

# A Novel Algorithm for Duplication-Loss Problem Based on NNI Local Search

Hui Wang<sup>a</sup>, Daming Zhu

School of Computer Science and Technology, Shandong University, Jinan 250101, China

<sup>a</sup>bzwdwh@163.com

**Abstract**—The gene-duplication problem is to infer a species tree from a collection of gene trees. The decision variant of this problem is NP-complete. The utility of this NP-hard problem for large-scale phylogenetic analyses has been largely limited by the high time complexity of existing heuristics. These methods search the whole possible subtrees. They solve a local search problem instance in each step. The NNI(nearest neighbor interchange) local search problem is a classical local search problem. In this paper, we study the duplication-loss problem and provide a novel algorithm for the k-NNI local search problem. Our improvement not only significantly enlarges the search space but also makes the time complexity better. Hence, the duplication-loss problem will be much more manageable for large scale phylogenetic analyses.

**Keywords**- Duplication-Loss; NNI; Local search; Computational phylogenetics; Computational complexity; Algorithm

## I. INTRODUCTION

Recently, the available genomic sequence data increase more and more quickly. They provide phylogenetic analyses with potential information. However, the large amounts of information that include many complex evolutionary processes have to be neglected by these analyses. One way to utilize these information is to reconcile the gene trees with species trees under the mutation cost criterion(duplication-loss cost)[1]. The duplication-loss problem that is the corresponding optimization problem is to find a minimum mutation cost tree for a set of given trees[2]. The decision version of this problem is NP-complete[3].

The duplication-loss problem is based on the Gene-Duplication model from Goodman et al.[1]. That model [8-13] explains the incompatible property between gene trees and species trees through the number of gene duplications. A gene tree is comparable with a species tree if there exists a leaf-mapping that maps every leaf-gene to the leaf-species. An extension of the leaf-mapping denoted by  $M$  describes the minimum number of gene duplications.  $M$  maps each gene to the least common ancestor of the species from which the leaves of the subtree were sampled. A gene in the gene tree is a duplication if a child of it has the same mapping under  $M$ . Similarly, the number of losses associated with a gene is roughly equal to the number of ancient species between the mapping of the gene and the mappings of its children. The duplication-loss cost for a gene tree and a species tree is the sum of duplications number and losses number. The duplication-loss cost for a set of given gene trees and a comparable species tree is the

sum of the duplication-loss costs for each gene tree and the species tree. The duplication-loss cost can be computed in linear time[14].

There exist some heuristics to answer the gene-duplication problem. These methods search the whole possible subtrees. They solve a local search problem instance in each step[4]. That is to find a species tree with the minimum duplication cost in the whole neighborhood of a species tree. The neighborhoods are the total trees that the given tree can be transformed into them by utilizing the tree edit operation. The nearest neighbor interchange(NNI) operation[5] has been proved to be effective for phylogenetic studies[6].

Bansal et al have already given effective solutions for the gene-duplication problem based on the NNI local search[7]. In this work we provide fast algorithm for solving the duplication-loss problem based on the k-NNI local search. The set of the k-NNI neighborhood can be obtained by executing at most  $k$  successive NNI operations on a given tree. We assume that the number of the given gene trees is  $r$  and the size of the resulting species tree is  $n$ . The difference between the size of the gene tree and the size of the species tree is a constant. For  $k \in \{1,2,3\}$ , the size of the neighborhood induced by the  $k$ -NNI local searches is  $\theta(n^k)$ . Hence, it requires  $O(rn^{k+1})$  time for solving the corresponding local search problems. In our paper, in order to get the costs of the trees that are produced by the second NNI operations, we reuse the results reduced in the  $\theta(n)$  possible 1-NNI operations on a tree. Our algorithm can solve the duplication-loss problem based on 2-NNI-neighborhoods in  $O(rn^2)$  time. It is not only enlarge the search neighborhoods but also improve on the time complexity. Thus, our improvement makes the duplication-loss problem much more manageable for large-scale phylogenetic analyses.

