Górecki and Eulenstein *BMC Bioinformatics* 2012, **13**(Suppl 10):S14
http://www.biomedcentral.com/1471-2105/13/S10/S14

BMC Bioinformatics

PROCEEDINGS Open Access

Algorithms: simultaneous error-correction and rooting for gene tree reconciliation and the gene duplication problem

Pawel Górecki[1*], Oliver Eulenstein[2]

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

Abstract

Background: Evolutionary methods are increasingly challenged by the wealth of fast growing resources of genomic sequence information. Evolutionary events, like gene duplication, loss, and deep coalescence, account more then ever for incongruence between gene trees and the actual species tree. Gene tree reconciliation is addressing this fundamental problem by invoking the minimum number of gene duplication and losses that reconcile a rooted gene tree with a rooted species tree. However, the reconciliation process is highly sensitive to topological error or wrong rooting of the gene tree, a condition that is not met by most gene trees in practice. Thus, despite the promises of gene tree reconciliation, its applicability in practice is severely limited.

Results: We introduce the problem of reconciling unrooted and erroneous gene trees by simultaneously rooting and error-correcting them, and describe an efficient algorithm for this problem. Moreover, we introduce an error-corrected version of the gene duplication problem, a standard application of gene tree reconciliation. We introduce an effective heuristic for our error-corrected version of the gene duplication problem, given that the original version of this problem is NP-hard. Our experimental results suggest that our error-correcting approaches for unrooted input trees can significantly improve on the accuracy of gene tree reconciliation, and the species tree inference under the gene duplication problem. Furthermore, the efficiency of our algorithm for error-correcting reconciliation is capable of handling truly large-scale phylogenetic studies.

Conclusions: Our presented error-correction approach is a crucial step towards making gene tree reconciliation more robust, and thus to improve on the accuracy of applications that fundamentally rely on gene tree reconciliation, like the inference of gene-duplication supertrees.

Background

The wealth of newly sequenced genomes has provided us with an unprecedented resource of information for phylogenetic studies that will have extensive implications for a host of issues in biology, ecology, and medicine, and promise even more. Yet, before such phylogenies can be reliably inferred, challenging problems that came along with the newly sequenced genomes have to be overcome. Evolutionary biologists have long realized that gene-duplication and subsequent loss, a fundamental evolutionary process [1], can largely obfuscate phylogenetic inference [2]. Gene-duplication can form complex evolutionary histories of genes, called gene trees, whose topologies are traditionally used to derive species trees. This approach relies on the assumption that the topologies from gene trees are consistent with the topology of the species tree. However, frequently genes that evolve from different copies of ancestral gene-duplications can become extinct and result in gene trees with correct topologies that are inconsistent with the topology of the actual species tree (see Figure 1). In many such cases phylogenetic information from the gene trees is indispensable and may still be recovered using gene tree reconciliation.

* Correspondence: gorecki@mimuw.edu.pl
[1]Institute of Informatics, University of Warsaw, Warsaw, 02-097, Poland
Full list of author information is available at the end of the article

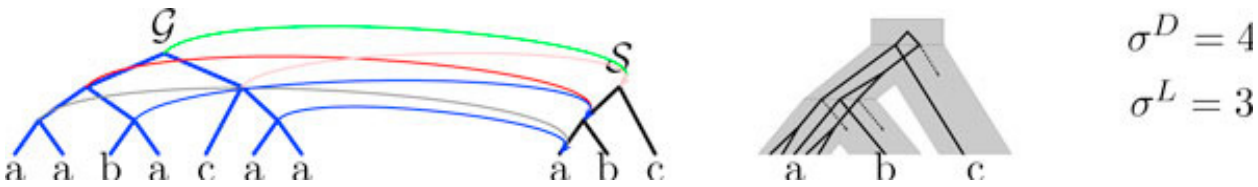


Figure 1 Rooted reconciliation. An lca-mapping *M* from the gene tree $\mathcal{G}$ into the species tree $\mathcal{S}$ and the corresponding embedding. *M* is shown for the internal nodes of $\mathcal{G}$.

Related work

Gene tree reconciliation is a well-studied method for resolving topological inconsistencies between a gene tree and a trusted species tree [2-7]. Inconsistencies are resolved by invoking gene-duplication and loss events that reconcile the gene tree to be consistent with the actual species tree. Such events do not only reconcile gene trees, but also lay foundation for a variety of evolutionary applications including ortholog/paralog annotation of genes, locating episodes of gene-duplications in species trees [8-10], reconstructing domain decompositions [11], and species supertree construction [8,12-14].

A major problem in the application of gene tree reconciliation is its high sensitivity to error-prone gene trees. Even seemingly insignificant errors can largely mislead the reconciliation process and, typically undetected, infer incorrect phylogenies (e.g., [7,15]). Errors in gene trees are often topological errors and rooting errors. Topological error results in an incorrect topology of the gene tree that can be caused by the inference process (e.g. noise in the underlying sequence data) or the inference method itself (e.g. heuristic results). This problem has been addressed for rooted gene trees by 'correcting the error'; that is, editing the given tree such that the number of invoked gene-duplications and losses is minimized [16,17]. However, most inference methods used in practice return only unrooted gene trees (e.g. parsimony and maximum likelihood based methods) that have to be rooted for the gene tree reconciliation process. Rooting error is a wrongly chosen root in an unrooted gene tree. Whereas rooting can be typically achieved in species trees by outgroup analysis, this approach may not be possible for gene trees if there is a history of gene duplication and loss [7]. Other rooting approaches like midpoint rooting or molecular clock rooting assume a constant rate of evolution that is often unrealistic. However, rooting problems can be bypassed by identifying roots that minimize the invoked number of gene duplications and losses [7,16-19].

In summary, even small topological error or a slightly misplaced root can incorrectly identify enormous numbers of gene duplications and losses, and therefore largely mislead the reconciliation process. Therefore, gene tree reconciliation requires gene trees that are free of error and correctly rooted at the same time [5]. However, as previous work has incorporated topological error-correction only separately from correctly rooting gene trees into the reconciliation process [16,18], this process can still be misled.

Our contribution

We address the problem of reconciling erroneous and unrooted gene trees by error-correcting and rooting them at the same time. Solving this problem efficiently is a crucial step towards making gene tree reconciliation more robust, and thus to improve on the accuracy of applications that rely on gene tree reconciliation like the construction of gene-duplication supertrees. We introduce the problem and design an efficient algorithm that facilitates a much more precise gene tree reconciliation, even for large-scale data sets. Our algorithm detects and corrects errors in unrooted gene trees, and thus we avoid the biologists' difficulty and uncertainty of handling erroneous gene trees and correctly rooting them. The presented experimental results suggest that our novel reconciliation algorithms can identify and correct topological error in unrooted input gene trees, and at the same time root them optimally.

Our algorithm is designed to search for the correct and rooted tree of a given unrooted tree in local search neighborhoods of the given tree. The size of these neighborhoods is described by a positive integer k that allows to fine-tune the search. While in theory k can be large it is assumed that gene trees have only small topological error, which typically can be captured by small values of k. For a fixed but freely choosable integer k the runtime of our algorithm is $O(l^k + \max(n, m))$, where n and m is the size of the gene tree and species tree respectively, and l is the number of edges in the gene tree that potentially contain an error (such edges will be called *weak*). Thus, for a small error, which is expressed by $k = 1$, our algorithm runs in linear time. Our experiments show that error-correction runs of the algorithm for $k = 3$ are still possible even for trees with large number of weak edges (e.g., $l = 200$) on a standard workstation configuration.

