Inferring phylogeny from whole genomes
Paweł Górecki* and Jerzy Tiuryn
Institute of Informatics, Warsaw University, Banacha 2, 02-678 Warsaw, Poland

ABSTRACT

Motivation: Inferring species phylogenies with a history of gene losses and duplications is a challenging and an important task in computational biology. This problem can be solved by duplication-loss models in which the primary step is to reconcile a rooted gene tree with a rooted species tree. Most modern methods of phylogenetic reconstruction (from sequences) produce unrooted gene trees. This limitation leads to the problem of transforming unrooted gene tree into a rooted tree, and then reconciling rooted trees. The main questions are ‘What about biological interpretation of choosing rooting?’, ‘Can we find efficiently the optimal rootings?’, ‘Is the optimal rooting unique?’.

Results: In this paper we present a model of reconciling unrooted gene tree with a rooted species tree, which is based on a concept of choosing rooting which has minimal reconciliation cost. Our analysis leads to the surprising property that all the minimal rootings have identical distributions of gene duplications and gene losses in the species tree. It implies, in our opinion, that the concept of an optimal rooting is very robust, and thus biologically meaningful. Also, it has nice computational properties. We present a linear time and space algorithm for computing optimal rooting(s). This algorithm was used in two different ways to reconstruct the optimal species phylogeny of five known yeast genomes from approximately 4700 gene trees. Moreover, we determined locations (history) of all gene duplications and gene losses in the final species tree. It is interesting to notice that the top five species trees are the same for both methods.

Availability: Software and documentation are freely available from http://bioputer.mimuw.edu.pl/~gorecki/urec

Contact: gorecki@mimuw.edu.pl

1 INTRODUCTION

The relationships between species cannot always be inferred from a single gene family. This important property is crucial in the species phylogeny reconstruction. In modern phylogenetics, gene trees are mostly computed by computer programs from gene sequences. Most of them produce unrooted gene trees (ML, MP, NJ, etc.) in which the common ancestor relation is undefined. Sometimes an ‘out-group’ species is added to ‘root’ the unrooted trees. In this paper, we use a different approach. The best ‘rooting’ is determined by ‘cost’ minimality criterion using the duplication-loss model (DL-model) and reconciled trees (Page and Charleston, 1997). The model explains the differences in terms of gene duplications and gene losses (Goodman et al., 1979; Page, 1994; Guigo et al., 1996). It is known that the ‘reconciled tree’ gives the minimal number of these events (Bonizzoni et al., 2003; Górecki and Tiuryn, 2005) required to reconcile a given gene tree with a given species tree.

The problem of rooting an unrooted gene tree, subject to minimization of a certain cost function, has been addressed in Chen et al. (2000). This is, to our knowledge, the only paper which deals with reconciliation of an unrooted gene tree with a (rooted) species tree. However, the authors of Chen et al. (2000) are not specific enough in explaining the algorithm which they use to solve this problem. Hence it is difficult to compare our approach with theirs. In particular they do not discuss the important problem of how the distribution of costs assigned to the nodes of a reconciled tree looks when we have two or more optimal rootings. As we will see, it follows from the theory presented in our paper that these distributions are in fact identical. Hence it is enough to find just one optimal rooting. Also a discussion of the role of weight constants in the cost function, which express relative importance of a duplication cost versus gene loss cost, is omitted in Chen et al. (2000). One of the main results of our paper states that these constants are not important (as long as they are positive), i.e. exactly the same edges are optimal for each choice of a cost function.

The problem of reconstructing the species phylogeny in DL-model is NP-complete (Ma et al., 2000). In our experiments, we enumerate all possible species trees. It should be noted that it is possible for a small number of species.

The paper is organized as follows. In Sections 2 and 3, we introduce the duplication-loss model and the general procedures of phylogeny reconstruction. Section 4 provides a detailed exposition of reconciling of unrooted gene tree and a rooted species tree. The main result of this paper is included in Theorem 7. Then, we present the algorithm of finding an optimal rooting. Finally, we present the results of experiments for five yeast genomes.

2 PHYLOGENY RECONSTRUCTION

2.1 Preliminaries

Let I be the set of species consisting of \( N > 0 \) elements. We consider a set \( P \) of protein sequences from species in \( I \). The first step is the classification of proteins into families. As a result a division \( C \) of \( P \) is obtained. Formally, \( C \) is a family of disjoint non-empty sets, called clusters, such that \( \cup C = P \). A cluster represents a set of homologous proteins.