## II. BASIC DEFINITIONS AND PRELIMINARIES

### A. Basic Definitions and Notation

A binary tree  $T$  consists of a node set  $V(T)$  and an edge set  $E(T)$ .  $T$  is rooted if it has a so-called root node denoted by  $Ro(T)$ . We define  $\leq T$  to be the partial order on  $V(T)$ . We write  $x \leq_T y$  if  $y$  is a node on the path between  $Ro(T)$  and  $x$ .  $Le(T)$  whose elements are called leaves denotes the set of minima under  $\leq T$ . Internal nodes  $I(T)$  is defined to be  $V(T) \setminus Le(T)$ . If  $\{x, y\} \in E(T)$ , and  $x \leq_T y$  then we call  $x$  a child of  $y$  and  $y$  the parent of  $x$  denoted by  $Pa(x)$ .  $Ch(y)$  denotes all of the children of  $y$ . We call two nodes that have

the same parent siblings. Every internal node has exactly two children in a (fully) binary tree.  $T_x$  induced by  $x$  and all its descendants is the subtree of  $T$  rooted at node  $x$ . The least common ancestor of a non-empty subset  $L \subseteq V(T)$ , denoted as  $\text{lca}(L)$ , is the unique smallest upper bound of  $L$  under  $\leq T$ . Given  $x, y \in V(T)$ ,  $x \rightarrow_T y$  denotes the path from  $x$  to  $y$  in  $T$ .  $d_T(x, y)$  denotes the number of edges on the path  $x \rightarrow_T y$ . Given  $T$  and  $L \subseteq \text{Le}(T)$ ,  $T[L]$  denotes the minimal subtree of  $T$  with  $L$ .  $T[L]$  is defined to be the tree obtained from  $T$  by successively removing every degree two node (not root) and connecting its two ends.

### B. The Duplication-Loss Problem

A species tree is a rooted full binary tree that depicts the ancestor-descendant relationships between a set of species. A gene tree is a tree that depicts the evolutionary relationships among the same gene family. A mapping that maps each gene  $g$  in  $G$  to the least common ancestor in  $S$  is required for comparing a gene tree  $G$  with a species tree  $S$ .

Definition 2.1 (Mapping). The leaf-mapping  $L_{G,S}: \text{Le}(G) \rightarrow \text{Le}(S)$  specifies the species  $M_{G,S}(g)$  from which gene  $g$  was sampled. The extension of  $L_{G,S}$  to  $M_{G,S}: V(G) \rightarrow V(S)$  is the mapping defined by  $M_{G,S}(g) = \text{lca}(L_{G,S}(\text{Le}(G_g)))$ .

We define  $M_{G,S}^{-1}(s)$  to denote the set of nodes in  $G$  that map to node  $s \in V(S)$  under the mapping  $M_{G,S}$ .

Definition 2.2 (Comparability). Given gene tree  $G$  and species tree  $S$ , if there exists a leaf-mapping  $L_{G,S}$  we say that  $G$  is comparable to  $S$ .

Definition 2.3 (Duplication). A node  $v \in V(G)$  is a duplication if  $M_{G,S}(v) = M_{G,S}(u)$  for some  $u \in \text{Ch}(v)$ , and we define  $\text{Dup}(G, S) = \{g \in V(G): g \text{ is a duplication}\}$ .

Definition 2.4 (Losses).

$$\text{Loss}(G, S, g) = \begin{cases} 0, & \text{if } M_{G,S'}(g) = M_{G,S'}(g') \forall g' \in \text{Ch}_G(g) \\ \left| \sum_{g' \in \text{Ch}_G(g)} \left| d_{S'}(M_{G,S'}(g), M_{G,S'}(g')) - 1 \right| \right|, & \text{otherwise} \end{cases}$$

Where  $S' = S[\text{Le}(G)]$ .

$$\text{Loss}(G, S) = \sum_{g \in \text{Le}(G)} \text{Loss}(G, S, g).$$

Definition 2.5 (Reconciliation cost).

$$\Delta(G, S) = |\text{Dup}(G, S)| + \text{Loss}(G, S).$$

$$\Delta(G', S) = \sum_{G \in G'} \Delta(G, S).$$

Let  $T = \{S: G' \text{ is comparable with } S\}$ . We define  $\Delta(G')$  to be the reconciliation cost of  $G'$ .

