BEN-GURION UNIVERSITY OF THE NEGEV

THE FACULTY OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

A new gene tree reconciliation algorithm suited for tandem paralogs

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE MASTER OF SCIENCES DEGREE

WRITTEN BY: Nir Gilad

UNDER THE SUPERVISION OF: Prof. Michal Ziv-Ukelson

December 2018

BEN-GURION UNIVERSITY OF THE NEGEV

THE FACULTY OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

A new gene tree reconciliation algorithm suited for tandem paralogs

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE MASTER OF SCIENCES DEGREE

WRITTEN BY: Nir Gilad

UNDER THE SUPERVISION OF: Prof. Michal Ziv-Ukelson

Signatures:

[illegible]	24.01.2019
Author: Nir Gilad	Date
מיכל זיו יוקלסון	19.02.2019
Advisor: Prof. Michal Ziv-Ukelson	Date
Dept. Committee Chairperson	Date

December 2018

A new gene tree reconciliation algorithm suited for tandem paralogs

Nir Gilad

Thesis submitted in partial fulfillment of the requirements for the Master of Sciences degree

Department of Computer Science, Faculty of Nature, Ben Gurion University, Beer Sheva, Israel

December 2018

Abstract

Motivation: Duplications of long DNA segments are constantly produced by rare mutations. They may become fixed in a population by selection or random drift, and are subject to divergent evolution of the paralogs sequences after fixation. Reconciliation is a method widely used to infer the evolutionary relationship between the members of a gene family. It consists of comparing a gene tree with a species tree, and interpreting the incongruence between the two trees as evidence of duplication, transfer, and loss. However, existing reconciliation algorithms ignore the evolutionary behavior dissimilarity between tandem duplications and other duplications forms.

Result: We address this problem by giving an efficient algorithm for computing gene tree and species tree reconciliation, based on the DTL model with an additional mapping and scoring model that is suited for monitoring tandem gene duplications and the tandem participation of the duplication product in gene evolutionary events such as horizontal gene transfer of the original gene jointly with its tandem duplicate and more.

Contents

1. Introduction....................................1
2. Background on Previous Tools and Methods.........3
  2.1 The Duplication-Transfer-Loss Model.........3
    2.1.I. DTL Scenario........................3
    2.1.II. Loses...............................4
  2.2 Previous works..............................5
3. Model breakdown..................................6
  3.1 Definitions.................................6
    3.1.I. Preliminaries.......................6
    3.1.II. The TP reconciliation problem.......6
  3.2 Intuitive explanation of Algorithm 3.......13
  3.3 Algorithm 3 Pseudocode.....................16
4. Methods..20
5. Experimental results.............................20
  5.1 The Data...................................20
  5.2 PreProcess Gene tree.......................21
  5.3 Scenario described by TP mapping...........23
6. Conclusions and Future Work......................23
7. Bibliography.....................................24

List of Figures and Tables:

Figure 1: Workflow of our proposed approach for solving the TP reconciliation problem. 2
Figure 2: An example of Pre Process output. 9
Figure 3: An example scenario involving both a speciation and an HGT event of TP genes. 12
Table 1: Example of the speciation-event scoring function. 15
Table 2: Infographic of genes location in species. 21
Figure 4: The Gene tree after Alg.1 and Alg.2 pre process. 22
Figure 5: Specie tree reflecting the major findings of the most optimal reconciliation. 22

1. Introduction

Phylogenetic Tree Reconciliation is a powerful approach for inferring the history of evolutionary events leading to the occurrence of multiple homologous copies of a gene among several species related by common ancestry, and to multiple occurrences of homologous genes within the same species. In the Gene\Specie tree Phylogenetic Reconciliation problem, a phylogenetic specie tree, T_{sp} is constructed based on a set of input species X, and then compared to an input phylogenetic gene tree T_g, that was constructed based on a specific gene from X. The difference between the trees may be the consequence of evolutionary events that took place in the gene sequences, which are not always consistent with specie evolution [1].

Gene duplication is an evolutionary phenomenon in which one or more genes in an organism duplicate. Both copies of the duplicated genes then evolve independently, and the newly duplicated gene may differentiate into new functional niches [2], Gene duplication are known to play a major role in the evolution of almost all life on Earth.

The simplest form of duplication is tandem duplication, such that the inserted DNA segment is identical or nearly- identical to a segment immediately bordering the site of insertion. In the case of perfect tandem duplication, the new and old copies of the duplicated segment are indistinguishable at the sequence level [3]. In prokaryotes, gene amplification is typically characterized as a step-wise process initiated by tandem duplication [4].

Existing tree reconciliation algorithms are able to recognize duplications and the undergoing evolutionary events on the duplicated genes, but they do not distinguish between tandem duplication resulting in adjacent paralogs genes, and other forms of duplications. In this work, we expand the DTL tree reconciliation problem to recognize tandem paralogous genes in the species involved, and trace their evolution with an extended mappings and scoring model. The goal is to be able to locate where on the specie tree the tandem paralogs were created via tandem duplication and to follow their horizontal transfer and speciation evolutionary events throughout the specie tree. More formally, the new phylogenetic problem is defined as follows.

Definition 1: (*The Tandem Paralogs (TP) reconciliation problem*). Given a specie tree S, a gene tree G, a labeling function from $leafs(G)$ to $leafs(S)$. Also given a binary relation for any two genes indicating whether they are tandem paralogs located on the same specie or not, and a scoring function F denoting the cost of duplication, loss, and horizontal gene transfer (HGT) events, Compute the minimal scoring scenario mapping each vertex of G to one of the vertices of S, taking into account the tandem paralogs information.

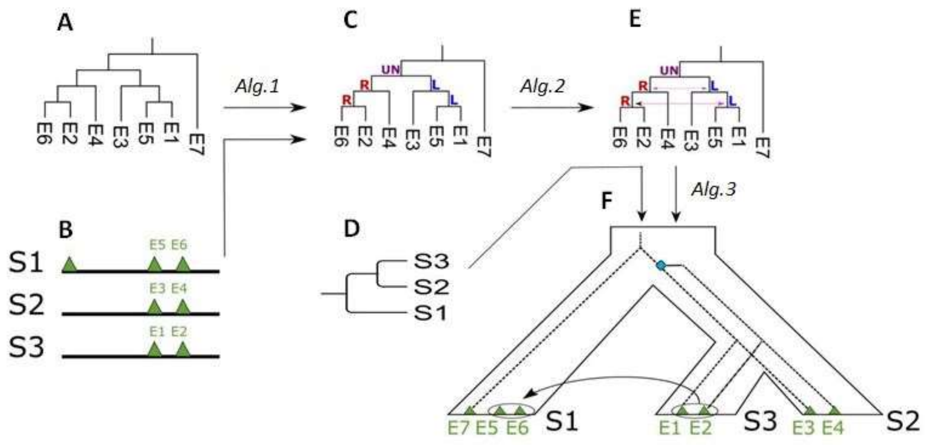


***Figure 1.** Workflow of our proposed approach for solving the TP reconciliation problem. Genes E1..E7 located on 3 species S1..S3. **A)** Input gene tree. **B)** The locations of the genes on the three input genomes (species). **C)** Algorithm 1: using gene location on species genomes for recursively labelling the gene tree inner nodes with OL values. **D)** Specie tree. **E)** Algorithm 2: linking gene tree nodes based on OL values. **F)** Finally, the processed gene tree and the specie tree are subjected to the TP-DLT algorithm (Alg.3) resulting in the tree reconciliation.*

Our Contribution and Workflow.

In this thesis we define a new problem, propose efficient algorithms to solve it, and demonstrate the application of our approach on a biological example.

Figure 1 illustrates the high-level workflow of our proposed solution to the TP Reconciliation problem on an artificial example. The example includes three species consisting each with two TP genes (e.g. genes $E3, E4$ on specie $S2$) and another single gene $E7$ located on $S1$. The tree reconciliation final results, as shown if Figure 1.F, implies that a single gene speciated to $E7$ on $S1$ and underwent a tandem duplication forming a TP on the ancestor of $S2$ and $S3$, marked by a blue polygon. The TP speciated into $E1E2$ in $S3$ and into $E3E4$ in $S2$. Finally an HGT (Horizontal gene transfer) event of a complete TP happened from $S3$ to $S1$.