In particular, the process consists of computing a similarity measure between sequences (e.g. by Smith–Waterman algorithm or BLAST) and then applying a clustering algorithm (e.g. MCL-method). In general, it can be more complex: in most cases an additional sophisticated analysis is employed to incorporate specific features of the clustered set (Dujon and Sherman, 2004). In our approach, we assume that the clustering is given.

2.2 Species and (un)rooted gene trees

Formally, 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 \( I \).
We now shortly introduce the concepts of the reconciled trees and the duplication-loss model. For a broader approach we refer reader to Goodman et al. (1979); Page (1994); and Guigo et al. (1996).

Let $S = \{V_S, E_S\}$ be a ‘species tree’. We view $S$ as an upper semilattice with + a binary least upper bound operation and $\mathcal{T}$ the top element (i.e. the root). In particular for $a, b \in V_S$, $a \leq b$ means that $a$ and $b$ are on the same path from the root, with $b$ being closer to the root than $a$ (unless $a \neq b$). We will use the ‘comparability predicate’ $D(a, b) = 1$, when $a \leq b$ or $b < a$ and $D(a, b) = 0$, when $a$ and $b$ are incomparable. The ‘distance function’ $\rho(a, b)$ denotes the number of edges on the unique path connecting $a$ and $b$.

Nodes $a, b \in V_S$ are called ‘siblings’ when $a + b$ is a parent of both $a$ and $b$. For $a, b \in V_S$, let $\mathbb{S}(a, b)$ denote the set of siblings defined as follows. $\mathbb{S}(a, b) := \{c\} \cup \mathbb{S}(a + c, b)$, where $c$ is a sibling of $a$ and either $a < b$ or $a + c < a + b$. In a similar way, we define $\mathbb{S}(a, b) := \{d\} \cup \mathbb{S}(a, d + b)$, where $d$ is a sibling of $b$ and either $b < a$ or $d + b < a + b$. The remaining cases are covered by the definition: $\mathbb{S}(a, b) = \emptyset$, when $a = b$, or $a$ and $b$ are siblings. It can be easily proved that the above definition is correct (i.e. it uniquely defines a set) and that $\mathbb{S}(a, b) = \mathbb{S}(b, a)$.

Let $L(a, b)$ denote the number of elements in $\mathbb{S}(a, b)$. The functions $L$ and $\rho$ are close to each other. When $a$ and $b$ are comparable, then $L(a, b) = \rho(a, b)$. While when $a$ and $b$ are incomparable, then we have $L(a, b) = \rho(a, b) - 2$. These observations are easy to prove by induction. The connection between $L$ and $\rho$ can be written concisely as follows:

$$L(a, b) = \rho(a, b) - 2 \cdot (1 - D(a, b)).$$ (1)

Let $M_G : V_G \rightarrow V_S$ be the ‘least common ancestor mapping’ from rooted $G$ into $S$ which preserves the labeling of the leaves. Clearly when $w_1, w_2 \in V_G$ are children of $v$, then we have

$$M_G(v) = M_G(w_1) + M_G(w_2).$$ (2)

In this general setting let us assume that we are given a ‘cost function’ $\xi : V_G \times V_S \rightarrow \mathbb{R}$ which for all nodes $v \in V_G$, $a \in V_S$ assigns a real $\xi(v, a)$ representing a contribution to node $a$ which comes from $v$ when reconciling $G$ with $S$. Having $\xi$ we can derive several values of interest. For $a \in V_S$, $\delta(a) = \sum_j \xi(v, a)$ is the total cost assigned to $a$. The function $\delta$ will be called a ‘cost distribution’ function. Dually, for $v \in V_G$ we can define $\kappa(v) = \sum_a \xi(v, a)$ which is a total contribution from $v$ due to reconciliation of $G$ with $S$. The function $\kappa$ will be called a ‘contribution’ function. Finally, $\sigma = \sum_v \kappa(v) = \sum_v \delta(a)$ is the total cost of reconciliation of $G$ with $S$. Let us give a few examples of a cost function which we will use later. Below we denote by $w_1$ and $w_2$ the children of a node $v \in V_G$ if $v \in int(G)$.