Problem 1 (Duplication-Loss).

Instance: A set of gene trees  $G'$ .

Find: A species tree  $S^*$  comparable with  $G'$ , such that  $\Delta(G', S^*) = \Delta(G')$ .

### C. Local Search Problems

Definition 2.6 (NNI operation). We define the valid nodes set in  $T$   $\text{valid}(T) = V(T) \setminus \{\text{Ro}(T)\} \cup \text{Ch}(\text{Ro}(T))$ . For  $y \in \text{valid}(T)$ ,  $\text{NNI}_T(y)$  denotes the tree obtained by swapping  $T_x$  and  $T_y$ , where  $x$  is the sibling of  $\text{Pa}(y)$ .  $\text{NNI}_T(y)$  is the tree that is obtained by one NNI operation on  $y$ .

Definition 2.7 (k-NNI neighborhood). The set of all the k-NNI neighborhood trees denoted by  $k\text{-NNI}_T$  can be obtained by performing at most  $k$  successive NNI operations on  $T$ . Then,  $1\text{-NNI}_T$  (or  $\text{NNI}_T$ ) is the set  $\{\text{NNI}_T(y): y \in \text{valid}(T)\}$ .

Problem 2 (k-NNI local search problem).

Instance: A set of gene trees  $G'$ , and a species tree  $S$  such that  $\bigcup_{G \in G'} \bigcup_{g \in \text{Le}(G)} L_{G,S}(g) = \text{Le}(S)$ .

Find: A species tree  $S^* \in k\text{-NNI}_S$ , such that  $\Delta(G', S^*) = \min_{T \in k\text{-NNI}_S} \Delta(G', T)$ .

## III. STRUCTURAL PROPERTIES

Let  $S' = \text{NNI}_S(y)$ ,  $v = \text{Pa}_S(y)$ ,  $u = \text{Pa}_S(v)$ ,  $z$  and  $x$  separately denote the siblings of  $y$  and  $v$  in  $S$ . Next, we will study the effects of NNI operation on the status of nodes of duplication and loss and on the mapping  $M_{G,S}$ . When we perform an NNI operation on a tree, there is only a small sized mappings will be affected. Then, the following lemmas come into existence.

Lemma 3.1. Let  $g \in V(G) \setminus (M_{G,S}^{-1}(u) \cup M_{G,S}^{-1}(v))$ . Then,  $M_{G,S'}(g) = M_{G,S}(g)$ .

Lemma 3.2. Let  $g \in M_{G,S}^{-1}(s) \setminus \{\text{Pa}_G(w)\}$ ,  $s \in V(S) \setminus \{u, v\}$ ,  $w \in M_{G,S}^{-1}(u) \cup M_{G,S}^{-1}(v)$ . Then,  $\text{Loss}(G, S, g) = \text{Loss}(G, S', g)$ .

Next, we need two definitions.

Definition 3.1. Given  $s \in \text{valid}(S)$ , let  $b$  and  $a$  separately be the siblings of  $s$  and  $\text{Pa}_S(s)$ . We define  $\text{dep}_S(s) = \{a, b, s, \text{Pa}_S(s)\} \cup \text{Ch}_S(s) \cup \text{Ch}_S(b) \cup \text{Ch}_S(a)$ ,  $\text{ind}_S(s) = \text{valid}(S) \setminus \text{dep}_S(s)$ . The nodes in  $\text{dep}_S(s)$  (or  $\text{ind}_S(s)$ ) are dependent (or independent) on node  $s$  in  $S$ .