Note that, in the original DLT tree reconciliation algorithm, the transfer of TP would have been described by two independent transfer events of each of the TP genes. Here, in contrast, we added a new mapping denoted TP-mapping, that is able to consider the fact that the two genes undergoing transfer are part of a TP, giving the event a better score based on this information.

All algorithms proposed in this thesis have a polynomial running time: The running times of both Algorithms 1, and Algorithm 2 is $O(|V(G)|)$, as they require a single binary tree traversal on the gene tree G. The main reconciliation algorithm, Algorithm 3, has a running time of $O(|V(G)| \cdot |V(S)|^2)$.

2. Background on Previous Tools and Methods

Algorithms for computing phylogenetic Tree Reconciliation are classically based on dynamic programming. The reconciliation algorithm takes as input a gene tree and a species tree and optimizes a reconciliation scenario between the two trees. The reconciliation classically considers cost free speciation events and three other evolutionary events with corresponding predefined optimization costs: deletion, Loss and Transfer (DLT scenario for short) [5]. This problem is naively solved via dynamic programming in $O(|V(G)| \cdot |V(S)|^2)$ time.

2.1 The Duplication-Transfer-Loss Model

In order to achieve a biologically valid tree reconciliation, we define a Duplication-Transfer-Loss scenario (DTL-scenario) [1], for G and S. In essence, DTL-scenario extends the agreement between the genes and species given by the leaf mapping function σ to a complete mapping of the every gene tree node to a specie node with respect to the constraints created by the topology of S and by that of appointing evolutionary event to each node in the gene tree.

Definition 2.1.I (DTL-Scenario). [6] [1] A DTL-scenario for G and S is a seven-tuple $< l, \mathrm{M}, \Sigma, \Delta, \Theta, \Xi, \tau >$, where $l: L(G) \to L(S)$ represents the leaf mapping from G to S, $\mathrm{M}: V(G) \to V(S)$ maps each nodeof G to a node of S, the sets $\Sigma, \Delta,$ and Θ partition $\dot{V}(T)$ into speciation, duplication and transfer nodes, respectively, Ξ is a subset of gene tree edges that represents transfer edges, and $\tau: \Theta \to V(S)$ specifies the recipient species for each transfer event, subject to the following constraints:

1. If $u \in L(G)$, then $\mathrm{M(u)} = \mathrm{l(u)}$.

2. If $u \in \dot{V}(G)$, and v and w denote the children of u, then,

(a). $\mathrm{M}(u) \nleq_S \mathrm{M(v)}$ and $\mathrm{M}(u) \nleq_S \mathrm{M(w)}$.

(b). At least one of $\mathrm{M(v)}$ and $\mathrm{M(w)}$ is a decadent of $\mathrm{M(u)}$.

3. Given any edge $(u, v) \in E(G), (u, v) \in \Xi$ if and only if $\mathrm{M(u)}$ and $\mathrm{M(v)}$ are incomparable.

4. If $u \in \dot{V}(G)$ and v and w denote the children of u, then

(a). $u \in \Sigma$ only if $\mathrm{M(u)} = \mathrm{lca_S}\big(\mathrm{M(v)}, \mathrm{M(w)}\big)$ and $\mathrm{M(u)}$ and $\mathrm{M(v)}$ are incomparable.

(b). $u \in \Delta$ only if $\mathrm{M(u)} \geq_S \mathrm{lca_S}\big(\mathrm{M(v)}, \mathrm{M(w)}\big)$.

(c). $u \in \Theta$ *if and only if either* $(u, v) \in \Xi$ or $(u, w) \in \Xi$.

(d). $u \in \Theta$ and $(u, v) \in \Xi$, then $\mathrm{M(u)}$ and $\tau(u)$ must be incomparable, and $\mathrm{M(v)}$ must be a descendant of $\tau(\mathrm{u})$, $i.e.$ $\mathrm{M(v)} \leq_S \tau(\mathrm{u})$.

Constraint 1 ensures that the mapping M is consistent with the leaf-mapping l. Constraint 2a imposes on M the temporal constraints implied by S. Constraint 2b implies that any internal node in G may represent at most one transfer event. Constraint 3 determines the edges of G that are transfer edges. Constraints 4a, 4b, and 4c state the conditions under which an internal node of G may represent a speciation, duplication, and transfer, respectively. Constraint 4d specifies which species are the recipient species for given transfer event. Note the overlap between Conditions 4b and 4c: given a mapping M, the set of gene tree vertices that may be labeled as speciation according to Condition 4b is a subset of those that may be labeled as duplications according to Condition 4c. Of course, no most parsimonious DTL-scenario will label a gene tree vertex as duplication if the vertex just as well labeled a speciation.

Definition 2.1.II (Losses).

The minimum number of losses inferred from a scenario is computed by considering each nontransfer edge (u, v) of the gene tree and the mapping of its vertices into the species tree. This is similar to how losses are computed in the duplication-loss model. One loss is inferred for each intermediate species tree vertex between $\mathrm{M}(u)$ and $\mathrm{M}(v)$. A loss is also inferred when $u \in \Delta$ and $\mathrm{M}(v) \neq \mathrm{M}(u)$. We can make this argument formal as follows: Let α be a scenario for S, G and l, define $I_\alpha(e)$, where $e = (u, v) \in E(G)$, to be the number of intermediate species tree vertices between $\mathrm{M}(u)$ and $\mathrm{M}(v)$: $I_\alpha(e) = |\{x \in V(S): \mathrm{M(v)} <_S \mathrm{x} <_S \mathrm{M(v)}\}|$.

Note that $I_\alpha(e) = 0$ when $e \in \Xi$. The number of losses inferred by e is

$$loss_a(e) = \begin{cases} I_\alpha(e) + 1, if\ u \in \Delta \text{ and } \mathrm{M(v)} \neq \mathrm{M(v)} \\ I_\alpha(e), \quad otherwise. \end{cases}$$

The total number of losses of the scenario is then

$$loss(\alpha) = \sum_{e \in E(G)} loss_\alpha(e).$$

We assume that a speciation event has zero cost and let $P_\Delta, P_\Theta, and\ P_\Lambda$ denote the assigned positive costs for duplication, transfer, and loss events, respectively. The cost of reconciling G and S according to a DTL-scenario α is defined as $P_\Delta \times |\Delta| + P_\Theta \times |\Theta| + P_{loss} \times loss(\alpha)$.

2.2 Previous works

Given two trees (a Gene tree and a Species tree), the problem of finding the optimal reconciliation has been studied for a long time. Most of its formulations are based on a parsimony approach [7] [8] [9] [10] [11] [12] [1]. In this work, we will use the parsimony formulation, in which each event of Speciation, Duplication, or Horizontal Transfer has a cost, and the goal is to find a reconciliation scenario with the lowest total cost. Let *m* denote the number of nodes in the Gene tree, and *n* the number of vertices in the Species tree. Biologically, genes can undergo horizontal transfer from one specie into a contemporary specie. The problem of finding a DLT scenario, which obeys this biological constraint, is referred to as the acyclic DLT reconciliation. We distinguish between two versions of the DLT problem: (1) the undated DLT-reconciliation in which the species are undated, and (2) the fully-dated DLT-reconciliation in which each vertices in the Species tree is associated with an estimated date or the Species tree vertices are associated with a total order, and any reconciliation must respect those dates, i.e., an HT event can occur only between co-existing species. [1] Have shown that the acyclic problem is NP-Hard, while the acyclic dated problem becomes polynomially solvable, as shown in [11]. In general, the Species tree can be undated, partially dated or fully dated. [13] used a Linear Programming approach to solve the undated acyclic problem, and [14] gave an $O(m \cdot \log(n))$ algorithm to decide whether an acyclic reconciliation mapping exists. [1], [5], and [8] studied a version of the problem that ignores losses and proposed a $O(m \cdot n^2)$ dynamic programming algorithm for the undated cyclic problem. They also gave an FPT algorithm for enumerating all optimal solutions. The time complexity was improved to $O(m \cdot n)$ in [5] (under the restricted model which ignores the losses) and in [6] (which does not ignore losses).