‘Duplication cost’ function is defined as follows:

$\xi^D(v, a) = \begin{cases} 1 & \text{if } M(v) = a, v \in int(G), \\ 0 & \text{otherwise} \end{cases}$

Loss cost function:

$\xi^L(v, a) = \begin{cases} 1 & \text{if } v \in int(G) \text{ and } a \in \mathbb{S}(M(w_1), M(w_2)) \\ 0 & \text{otherwise} \end{cases}$

We have the following result. Its proof follows immediately from our definitions.

**Lemma 1.** Let $v \in V_G$ and let $w_1, w_2$ be children of $v$. Then,

$$\kappa^D(v) = \begin{cases} D(M(w_1), M(w_2)) & \text{if } v \in int(G) \\ 0 & \text{if } v \text{ is a leaf} \end{cases}$$ (3)

$$\kappa^L(v) = \begin{cases} L(M(w_1), M(w_2)) & \text{if } v \in int(G) \\ 0 & \text{if } v \text{ is a leaf} \end{cases}$$ (4)

It can be proved that for a gene node $v$, $\kappa^D(v) = l(v)$, where $l(v)$ is the number of gene losses associated to $v$ (Ma et al., 2000; Eulenstein et al., 1998). Also, $\kappa^D(v) = 1$ if $v$ is a duplication node (Ma et al., 2000; Eulenstein et al., 1998). Distributions $\delta^D$ and $\delta^L$ show the location of gene duplications and gene losses, respectively, in the species tree. The example is shown in Figure 1.

The reconciliation model is known to be biologically meaningful (Mirkin et al., 1995). The crucial notion is the ‘reconciled tree’ which can be visualized as an embedding of the gene tree into the species tree. It follows easily that $\delta^D$ and $\delta^L$ are the minimal number
of gene duplications and gene losses (respectively) required to reconcile $G$ with $S$. Here we omit the details. Refer Figure 1 for example of embedding.

4 RECONCILING UNROOTED GENE TREE WITH A SPECIES TREE

4.1 Unrooted gene trees

From now on, let $G = (V_G, E_G)$ be unrooted gene tree. Choosing a position of the root in $G$ amounts to selecting an edge $e \in E_G$ on which the root is to be placed. Such a rooted tree we denote by $G_e$, with $v_e$ a new node denoting the root. To distinguish between rootings of $G$ all symbols defined in previous section for a rooted gene tree will be extended by inserting index $e$, for instance $M_e$ is the mapping from $G_e$ to $S$, etc.

Without loss of generality, we assume

(A1) $S$ and $G$ have at least one internal node,

(A2) $M_e(v_v) = \top$, i.e. the root of every rooting is mapped into the root of $S$ (it is independent of $e$).

Note that from these conditions $|L(G)| > 1$.

The proof of the next lemma follows immediately from the definition of duplication and gene loss costs.

**Lemma 2.** Let $e$ and $e'$ be any two edges in $G$. If for any pair of nodes $a, b \in V_S$ the number of siblings $w_1, w_2$ in $G_e$ which are mapped to $a$ and $b$, respectively, is the same as the number of siblings $u_1, u_2$ in $G_{e'}$ which are mapped to $a$ and $b$, respectively, then $\delta^D_e = \delta^D_{e'}$ and $\delta^L_e = \delta^L_{e'}$.

4.2 Orientation and labeling of $G$

We transform $G$ into a directed graph $\hat{G} = (V_G, E_G)$, where $E_G = \{(v, w) \mid \{v, w\} \in E_G\}$. In other words each undirected edge $\{v, w\}$ in $G$ is replaced in $\hat{G}$ by a pair of directed edges $(v, w)$ and $(w, v)$.

Edges in $\hat{G}$ are labeled by nodes of $S$ as follows. If $v \in V_G$ is a leaf labeled by $a$, then the edge $(v, w) \in E_G$ is labeled by $a$. For the induction step let us assume that $(v, w)$ and $(w, v)$ are labeled by $b_1$ and $b_2$, respectively, then the edge $(v, w) \in E_G$, such that $w_1 \neq w_2$ is labeled by $b_1 + b_2$. The intuition behind the label $a$ of an edge $(v, w)$ of $\hat{G}$ is that in the rooted tree $G_e$, which is isomorphic to $\hat{G}$, $a$ is either $(v, w)$, or it is located in the part of $G$ which is past $w$, the node $v$ is mapped to $a$. Equation (2) justifies this intuition.

