in bottom-up manner. Step 3 computes the duplication, duplication-loss or deep coalescence score of $\overline{G}$ and S by calling Algo-Comp-Score. In Algo-Comp-Score, step 1 and step 2 runs for $O(1)$ and $O(n)$ time, respectively. Step 3 calls Algo-G-Score in each iteration of for-loop. Algo-G-Score runs for $O(1)$ time for $\Gamma \in \{D, DL, DC\}$.

When Γ is DC, steps 4 and 5 are further executed in Algo-Comp-Score for constant time. Thus in Algo-R-SEC-Γ, step 3 runs for $O(n)$ time. Further, steps 4 and 5 take constant time. The loop of step 6 runs for $\Theta(n)$ time. If condition of step 7 is true, steps 8-10 executes in constant time. With precomputed LCA values from step 2, step 11 executes in constant time. Algo-G-Score runs for constant time for $\Gamma \in \{D, DL, DC\}$, and lets step 12 to execute in constant time. Further, steps 13-15 execute for constant time too. If the condition in step 7 is false, then steps 17-22 execute in constant time, similarly. Finally, step 23 runs for constant time, and hence, the R-SEC-Γ problem can be solved in $\Theta(n)$ time. From Observation 1, Algo-R-SEC-Γ is called $\Theta(n)$ time to solve SEC-Γ problem. Thus, the SEC-Γ problem can be solved in $\Theta(n^2)$ time. □

Solving the TEC-Γ problems

In this section we study the TBR based error-correction problems, for duplication (D), duplication-loss (DL), and deep coalescence (DC). More precisely, we extend our solution for the SEC-Γ problems to solve the TEC-Γ problems. A TBR operation can be viewed as an SPR operation, except that the pruned subtree can be rerooted before it is regrafted. Our speed-up for the SEC-Γ problems is achieved by observing that the scores Γof any re-rooted pruned subtree and its remaining pruned tree are independent of each other. We define the R-TEC-Γ problems for the TEC-Γ problems, as we defined the R-SEC-Γ problems for the SEC-Γ problems. We will show that the R-TEC-Γ problems can be solved by solving two smaller problems separately and combining their solutions.

Definition 12. *Let T be a tree and $x \in V(T)$. $RR(T, x)$ is defined to be the tree T , if $x = Ro(T)$ or $x \in Ch(Ro(T))$. Otherwise, $RR(T, x)$ is the tree obtained by suppressing $Ro(T)$, and subdividing the edge $(Pa(x), x)$ by the new root node.*

Lemma 8. *Given a tuple* $\langle G, S, v\rangle$, and G":= $TBR_G(v, x, y)$, *for* $x \in V(G_v)$, $y \in V(G)\backslash V(G_v)$. *Then,* $\mathcal{C}_\Gamma(G'', S) \leq_{G' \in TBR_G(v)} \mathcal{C}_\Gamma(G', S)$ *iff* $\mathcal{C}_\Gamma(RR(G_v, x), S) \leq_{x' \in V(G_v)} \mathcal{C}_\Gamma(RR(G_v, x'), S)$ *and* $\mathcal{C}_\Gamma(G'', S) \leq_{G' \in TBR_G(v,x)} \mathcal{C}_\Gamma(G', S)$.

Proof. ($\Rightarrow$) Let $G^1 := TBR_G(v, x_1, y)$, for $x_1 \in V(G_v)$, and $x_1 \neq x$. Now observe that, $\forall g \in V(G)\backslash V(G_v)$, $\mathcal{C}_\Gamma(G'', S, g) = \mathcal{C}_\Gamma(G^1, S, g)$ Also, let $G^2 := TBR_G(v, x, y_1)$, for $y_1 \in V(G)\backslash V(G_v)$, and $y_1 \neq y$. Observe that, $\forall g \in V(G_v)$, $\mathcal{C}_\Gamma(G'', S, g) = C_\Gamma(G^2, S, g)$. Thus, if G" gives the minimum duplication, duplication-loss, or deep coalescence score among all trees in $TBR_G(v)$, then the score contribution of vertices in $V(G_v)$ and $V(G)\backslash V(G_v)$ is independent. Now looking at vertices of G, the best score is achieved when G_v is rooted at x, i.e. $\mathcal{C}_\Gamma(RR(G_v, x), S) \leq_{x' \in V(G_v)} \mathcal{C}_\Gamma(RR(G_v, x'), S)$; also the best score is achieved when $RR(G_v, x)$ is regrafted at y, i. e., $\mathcal{C}_\Gamma(G'', S) \leq_{G' \in \mathrm{TBR}_G(v,x)} \mathcal{C}_\Gamma(G', S)$. ($\Leftarrow$). This follows similarly. □