3. Model breakdown

3.1 Definitions

3.1.I. PRELIMINARIES

In this thesis, all the trees are considered rooted (we omit to mention it each time). For tree $T,$ we let $V(T),$ and $\dot{V}(T)$, denote the sets of vertices and internal vertices. Also $L(T)$, and $l(T)$ denote the set of leaf nodes, and the set of labels representing the leafs respectively, and such that only its leaves are labeled. The root of Tree T is denoted by $R(T)$ and the Subtree of T rooted at $x \in V(T)$ is denoted by T_x. We denote edges of tree T by $E(T)$ and consider them to be directed from the root out. An edge of a rooted tree is denoted by an order pair of vertices (u, v), where u is closer to the root than v.

Let T be a rooted tree. if (u, v) is an edge of T, then u is called a child of u, and u is called the parent of v denoted by $Pt(v) = u$. Two distinct vertices u and v are siblings $iff\ Pt(u) = Pt(v)$. A directed path from vertex u to vertex v denoted by $v \leq_T u$ exist $iff\ v$ is a descendent of u, in that case we also say u is an ancestor of u, denote by $u \geq_T v$. Two vertices u and v are incomparable iff $u \nleq_T v$ and $v \nleq_T u$.

A specie tree S is a binary tree such that each $x \in l(S)$ represents extant specie matching exactly one of S leafs, a gene tree G is a binary tree such that each $u \in l(G)$ represent an extant gene. The function $l: L(G) \rightarrow L(S)$ maps each leaf of G to a leaf of S containing the gene. Trees will be described using the newick format.

3.1.II. The TP reconciliation problem definition

We use the term “tandem paralogs” (TP) to denote two paralogous genes that occur tandemly in a genome. Due to the high sequence homology between the two parts of a TP and the fact that some species have only one copy of the paralogs, we do not know in advance which part of the TP to map it to. Thus, we use a single gene tree to describe the homology of all genes participating in a TP.

In this thesis, we consider a new problem in the reconciliation of gene trees with species trees. In our problem variant, we focus on the creation and evolution of TP's. Our assumption is that TP's evolve tandemly and are tandemly transported in horizontal transfer events. Our goal is to design an algorithm that computes reconciliation scenarios for a gene tree with a species tree under this model.

Our phylogenetic assumptions are as follows:

(1) Both genes in a TP were formed by a duplication event from the same ancestor gene.

(2) Once a TP is formed, both genes comprising the TP will be laterally transferred together.

(3) We do not handle in this work the splitting of a TP, which is quite an uncommon event.

(4) We do handle the deletion of one of the genes comprising the TP.

From the moment the TP is formed, each gene will undergo speciation separately resulting in descendents who are genetically closer to their origin TP gene.

Based on the assumption above, we set out to solve the following problem,

Definition 1: (The Tandem Paralogs (TP) reconciliation problem).

Given a specie tree S, a gene tree G, a labeling function from $leafs(G)$ to $leafs(S)$, Also given a binary relation for any two genes indicating whether they are tandem paralogs located on the same specie or not, and a scoring function F denoting the cost of duplication, loss, and horizontal transfer events. Compute the minimal scoring scenario mapping each vertex of G to one of the vertices of S, taking into account the tandem paralogs information.

During a preprocessing stage, two algorithms are used to reflect the information regarding the participation of the genes in a TP into the gene tree.

Algorithm 1 receives as input, together with the gene tree itself, the binary relation between each two genes indicating whether they are both part of TP or not. Based on this, the algorithm labels the gene tree nodes based on this information with a label called OL, defined below.

Definition 2 (Occurrence location OL).

OL is a labeling on the nodes of $T(G)$, that recursively reflects information regarding the participation of the gene, represented by the node, in a TP.

Four possible labels are used for this annotation (denoted OL):

S – denotes that this gene is not part of a TP

R – denotes that this gene is a right part of a TP

L – denotes that this gene is a left part of a TP

UN – denotes the creation of a TP.

The algorithm to annotate the nodes of G with OL proceeds as follows: During initialization, the gene tree leafs, each represent a gene, are annotated with one of the first 3 labels, $S, R,$ or L based on the binary relation between the genes, that indicating if their occurrence is single or part of a TP. If two genes found to be part of a TP, the final OL label (R or L) is determined using the specie genome map. Next, the gene tree is traversed in postorder, and the following pseudo code is applied to compute the OL of each traversed node:

```
Algorithm 1. $OL$ Labeling of the gene tree inner vertices.
  input: $\boldsymbol{G}$ with $OL$ labeled leafs.
  Output: $OL$ labeled gene tree $G$.
1.for $u \in \dot{V}(G1)$ $in\, postorder$ do
2.Let $v,w$ be the children's of $u$
3.  if $\boldsymbol{OL(v) = S}$ and $\boldsymbol{OL(w) = S}$ then
4.    $OL(u) = S$
5.  else if $OL(v) = R\ or\ S$ and $OL(w) = R\ or\ S$ then
6.    $OL(u) = R$
7.  else if $OL(v) = L\ or\ S$ and $OL(w) = L\ or\ S$ then
8.    $OL(u) = L$
9.  if ($OL(v) = R$ and $OL(w) = L$) or
        ($OL(w)$=L and $OL(v) = R$) then
10.   $OL(u) = UN$
11.end
```

Creation of a new TP by a gene duplication event will be followed by different speciation of the two adjacent genes; this will generally result in two separated subtrees in the gene tree, one for each TP gene speciation (see Figure 2 nodes 3 and 4 in the gene tree). The creation of TP's might be located in the input gene tree, or further up in evolution, (in that case, the gene tree root itself holds an ancestor of the TP). If the TP was created within the gene tree evolution boundaries at some node $u \in, \dot{V}(G)$, node u children's will be labeled with the two TP opposite OL labels R and L and u will be labeled with UN, indicating the creation of a TP in that location.

The OL labels are used both in the next definition of $Glink$ and during the algorithm run to verify that a gene tree node indeed holds an ancestral TP.

If vertex u's children are both leafs labeled with $OL = R$ $(w.l.o.g)$, the genes represented by the children nodes speciated from the same right gene in a TP and so does their parent ancestral gene represented by the vertex u, so we label $OL(u) = R$. If only one of u children's is marked with R and the other children with S, u would still be marked with R: by this we are taking into consideration the case where Deletion of a gene occurs on the path between u and their children labeled S (See Table 1 for an illustrated example of this point).

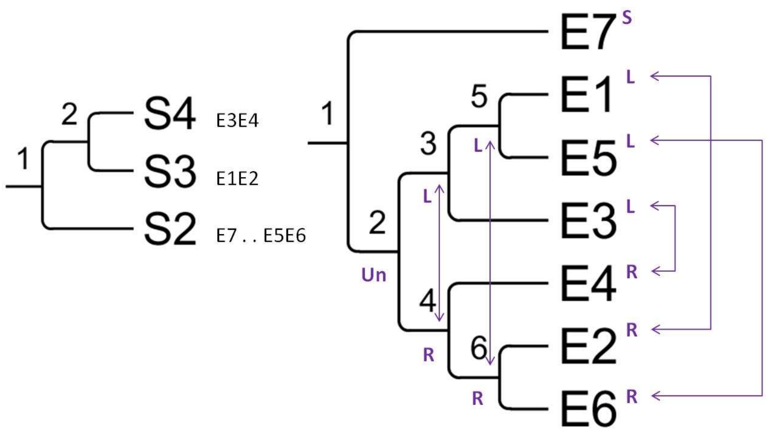


Figure 2 *An example of Pre Process output. This Figure continues the Example of Figure 1, and illustrates TP speciation throughout evolution. S is a specie tree over L(S)={S2,S3,S4}. Each species contain single TP occurrence consisting of two genes, and S2 holds an additional isolated gene copy. G is the gene tree over L(G)={E1,..,E7}.G's vertices also show their corresponding OL() function value, the leafs OL() value is obtained directly from their location in a TP, while the inner nodes OL() value is deduced from its children's. The arrows between G's vertices represent the Gene linkage produced by Alg.2: Glink(6)=5, Glink(4)=3.*

Definition 3 (Gene Linkage ($Glink$)).