4.3 Classification of stars

Every internal node $v \in V_G$ defines a ‘star’ with center $v$ as indicated in Figure 2. The edges $(v, w)$ will be called ‘outgoing’, while the edges $(w, v)$ will be called ‘incoming’. We will refer to the undirected edge $(v, w)$ as $e_i$ for $i = 1, 2, 3$.

**Lemma 3.** Let $b_i = \top$ for all $i = 1, 2, 3$. Then for exactly one $a_i$ say for $a_1$, we have $a_1 \neq \top$ (i.e. $a_2 \neq a_3 = \top$).

**Proof.** It follows from (A2) that for every edge $(v, w)$ of $G$, if the label of $(v, w)$ is $a$ and the label of $(w, v)$ is $b$, then $a + b = \top$. Thus, we have $b_1 + b_2 + b_3 = a_1 + b_3 = \top$. Since none of $b_i$'s is $\top$, it follows that if $x, y \in V_S$ are the children of $\top$, then two elements (say $b_1$ and $b_2$) are below $x$, and the third (i.e. $b_3$) is below $y$. Hence $a_1 = b_2 + b_3 \leq x$ and therefore $a_1 \neq \top$. On the other hand we have $a_2 = b_1 + b_2 = \top$ and $a_3 = b_1 + b_3 = \top$. This completes the proof.

Observe that placing the root on the edge $e_i$ results in a rooted tree as presented in Figure 2. In square bracket next to a node $u$ we indicate the node of $S$ to which $u$ is mapped from $G_e$. The trees below each node $w_i$ are the same in all trees of Figure 2, for each $i = 1, 2, 3$, and therefore we omit them.

We will use the following classification of stars (Fig. 3). A star is said to be of type S1, if it has exactly one incoming edge labeled $\top$ and exactly two outgoing edges labeled $\top$ and these edges are connected to the three siblings of the center. A star is said to be of type S2, if it has no incoming edges labeled $\top$ (it follows from Lemma 2 that it must have exactly two outgoing edges labeled $\top$). A star is said to be of type S3, if it has exactly one incoming edge labeled $\top$ and all outgoing edges labeled $\top$. Finally, a star is said to be of type S4, if it has no outgoing edges labeled $\top$ and at least two incoming edges labeled $\top$. It is easy to show that there are no other types of stars.

**Lemma 4.** For every gene tree [satisfying (A1) and (A2)] $G$ we have the following mutually exclusive situations.

(i) $\hat{G}$ has exactly one star of type S2 and all other stars are of type S1.

(ii) $\hat{G}$ has exactly two stars of type S2. They share a common edge. All other stars are of type S1.

(iii) $\hat{G}$ has exactly two stars of type S3. They share a common edge. All other stars are of type S1.

(iv) $\hat{G}$ has an occurrence of a star of type S4. All other stars are of type S1 or S3.

**Proof.** Suppose $G$ does not contain stars of type S3 or S4. We will show that it has to contain a star of type S2. Suppose on the contrary
that $G$ has only stars of type $S1$. Take any such a star and follow a path which starts at its center and goes against the direction of the edges labeled $\mathcal{T}$, i.e., move from node $v$ to $v'$ whenever $(v', v) \in E$ is labeled $\mathcal{T}$. Clearly this path is unique and does not contain a cycle. So we have to arrive at a node, say $u$, from which we cannot continue. This must be a leaf. This is a contradiction since then $G$ contains an edge starting from a leaf and labeled $\mathcal{T}$. This proves that $G$ must contain a star of type $S2$, and in fact the node $u$ where we got stuck is the center of a star of type $S2$. Let $u'$ be a node of this star such that neither $(u, u')$ nor $(u', u)$ is labeled $\mathcal{T}$. Observe that every node $v \in V_G$ is reachable either from $u$ or from $u'$ by a directed path along edges labeled $\mathcal{T}$. Hence all internal nodes different from $u$ and $u'$ are centers of stars of type $S1$. Clearly if $u$ or $u'$ is leaf, then $G$ has only one occurrence of a star of type $S2$. Otherwise $G$ has exactly two occurrences of stars of type $S2$ and these stars share the same edge $(u, u')$.