Lemma 8 implies that a tree in $\mathrm{TBR}_G(v)$ with the minimum duplication, duplication-loss, or deep coalescence cost can be obtained by optimizing the rooting for the pruned subtree, and the regraft location, separately. A best rooting for the pruned subtree is linear time computable [17,25], and the solution to the R-SEC problem identifies a best regraft location in $\Theta(n)$ time. This allows to obtain a tree in $\mathrm{TBR}_G(v)$ with the minimum duplication, duplication-loss, or deep coalescence cost by evaluating only $\Theta(n)$ trees. Thus the R-TEC-Γ problem can be solved in Θ(n) time. The TEC-Γ problem can be solved by calling the solution of R-TEC-Γ problem $\Theta(n)$ times, and Theorem 4 follows.

Theorem 4. *The TEC-*Γ *problem can be solved in* $\Theta(n^2)$ *time.*

Experimental results

We tested the performance of the gene tree rearrangement algorithms on a set of 106 gene alignments containing sequences from 8 yeast taxa from Rokas et al. [26]. There is a well accepted phylogeny for the yeast species, and the data set has been used to test algorithms for gene tree parsimony based on the deep coalescence problem [27,28]. We constructed maximum likelihood gene trees for each gene using RAxML-VI-HPC version 7.0.4 [29], the gene trees were rooted with the outgroup *Candida albicans*. We used the new error correction algorithms to examine how much a single SPR rearrangement in the gene tree reduces the reconciliation cost based on deep coalescence and also gene duplications and losses. Over all genes the SPR error correction reduced the total deep coalescence cost from 151 to 53 (Table 1) and the duplication and loss cost from 481 to 175 (Table 2). Both the algorithms took only seconds to run for all 106 genes on a standard laptop.

Table 1 Error correction based on deep coalescence model

Reconciliation Cost	Original	Post-Correction
0	45	77
1	32	15
2	6	8
3	9	5
4	8	0
>4	6	1

The number of yeast gene trees with different reconciliation costs based on the deep coalescence model both before (Original) and after (Post-Correction) the SPR error correction.

We also implemented a protocol to use the gene rearrangement algorithm to correct for gene tree error in gene tree parsimony phylogenetic analyses. We first took a collection of input gene trees and performed a SPR species tree search using Duptree [30], which seeks the species tree with the minimum gene duplication cost. We used the duplication only cost (instead of duplications and losses) because when there is no complete sampling of all existing genes, the loss estimates may be inflated by missing sequences. After finding the locally optimal species tree, we used our SPR gene tree rearrangement algorithm to find gene tree topologies with a lower duplication cost. We then performed another SPR species tree search using Duptree, starting from the locally optimal species tree and using the new gene tree topologies. This search strategy is similar to re-rooting protocol in Duptree, which checks for better gene tree roots after a SPR species tree search [30,31]. We tested this protocol on data set of 6,084 genes (with a combined 81,525 leaves) from 14 seed plant taxa. This is the same data set used by [31], except that all gene tree clades containing sequences from a single species were collapsed to a single leaf. Our original SPR tree search found a species tree with 23,500 duplications. The SPR tree search after the gene tree rearrangements identified the same species tree, but the new gene trees had a reconciliation cost of only 18,213. This tree search

Table 2 Error correction based on duplication and loss model

Reconciliation Cost	Original	Post-Correction
0	45	77
1-5	32	15
6-10	15	13
11-15	8	0
16-20	5	1
>20	1	0

The number of yeast gene trees with different reconciliation costs based on the duplication and loss model both before (Original) and after (Post-Correction) the SPR error correction.

protocol took just under 4 hours on a Mac Powerbook with a 2 GHz Intel Core 2 Duo processor and 2 GB memory.

Conclusion

GT-ST reconciliation provides a powerful approach to study the patterns and processes of gene and genome evolution. Yet it can be thwarted by the error that is an inherent part of gene tree inference. Any reliable method for GT-ST reconciliation must account for gene tree error; however, any useful method also must be computationally tractable for large-scale genomic data. We introduce fast and effective algorithms to correct error in the gene trees. These algorithms, based on SPR and TBR rearrangements, greatly extend upon the range of possible errors in the gene tree from existing algorithms [17,18], while remaining fast enough to use on data sets with thousands of genes. These algorithms can correct trees based on a broad variety of evolutionary factors that can cause conflict between gene trees and species trees, including gene duplication, duplications and losses, and deep coalescence.