In our model, either a TP is generated once during a tandem duplication event in S, and then the components of the TP travel together through the species tree till they reach a leaf, or one or both of the genes participating in the TP are lost. Based on this assumption, one could annotate the nodes of G in advance by linking together genes that are paired up in a TP, based on the evidence found in the leafs of S. This pairing, which we denote "linkage", is also computed during preprocessing, by initializing the genes in $G's$ leafs based on the binary relations between the genes concerning TP's given as input, and then propagating this linkage recursively up the tree G.

$Glink: V(G) \rightarrow V(G)$ is a function connecting two gene tree nodes where each the nodes represent half of a TP. Given a gene binary tree labeled by $Alg. 1$ OL labels, and the relation between each two genes regarding their occurrence in a TP together (see introduction), first $Glinks$ are created between gene tree leafs where the genes represented by the leafs are part of the same TP, Following the example given in Figure 2, if the genes $E3, E4 \in l(G)$ form together a TP in the specie $S4 \in l(S)$, we link the gene leaf vertices representing the genes to each other $Glink(L(E3)) = L(E4)$.
Next the inner vertices are traversed in postorder, and the following pseudo code is applied to link the inner vertices:

```
Algorithm 2. Gene tree inner vertices linking.
  input: $OL$ labeled gene tree $G$.
  Output: Glink gene tree $G$.
1.for $u \in \dot{V}(G)$ $in\, postorder$ do
2.Let $v, w$ be the children's of $u$
3. if $OL(v) = OL(w)$ then
4.     $Glink(u) = LCA(T_G).(Glink(v), Glink(w))$
5. if $OL(v) = (R/L)$ and $OL(w) = S$ (w.l.o.g) then
6.     $Glink(u) = Glink(v)$
7.end
```

For $u \in \dot{V}(G)$, let v, w be the children of u. If both children of u are linked, $Glink(v) = v', Glink(w) = w'$, and the linked nodes $v', w' \in V(G)$ are siblings, $Pt(v') = Pt(w')$, then u is linked to the parent of the linked sibling nodes, $Glink(u) = P(Glink(v))$. Note that only vertices labeled with either $OL(u) = L$, or $OL(u) = R$ can be linked. In Figure 2, vertex 6 in the gene tree G is linked to vertex 5 as an example. By linking the inner vertices, we are able to connect and follow the two genes of the same TP through evolution up to possibly the point of creation of the tandem gene duplication at some ancestral inner vertex.
If $OL(v) = R$ and $OL(w) = S$ (w.l.o.g) and v is linked, $Glinked(v) = v'$, then we link u to the same vertex his children is linked to, $Glink(u) = Glink(v) = v'$.This special case is used in case a gene loss occurs and as a result the gene is no longer part of a TP.

Definition 4 (TP Mapping).

Mapping of gene tree vertex $u \in V(G)$ to a specie tree vertex $x \in V(S)$ will hold an additional mapping and score suited for the case of evolution of a TP. The TP-mapping requires from the participating gene tree node certain OL and $Glink$ values: $OL(u) = R$ or $OL(u) = L$ and that u is linked to another gene tree node with an opposite OL value, describing the other gene belong to the same TP. If all the requirements are satisfied, $c_{\Sigma}(u, x)[TP]$, and $c_{\Theta}(u, x)[TP]$ will be constructed based on u children's TP mappings scores. Both mappings will then be offered to the next mappings in the algorithm for use, yet only further TP-mappings or a Unifying node mapping (See definition 5) will be able to use these scores.

In the case of transfer, TP-mapping will offer a better score for the cases where the two genes in the TP are both part of a single HGT event, in comparison to two separated HGT events of each gene in the original mapping scoring.

In the case of speciation, while not necessarily giving score benefits (both TP mapping and original mapping speciation cost are 0), TP mappings are still necessary since they will be used by other TP-mappings in the tree and together will fully describe the evolutionary scenario that the TP went through. Example of this scenario is shown in figure 3, where TP HGT event, described by the mappings $(6g, 5s)$ and $(7g, 5s)$, is based on the TP speciation mapping of the gene tree node children's $(8g, 9g, 10g, and\ 11g)$.

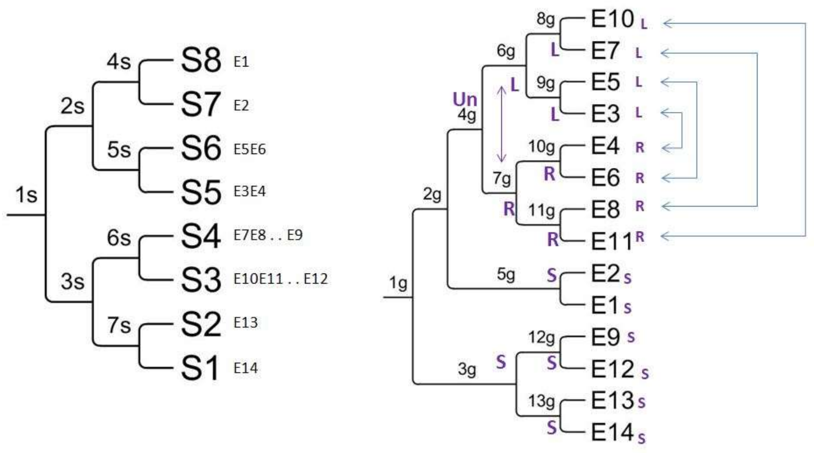


Figure 3 *An example scenario involving both a speciation and an HGT event of TP genes. Samples S3 - S6 hold TP genes. The mappings c(9g,5s) and c(10g,5s) describe together the speciation of a TP from node 5s leading current species S5, and S6. Similarly mappings (8g,6s) and (11g,6s) describe together the speciation from node 6s to the current species S4, and S3. The next nodes in the algorithm, 6g and 7g, will be able to use the mentioned TP speciation mappings to describe a TP HGT event. The left part of the TP HGT event is described by the mapping (6g,5s)=(9g,5s)+(8g,6s), and the complementary right HGT TP mapping event is describe by in (7g,5s) = (10g,5s)+(11g,6s).*

Definition 5 (Unifying node Mapping (UN)).

A Unifying Node Mapping (UN mapping) is a special case of duplication mapping, used to combine two TP mappings to the single gene that created the TP.

Figure 3 showcases a scenario where TP genes undergo an HGT event from the specie node $5s$ to node $6s$, then a speciation event leading to the TP genes in current existing species ($E5E6, E3E4, E7E8, E10E11$). Since the mapping algorithm traverses the tree nodes in postorder, the HGT of TP described in the mappings $(6g, 5s), (7g, 5s)$ was based on TP's speciation located in the mappings $(9g, 5s), (10g, 5s), (8g, 6s)$, and $(11\text{g}, 6\text{s})$. Inner gene tree node $u \in \dot{V}(G)$, with children's $v, w \in V(G)$, must be labeled as $OL(u) = U$, meaning his two children's are labeled one with R and L $OL()$ values. We also required that v and w will be $Glinked$ to each other. Together, these two requirements indicating that node u is the location on the gene tree where the TP was created. For mapping (u, x) $x \in T(sp)$, to hold UN mapping, v and w must both hold TP-mappings to x themselves, describing the evolution of TP after the creation at location x in the specie tree and location u in the gene tree. Hence, UN mapping of (u, x) will always be in the form of $(u, x) = (v, x)[TP - mapping] + (w, x)[TP - mapping]$.