Observe that if $G$ contains a star of type $S3$ or $S4$, then it cannot contain a star of type $S2$. Suppose now that $G$ contains a star of type $S3$ with center $u$, and no star of type $S4$. Let $u'$ be such that both $(u, u')$ and $(u', u)$ are labeled $\mathcal{T}$. We will show that $G$ has exactly two occurrences of a star of type $S3$ and these stars share the edge $(u, u')$. Clearly neither $u$ nor $u'$ is a leaf and the outgoing edges leaving $u$ or $u'$ must be labeled $\mathcal{T}$. This implies that we have two occurrences of a star of type $S3$ and they share the same edge $(u, u')$.

If $G$ contains another star of type $S3$ and $v$ is its center, then all edges on the path connecting $u$ and $v$ have labels in both directions equal $\mathcal{T}$. It follows that on this path there must be at least one occurrence of a center of a star of type $S4$. This yields contradiction and completes the proof of the Lemma.

4.4 Duplication and loss cost analysis

In this section we will analyze how the total cost changes when we move a position of the root in $G$. The analysis will be done for loss cost, as well as for the gene duplication cost. We will use throughout this section the notation of a star introduced in Figure 2.

**Proposition 5.** Consider the following two cases.

(I) $b_1 = \mathcal{T}$.

(II) $b_i \neq \mathcal{T}$, for all $i = 1, 2, 3$ and let $a_i \neq \mathcal{T}$ (see Lemma 2).

Then in each case we have

$$\sigma_{e_1}^D \leq \sigma_{e_1}^L = \sigma_{e_1}^P.$$ 

And

$$\Delta_L = \sigma_{e_1}^L - \sigma_{e_1}^P = \rho(a_1, \mathcal{T}) + 2 \cdot (1 - D(b_2, b_3)).$$

Moreover, when $b_1 = a_1 = \mathcal{T}$ and either $b_2 = \mathcal{T}$, or $b_3 = \mathcal{T}$, then

$$\Delta_L = 0 \quad \text{and} \quad \delta_{e_1}^L = \delta_{e_1}^P = \delta_{e_1}^P.$$ 

**Proof.** We start with Case (I) and use notation of Figure 2. We have $b_1 = a_2 = a_3 = \mathcal{T}$ and by Lemma 1 we obtain $\delta_{e_1}^L = \delta_{e_1}^P$ and thus $\sigma_{e_1}^L = \sigma_{e_1}^P$. Moreover, when $a_1 = \mathcal{T}$ and $b_2 = b_3 = \mathcal{T}$, then again by Lemma 2 we obtain $\delta_{e_1}^L = \delta_{e_1}^P$, which implies $\Delta_L = 0$.

It remains to compute $\Delta_L = \sigma_{e_1}^L - \sigma_{e_1}^P$, when $b_1 = a_2 = a_3 = \mathcal{T}$. We have $\kappa_{e_1}^L(v) = L(a_1, \mathcal{T}) - \rho(a_1, \mathcal{T})$, $\kappa_{e_1}^P(v) = L(b_2, b_3) = \rho(b_2, b_3) - 2 \cdot (1 - D(b_2, b_3))$, and $\kappa_{e_1}^L(v) = L(b_2, b_3) = \rho(b_2, b_3)$. Then

$$\Delta_L = \rho(b_2, b_3) + \rho(b_2, \mathcal{T}) - 2 \cdot (1 - D(b_2, b_3)) \quad \text{and} \quad \delta_{e_1}^L = \delta_{e_1}^P.$$ 

The second equality above follows from $a_1 = b_2 + b_3$. Thus $\Delta_L \geq 0$ and therefore $\sigma_{e_1}^L \leq \sigma_{e_1}^P$. This completes the proof of Case (I). For the proof of Case (II) we have $a_i \neq \mathcal{T}$ and $a_1 = a_3$. Then $\kappa_{e_1}^L(v) = L(b_2, b_3) = \rho(b_2, b_3)$, $\kappa_{e_1}^P(v) = L(b_1, b_3) = \rho(b_1, b_3) - 2 \cdot (1 - D(b_2, b_3))$ and similarly $\kappa_{e_1}^P(v) = L(b_2, b_3) = \rho(b_2, b_3)$.

The reader may wish to consult the above picture from which the equality follows immediately. Since $\kappa_{e_1}^L$ and $\kappa_{e_1}^P$ coincide on all other nodes, we conclude that $\sigma_{e_1}^L = \sigma_{e_1}^P$.