Then we can have the next lemma directly.

Lemma 3.3.  $|\text{dep}_S(s)| = |\text{valid}(S) \setminus \text{ind}_S(s)| \leq 10$ .

Definition 3.2. For any  $s \in \text{valid}(S)$ , we define  $\text{diff}_S(s) = \Delta(G', S) - \Delta(G', \text{NNI}_S(s))$ .

Then, we give an important lemma of our work.

Lemma 3.4. For any  $s \in \text{ind}_S(y) \cap \text{valid}(S')$ , we have  $\text{diff}_S(s) = \text{diff}_{S'}(s)$ .

Proof. Let

$$m = \text{Pa}_S(s), \quad w = \text{Pa}_S(m), \quad m' = \text{Pa}_{S'}(s), \quad w' = \text{Pa}_{S'}(m')$$

$$T = \text{NNI}_S(s), \quad T' = \text{NNI}_{S'}(s)$$

$$mw = \{\text{Pa}_S(r), r \in M_{G,S}^{-1}(m) \cup M_{G,S}^{-1}(w)\}$$

$$m'w' = \{\text{Pa}_{S'}(r'), r' \in M_{G,S'}^{-1}(m') \cup M_{G,S'}^{-1}(w')\}$$

$$MW = M_{G,S}^{-1}(m) \cup M_{G,S}^{-1}(w) \cup mw$$

$$M'W' = M_{G,S'}^{-1}(m') \cup M_{G,S'}^{-1}(w') \cup m'w'$$

For a species tree  $R$ , we define  $d_{G,R}(g) = 0$  if  $g \in V(G)$  is not a duplication under  $M_{G,R}$ , otherwise  $d_{G,R}(g) = 1$ . Thus, we have the next equation.

$$\begin{aligned} \text{diff}_S(s) &= |\text{Dup}(G, S)| - |\text{Dup}(G, T)| + \text{Loss}(G, S) - \text{Loss}(G, T) \\ &= \sum_{G \in G'} \sum_{g \in V(G)} (d_{G,S}(g) - d_{G,T}(g)) \\ &+ \sum_{G \in G'} \sum_{g \in V(G)} (\text{Loss}(G, S, g) - \text{Loss}(G, T, g)) \end{aligned}$$

Then, by Lemma 3.1 and Lemma 3.2, we have

$$\begin{aligned} \text{diff}_S(s) &= \sum_{G \in \mathcal{G}} \sum_{g \in M_{G,S}^{-1}(m) \cup M_{G,S}^{-1}(w)} (d_{G,S}(g) - d_{G,T}(g)) \\ &\quad + \sum_{G \in \mathcal{G}'} \sum_{g \in MW} (\text{Loss}(G, S, g) - \text{Loss}(G, T, g)) \end{aligned}$$

Similarly, we must also have

$$\begin{aligned} \text{diff}_{S'}(s) &= \sum_{G \in \mathcal{G}'} \sum_{g \in M_{G,S'}^{-1}(m') \cup M_{G,S'}^{-1}(w')} (d_{G,S'}(g) - d_{G,T'}(g)) \\ &\quad + \sum_{G \in \mathcal{G}'} \sum_{g \in M'W'} (\text{Loss}(G, S', g) - \text{Loss}(G, T', g)) \end{aligned}$$

Next, we let  $n$  and  $t$  be the siblings of  $s$  and  $m$  in  $S$ , and  $n'$  and  $t'$  be the siblings of  $s$  and  $m'$  in  $S'$ . If  $s \in \text{ind}_S(y) \cap \text{valid}(S')$ , we must have  $\text{Le}(S_s) = \text{Le}(S'_s), \text{Le}(S_n) = \text{Le}(S'_n), \text{Le}(S_t) = \text{Le}(S'_t), \text{Le}(S_n) = \text{Le}(S'_n)$ .