3.2 Intuitive explanation of Algorithm 3

In this section we propose a polynomial time and space algorithm that uses the DLT traceable reconciliation model (see previous works section) and is built on the algorithm proposed in [1], that computes an optimal reconciliation scenario to solve the TP tree reconciliation problem. Specifically, we begin by traversing the undated reconciliation graph in postorder – that is, first the leaves, then their parents, and so forth towards the roots. We compute multiple values under the same division to evolutionary events as in the DLT scenario: Σ, Δ, Θ (speciation, duplication, and transfer) for each $u \in \dot{V}(G)$ *and* $x \in V(S)$, by performing a nested post-order traversal of *G* and *S*. In addition to the values computed in the original algorithm, we add new scores computed by additional new mappings, serving the purpose of tracing the evolution of TP genes:

Initialization

If $u \in L(G)$, $c\big(u, \sigma(u)\big)[E] \leftarrow \begin{cases} 0 & \boldsymbol{if}\ OL(u) = S \\ c & otherwise \end{cases}$

$$c\big(u, \sigma(u)\big)[EE] \leftarrow \begin{cases} c & \boldsymbol{if}\ OL(u) = S \\ 0 & otherwise \end{cases}$$

The initialization of leaf mappings is based on both matching to correct specie nodes, and on the $OL(u)$ labels. Each leaf matching will hold score for two fields: E,EE if the gene represented by this gene tree leaf $u \in L(G)$ is part of a TP then $OL(u)$ will be R, or L and the EE score field will hold the score 0.

Each of the three events mapping Σ, Δ, Θ will keep the original calculation suited for a single gene, in this work marked by 'E', e.g. $c_\Sigma(u, x)[E]$ and originally $c_\Sigma(u, x)$. Moreover, we added new scoring type marked 'EE' e.g. $c(u, x)[EE]$, available only for speciation and transfer events, representing the state where two adjacent copies of the gene exist on node x in the specie tree. When calculating the score of $c_\Sigma(u, x)[E]$, the two mappings of u children's can be composed of any combination of E and EE scores, each combination will represent a different case and result in a different score from the scoring function. See Table 1 for illustration of number of different scoring cases.

For $u \in \dot{V}(G)$ and $x \in V(S)$, let $v, w \in V(G)$ be the children's of u;

(a)Speciation:

(a).1. $c_{\Sigma}(u,x)[E] = min\{score\ (E, c(v,y)[E,EE] + c(w,z)[E,EE]) :$
$y\ incomparable\ to\ z\ and\ lca\{y,z\} = x\}$

(a).2. $c_{\Sigma}(u,x)[EE] = min\{score\ (EE, c(v,y)[E,EE] + c(w,z)[E,EE]) :$
$y\ incomparable\ to\ z\ and\ lca\{y,z\} = x\}$

(b)Transfer:

(b).1. $c_{\Theta}(u,x)[E] = \mathrm{m}in\{score\ (c(v,y)[E,EE] + c(w,z)[E,EE]) :$
$y\ incomparable\ to\ x\ and\ z \leq_S x\}$

(b).2. $c_{\Theta}(u,x)[EE] = \mathrm{m}in\{score\ (c(v,y)[E,EE] + c(w,z)[E,EE]) :$
$y\ incomparable\ to\ x\ and\ z \leq_S x\}$

For now, the algorithm supports only the occurrence of TP's consisting of two genes, due to that fact there is no use for an 'EE' score field under duplication event.

(c)Duplication

(c).1. $c_{\Delta}(u,x)[E] = \{Punish(c(v,y)[E,EE] + c(w,z)[E,EE]) : z, y \leq_S x\}$

Speciation and Transfer events of TP genes will be evaluated by TP-mappings, using the OL labels and $Glink$ linked gene tree nodes, produced by $Alg.1$ and $Alg.2$ in preprocess, to assure that there exists a gene tree node u' to complete the other half of the TP mapping, as described below:

(a) Speciation:

(a).1. $c_{\Sigma}(u,x)[TP] = min\{scoreTP(c(v,y)[TP] + c(w,z)[TP])$:
$y\ incomparable\ to\ z,\ lca\{y,z\} = x, OL(u) = R\ or\ L, and\ \exists u' \in V(G)$
$s.t. Glink(u) = u'\ and\ OL(u') = L\ or\ R\}$

(b)Transfer:

(b).1.$c_{\Theta}(u,x)[TP] = scoreTP(c(v,y)[TP] + c(w,z)[TP])$:
$y\ incomparable\ to\ x, z \leq_S x, OL(u) = R\ , and\ \exists u' \in V(G)$
$s.t. Glink(u) = u'\ and\ OL(u') = L\ \}$

Lastly, a special case of duplication named Unifying Node Mapping, ties up two TP mappings, that are mapped to the same specie tree node as u, together.

(c)Duplication

(c).1. $c_{\Delta}(u,x)[UN] = min\{scoreUN(c(v,x)[TP] + c(w,x)[TP]): OL(u) = UN\}$

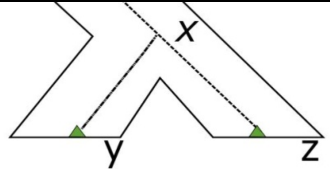
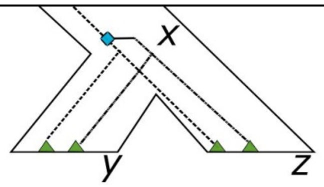
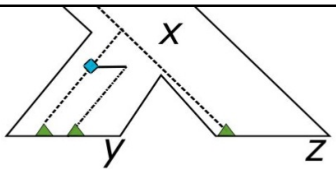
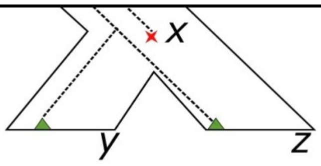
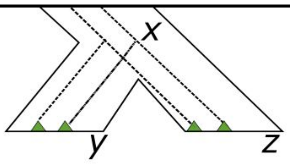
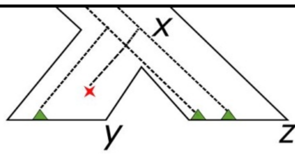
Mapping calculated	**Child v mapping**	**Child w mapping**	**Score**	**Illustration**
$\boldsymbol{c_{\Sigma}(u,x)[E]}$	$c(v,y)[E]$	$c(w,z)[E]$	0	
$\boldsymbol{c_{\Sigma}(u,x)[E]}$	$c(v,y)[EE]$	$c(w,z)[EE]$	Duplication at node	
$\boldsymbol{c_{\Sigma}(u,x)[E]}$	$c(v,y)[EE]$	$c(w,z)[E]$	Duplication after node	
$\boldsymbol{c_{\Sigma}(u,x)[EE]}$	$c(v,y)[E]$	$c(w,z)[E]$	Deletion at node	
$\boldsymbol{c_{\Sigma}(u,x)[EE]}$	$c(v,y)[EE]$	$c(w,z)[EE]$	0	
$\boldsymbol{c_{\Sigma}(u,x)[EE]}$	$c(v,y)[EE]$	$c(w,z)[E]$	Deletion after node	

***Table 1**. Example of the speciation-event scoring function. $\boldsymbol{c_{\Sigma}(u,x)[E]}$ will hold the value as if a single copy of the gene exist on the node $\boldsymbol{x \in V(S)}$, while $\boldsymbol{c_{\Sigma}(u,x)[EE]}$ will hold the value as if a TP existed on x. The calculations differ and represent a different evolutonery events based on the score used on node u's children's mappings $\boldsymbol{c(v,y)}$, and $\boldsymbol{c(w,z)}$.*