Further, we address the problem of constructing rooted supertrees by reconciling unrooted and erroneous gene trees with assigned weak edges, a key

problem in illuminating the role and effect of gene duplication and loss in shaping the evolution of organisms. We introduce the problem and develop an effective local search heuristic that makes the construction of more accurate supertrees possible and allows a much better postulation of gene duplication histories. Our experimental results demonstrate that our approach is effective in identifying gene duplication histories given erroneous gene trees and producing more accurate supertrees under gene tree reconciliation.

Duplication-loss model

We introduce the fundamentals of the classical duplication-loss model. Our definitions are mostly adopted from [18]. For a more detailed introduction to the duplication-loss model we refer the interested reader to [2,5,10,20].

Let $\mathcal{J}$ be the set of species consisting of $N > 0$ elements. The *unrooted gene tree* is an undirected acyclic graph in which each node has degree 3 (internal nodes) or 1 (leaves), and the leaves are labeled by the elements from $\mathcal{J}$. A *species tree* $\mathcal{S}$ is a rooted binary tree with N leaves uniquely labeled by the elements from $\mathcal{J}$. In some cases, a node of a tree will be referred by "cluster" of labels of its subtree leaves. For instance, a species tree $(a, (b, c))$ has 5 nodes denoted by: *a, b, c, bc* and *abc*. *A rooted gene tree* is a rooted binary tree with leaves labeled by the elements from $\mathcal{J}$. The internal nodes of a tree T we denote by int(T).

Let $\mathcal{S} = \langle V_S, E_S \rangle$ be a *species tree*. $\mathcal{S}$ can be viewed as an upper semilattice with + a binary least upper bound operation and $\top$ the top element, that is, the root. In particular for $a, b \in V_S$, $a < b$ means that a and b are on the same path from the root, with b being closer to the root than a. We define the *comparability predicate* $D(a, b) = 1$, if $a \leq b$ or $b \leq a$ and $D(a, b) = 0$, when a and b are incomparable. The *distance function* $\rho(a, b)$ is used to denote the number of edges on the unique (non-directed) path connecting a and b.

We call distinct nodes $a, b \in V_S$ *siblings* when $a + b$ is a parent of a and b. For $a, b \in V_S$ let $\mathbf{Sb}(a, b)$ be the set of nodes defined by the following recurrent rule: **(i)** $\mathbf{Sb}(a, b) = \emptyset$ if $a = b$ or a and b are siblings, **(ii)** $\mathbf{Sb}(a, b) = \{c\} \cup \mathbf{Sb}(a + c, b)$, if $a < b$ or $a + c < a + b$; here c is the sibling of a, and **(iii)** $\mathbf{Sb}(a, b) = \mathbf{Sb}(b, a)$ otherwise.

By $L(a, b)$ we denote the number of elements in $\mathbf{Sb}(a, b)$. Observe that $L(a, b) = \rho(a, b) - 2 \cdot (1 - D(a, b))$. Let $M : V_\mathcal{G} \to V_S$ be the *least common ancestor (lca) mapping*, from rooted $\mathcal{G}$ into $\mathcal{S}$ that preserves the labeling of the leaves. Formally, if v is a leaf in $\mathcal{G}$ then $M(v)$ is the node in $\mathcal{S}$ labeled by the label of v. If v is internal node in $\mathcal{G}$ with two children a, b, then $M(v) = M(a) + M(b)$. An example is depicted in Figure 1.

In this general setting let us assume that we are given a *cost function* $\xi : V_\mathcal{G} \times V_S \to \mathbf{R}$ which for all nodes $a \in V_S$, $a \in V_S$ assigns a real $\xi(v, a)$ representing a contribution to node a which comes from v when reconciling $\mathcal{G}$ with $\mathcal{S}$. Having ξ we can define $k(v) = \sum_a \xi(v, a)$ to be a total contribution from v in the reconciliation of $\mathcal{G}$ with $\mathcal{S}$. We call κ a *contribution* function. Finally, $\sigma = \sum_v k(v)$ is the total cost of reconciliation of $\mathcal{G}$ with $\mathcal{S}$.

Now we present examples of cost functions that are used in the duplication model. We assume that if v is an internal node in $\mathcal{G}$ then w_1 and w_2 are its children. The *Duplication cost* function is defined as follows: $\xi^D(v, a) = 1$ if $v \in \text{int}(\mathcal{G})$ and $M(v) = M(w_i) = a$ for some i, and $\xi^D(v, a) = 0$ otherwise. The *Loss cost* function: $\xi^L(v, a) = 1$ if $v \in \text{int}(\mathcal{G})$ and $a \in \mathbf{Sb}(M(w_1), M(w_2))$, and $\xi^L(v, a) = 0$ otherwise. It can be proved that if $v \in \text{int}(\mathcal{G})$ then $\kappa^D(v) = D(M(w_1), M(w_2))$ and $\kappa^L(v) = L(M(w_1), M(w_2))$ (in both cases 0 if v is a leaf).

The *Duplication cost* function is defined as follows: $\xi^D(v, a) = 1$ if $v \in \text{int}(\mathcal{G})$ and $M(v) = M(w_i) = a$ for some i, and $\xi^D(v, a) = 0$ otherwise. Loss cost function: $\xi^L(v, a) = 1$ if $v \in \text{int}(\mathcal{G})$ and $a \in \mathbf{Sb}(M(w_1), M(w_2))$, and $\xi^L(v, a) = 0$ otherwise. It can be proved that if $v \in \text{int}(\mathcal{G})$ then $\kappa^D(v) = D(M(w_1), M(w_2))$ and $\kappa^L(v) = L(M(w_1), M(w_2))$ (in both cases 0 if v is a leaf).

Observe that a node $v \in V_\mathcal{G}$ is called a duplication [4,13] if $\kappa^D(v) = 1$. Moreover, $\kappa^L(v) = l(v)$, where $l(v)$ is the number of gene losses associated to v. It can be proved that σ^D and σ^L are the minimal number of gene duplications and gene losses (respectively) required to reconcile (or to embed) $\mathcal{G}$ with $\mathcal{S}$. Please refer to [18] for more details. The example of an embedding is depicted in Figure 1.

Introduction to unrooted reconciliation

Here we highlight some results from [18] that are used for the design of our algorithm. From now on, we assume that $\mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G} \rangle$ is an unrooted gene tree. We define a rooting of $\mathcal{G}$ by selecting an edge $e \in E_\mathcal{G}$ on which the root is to be placed. Such a rooted tree will be denoted by $\mathcal{G}_e$, where v_* is a new node defining the root. To distinguish between rootings of $\mathcal{G}$, the symbols defined in previous section for rooted gene trees will be extended by inserting index e. Please observe, that the mapping of the root of $\mathcal{G}_e$ is independent of e. Without loss of generality the following is assumed: **(A1)** $\mathcal{S}$ and $\mathcal{G}$ have at least one internal node and **(A2)** $M_e(v_*) = \top$; that is, the root of every rooting is mapped into the root of $\mathcal{S}$ (we may always consider the subtree of the species tree rooted in $M_e(v_*)$ with no change of the cost).

First, we transform $\mathcal{G}$ into a directed graph $\widehat{\mathcal{G}} = \langle V_\mathcal{G}, \widehat{E_\mathcal{G}} \rangle$ where $\widehat{E_\mathcal{G}} = \{\langle v, w \rangle | \{v, w\} \in E_\mathcal{G}\}$. In other

words each edge $\langle v, w\rangle$ in $\mathcal{G}$ is replaced in $\widehat{\mathcal{G}}$ by a pair of directed edges $\langle v, w\rangle$ and $\langle w, v\rangle$.