Our analysis on 106 yeast gene trees demonstrates that even a single SPR correction on the gene trees can radically improve upon the reconciliation cost. Our results in the yeast analysis are very similar to the 2-3 fold improvement in implied duplications and losses reported from the parametric gene tree estimation and reconciliation method of Rasmussen and Kellis [5]. However, in contrast, to this computationally complex method, the gene tree rearrangement algorithm is extremely fast, does not require assumption about the rates of duplication and loss, and is also amenable to corrections based on deep coalescence and duplications as well as duplications and losses. We do not claim that the gene correction algorithms produce a more accurate reconciliation than these parametric methods. However, they do present an extremely fast and flexible alternative.

We also demonstrated that this error correction protocol could easily be incorporated into a gene tree parsimony phylogenetic analysis. Previous studies have emphasized that gene tree parsimony is sensitive to the topology of the input trees. For example, the species tree may differ whether the gene trees are made using parsimony or maximum likelihood [8,10]. In our study, although the gene tree rearrangement did not affect the species tree inference, it did greatly reduce the gene duplication reconciliation cost.

While the results of the experiments are promising, they also suggest several directions for future research. First, further investigation is needed to characterize the effects of error on gene tree topologies. For example, it seems likely that gene tree errors may extend beyond a single SPR or TBR neighborhood. Yet, if we allow unlimited rearrangements, the gene trees will simply converge on the species tree topology. One simple improvement may be to weight the possible gene tree rearrangements based on support for different clades in the gene tree. Thus, well-supported clades may be rarely or never be subject to rearrangement, while poorly supported clades may be subject to extensive rearrangements. Finally, these approaches implicitly assume that all differences between gene trees and species trees are due to either coalescence, duplications, or duplications and losses. Future work will seek to combine these objectives and also address lateral transfer.

Acknowledgements

The authors thank André Wehe for his generous support with the implementation. This work was conducted in parts with support from the Gene Tree Reconciliation Working Group at NIMBioS through NSF award EF-0832858, with additional support from the University of Tennessee. R.C. and O.E. were supported in parts by NSF awards #0830012 and #10117189.
This article has been published as part of *BMC Bioinformatics* Volume 13 Supplement 10, 2012: "Selected articles from the 7th International Symposium on Bioinformatics Research and Applications (ISBRA'11)". The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/13/S10.

Author details

[1]Department of Computer Science, Iowa State University, Ames, IA 50011, USA. [2]Department of Biology, University of Florida, Gainesville, FL 32611, USA.

Authors' contributions

RC was responsible for algorithm design and program implementation, and wrote major parts of the paper. JGB performed the experimental evaluation and the analysis of the results, and contributed to the writing of the paper. OE supervised the project, contributed to the algorithmic design and writing of the paper.
All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Published: 25 June 2012

References

1. Maddison WP: **Gene Trees in Species Trees.** *Systematic Biology* 1997, **46**:523-536.
2. Goodman M, Czelusniak J, Moore GW, Romero-Herrera AE, Matsuda G: **Fitting the gene lineage into its species lineage. A parsimony strategy illustrated by cladograms constructed from globin sequences.** *Systematic Zoology* 1979, **28**:132-163.
3. Guigó R, Muchnik I, Smith TF: **Reconstruction of Ancient Molecular Phylogeny.** *Molecular Phylogenetics and Evolution* 1996, **6(2)**:189-213.
4. Slowinski JB, Knight A, Rooney AP: **Inferring Species Trees from Gene Trees: A Phylogenetic Analysis of the Elapidae (Serpentes) Based on the Amino Acid Sequences of Venom Proteins.** *Molecular Phylogenetics and Evolution* 1997, **8**:349-362.
5. Rasmussen MD, Kellis M: **A Bayesian approach for fast and accurate gene tree reconstruction.** *Molecular Biology and Evolution* 2011, **28**:273-290.
6. Burleigh JG, Bansal MS, Wehe A, Eulenstein O: **Locating Large-Scale Gene Duplication Events through Reconciled Trees: Implications for Identifying Ancient Polyploidy Events in Plants.** *Journal of Computational Biology* 2009, **16**:1071-1083.
7. Hahn MW: **Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution.** *Genome Biology* 2007, **8**:R141.
8. Burleigh JG, Bansal MS, Eulenstein O, Hartmann S, Wehe A, Vision TJ: **Genome-scale phylogenetics: inferring the plant tree of life from 18,896 discordant gene trees.** *Systematic Biology* 2011, **60(2)**:117-125.