On the other hand we have $\kappa_{e_1}^L(v) = L(a_1, b_1) - \rho(a_1, b_1) = \rho(b_2, b_3) - 2 \cdot (1 - D(b_2, b_3))$.

Therefore

$$\Delta_L = \rho(b_2, \mathcal{T}) + \rho(b_2, b_3) - 2 \cdot (1 - D(b_2, b_3)) \quad \text{and} \quad \delta_{e_1}^L = \delta_{e_1}^P.$$ 

Again it may be helpful to consult the picture above for justification of the second equation. Since $\Delta_L > 0$ we have $\sigma_{e_1}^L \leq \sigma_{e_1}^P$. This completes the proof of Case (II).

It follows from Proposition 5 that for stars of type $S1$, $S2$ or $S3$ we have $\Delta_L > 0$, and for stars of type $S4$ we have $\Delta_L = 0$.

**Proposition 6.** Consider the following two cases.

(I) $b_1 = \mathcal{T}$. 

(II) $b_i \neq \mathcal{T}$, for all $i = 1, 2, 3$ and let $a_i \neq \mathcal{T}$ (see Lemma 3).

Then in each case we have

$$\sigma_{e_1}^D \leq \sigma_{e_1}^P = \sigma_{e_1}^D.$$ 

And

$$\Delta_D = \sigma_{e_1}^L - \sigma_{e_1}^P = 1 - D(b_2, b_3).$$

Moreover, when $b_1 = a_1 = \mathcal{T}$ and either $b_2 = \mathcal{T}$, or $b_3 = \mathcal{T}$, then

$$\Delta_D = 0 \quad \text{and} \quad \delta_{e_1}^L = \delta_{e_1}^D = \delta_{e_1}^P.$$ 

**Proof.** The proof is similar to the previous one.

It follows from Proposition 6 that for stars of type $S4$ we have $\Delta_D = 0$ and the cost distribution function is the same for any of the three undirected edges. It also follows that for a star of type $S3$ we have $\Delta_D = 1$.

4.5 Finding minimal edges

Here we present the main theoretical results of our paper. First we introduce some definitions and notation. A subset $X$ of edges of $E_G$ is called a full subtree of $G$ if any two nodes incident to edges of $X$ (these nodes do not have to be incident to the same edge) are connected by unique path consisting entirely of edges from $X$, and every internal node of $X$ (i.e. incident to at least two edges of $X$) has degree 3 (i.e. it is incident to three edges of $X$).

Let $a, \ b \in R$ be two positive reals. In this section we will consider a weighted combination of duplication and loss costs of
reconciliation. This is represented by the following cost function called a ‘weighted mutation’ cost.

$$\xi_{v}^{M,\beta}(v, a) = \alpha \cdot \xi_{v}^{D} + \beta \cdot \xi_{v}^{L}.$$ 

The usual mutation cost is obtained for $$\alpha = \beta = 1.$$

**Theorem 7.** Let $$S$$ and $$G$$ be species and gene trees ($$S$$ rooted and $$G$$ unrooted) and let $$\alpha, \beta \in \mathbb{R}$$ be any positive reals. Let

$$\text{Min}_{G} = \{ e \in E_{G} | \sigma^{M,\beta}_{e} \text{ is minimal} \}.$$ 

Then

(i) $$| \text{Min}_{G} | > 1$$ iff $$\tilde{G}$$ contains a star of type S4. Moreover, if $$| \text{Min}_{G} | > 1,$$ then $$\text{Min}_{G}$$ consists of all and only edges which occur in stars of type S4. All others stars in this case are of type S1 or S3.

(ii) $$| \text{Min}_{G} | > 1$$ iff $$\tilde{G}$$ has exactly one pair of edges $$\{v, w\}$$ and $$\{w, v\}$$ whose labels are either both equal to $$\top$$ or both are not equal to $$\top.$$ For such edges we have that the unordered pair $$\{v, w\}$$ is the only element of $$\text{Min}_{G}.$$ All stars of $$\tilde{G}$$ which do not contain the edge $$\{v, w\}$$ are of type S1.

(iii) For all edges $$e, e' \in \text{Min}_{G}$$ we have $$\delta^{M,\beta}_{e} = \delta^{M,\beta}_{e'}$$

(iv) The set $$\text{Min}_{G}$$ is independent of the choice of $$\alpha$$ and $$\beta,$$ as long as they are positive.