Edges in $\widehat{\mathcal{G}}$ are labeled by nodes of $\mathcal{S}$ as follows. If $v \in V_{\mathcal{G}}$ is a leaf labeled by a, then the edge $\langle v, w\rangle \in \widehat{E_{\mathcal{G}}}$ is labeled by a. When v is an internal node in $\widehat{\mathcal{G}}$ we assume that $\langle w_1, v\rangle$ and $\langle w_2, v\rangle$ are labeled by b_1 and b_2, respectively. Then the edge $\langle v, w_3\rangle \in \widehat{E_{\mathcal{G}}}$, such that $w_3 \neq w_1$ and $w_3 \neq w_2$ is labeled by $b_1 + b_2$. Such labeling will be used to explore mappings of rootings of $\mathcal{G}$. An edge $\{v, w\}$ in $\mathcal{G}$ is called *asymmetric* if exactly one of the labels of $\langle v, w\rangle$ and $\langle w, v\rangle$ in $\widehat{\mathcal{G}}$ is equal to $\top$, otherwise it is called *symmetric*.

Every internal node v, and its neighbors in $\widehat{\mathcal{G}}$ define a subtree of $\widehat{E_{\mathcal{G}}}$, called a *star* with a center v, as depicted in Figure 2. The edges $\langle v, w_i\rangle$ are called *outgoing*, while the edges $\langle w_i, v\rangle$ are called *incoming*. We will refer to the undirected edge $\{v, w_i\}$ as e_i, for $i = 1, 2, 3$.

The are several types of possible star topologies based on the labeling (for proofs and details see [18]): (S1) a star has one incoming edge labeled by $\top$ and two outgoing edges labeled $\top$ and these edges are connected to the three siblings of the center, (S2) a star has exactly two outgoing edges labeled by $\top$, (S3) a star has all outgoing edges and exactly one incoming edgd labeled by $\top$, (S4) a star has all edges labelled by *top*, and (S5) a star has all outgoing edges and exactly two incoming edges labeled by $\top$. Figure 2 illustrates the star topologies.

In summary stars are basic 'puzzle-like' units that can be used to assemble them into unrooted gene trees. However, not all star compositions represent a gene tree. For instance, there is no gene tree with 3 stars of type S2. It follows from [18] (see Lemma 4) that we need the following additional condition: (C1) if a gene tree has two stars of type S2 then they share a common edge.

Now we overview the main result of [18] (see Theorem 1 for more details). Let $\mathcal{S}$ be a species tree and $\mathcal{G}$ be unrooted gene tree. The set of optimal edges, that is, candidates for best rootings, is defined as follows: $\mathbf{Min}_{\mathcal{G}} = \{e \in E_{\mathcal{G}} | \sigma_e^{M_{\alpha,\beta}}$ is minimal$\}$, where $\sigma_e^{M_{\alpha,\beta}}$ is the total cost for the weighted mutation cost defined by , e is an edge in $\mathcal{G}$ and α, β are two positive reals. Then **(M1)** if $|\mathbf{Min}_{\mathcal{G}}| > 1$, then $\mathbf{Min}_{\mathcal{G}}$ consists of all edges present in all stars of type S4 or S5, **(M2)** if $|\mathbf{Min}_{\mathcal{G}}| = 1$, then $\mathbf{Min}_{\mathcal{G}}$ contains exactly one symmetric edge that is present in star of type S2 or S3. From the above statements, (C1) and star topologies we can easily determine $\mathbf{Min}_{\mathcal{G}}$. More precisely, the star edges outside $\mathbf{Min}_{\mathcal{G}}$ are asymmetric and share the same direction. Thus, to find an optimal edge it is sufficient to follow the direction of non $\top$ edges in $\widehat{\mathcal{G}}$.

Now we summarize the time complexity of this procedure. It follows from [21] that a single lca-query (that, is $a + b$ for nodes a and b in $\mathcal{S}$) can be computed in constant time after an initial preprocessing step requiring $O(|\mathcal{S}|)$ time. Other structures like $\widehat{\mathcal{G}}$ with the labeling can be computed in $O(|\mathcal{G}|)$ time. The same complexity has the procedure of finding an optimal edge in $\mathcal{G}$. In summary an optimal edge/rooting and the minimal cost can be computed in linear time. See [18] for more details and other properties.

Methods

First we describe our algorithm for computing the optimal cost and the set of optimal edges after one nearest neighbor interchange (NNI) operation performed on an unrooted gene tree, and then extend it to a general case with k NNI operations. For the definition of NNI please refer to Def. 1 and Figure 3.

Algorithm

Now we show that a single NNI operation can be completed in constant time if all structures required for computing the optimal rootings are already constructed. First, let us assume that the following is given: (a) two positive reals α and β, a species tree $\mathcal{S}$, (b) lca structure for $\mathcal{S}$ that allows to answer lca-queries in constant time, (c) an unrooted gene tree $\mathcal{G}$, (d) $\widehat{\mathcal{G}}$ with the labeling of edges, (e) $\mathbf{Min}_{\mathcal{G}}$ - the set of optimal edges, and (f) σ - the minimal total weighted mutation cost. As observed in the previous section (b),(d)-(f) can be computed in $O(\max(|\mathcal{S}|, |\mathcal{G}|))$. Now we show that (c)-(f)

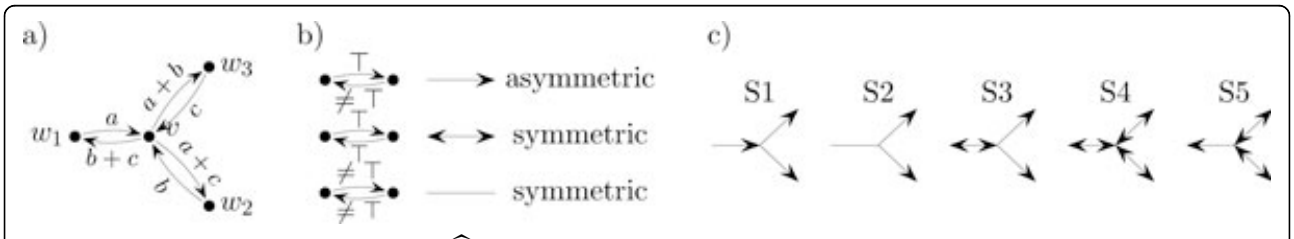


Figure 2 Unrooted reconciliation. a) A star in $\widehat{\mathcal{G}}$. b) Types of edges. c) All possible types of stars. We use simplified notation instead of the full topology.

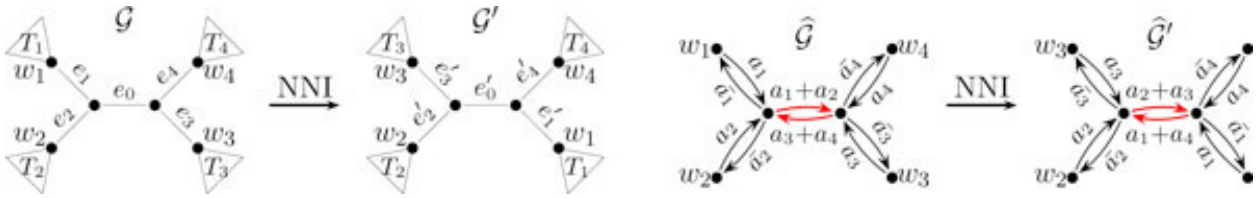


Figure 3 NNI. A single NNI on $\mathcal{G}$ and $\widehat{\mathcal{G}}$. On the left e_i and e'_i (for i = 0, ... , 4) denote edges in $\mathcal{G}$ and its NNI-neighbor $\mathcal{G}'$, respectively. On the right each node a_i denote the labeling of edges in $\widehat{\mathcal{G}}$. Notation $\bar{a}_i$ denote the lca-mapping of complementary subtrees, for instance, $\bar{a}_3 = a_1 + a_2 + a_4$, etc. For brevity, we omit each subtree T_i attached to w_i in the left diagram.

can be computed in constant time after a single NNI operation.

NNI operation (c) and the update of lca-mappings (d).