9. Huang H, Knowles LL: **What Is the Danger of the Anomaly Zone for Empirical Phylogenetics?** *Systematic Biology* 2009, **58**:527-536.
10. Sanderson MJ, McMahon MM: **Inferring angiosperm phylogeny from EST data with widespread gene duplication.** *BMC Evolutionary Biology* 2007, **7(suppl 1:S3)**.
11. Berglund-Sonnhammer A, Steffansson P, Betts MJ, Liberles DA: **Optimal Gene Trees from Sequences and Species Trees Using a Soft Interpretation of Parsimony.** *Journal of Molecular Evolution* 2006, **63**:240-250.
12. Vernot B, Stolzer M, Goldman A, Durand D: **Reconciliation with non-binary species trees.** *Computational Systems Bioinformatics* 2007, **53**:441-452.
13. Yu Y, Warnow T, Nakhleh L: **Algorithms for MDC-Based Multi-locus Phylogeny Inference.** In *RECOMB, Volume 6577 of Lecture Notes in Computer Science*. Springer;Bafna V, Sahinalp SC 2011:531-545.
14. Cotton JA, Page RDM: **Going nuclear: gene family evolution and vertebrate phylogeny reconciled.** *P Roy Soc Lond B Biol* 2002, **269**:1555-1561.
15. Joly S, Bruneau A: **Measuring Branch Support in Species Trees Obtained by Gene Tree Parsimony.** *Systematic Biology* 2009, **58**:100-113.
16. Arvestad L, Berglund A, Lagergren J, Sennblad B: **Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution.** *RECOMB* 2004, 326-335.
17. Chen K, Durand D, Farach-Colton M: **Notung: a program for dating gene duplications and optimizing gene family trees.** *Journal of Computational Biology* 2000, **7**:429-447.
18. Durand D, Halldórsson BV, Vernot B: **A Hybrid Micro-Macroevolutionary Approach to Gene Tree Reconstruction.** *Journal of Computational Biology* 2006, **13(2)**:320-335.
19. Allen BL, Steel M: **Subtree transfer operations and their induced metrics on evolutionary trees.** *Annals of Combinatorics* 2001, **5**:1-13.
20. Bordewich M, Semple C: **On the computational complexity of the rooted subtree prune and regraft distance.** *Annals of Combinatorics* 2004, **8**:409-423.
21. Zhang L: **On a Mirkin-Muchnik-Smith Conjecture for Comparing Molecular Phylogenies.** *Journal of Computational Biology* 1997, **4(2)**:177-187.
22. Page RDM: **Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas.** *Systematic Biology* 1994, **43**:58-77.
23. Eulenstein O: **Predictions of gene-duplications and their phylogenetic development.** *PhD thesis* , University of Bonn, Germany 1998. [GMD Research Series No. 20/1998, ISSN: 1435-2699].
24. Bender MA, Farach-Colton M: **The LCA Problem Revisited.** *LATIN* 2000, 88-94.
25. Górecki P, Tiuryn J: **Inferring phylogeny from whole genomes.** *ECCB (Supplement of Bioinformatics)* 2006, 116-122.
26. Rokas A, Williams BL, King N, Carroll SB: **Genome-scale approaches to resolving incongruence in molecular phylogenies.** *Nature* 2003, **425(6960)**:798-804.
27. Than C, Nakhleh L: **Species tree inference by minimizing deep coalescences.** *PLoS Comput Biol* 2009, **5(9)**:e1000501.
28. Bansal MS, Burleigh JG, Eulenstein O: **Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models.** *BMC Bioinformatics* 2010, **11(Suppl 1)**:S42.
29. Stamatakis A: **RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models.** *Bioinformatics* 2006, **22(21)**:2688-2690.
30. Wehe A, Bansal MS, Burleigh JG, Eulenstein O: **DupTree: a program for large-scale phylogenetic analyses using gene tree parsimony.** *Bioinformatics* 2008, **24(13)**.
31. Chang W, Burleigh JG, Fernández-Baca D, Eulenstein O: **An ILP solution for the gene duplication problem.** *BMC Bioinformatics* 2011, **12(Suppl 1)**:S14.

doi:10.1186/1471-2105-13-S10-S11
Cite this article as: Chaudhary *et al.*: Efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence. *BMC Bioinformatics* 2012 **13**(Suppl 10):S11.