(v) $$\text{Min}_{G}$$ is a full subtree of $$G.$$

(vi) The edges of $$\text{Min}_{G}$$ can be found by a greedy method of gradient descent: leave a star with $$\Delta_{\text{Min}_{G}} = 0$$ through an edge $$\{e\}$$ with a smaller $$\sigma^{M,\beta}_{e}.$$ Arriving at a star with $$\Delta_{\text{Min}_{G}} = 0$$ means that we have arrived at $$\text{Min}_{G}.$$

**Proof.** It follows from Propositions 5 and 6 that the only stars in $$\tilde{G}$$ which have $$\Delta_{\text{Min}_{G}}$$ are of type S4, whenever $$\alpha, \beta > 0.$$ Hence $$| \text{Min}_{G} | = 1$$ iff $$\tilde{G}$$ has no stars of type S4. The second part of (i) follows from Lemma 4. The ‘only if’ part of (ii) is equivalent, by Lemma 4, to the statement that $$\tilde{G}$$ has no stars of type S4. This proves (ii). (iii) follows from (i), Propositions 5 and 6. Condition (iv) follows from (i), since the types of stars of $$\tilde{G}$$ are independent of $$\alpha$$ and $$\beta.$$ Condition (v) follows from (i) and Lemma 4. Finally (vi) follows immediately from (i) and (ii). This completes the proof.

It follows from (ii) in the above theorem that if $$\text{Min}_{G}$$ has only one element, then the star which contains the edge $$\{v, w\}$$ is either of type S2, or of type S3.

The above result can be nicely explained in terms of a landscape of a cost function. It follows that in case of more than one optimal edge, all such edges are clumped together and form a flat ‘valley’ which consists of all and only stars of type S4. The boundary of this valley is built of stars of type S3 and the ‘slopes’ of the surrounding ‘hills’ are stars of type S1. There are no local minima in this landscape, except the global minimum located in the valley (Fig. 4). A similar picture can be drawn for the case of a unique optimal edge.

In Figure 4 we present an illustration of unrooted gene tree with an example of embedding one of the optimal edges. It follows from our results that there are five different optimal rootings. Moreover, all of them have the same distributions $$\delta^{L}$$ and $$\delta^{D}. \text{Also, all differences between optimal embeddings are located in the hat (Fig. 4).}$$

This fact can be formally proved for the reconciled trees. We omit the details which follow quite easily from the model. This property is important from biological point of view. It ensures that the embeddings of optimal rootings are almost identical, and thus the concept of searching optimal rooting is biologically correct also when $$| \text{Min}_{G} | > 1.$$

**4.6 Algorithm**

As proved in Theorem 7 it is enough to use a greedy method of gradient descent to find an optimal edge. Thus, it is sufficient to know only the ‘starred’ topology of unrooted gene tree, i.e. the labels of edges in $$G.$$ First, we need an efficient algorithm for computing least common ancestors in $$S.$$ There is an algorithm which requires $$O(N)$$ time for preprocessing and then $$O(1)$$ time for LCA-query (Bender and Farach-Colton, 2000). Note that there is no cycle between computing the labels. When a value is evaluated then it is stored in the attribute. Thus, when we refer to already computed value, we do not have to recompute it. Thus, all labels in $$G$$ can be computed in $$O(n)$$ steps, where $$n$$ is the size of $$G$$ if LCA-preprocessing is given.

Computing cost of a rooted reconciliation is linear (Ma et al., 2000). For computing number of losses, we use depth of nodes in $$S,$$ which can be computed in $$O(1)$$ after $$O(N)$$ preprocessing, where $$N$$ is the size of $$S.$$ For duplications use the labels of edges in $$G.$$ Finally, computing costs of all rootings has time complexity $$O(max(n, N)).$$ The same time complexity is for computing an optimal cost.

Note also that starting from an optimal edge the set of all optimal edges (and also optimal gene trees) can be easily enumerated.