This further implies that

$$\begin{aligned} M_{G,S}^{-1}(m) &= M_{G,S}^{-1}(m'), M_{G,S}^{-1}(w) = M_{G,S}^{-1}(w'), mw = m'w', \\ MW &= M'W' \text{ and furthermore for any } g \in M_{G,S}^{-1}(m) \cup \\ M_{G,S}^{-1}(w), &\text{ we must have } d_{G,S}(g) = d_{G,S'}(g), d_{G,T}(g) = d_{G,T'}(g), \\ \text{Loss}(G, S, g) &= \text{Loss}(G, S', g), \text{Loss}(G, S', g) = \text{Loss}(G, T', g). \end{aligned}$$

It is obvious that the change of the structure of the species tree between from  $S$  to  $S'$  and from  $T$  to  $T'$  is similar, so we have

$$\begin{aligned} &\sum_{G \in \mathcal{G}} \sum_{g \in mw} \text{Loss}(G, S, g) - \sum_{G \in \mathcal{G}} \sum_{g \in m'w'} \text{Loss}(G, S', g) \\ &= \sum_{G \in \mathcal{G}} \sum_{g \in mw} \text{Loss}(G, T, g) - \sum_{G \in \mathcal{G}} \sum_{g \in m'w'} \text{Loss}(G, T', g), \end{aligned}$$

that is

$$\begin{aligned} &\sum_{G \in \mathcal{G}'} \sum_{g \in mw} (\text{Loss}(G, S, g) - \text{Loss}(G, T, g)) \\ &= \sum_{G \in \mathcal{G}'} \sum_{g \in m'w'} (\text{Loss}(G, S', g) - \text{Loss}(G, T', g)). \end{aligned}$$

Thus, the lemmas follows.

#### IV. THE ALGORITHM

Our algorithm for solving the 2-NNI local search problem will be described in the following chapter. By definition 2.6, we can separately compute a minimum cost tree in  $\text{NNI}_S$  and  $2\text{-NNI}_S \setminus \text{NNI}_S$ .  $2\text{-NNI}_S \setminus \text{NNI}_S$  are obtained by performing exactly two successive NNI operations on tree  $S$ . So the first step is to give a tree to the 1-NNI. Then, the optimal tree in  $2\text{-NNI}_S \setminus \text{NNI}_S$  will be computed by our algorithm.

Consider some tree  $T \in 2\text{-NNI}_S \setminus \text{NNI}_S$ . There must exist two nodes  $r, s \in V(S)$ , such that  $T' = \text{NNI}_S(s), T = \text{NNI}_{T'}(r)$ . And  $r$  has two cases:  $r \notin \text{ind}_S(s)$  or  $r \in \text{ind}_S(s)$ . For case 1 ( $r \notin \text{ind}_S(s)$ ), for each of the possible  $O(n)$   $s \in \text{valid}(S)$ , there are only 10 possible candidates for  $r$ . For case 2 ( $r \in \text{ind}_S(s)$ ), from lemma 3.4 we can reuse the previous information. The next lemma shows how to effectively solve case 2.

Lemma 4.1. Assume that the valid nodes in  $S$  have been arranged according to the values of  $\text{diff}_S(s)$  decreasingly. And let  $F$  denotes the first eleven nodes and

$\Gamma = \{T = \text{NNI}_{T'}(r), T' = \text{NNI}_S(s), r \in \text{ind}_S(s)\}$ ,  $R^* = \text{argmin}_{R \in \Gamma} \Delta(G', R)$ . Then, there must exist a pair of nodes  $p, q \in F$  such that  $q \in \text{ind}_S(p), P = \text{NNI}_S(p), Q = \text{NNI}_P(q)$ , and  $\Delta(G', R^*) = \Delta(G', Q)$ .