Definition 1. *(Single NNI operation) An NNI operation transforms a gene tree* $\mathcal{G} = ((T_1, T_2), (T_3, T_4))$ *into* $\mathcal{G}' = ((T_2, T_3), (T_1, T_4))$ *, where* T_i*-s are (rooted) subtrees of* $\mathcal{G}$ *. The edge that connects the roots of* (T_1, T_2) *and* (T_3, T_4) *in* $\mathcal{G}$ *is denoted by* e_0 *and called the* center edge. *For each* i = 1, 2, 3, 4 *we assume the following:* w_i *is the root of* T_i*,* e_i *is the edge connecting* w_i *with* e_0 *and* a_i *is the lca-mapping of* T_i*. Similarly, we define the* center edge e'_0 *and* e'_i *in* $\mathcal{G}'$.

An NNI operation is depicted in Figure 3 with the transformation of $\widehat{\mathcal{G}}$ into $\widehat{\mathcal{G}}'$. The notation will be used from now on. Note that there is a second NNI operation, when $\mathcal{G}$ is replaced with $((T_1, T_3), (T_2, T_4))$. However, it can be easily defined and therefore it is omitted here. Observe that the NNI operation (without updating of lca-mappings) can be performed in constant time for both trees.

The right part of Figure 3 depicts the transformation of $\widehat{\mathcal{G}}$. Observe that the labels of the incoming and outgoing edges attached to each w_i in $\widehat{\mathcal{G}}$ do not change during this operation. Lemma 1 follows directly from this observation.

Lemma 1. *An NNI operation changes only the labels of the center edge.*

We conclude that updating $\widehat{\mathcal{G}}$ requires only two lca-queries, and therefore can be performed in constant time.

Reconstruction of optimal edges (e). We analyze the changes of the optimal set of edges $Min_{\mathcal{G}}$. To this end we consider a number of cases depending on the relation between the optimal set of edges and the set of edges, incident to the nodes of the center edge. Let $C_{\mathcal{G}} = \{e_i\}_{i=0,\ldots,4}$.

For convenience, assume that the NNI operation replaces e_i with e'_i as indicated in Figure 3. We call two disjoint edges from $C_{\mathcal{G}}$ *semi-alternating* if they share a common node after the NNI operation. In Figure 3 $\{e_1, e_4\}$ and $\{e_2, e_3\}$ are semi-alternating. For two edges a and b that are incident to the same node let $\star(a, b)$ be the set of three edges defining the unique star that contains a and b.

Lemma 2. *Assuming that* e_i *is replaced by* e'_i *after the NNI operation the set of optimal edges does not require additional changes if and only if one of the following conditions is satisfied:* **(EQ1)** $\mathbf{Min}_{\mathcal{G}} \cap C_{\mathcal{G}} = \emptyset$,

(EQ2) $\mathbf{Min}_{\mathcal{G}} \supseteq C_{\mathcal{G}}$ *and each pair of semi-alternating edges contains at least one symmetric edge,*

(EQ3) $\mathbf{Min}_{\mathcal{G}}$ *consists of only the center edge,*

(EQ4) $\mathbf{Min}_{\mathcal{G}} \cap C_{\mathcal{G}} = \{e_i\}$ *for some* i >0 *and the center is asymmetric after the NNI operation.*

Proof: (EQ1) All edges in $C_{\mathcal{G}}$ are asymmetric (2 stars S1). Then, after the NNI operation e'_0 is asymmetric and ($C_{\mathcal{G}'}$ has 2 stars S1). (EQ2) $C_{\mathcal{G}}$ consists of 2 stars of type S4/S5 and at most two asymmetric edges. It follows from EQ2 that the asymmetric edges in $C_{\mathcal{G}'}$ cannot form a star of type other than S5. Together with M1 it follows that $C_{\mathcal{G}'}$ is optimal. (EQ3) By M1 the center is symmetric in $\mathcal{G}$. It remains symmetric after NNI. From C1 and M2, $\mathbf{Min}_{\mathcal{G}'}$ consists of the center edge. (EQ4) Note, that the type of $\star(e'_i, e'_0)$ is *S*1, *S*2 or *S*3.

Lemma 3 (NE1). *If* $\mathbf{Min}_{\mathcal{G}} \supseteq C_{\mathcal{G}}$ *and there exists a pair* $\{e_i, e_j\}$ *of asymmetric semi-alternating edges, then* $\mathbf{Min}'_{\mathcal{G}} = \mathbf{Min}_{\mathcal{G}} \setminus C_{\mathcal{G}} \cup (C_{\mathcal{G}'} \setminus \{e'_i, e'_j\})$.

Proof: The type of $\star(e'_i, e'_j)$ is S1 or S3 and the other star has type S4 or S5. By M2 e'_i and e'_j are not optimal.

Lemma 4 (NE2). *If* $\mathbf{Min}_{\mathcal{G}} \cap C_{\mathcal{G}} = \{e_i\}$ *for some* i >0 *and the center is symmetric after the NNI operation then* $\mathbf{Min}'_{\mathcal{G}} = \mathbf{Min}_{\mathcal{G}} \setminus \{e_i\} \cup \star(e'_0, e'_j)$.

Proof: In this case e'_0 has two arrows and $\star(e'_0, e'_i)$ is of type S5.

Lemma 5. *Assume that* $\mathbf{Min}_{\mathcal{G}} \cap C_{\mathcal{G}} = \{e_0, e_i, e_j\}$*, where* $i \neq 0$,

(NE3) *If both* e_i *and* e_j *are symmetric then* $\mathbf{Min}_{\mathcal{G}'} = \mathbf{Min}_{\mathcal{G}} \setminus C_{\mathcal{G}} \cup C_{\mathcal{G}'}$,

(NE4) *If* e_j *is asymmetric and* e'_0 *is symmetric then* $\mathbf{Min}_{\mathcal{G}'} = \mathbf{Min}_{\mathcal{G}} \setminus C_{\mathcal{G}} \cup \star(e'_0, e'_i)$.

(NE5) *If both* e_j *and* e'_0 *are asymmetric then* $\mathbf{Min}_{\mathcal{G}'} = \mathbf{Min}_{\mathcal{G}} \backslash C_{\mathcal{G}} \cup \{e'_i\}$.

Proof: Note that $\{e_0, e_i, e_j\}$ must be a star in $\mathcal{G} \cdot (\text{NE3}) \star (e_i, e_j)$ has type S4 or S5. After the transformation the two stars $\star(e'_0, e'_i)$ and $\star(e'_0, e'_j)$ have type S5. Both are optimal in $\mathcal{G}' \cdot (\text{NE4}) \star (e_i, e_j)$ has type S5. After the transformation $\star(e'_0, e'_i)$ has type S5 and $\star(e'_0, e'_j)$ has type S3. Only the first is optimal in $\mathcal{G}' \cdot (\text{NE5}) \star (e_i, e_j)$ has type S5 while the other star in $C_{\mathcal{G}}$ has type S3. After the transformation only e'_i remains symmetric in $C_{\mathcal{G}'}$ therefore it is the only optimal edge in $C_{\mathcal{G}'}$.

Computing the optimal cost (f). Observe that from Lemmas 2-5 at least one optimal edge remains optimal after the NNI operation. Therefore, to compute the difference in costs between optimal rootings of $\mathcal{G}$ and $\mathcal{G}'$ we start with the cost analysis for the rootings of such edge.