**5 EXPERIMENTS**

**5.1 Optimal species tree by total cost**

The algorithms described in previous sections were used to compute optimal species tree for the set of gene families of five yeasts from Dujon and Sherman (2004). In our example $$I = \{c, k, d, y, z\}$$ where
Candida glabrata; K. lactis; Debaryomyces hansenii; Y. lipolytica and Saccharomyces cerevisiae. We started from the clusterings taken from Dujon and Sherman (2004). We aligned them by ClustalW. The ClustalW trees were used as unrooted gene trees. The total number of gene trees equals 4708. For our small set of species we computed all possible species trees (there are 105 rooted trees). Then, we calculate costs of reconciling 4708 unrooted gene trees with every species tree (we started our algorithm \(4708 \times 105 = 494340\) times which consumed \(\approx 9\) s on Athlon 1800+; program was written in C++. The summary of computations is presented in Figure 5. Here, we show only the best nine species trees. The total weighted mutation cost is shown (with weights 1).

The optimal species tree is \((y, (d, (k, (s, c))))\). It was reconciled with 4708 unrooted gene trees with the mutation cost 14 266. This breaks into 6214 gene duplications and 8052 gene losses. The optimal tree agrees with the tree computed for 25S rDNA sequences in Dujon and Sherman (2005).

Now, we can shortly compare the major evolutionary events in the hemiascomycetes shown in Dujon and Sherman (2004) with the results obtained in the present experiment.

- between \(s\) \(S\).cerevisiae and \(sc\) (i.e. between the least common ancestor of \(S\).cerevisiae and \(C\).glabrata) there is 1110 genes loss; this agrees with the duplicated gene loss as indicated in Dujon and Sherman (2004)
- 282 (4.5%) duplications between \(sc\) and \(sk\); the authors of Dujon and Sherman (2004) detected massive duplications and suggest whole-genome duplication; our analysis does not support this hypothesis;
- 983 (15.8%) duplications between \(sk\) and \(skd\); this is more probable whole-duplication location (this hypothesis requires an additional analysis);
- only 95 (1.5%) gene duplications between \(k\) and \(ks\); in Dujon and Sherman (2004) authors detected few duplicated blocks;
- 531 (10.7%) gene losses between \(k\) and \(ks\), and 1429 (10.0%) gene losses between \(c\) and \(sc\) may suggest low level of genome redundancy in \(C\).glabrata and \(K\).lactis; this agrees with Dujon and Sherman (2004).

<table>
<thead>
<tr>
<th>Species tree 1–6 Points</th>
<th>Species tree 7–12 Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>((y, (d, (k, (s, c))))) 0.1159</td>
<td>((s, (c, (k, (d, y))))) 0.0442</td>
</tr>
<tr>
<td>((d, (y, (k, (s, c))))) 0.0772</td>
<td>((y, (d, (c, (s, k))))) 0.0311</td>
</tr>
<tr>
<td>((d, (y, (k, (s, c))))) 0.0630</td>
<td>((d, (y, (c, (s, k))))) 0.0307</td>
</tr>
<tr>
<td>((s, (c, (k, (d, y))))) 0.0478</td>
<td>((d, (y, (c, (s, k))))) 0.0264</td>
</tr>
<tr>
<td>((c, (s, (k, (d, y))))) 0.0455</td>
<td>((y, (d, (c, (s, k))))) 0.0216</td>
</tr>
</tbody>
</table>

The species in bold are present in Figure 5.

5.2 Optimal species tree by ‘voting’
We applied ‘voting’ algorithm (see Section 2.3) to trees from previous section. The result is summarized in Table 1. Again, the optimal species tree is the 25S rDNA tree. Moreover, the first five species trees are the same for both algorithms.

6 DISCUSSION
This paper contains several theoretical and practical results on the problem of phylogeny reconstruction. First, we show that the model reconciled trees can be applied to unrooted gene trees. In particular, we formally prove that the concept of choosing an optimal rooting leads to biologically correct scenarios even if more than one optimal solution exists. This property, which is a contribution of the present paper, together with the efficient algorithm of computing an optimal cost or reconciliations allow us to perform fast experiments on large datasets available for yeast genomes. Our experimental results confirm (with some exceptions as shown in the previous section) the known facts from literature. We propose, therefore, in the present paper that the model of ‘unrooted reconciliation’ can be used to infer correctly phylogeny of species.

In the basic approach, unrooted gene family trees generated from publicly available phylogenetic software can be treated as input to the ‘unrooted’ reconciliation in order to obtain a set of optimal rootings. This can be viewed as an alternative for ‘species outgroup’ methodology used commonly when the rooting is unknown.
ACKNOWLEDGEMENTS

The financial support for the study is provided by KBN Grant 3 T11F 021 28.

Conflict of Interest: none declared.

REFERENCES