Proof. Assume that  $p', q' \in \text{valid}(S), q' \in \text{ind}_S(p')$  such that  $R^* \in \Gamma$  has minimum cost. Therefore, we must to prove that there must have a pair of nodes  $p, q \in F, q \in \text{ind}_S(p)$  such that  $\text{diff}_S(p) + \text{diff}_{\text{NNI}_S(p)}(q) \geq \text{diff}_S(p') + \text{diff}_{\text{NNI}_S(p')}(q')$ .

Lemma 3.4 allows us to have  $\text{diff}_S(q) = \text{diff}_{\text{NNI}_S(p)}(q)$  and  $\text{diff}_S(q') = \text{diff}_{\text{NNI}_S(p')}(q')$ . Therefore, we should show that  $\text{diff}_S(p) + \text{diff}_S(q) \geq \text{diff}_S(p') + \text{diff}_S(q')$ .

By lemma 3.3, it is true that there must exist nodes  $(a, b) = \text{argmax}_{p, q \in F, q \in \text{ind}_S(p)} (\text{diff}_S(p) + \text{diff}_S(q))$ . Set  $(p, q) = (a, b)$ .

If both  $p', q'$  are in (or not in) the set  $F$ , by the definition of set  $F$  and the choice of  $p$  and  $q$ , we must have  $\text{diff}_S(p) + \text{diff}_S(q) \geq \text{diff}_S(p') + \text{diff}_S(q')$ .

If  $p' \in F, q' \notin F$ , by the definition of set  $F$  and Lemma 3.3, there must exist a node  $c \in F, c \in \text{ind}_S(q')$  such that  $\text{diff}_S(c) \geq \text{diff}_S(q')$ . So, we must have  $\text{diff}_S(p) + \text{diff}_S(q) \geq \text{diff}_S(p') + \text{diff}_S(q')$ .

Similarly, if  $p' \notin F, q' \in F$ , we also have  $\text{diff}_S(p) + \text{diff}_S(q) \geq \text{diff}_S(p') + \text{diff}_S(q')$ .

The lemma follows.

Now we present our algorithm (TABLE I) for the 2-NNI local search problem.

TABLE I. THE ALGORITHM

#### Algorithm 1.

Instance: A set of gene trees  $G'$ , a species tree  $S$ .  
Find: A species tree  $S^* \in 2\text{-NNI}_S$ , such that  $\Delta(G', S^*) = \min_{T \in 2\text{-NNI}_S} \Delta(G', T)$ .

- 1: for any  $s \in \text{valid}(S)$  do
- 2:   Compute the values  $\text{diff}_S(s)$ .
- 3: endfor
- 4: Find  $y \in \text{valid}(S)$  such that  $\text{diff}_S(y)$  is the maximum.  
   Set  $S_1 = \text{NNI}_S(y)$ .
- 5: Set  $S_2 = S$ .
- 6: for every  $s \in \text{valid}(S)$  do
- 7:   Set  $S' = \text{NNI}_S(s)$ .
- 8:   for  $t \in \text{valid}(S') \cap \text{dep}_S(s)$  do
- 9:      $R = \text{NNI}_{S'}(t)$ .
- 10:     if  $\Delta(G', R) < \Delta(G', S_2)$  then set  $S_2 = R$ .
- 11:   endfor
- 12: endfor
- 13: Let  $F$  denotes the first eleven nodes in  $S$  arranged according to  $\text{diff}_S(s)$  decreasingly.
- 14: Find  $p, q \in F, q \in \text{ind}_S(p)$  such that  $\text{diff}_S(p) + \text{diff}_S(q)$  is the maximum.
- 15: Set  $S_3 = \text{NNI}_{\text{NNI}_S(p)}(q)$ .
- 16: return  $\text{argmin}_{T \in \{S_1, S_2, S_3\}} \Delta(G', T)$ .

Let  $n = |\text{Le}(S)|$ ,  $m = |\text{Le}(S) + \text{Le}(G)|$ ,  $r = |G'|$ . Then, we can calculate the complexity of the algorithm.