First, we introduce a function for computing the cost differences. Consider three nodes *x, y, z* of some rooted gene tree such that *x* and *y* are siblings and the parent of them (denoted by *xy*), is a sibling of *z*. In other words we can denote this subtree by ((*x, y*), *z*). Then, the partial contribution of ((*x, y*), *z*) to the total weighted mutation cost can be described as follows: $\sum_{a \in \mathcal{S}} \alpha * (\xi^D(xy, a) + \xi^D(xyz, a)) + \beta * (\xi^L(xy, a) + \xi^L(xyz, a))$. Assume that *x, y* and *z* are mapped into *a, b* and *c* (from the species tree), respectively. It can be proved from the defnition of ξ^D and ξ^L that the above contribution equals: $\varphi(a, b, c) = \alpha * (D(a, b) + D(a + b, c)) + \beta * (L(a, b) + L(a + b, c))$. Now, assume that a single NNI operation changes ((*x, y*), *z*)) into (*x*, (*y, z*)). It should be clear that the cost difference is given by: $\Delta_3(a, b, c) = \varphi(c, b, a) - \varphi(a, b, c)$. Similarly, we can define a cost difference when a single NNI operation changes ((*x, y*), (*z, v*)) into ((*x, v*), (*y, z*)). Assume, that *v* is mapped into *d*. Then, the cost contribution of the first subtree is $\varphi'(a, b, c, d) = \varphi(a, b, c + d) + \alpha * (D(c, d) + \beta * L(c, d)$. The cost difference is given by: $\Delta_4(a, b, c, d) = \varphi'(a, d, b, c) - \varphi'(a, b, c, d)$.

Lemma 6. *If the center edge is optimal and remains optimal after the NNI operation then the cost difference equals* $\Delta_4(a_1, a_2, a_3, a_4)$, *where* a_i *(for* $i = 1, 2, 3, 4$*) is the mapping as indicated in* Figure 3.

As mentioned the above lemma can be proved by comparing the rootings placed on the center edges in $\mathcal{G}$ and $\mathcal{G}'$. Lemma 6 gives a solution for cases: EQ2, EQ3, NE1 and NE3. The next lemma gives a solution for the remaining cases.

Lemma 7. *If for some* $i > 0$ *there exists an optimal edge in* $T_i \cup \{e_i\}$ *that remains optimal after the NNI operation (under assumption that* e_i *is replaced by* e'_i*) then the cost difference is* $\Delta_3(a_4, a_3, a_2)$ *if* $i = 1$, $\Delta_3(a_3, a_4, a_1)$ *if* $i = 2$, $\Delta_3(a_2, a_1, a_4)$ *if* $i = 3$ *and* $\Delta_3(a_1, a_2, a_3)$ *if* $i = 4$.

Similarly to Lemma 6 we can prove Lemma 7 by comparing the rootings of e_i and e'_i.

Error correction algorithm. Finally, we can present the algorithm for computing the optimal weighted mutation cost for a given gene tree and its *k*-NNI neighborhood. See Figure 4 for details. It should be clear that the complexity of this algorithm is $O(|\mathcal{G}|^k + \max(|\mathcal{G}|, |\mathcal{S}|))$. We write that a gene tree has *errors* if the optimal cost is computed for one of its NNI variants. Otherwise, we write that a gene tree *does not require corrections*. Please note that it for a special case of $k = 1$, this algorithm is linear in time (see also our preliminary article [22]).

General reconstruction problems

We present several approaches to problems of error correction and phylogeny reconstruction. Let us assume that $\sigma_{\alpha,\beta,k}(\mathcal{S}, \mathcal{G})$ is the cost computed by algorithm from Figure 4, where $\alpha, \beta > 0$, $k \geq 0$, $\mathcal{S}$ is a rooted species tree and $\mathcal{G}$ is an unrooted gene tree.

Problem 1 (*k*NNIC). *Given a rooted species tree* $\mathcal{S}$ *and a set of unrooted gene trees, G compute the total cost* $\sum_{\mathcal{G} \in G} \sigma_{\alpha,\beta,k}(\mathcal{S}, \mathcal{G})$.

The *k*NNIC problem can be solved in polynomial time by an iterative application of our algorithm. Additionally, we can reconstruct the optimal rootings as well as the correct topology of each gene tree. Please note that for $k = 0$ (no error correction), we have the cost inference problem for the reconciliation of an unrooted gene tree with a rooted species tree [18].

Problem 2 (*k*NNIST). *Given a set of unrooted gene trees G find the species tree* $\mathcal{S}$ *that minimizes the total cost* $\sum_{\mathcal{G} \in G} \sigma_{\alpha,\beta,k}(S, \mathcal{G})$.

The complexity of the *k*NNIST problem is unknown. However, similar problems for the duplication model are NP-hard [13]. Therefore we developed heuristics for the *k*NNIST problem to use them in our experiments.

In applications there is typically no need to search over all NNI variants of a gene tree. For instance, a good candidate for an NNI operation is *a weak edge*. A weak edge is usually defined on the basis of its length, where short length indicates weakness. To formalize this property, let us assume that each edge in a gene tree $\mathcal{G}$ has length. We call an edge *e* in $\mathcal{G}$ *weak* if the length of *e* is smaller than ω, where ω is a non-negative real. Now we can define variants of *k*NNIC and *k*NNIST denoted by ω-*k*NNIC and ω-*k*NNIST, respectively, where the NNI operations are performed on weak edges only. These straighforward definitions are omitted. Please note that the time complexity of the algorithm with NNIs limited

1. **Input** A species tree $\mathcal{S}$, an unrooted gene tree $\mathcal{G}$, $\alpha, \beta > 0$, $k \geq 0$.
2. **Output** Optimal weighted cost for $\mathcal{G}$ and its k-NNI neighborhood.
3. **Data preparation** compute: the optimal weighted mutation cost σ, $\mathbf{Min}_{\mathcal{G}}$, lca structure for $\mathcal{S}$ and $\widehat{\mathcal{G}}$ by the unrooted reconciliation algorithm. Let $mincost := \sigma$.
4. **for** each sequence $e^1, \ldots, e^k$ of internal edges in $\mathcal{G}$ **do** $mincost := min(mincost, nnicost(e^1, \ldots, e^k))$.
5. **return** mincost
6. **Procedure** $nnicost(e^1, \ldots, e^j)$
7. **if** j=0 return $+\infty$
8. Transform $\mathcal{G}$ into $\mathcal{G}'$ and $\widehat{\mathcal{G}}$ into $\widehat{\mathcal{G}}'$ (in situ).
9. Update $\mathbf{Min}_{\mathcal{G}}$ according to cases NE1-NE5 and adjust the cost σ (Lemma 6, 7).
10. $mincost := min(mincost, \sigma, nnicost(e^2, \ldots, e^j))$
11. Perform the reverse transformation to reconstruct $\mathcal{G}$, $\widehat{\mathcal{G}}$ and σ.
12. Execute all steps 8-11 for the second NNI operation on e_0.
13. **return** *mincost*

Figure 4 Algorithm. Optimal weighted cost for $\mathcal{G}$ and its k-NNI neighborhood.

to weak edges is $O(l^k + \max(|\mathcal{G}|, |\mathcal{S}|))$, where l is the number of weak edges in $\mathcal{G}$.

Software

The unrooted reconciliation algorithm [18] and its data structures are implemented in program URec [23]. Our algorithm partially depends on theses data structures and therefore was implemented as a significantly extended version of URec. Additionally, we implemented a hill climbing heuristic to solve kNNIST and ω-kNNIST.

Software and datasets from our experiments are made freely available through http://bioputer.mimuw.edu.pl/~gorecki/ec.

Experimental results and discussion

Data preparation

First, we inferred 4133 unrooted gene trees with branch lengths from nine yeast genomes contained in the Genolevures 3 data set [24], which contains protein sequences from the following nine yeast species: *C. glabrata* (4957 protein sequences, abbreviation CAGL), *S. cerevisiae* (5396, SACE), *Z. rouxii* (4840, ZYRO), *S. kluyveri* (5074, SAKL), *K. thermotolerans* (4933, KLTH), *K. lactis* (4851, KLLA), *Y. lipolytica* (4781, YALI), *D. hansenii* (5006, DEHA) and *E. gossypii* (4527, ERGO).