3.3 Algorithm 3 Pseudocode

```
Algorithm 3. Dynamic programming algorithm initialization.
  input: $\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{\sigma}$, and $\boldsymbol{OL}$.
  output: Initialized DP algorithm array $\boldsymbol{c}$.
1. $c \leftarrow Array\,[1..|V(G)|, 1..|V(S)|, 1..4]$ initialized to ∞.
2. for $u \in L(G)$ do
3.     $c(u, \sigma(u))[E] \leftarrow \begin{cases} 0 & \boldsymbol{if}\ OL(u) = S \\ c & otherwise \end{cases}$
4.     $c(u, \sigma(u))[EE] \leftarrow \begin{cases} c & \boldsymbol{if}\ OL(u) = S \\ 0 & otherwise \end{cases}$
5.     $c(u, \sigma(u))[TP] \leftarrow p$
6.     $c(u, \sigma(u))[UN] \leftarrow \infty$
7. end
Algorithm 3. Dynamic programming algorithm.
  input: $\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{\sigma}$, $\boldsymbol{OL}$, and $\boldsymbol{GL}$.
  Output: Minimum cost of any DTL-scenario for $\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{\sigma}$, $\boldsymbol{OL}$, and $\boldsymbol{GL}$.
1. for $u \in \dot{V}(G1)$ $in\ postorder$ do
2.   for $\boldsymbol{x \in V(S)}$ $\boldsymbol{in\ postorder}$ do
3.     if $x \in \dot{V}(S)$ then
4.         $c_1 \leftarrow \mathrm{CS(u, x)}$
5.     else
6.         $c_1 \leftarrow \infty$
7.     $c_2 \leftarrow CD(u, x)$
8.     $c_3 \leftarrow CT(u, x)$
9.     $c(u, x)[E] \leftarrow \min\,(c_1[\mathrm{E}], c_2[\mathrm{E}], c_3[\mathrm{E}])$
10.    $c(u, x)[EE] \leftarrow \min\,(c_1[\mathrm{EE}], c_2[\mathrm{E}], c_3[\mathrm{EE}])$
11.    $c(u, x)[TP] \leftarrow \min\,(c_1[\mathrm{TP}], c_3[\mathrm{TP}])$
12.    $c(u, x)[UN] \leftarrow c_2[\mathrm{UN}]$
13.  end
14. end
15. return $min_{x \in V(S)} c(root(G), x)[E, EE, UN]$
```

Algorithm 3 subroutines:

Subroutine **CS(u,x)** is implemented as follows:

```
CS(u,x)
1.CS←Array[E,EE,TP] initialized to ∞
1.Let $v,w$ be the children's of $u$
2.  Let $y,z$ be the children's of $x$
3.    currCS[E,EE]← $mapping_{y'\leq y,z'\leq z}(CS,v,y',w,z')$
4.    CS[E]← $\min(CS[E],currCS[E])$
5.    CS[EE]← $\min(CS[EE],currCS[EE])$
6.    **if** is_TP_vertex(u) **then**
7.      currCS_TP ← $mapping_{TP_{y'\leq y,z'\leq z}}(CS,v,y',w,z')$
8.      CS[TP]← $\min(CS[TP],currCS\_TP)$
9.  end
10.end
11.**return** CS
```

Subroutine **CD(u,x)** is implemented as follows:

```
CD(u,x)
1.CD←Array[E,UN] initialized to ∞
2.Let $v,w$ be the children of $u$
3.  Let $y,z$ be descendents of $x$, $y,z \leq x$.
4.    currCD[E]← $mapping(CD,v,y,w,z)$
5.    CD[E]← $\min(CD[E],currCD[E])$
6.  **end**
7.**end**
8.**if** $OL(u) = UN$ **then**
9.  CD[UN]← $mapping_{UN}(u,x)$
10.**return** CD
```

Subroutine **CT(u,x)** is implemented as follows:

```
CT(u,x)
1.CT←Array[E,EE,TP] initialized to ∞
2.Let $v,w$ be the children of $u$
3.  Let $y \leq x$, and $z \in V(S)$ is incomparable to $x$
4.    currCT[E,EE] $\leftarrow mapping(CT,v,y,w,z)$
5.    CT[E] $\leftarrow \min(CT[E], currCT[E])$
6.    CT[EE] $\leftarrow \min(CT[EE], currCT[EE])$
7.    **if** is_TP_vertex(u) **then**
8.      currCT_TP $\leftarrow mapping_{TP}(CT,v,y,w,z)$
9.      CT[TP] $\leftarrow \min(CT[TP], currCT\_TP)$
10.  end
11.end
12.**return** CT
```

Subroutine **mapping(event_type,v,y,w,z)** is implemented as follows:

```
mapping(event_type,v,y,w,z)
1.final_score←Array[E,EE] initialized to ∞
1.**for** case in cases **do**
2.  score=$c(v,y).score(case) + c(w,z).score(case)$
3.  pscore[E]=score+$punish(case,E,event\_type)$
4.  pscore[EE]=score+$punish(case,EE,event\_type)$
5.  final_score[E]=$\min(final\_score[E], pscore[E])$
6.  final_score[EE]=$\min(final\_score[EE], pscore[EE])$
7. **end**
8. return final_score
```

Subroutine **score(case)** for event $c(v,y)$ is implemented as follows:

```
score(case)
1.**if** case=E **then**
2.   score=$\min(c(v,y)[E], c(v,y)[UN])$
3.**else**
4.   score=$c(v,y)[EE]$
5.**return** score
6.end
```

Subroutine **is_TP_vertex(u)** is implemented as follows:

```
is_TP_vertex(u)
1.**if** $OL(u) = S$ or $GL(u)$=null or $OL(GL(u)) = S$ **then**
2.   **return** false
3.**if** $OL(u) = R$ and $OL(GL(u)) = L$ **then**
4.   **return** true
5.**if** $OL(u) = L$ and $OL(GL(u)) = R$ **then**
6.   **return** true
7.**return** false
8.end
```

Subroutine $mapping_{TP}$**(event_type,v,y,w,z)** is implemented as follows:

```
$mapping_{TP}$(event_type,v,y,w,z)
1.final_score← initialized to ∞
2.**for** case in cases **do**
3.   score=$c(v,y)[TP] + c(w,z).[TP]$
4.   pscore=score+$punish_{TP}(case, E, event\_type)$
5.   final_score=$\min(final\_score, pscore)$
6. **end**
7. **return** final_score
```

Subroutine $mapping_{UN}$**(u,x)** is implemented as follows:

```
mapping_UN(u,x)
1.final_score← initialized to ∞
2.Let $v,w$ be the children of u
3.  **for** case in cases **do**
4.    score=$c(v,x)[TP] + c(w,x)[TP]$
5.    pscore=score+$punish_{UN}(case, E)$
6.    final_score=$\min(final\_score, pscore)$
7. **end**
8. return final_score
```

4. Methods

The algorithms we propose in this work were implemented in Java; source code is available on demand. The NWK tree parser [15] was used to build the program. Sequences for the test example were extracted from StringDB [16] , then aligned with the MAFFT multiple sequence alignment online tool [17] using L-INS-I Iterative refinement method [18]. Phylogenetic gene trees based on the aligned sequences were created with MAFFT as well, using the UPGMA method. Specie trees were created using the NCBI taxonomy database [19] [20] .

5. Experimental results

5.1 The Data

In this section, we demonstrate the utility of our algorithms on a dataset of COG2165 sequences collected from StringDB [16]. COGS are homologous genes that have diverged in different species from a common ancestral gene [21]. Genes belonging to COG2165 orthology group are occasionally found in Tandem Duplicates as a vital part of the Type-II secretion systems found in variety gram-negative bacteria, including human pathogens [22] . We narrow down the dataset to contain sequences solely from Gammaproteobacteria species. Sequences were aligned and a phylogenetic binary gene tree was created (see Methods for more details). The gene tree was preprocessed by $Alg. 1$ and $Alg. 2$, see Figure 4. The specie tree was taken from the StringDB database and crossed with NCBI taxonomy to showcase the relationship between the different Gammaproteobacteria species, see Figure 5. Table 2 is an inforgraphic showing the origin specie for each gene, and whether the gene is part of a TP or a single occurrence. In the following sections 5.2 and 5.3 we continue to discuss the example illustrated in Figures 4 and 5.