Theorem 4.1. Algorithm 1 solves the Duplication-Loss problem based on the 2-NNI-Search in  $O(rmn)$  time.

## V. OUTLOOK AND CONCLUSION

The gene-duplication problem has been an effective way for inferring species evolutionary process[15]. In our work, we introduced novel algorithm for the duplication-loss problem based on k-NNI local search. Our new algorithm for solving the duplication-loss problem can greatly improve on the performance of the already algorithms by a factor of  $n$ . Our improvement makes the duplication-loss problem much more manageable for large-scale phylogenetic analyses.

## REFERENCES

- [1] M. Goodman, J Czelusniak, G. W. Moore, A. E. Romero-Herrera, and G. Matsuda, "Fitting the gene lineage into its species lineage: A parsimony strategy illustrated by cladograms: constructed from globin sequences," *Systematic Zoology*, vol. 28, pp. 132–163, Nov. 1979.
- [2] R. Guigo', L Muchnik, and T F. Smith, "Reconstruction of ancient molecular phylogeny," *Molecular Phylogenetics and Evolution*, vol. 6, pp. 189–213, Nov. 1996.
- [3] B. Ma, M. Li, and L. Zhang, "From gene trees to species trees," *SIAM J. Comput.*, vol. 30, pp. 729–752, Nov. 2000.
- [4] R. D. M. Page, "Gene Tree: Comparing gene and species phylogenies using reconciled trees," *Bioinformatics*, vol. 14, pp. 814–820, Nov. 1998.
- [5] M. Bordewich and C Semple, "On the computational complexity of the rooted subtree prune and regraft distance," *Annals of Combinatorics*, vol. 8, pp. 409–423, Nov. 2004.
- [6] R. D. M. Page and M. A. Charleston, "From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem," *Molec. Phyl. and Evol.*, vol. 7, pp. 231–240, Nov. 1997.
- [7] M. S. Bansal, O. Eulenstein, and A. Wehe, "The gene-duplication problem: Near-linear time algorithms for NNI-based local searches," in *IEEE/ACM TCBB*, 2009, pp. 221–231.
- [8] R.D.M. Page, " Maps between Trees and Cladistic Analysis of Historical Associations among Genes, Organisms, and Areas," *Systematic Biology*, vol. 43, no. 1, pp. 58–77, Nov. 1994.
- [9] B. Mirkin, I. Muchnik, and T.F. Smith, "A Biologically Consistent Model for Comparing Molecular Phylogenies," *J. Computational Biology*, vol. 4, no. 4, pp. 493–507, Nov. 1995.
- [10] L. Zhang, "On a Mirkin-Muchnik-Smith Conjecture for Comparing Molecular Phylogenies," *J. Computational Biology*, vol. 4, no. 2, pp. 177–187, Nov. 1997.
- [11] K. Chen, D. Durand, and M. Farach-Colton, "NOTUNG: A Program for Dating Gene Duplications and Optimizing Gene Family Trees," *J. Computational Biology*, vol. 7, no. 3/4, Nov. 2000.
- [12] P. Bonizzoni, G.D. Vedova, and R. Dondi, "Reconciling a Gene Tree to a Species Tree under the Duplication Cost Model," *Theoretical Computer Science*, vol. 347, pp. 36–53, Nov. 2005.
- [13] M.S. Bansal, J.G. Burleigh, O. Eulenstein, and A. Wehe, "Heuristics for the Gene-Duplication Problem: A  $\theta(n)$  Speedup for the Local Search," in *Proc. RECOMB*, 2007, pp. 238–252.
- [14] M.A. Bender and M. Farach-Colton, "The LCA Problem Revisited," in *Proc. LATIN*, 2000, pp. 88–94.
- [15] J.A.Cotton and R.D.M.Page, "Phylogenetic supertrees:Combining information to reveal the tree of life, ch.Tangled tale from multiple markers: reconciling conflict between phylogenies to build molecular supertrees," *Springer-Verlag*, pp.107–125, Nov. 2004.