We aligned the protein sequences of each gene family by using the program TCoffee [25] using the default parameter setting. Then maximum likelihood (unrooted) gene trees were computed from the alignments by using proml from the phylip software package. The original species tree of these yeasts [24], here denoted by G3, is shown in Figure 5.

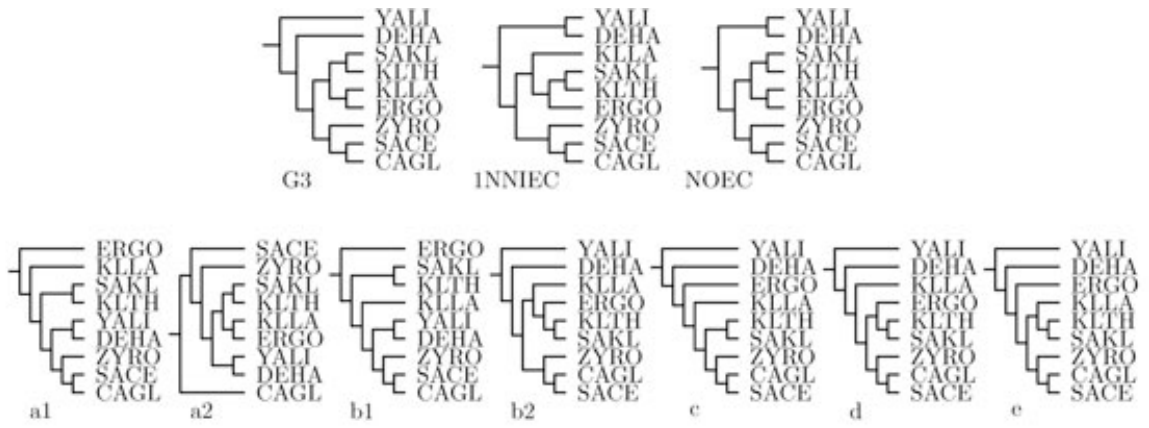
Inferring optimal species trees

The optimal species tree reconstructed with error corrections (1NNIST optimization problem) is depicted in Figure 5 and denoted by 1NNIEC. This tree differs from G3 in the rooting and in the middle clade with KLLA and ERGO. Additionally, we inferred by the heuristic an optimal species tree, denoted here by NOEC, with no error corrections (0NNIST optimization). All the trees from this figure are highly scored in each of the optimization schemas.

From weak edges to species trees

In the previous experiment, the NNI operations were performed on almost every gene tree in the optimal solution and with no restrictions on the edges. In order to reconstruct the trees more accurately, we performed experiments for ω-kNNIST optimization with various ω parameters and subsets of gene trees. The filtering of gene trees was determined by an integer $\mu > 0$ that defines the maximum number of allowed weak edges in a single gene tree. Each gene tree that did not satisfy such condition was rejected.

Figures 6 and 7 depict a summary of error correction experiments for weak edges. For each ω and μ we performed 20 runs of the ω-kNNIST heuristic for finding the optimal species tree in the set of gene trees filtered by μ. The optimal species trees are depicted in the diagram, where each cell represents the result of a single ω-kNNIST experiment. We observed that G3, 1NNIEC and NOEC are significantly well represented in the set of optimal species trees in ω-1NNIST experiments, while in ω-2NNIST and ω-3NNIST experiments only G3 and NOEC were detected. Note that the original yeast phylogeny (G3, black squares in Figures 6 and 7) is inferred for ω = 0.1-0.2 (in other words approx. 30-40% of edges are weak, see Figure 8) and $\mu \geq 10$ in most experiments. In particular for ω = 0.15 and μ = 10, 364 gene trees were rejected (see Figure 9). These results significantly support the G3 phylogeny. Please note that the results for the standard unrooted reconciliation



Tree	0NNIST cost (rank)	1NNIST cost (rank)	1NNIST Errors	2NNIST cost (rank)	2NNIST Errors
G3	66407 (4)	50653 (5)	3680	39119 (4)	6746
1NNIEC	64665 (2)	**48869 (1)**	3565	38088 (2)	6478
NOEC	**64413 (1)**	48909 (2)	3684	**37538 (1)**	6757

Figure 5 Yeasts phylogeny. Species tree topologies. G3 - original phylogeny of Genolevures 3 data set [24]. 1NNIEC - optimal rooted species tree inferred from gene trees with all possible 1-NNI error corrections. NOEC - optimal species tree for the yeast gene trees with no NNI operations (cost 64413, no corrections). Rank denotes a position of a tree on the sorted list of the best trees. The trees below are inferred from other ω-*k*NNIST (see next figures). Please note that NOEC, G3, a1 and a2 are rooted variants of the same unrooted tree. Similar property holds for 1NNIEC, b1 and b2.

algorithms without error correction are located in the first column of diagrams (ω = 0).

From trusted species tree to weak edges in gene trees - automated and manual curation

Assume that the set of unrooted gene trees and the rooted (trusted) species tree $\mathcal{S}$ are given. Then we can state the following problem: find ω and μ such that $\mathcal{S}$ is the optimal species tree in ω-NNIST problem for the set of gene trees filtered by μ. For instance in our dataset, if we assume that G3 is a given correct phylogeny of yeasts, then from the diagrams (Figure 6 and 7) one can determine appropriate values of ω and μ that yield G3 as optimal. In other words we can automatically determine weak edges by ω and filter gene trees by μ. This approach can be applied in tree curation procedures to correct errors in an automated way as well as to find candidates (rejected trees) for further manual curation. For instance, in the previous case, when ω = 0.1 and μ = 10, we have 3164 trees that can be corrected and rooted by our algorithm, while the 364 rejected trees could be candidates for further manual correction.

Discussion

We present novel theoretical and practical results on the problem of error correction and phylogeny reconstruction. In particular, we describe a polynomial time and space algorithm that simultaneously solves the problem of correction topological errors in unrooted gene trees and the problem of rooting unrooted gene trees. The algorithm allows us to perform efficiently experiments on truly large-scale datasets available for yeast genomes. Our experiments suggest that our algorithm can be used to (i) detect errors, (ii) to infer a correct phylogeny of species under the presence of weak edges in gene trees, and (iii) to help in tree curation procedures.

Conclusion

We introduced a novel polynomial time algorithm for error-corrected and unrooted gene tree reconciliation. Experiments on yeast genomes suggests that an implementation of our algorithm can greatly improve on the accuracy of gene tree reconciliation, and thus, curate error-prone gene trees. Moreover, we use our error-corrected reconciliation to make the gene duplication problem, a standard application of gene tree reconciliation, more robust. We conjecture that the error-corrected gene duplication problem is intrinsically hard to solve, since the gene duplication problem is already NP-hard. Therefore, we introduced an effective heuristic for