Specie	Occurrences of TP's and single genes			
Enterobacter cloacae SCF1	E11	E12		
Erwinia amylovora	E14	E15		
Salmonella enterica LT2	E16			
Salmonella bongori	E17	E18		
Citrobacter rodentium	E25-E26	E75	E74	
Citrobacter sp. 302	E1E2	E3		
Yersinia pestis KIM10	E28-E72	E29		
Yersinia enterocolitica	E30-E31	E32-E33	E34	
Rahnella sp. Y9602	E39-E40	E41	E42	
Pseudomonas aeruginosa	E55	E57	E58-E59	E60
Vibrio fischeri	E64	E65		

Table 2 *Infographic of gene occurrences in species. The gene order is sorted by sequence start index of occurence in the specie. Two adjecent genes in a TP appear togther in the same cell, seperated by '-' for example E25-E26.*

5.2 PreProcess Gene tree

The resulting gene tree is split to two main sub-trees directly from the root. The sub-tree rooted in node 2 consists of single genes that most likely were never part of a TP and speciated separately. Node 3, marked by the $OL = UN$ label is the ancestor origin of the gene where the adjacent duplication first happened and the TP was formed, followed by independent speciation of each of the gene paralogs, forming an L- sub-tree rooted in node 11 and R sub-tree rooted in node 12.

The L sub-tree also consists of an inner sub-tree of single genes rooted in node 13, that most likely originated by deletion of the R copy or HGT event of the L copy only in one of the ancestor species.

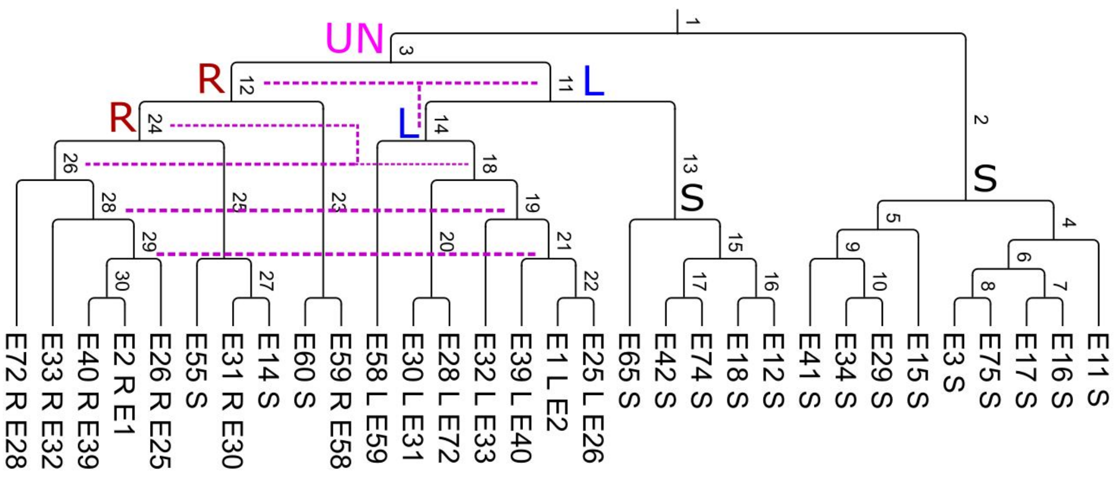


Figure 4 *The Gene tree after Alg.1 and Alg.2 pre process. Gene names consist also the OL label and the second gene in the TP (if exist). Glink's shown by dashed line and OL labels of some of the inner nodes are also shown.*

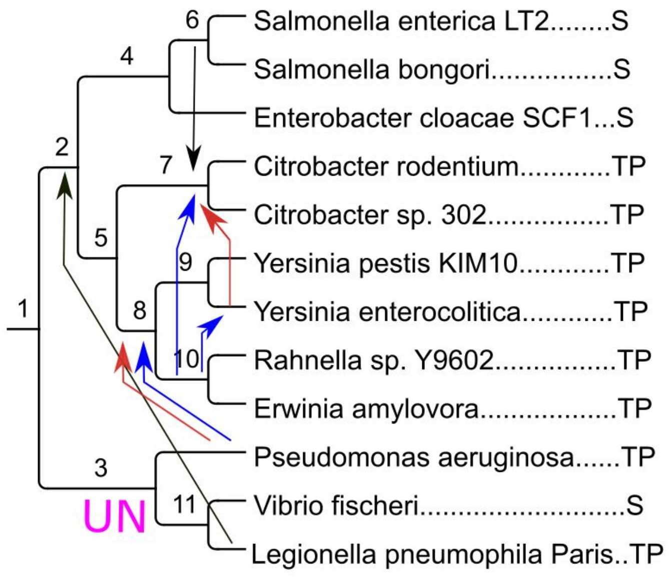

Salmonella enterica LT2........S
Salmonella bongori...............S
Enterobacter cloacae SCF1...S
Citrobacter rodentium...........TP
Citrobacter sp. 302...............TP
Yersinia pestis KIM10...........TP
Yersinia enterocolitica...........TP
Rahnella sp. Y9602................TP
Erwinia amylovora.................TP
Pseudomonas aeruginosa......TP
Vibrio fischeri..........................S
Legionella pneumophila Paris..TP
UN

Figure 5 *Specie tree reflecting the major findings of the most optimal reconciliation. Each specie label is also labelled with TP if the specie contains tandem paralogs of the gene or S otherwise, blue arrows showing HGT of the L gene, red arrows of the R gene, and black arrow for a single gene.*

5.3 Scenario described by TP mapping

In the optimal scenario found by the algorithm, the mapping of the gene tree root to the specie tree root (1, 1) hold a single gene copy and composed of speciation event mapping gene tree node children's to the children of specie tree node 1 $(2, 2), (3, 3)$).

The score selected by the algorithm for mapping $(3, 3)$ is the result of unifying node duplication, which by definition means the two children's of gene tree node 3, nodes 11 and 12, are both matched by TP mapping to the same specie node as their parent ($(3, 3) = (11, 3)[TP] + (12, 3)[TP]$), the unifying node marking the origin tandem duplication on the specie tree , created the first TP from a single gene.

The mappings leading to (3, 3) are all composed of TP-mappings speciation's and HGT transfers.

A TP-HGT event happened from the ancestor of Pseudomonas aeruginosa and below node 3 to the ancestors of above node 8 in the specie tree, The L gene copy transferred from the ancestor of node 14 while the R copy transferred from the ancestor of node 25 in the gene tree. Another TP-HGT event happened to node 7 in the specie tree, the R gene originated from the ancestors of Yersinia enter' and the L gene from node 10 in the specie tree, the corresponding locations in gene tree where this transfer originated from are node 21 for the L gene and node 29 for the R gene.

On the other hand the mappings leading to (2,2) consist of single gene evolutionary events, mainly speciation and couple of HGT's which are not shown in Figure 5.

6. Conclusions and Future Work

In this work, we have described new algorithms for understanding the creation and evolution of tandem duplicate paralogous genes through, parsimonious gene tree and species tree reconciliation.

In particular, we have proposed a new algorithm based on an existing DLT model [6] [1] tree reconciliation algorithm with the ability to differentiate between gene duplicated and adjacent gene duplicates forming tandem paralogs. The algorithm uses specialized mapping and scoring functions for genes that are part of tandem paralogs tuple.

We successfully tested the algorithm on variety of artificial examples, testing various parsimonious scenarios of both TP and single gene evolution (see examples illustrated in

Figures 1 and 2) and are currently working on the results of a real case example (see Results section).

There are numerous interesting directions for future work. The clear continuation is expanding the algorithm to handle TP consisting of more than two genes. Nevertheless, much work still needs to be done to perfect the existing algorithm, including tweaking the scoring function (which is for now mostly arbitrary) to better represent the biological cost of the various evolutionary events supported. This includes: deletion of a gene from a TP, the cost of a TP HGT event, and the forming of a TP by adjacent evolution. Another important challenge would be to support splitting of TP into single genes and the merging of those to a new TP, however this extension may be less urgent than the others, due to the fact this are rare biological events.

7.Bibliography

[1] A. M. H. a. J. L. Tofigh, "Simultaneous identification of duplications and lateral gene transfers," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 8.2,* pp. 517-535, 2011.

[2] O. M. W. B. Lynch M, "The Probability of Preservation of a Newly Arisen Gene Duplicate," *Genetics 159,* p. 1789–1804, 2001.

[3] F. A. &. K. A. S. Kondrashov, "Role of selection in fixation of gene duplications," *J. Theor. Biol 239,* p. 141–151, 2006.