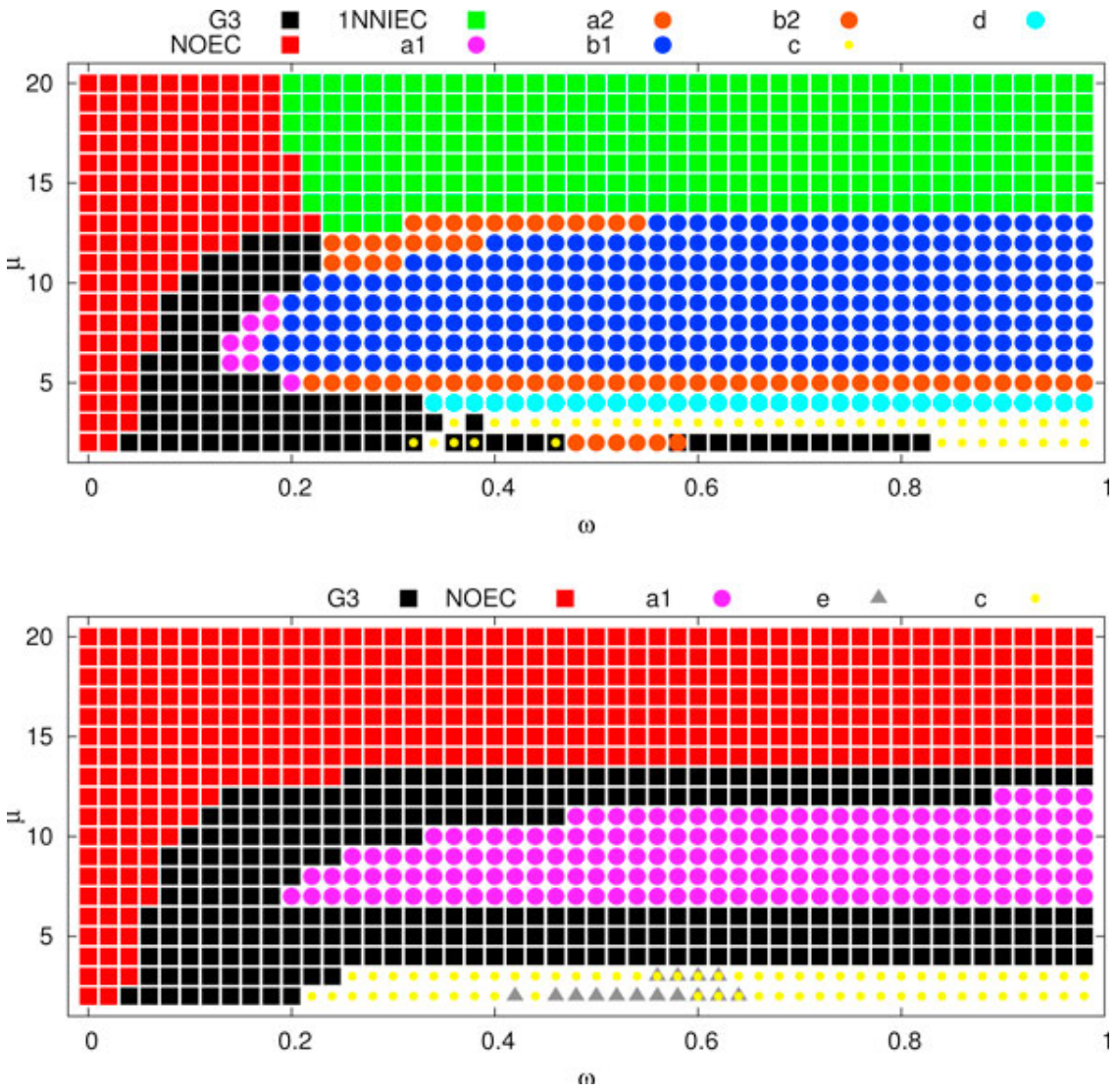


Figure 6 ω-1NNIST and ω-2NNIST experiments. A summary of ω-1NNIST (top) and ω-2NNIST experiments (bottom) for ω = 0, 0.02, 0.04, ... , 0.98, μ = 2, 3, ... , 20. Optimal species trees found by the heuristics. Please note that in some cases two optimal trees were found.

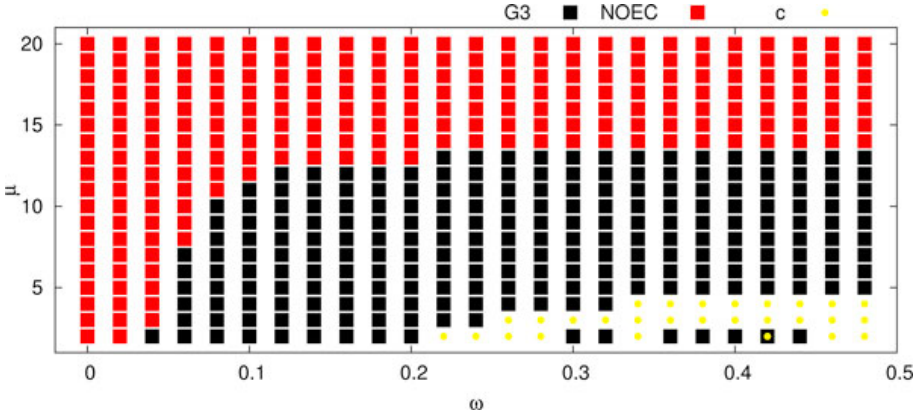


Figure 7 ω-3NNIST experiments. A summary of ω-3NNIST experiments for ω = 0, 0.02, 004, ... , 0.48 and μ = 2, 3, ... , 20.

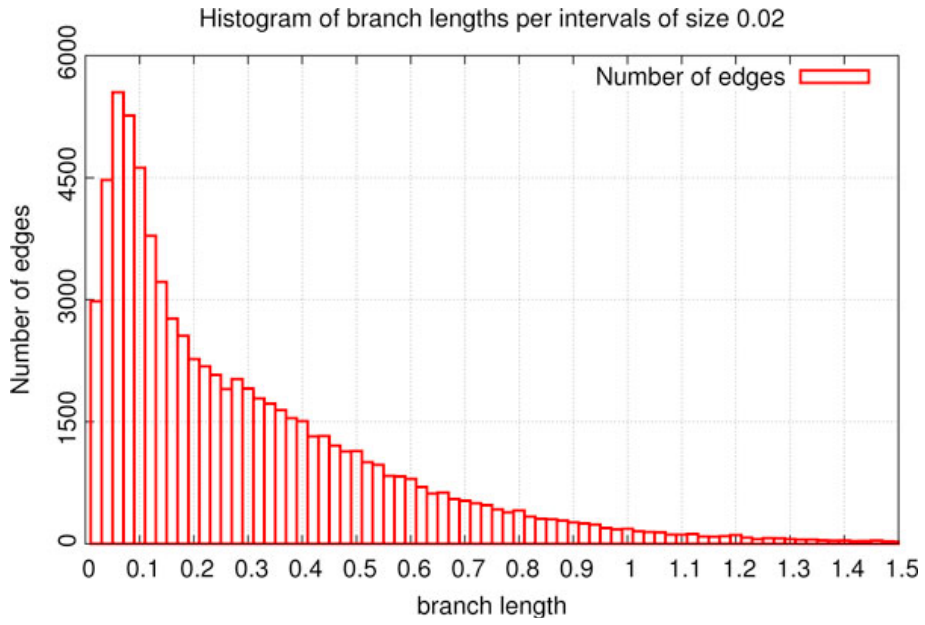


Figure 8 Branch lengths. Histogram of branch lengths.

error-corrected gene duplication problem. Our experimental results for a wide range of error-correction tests on yeasts phylogeny show that our error-corrected reconciliations result in improved predictions of invoked gene duplication and loss events that then allow to infer more accurate phylogenies.

The presented error correction is based on gene-species tree reconciliation using gene duplication and loss.

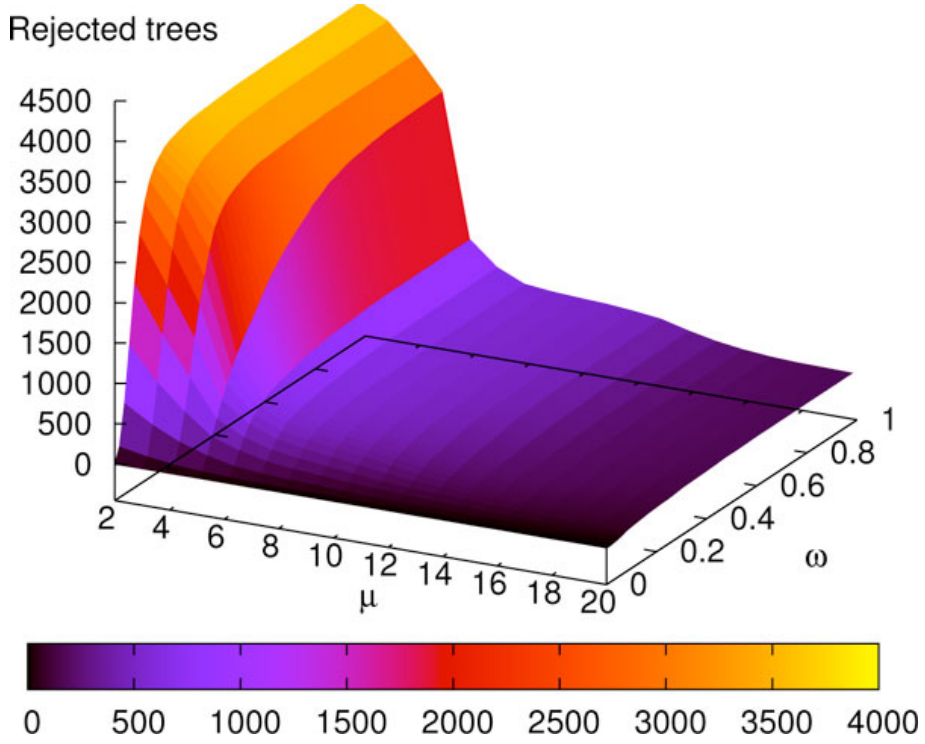


Figure 9 Rejected gene trees. The number of rejected trees as a function of μ and ω.

However, there are other major evolutionary mechanism that infer gene tree topologies that are inconsistent with the actual species tree topology, like horizontal gene transfer and deep coalescence. Gene tree reconciliation using these mechanisms is highly sensitive to topological error, similar to gene tree reconciliation under gene duplication and loss. Future work will focus on the development of algorithms that can also reconcile unrooted and erroneous gene trees using horizontal gene transfer and deep coalescence.

Acknowledgements
The reviewers have provided several valuable comments that have improved the presentation. This work was conducted in parts with support from the Gene Tree Reconciliation Working Group at NIMBioS through NSF award #EF-0832858. PG was partially supported by the grant of MNiSW (N N301 065236) and OE was 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]Institute of Informatics, University of Warsaw, Warsaw, 02-097, Poland. [2]Department of Computer Science, Iowa State University, Ames, 50011, USA.

Authors' contributions
PG and OE were responsible for algorithm design and writing the paper. PG implemented the programs, and performed the experimental evaluation and the analysis of the results. Both authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Published: 25 June 2012

References

1. Graur D, Li WH: *Fundamentals of Molecular Evolution.* 2 edition. Sinauer Associates; 2000 [http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0878932666].
2. Page RDM: **Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas.** *Systematic Biology* 1994, **43**:58-77.
3. Bonizzoni P, Della Vedova G, Dondi R: **Reconciling a gene tree to a species tree under the duplication cost model.** *Theoretical Computer Science* 2005, **347(1-2)**:36-53.
4. Eulenstein O, Mirkin B, Vingron M: **Duplication-Based Measures of Difference Between Gene and Species Trees.** *J Comput Biol* 1998, **5**:135-148.
5. 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(2)**:132-163.
6. Mirkin B, Muchnik IB, Smith TF: **A Biologically Consistent Model for Comparing Molecular Phylogenies.** *J Comput Biol* 1995, **2(4)**:493-507.
7. Sanderson M, McMahon M: **Inferring angiosperm phylogeny from EST data with widespread gene duplication.** *BMC Evolutionary Biology* 2007, **7(Suppl 1)**[http://dx.doi.org/10.1186/1471-2148-7-S1-S3].
8. Bansal MS, Eulenstein O: **The multiple gene duplication problem revisited.** *Bioinformatics* 2008, **24(13)**:i132-8.
9. Fellows MR, Hallett MT, Stege U: **On the Multiple Gene Duplication Problem.** In *ISAAC, Volume 1533 of LNCS* Chwa KY, Ibarra OH, Springer 1998, 347-356.
10. Guigó R, Muchnik IB, Smith TF: **Reconstruction of ancient molecular phylogeny.** *Molecular Phylogenetics and Evolution* 1996, **6(2)**:189-213.
11. Behzadi B, Vingron M: **Reconstructing Domain Compositions of Ancestral Multi-domain Proteins.** In *Comparative Genomics, Volume 4205 of LNCS.* Springer;Bourque G, El-Mabrouk N 2006:1-10.
12. Bansal MS, Burleigh GJ, Eulenstein O, Wehe A: **Heuristics for the Gene-Duplication Problem: A Θ(*n*) Speed-Up for the Local Search.** *RECOMB, Volume 4453 of LNCS* Springer; 2007, 238-252.
13. Ma B, Li M, Zhang L: **From Gene Trees to Species Trees.** *SIAM Journal on Computing* 2000, **30(3)**:729-752.
14. Page RDM: **GeneTree: comparing gene and species phylogenies using reconciled trees.** *Bioinformatics* 1998, **14(9)**:819-820.
15. Hahn MW: **Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution.** *Genome biology* 2007, **8(7)**:R141[http://dx.doi.org/10.1186/gb-2007-8-7-r141].
16. Chen K, Durand D, Farach-Colton M: **NOTUNG: a program for dating gene duplications and optimizing gene family trees.** *J Comput Biol* 2000, **7(3-4)**:429-447.
17. Durand D, Halldorsson BV, Vernot B: **A Hybrid Micro-Macroevolutionary Approach to Gene Tree Reconstruction.** *J Comput Biol* 2006, **13(2)**:320-335 [http://dx.doi.org/10.1089/cmb.2006.13.320].
18. Górecki P, Tiuryn J: **Inferring phylogeny from whole genomes.** *Bioinformatics* 2007, **23(2)**:e116-22.
19. Wehe A, Bansal MS, Burleigh GJ, Eulenstein O: **Dup-Tree: a program for large-scale phylogenetic analyses using gene tree parsimony.** *Bioinformatics* 2008, **24(13)**:1540-1541.
20. Eulenstein O, Huzurbazar S, Liberles D: **Reconciling phylogenetic trees.** *Evolution After Gene Duplication* Dittmar, Liberles, Wiley; 2010.
21. Bender MA, Farach-Colton M: In *The LCA Problem Revisited LATIN, Volume 1776 of LNCS.* Springer;Gonnet GH, Panario D, Viola A 2000:88-94.
22. Górecki P, Eulenstein O: **A Linear Time Algorithm for Error-Corrected Reconciliation of Unrooted Gene Trees.** In *Bioinformatics Research and Applications, Volume 6674 of Lecture Notes in Computer Science.* Springer Berlin/Heidelberg;Chen J, Wang J, Zelikovsky A 2011:148-159.
23. Górecki P, Tiuryn J: **URec: a system for unrooted reconciliation.** *Bioinformatics* 2007, **23(4)**:511-512.
24. Sherman DJ, Martin T, Nikolski M, Cayla C, Souciet JL, Durrens P: **Gènolevures: protein families and synteny among complete hemiascomycetous yeast proteomes and genomes.** *Nucleic Acids Research* 2009, **37(suppl 1)**:D550-D554[http://nar.oxfordjournals.org/content/37/suppl_1/D550.abstract].
25. Notredame C, Higgins DG, Jaap H: **T-coffee: a novel method for fast and accurate multiple sequence alignment.** *J Mol Biol* 2000, **302**:205-217 [http://dx.doi.org/10.1006/jmbi.2000.4042].

doi:10.1186/1471-2105-13-S10-S14
Cite this article as: Górecki and Eulenstein: Algorithms: simultaneous error-correction and rooting for gene tree reconciliation and the gene duplication problem. *BMC Bioinformatics* 2012 **13**(Suppl 10):S14.