[4] N. E. L. Craven S. H., "Double trouble: medical implications of genetic duplication and amplification in bacteria," *Future Microbiol. 2,* p. 309–321, 2007.

[5] A. Tofigh, "Using trees to capture reticulate evolution: lateral gene transfers and cancer progression," 2009.

[6] e. a. Bansal MS, "Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss," *Bioinformatics, vol. 28,* pp. 283-291, 2012.

[7] M. Charleston, "Jungles: a new solution to the host/parasite phylogeny reconciliation problem.," *Mathematical biosciences 149,* p. 191–223, 1998.

[8] L. A. a. A. E. J. David, "Rapid evolutionary innovation during an archaean genetic expansion," *Nature, 469(7328),* p. 93, 2011.

[9] J.-P. S. C. G. K. Y. S. G. J. R. V. a. B. V. Doyon, "An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers.," *RECOMB International Workshop on Comparative Genomics, Springer,* p. 93–108, 2010.

[10] K. a. L. V. Gorbunov, "Reconstructing genes evolution along a species tree.," *Molekuliarnaia biologiia, 43(5),* p. 946–958, 2009.

[11] R. a. C. M. A. Libeskind-Hadas, "On the computational complexity of the reticulate cophylogeny reconstruction problem.," *Journal of Computational16(1),* p. 105–117, 2009.

[12] D. a. M. M. Merkle, "Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information.," *Theory in Biosciences, 123(4),* p. 277–299, 2005.

[13] L. S. C. a. K. S. Van Iersel, " Exact reconciliation of undated trees," *arXiv preprint 1410.7004.,* 2014.

[14] N. G. M. S. P. F. M. D. W. N. a. H. M. (. Nøjgaard, "Forbidden time travel: Characterization of time-consistent tree reconciliation maps," *arXiv preprint arXiv:1705.02179,* 2017.

[15] D. Dazhi, "lightweight JavaCC Newick parser," [Online]. Available: https://bitbucket.org/djiao/a-lightweight-javacc-newick-parser/src.

[16] D. e. a. Szklarczyk, "The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible.," *Nucleic Acids Res. 45,* p. D362–D368, 2017.

[17] S. D. Katoh K, " MAFFT multiple sequence alignment software version 7: improvements in performance and usability," *Mol Biol Evol. 30,* p. 772–780, 2013.

[18] T. H. M. T. Katoh K, "MAFFT version 5: improvement in accuracy of multiple sequence alignment.," *Nucleic Acids Res , 33,* p. 511–518, 2005.

[19] "NCBI Resource CoordinatorsDatabase resources of the National Center for Biotechnology Information," *Nucleic Acids Res.41,* pp. D8-D20, 2013.

[20] K.-M. I. C. K. L. D. O. J. S. E. Benson DA, "GenBank," *Nucleic Acids Res.vol 40,* pp. D48-D53, 2012.

[21] R. K. E. a. L. D. Tatusov, "A genomic perspective on protein families.," *Science 278,* pp. 631-637, 1997.

[22] T. L. A. J. H. W. G. J. &. S. M. Johnson, " Type II secretion: from structure to function.," *FEMS Microbiol. Lett. 255,* p. 175–186, 2006.

[23] M. a. M. I. Cs˝ur¨os, " A probabilistic model for gene content evolution with duplication, loss, and horizontal transfer.," *Annual International Conference on Research in Computational Molecular Biology, Springer,* pp. 206-220, 2006.

[24] C. P. Y. a. Z. J. P. P. Chauve, "Evolution of genes neighborhood within reconciled phylogenies: an ensemble approach.," *BMC bioinformatics 16(19), S6,* 2015.

[25] L. a. C. D. Huang, "Better k-best parsing," *In Proceedings of the Ninth International Workshop on Parsing Technology, Association for Computational Linguistics.,* pp. 53-64, 2005.

[26] R. a. K. C. Patro, " Predicting protein interactions via parsimonious network history inference.," *Bioinformatics, 29(13),* p. 237–246, 2013.

תוכן העניינים:

1. מבוא..1

2. רקע על כלים ושיטות קודמות..................3

   2.1 מודל ה-DTL...........................3

      I.2.1. תרחיש DTL..........................3

      I.2.1. הפסדים..............................4

   2.2 עבודות קודמות..............................5

3. הבנת המודל..................................6

   3.1 הגדרות..................................6

      I.3.1. הגדרות מוקדמות..........................6

      I.3.1. בעיית פיוס עצים לגנים פרלוגים סמוכים......6

   3.2 הסבר אינטואיטיבי לאלגוריתם 3..................13

   3.3 פסאודו קוד אלגוריתם 3........................16

4. שיטות......................................20

5. תוצאות ניסוייות...............................20

   5.1 הנתונים....................................20

   5.2 עיבוד מוקדם של עץ הגנים......................21

   5.3 תאור תרחיש באמצעות מיפויי גנים סמוכים..........23

6. דיון ועבודה עתידית............................23

7. רשימת מקורות................................24

אלגוריתם פיוס עצים חדש המותאם לגנים פרלוגים סמוכים

חיבור זה מהווה חלק מהדרישות לקבלת התואר "מוסמך למדעי טבע" (M.Sc.)

מאת: ניר גלעד
המחלקה למדעי המחשב, פקולטה למדעי הטבע, אוניברסיטת בן גוריון, באר שבע, ישראל.
אוקטובר 2018.

תקציר

מוטיבציה: שכפול של קטעי ד.נ.א. ארוכים נוצרים בקביעות על ידי מוטציות.

אלו יכולים להתקבע באוכלוסיה על ידי סלקציה או סחף גנטי, ולהפוך לגנים פרלוגים אשר כפופים לאבולוציה בנפרד. Tree Reconciliation (פיוס עצים) הינה שיטה רווחת להסקת יחסים אבולוציוניים בין גנים השייכים לאותה משפחה. בשיטה זו משווים את עץ האבולוציה הגן לעץ האבולוציה של המינים מהם העותקים של הגן רוצפו, ומפרשים את אי ההתאמות שמתקבלות בין העצים כעדויות למאורעות אבולוציוניים כגון: התמיינות, שכפול, אובדן גנים, והעברה רוחבית. עם זאת, אלגוריתמים קיימים לפיוס עצים מתעלמים משונות ההתנהגות האבולוציונית בין שכפול גנים היוצר שני עותקים צמודים, לבין שכפול גנים העוצר שני עותקים במקומות שונים בגנום.

תוצאות: אנו מטפלים בבעיה זו על ידי נתינת אלגוריתם יעיל לחישוב פיוס עצים בתהבסס על מודל DTL בתוספת מיפויים ופונקציות ניקוד חדשות אשר מותאמים למעקב אחר יצירתם של גנים משוכפלים צמודים והשתתפותם במאורעות אבולוציוניים כמקשה אחת.

אוניברסיטת בן- גוריון בנגב
הפקולטה למדעי הטבע
המחלקה למדעי המחשב

נושא החיבור: אלגוריתם פיוס עצים חדש המותאם לגנים פרלוגים סמוכים

חיבור זה מהווה חלק מהדרישות לקבלת התואר "מוסמך למדעי טבע" (M.Sc.)

מאת: ניר גלעד
מנחה: פרופסור מיכל זיו יוקלסון

חתימת הסטודנט: [illegible] תאריך: 24.01.2019

חתימת המנחה: מיכל זיו יוקלסון תאריך: 19.02.2019

חתימת יו"ר הועדה המחלקתית: __________ תאריך__________

טבת ה'תשע"ט   דצמבר 2018

אוניברסיטת בן- גוריון בנגב
הפקולטה למדעי הטבע
המחלקה למדעי המחשב

נושא החיבור: אלגוריתם פיוס עצים חדש המותאם לגנים פרלוגים סמוכים

חיבור זה מהווה חלק מהדרישות לקבלת התואר "מוסמך למדעי טבע" (M.Sc.)

מאת: ניר גלעד
מנחה: פרופסור מיכל זיו יוקלסון

טבת ה'תשע"ט

דצמבר 2018