A Supervised Machine Learning Approach for Distinguishing Between Additive and Replacing Horizontal Gene Transfers

Abhijit Mondal
Department of Computer Science and Engineering
University of Connecticut
Storrs, CT, USA
abhijit.mondal@uconn.edu

Misagh Kordi*
Department of Computer Science and Engineering
University of Connecticut
Storrs, CT, USA
misagh.kordi@uconn.edu

Mukul S. Bansal
Department of Computer Science and Engineering
University of Connecticut
Storrs, CT, USA
mukul.bansal@uconn.edu

ABSTRACT

Horizontal gene transfer is one of the most important drivers of microbial gene and genome evolution. Despite its central role in microbial evolution, several aspects of horizontal gene transfer remain poorly understood. In particular, transfers can be either *additive* or *replacing* depending on whether the transferred gene adds itself as a new gene in the recipient genome or replaces an existing homologous gene. However, despite recent efforts, there do not yet exist effective computational approaches for classifying inferred transfers as being additive or replacing.

In this work, we address this gap by devising a novel supervised machine learning approach for classifying transfers as being either additive or replacing. Our approach is based on phylogenetic reconciliation, a standard computational technique for inferring transfers. Our classifier, named *ARTra*, uses as features the classifications provided by several simple reconciliation-based classification rules, along with topological information from the gene tree, and ensembles them to produce a more accurate classification.

ARTra is efficient and robust and significantly improves upon the classification accuracy of the only existing computational approach for this problem. We demonstrate the accuracy of ARTra by applying it to a wide range of simulated datasets and to a large biological dataset. Our results show that ARTra performs well over a broad range of evolutionary conditions and on real data, and that it does so even when trained only on a narrow range of such conditions and only using simulated data.

An open-source implementation of ARTra is freely available from https://compbio.engr.uconn.edu/software/ARTra/.

CCS CONCEPTS

• **Applied computing** → **Molecular evolution**; • **Computing methodologies** → **Machine learning algorithms**.

*Currently with the Department of Computer Science at University of California, Los Angeles.

BCB '20, September 21–24, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7964-9/20/09…$15.00
https://doi.org/10.1145/3388440.3412428

KEYWORDS

Horizontal gene transfer, additive transfer, replacing transfer, xenologous gene displacement, classification, supervised machine learning, DTRL reconciliation, microbial gene family evolution

ACM Reference Format:
Abhijit Mondal, Misagh Kordi, and Mukul S. Bansal. 2020. A Supervised Machine Learning Approach for Distinguishing Between Additive and Replacing Horizontal Gene Transfers. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '20), September 21–24, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3388440.3412428

1 INTRODUCTION

The transfer of genetic material between two organism not related by an ancestor-descendant relationship, known as *Horizontal Gene Transfer*, or just *transfer* for short, is a fundamental process in microbial evolution. Given its importance to understanding microbial evolution and adaptation, many computational methods have been developed for inferring transfer events in the evolutionary histories of organisms. These include a variety of sequence composition based approaches as well as phylogenetic approaches; see, e.g., [28] for a review. Sequence composition methods look for atypical sequence characteristics, such as dinucleotide frequencies or codon usage biases, that might indicate instances of horizontally acquired genes, but such methods are only effective at short evolutionary time scales and cannot generally be used to identify donors and recipients of transfer events [10, 28]. Phylogenetic approaches for transfer inference are based on the fact that horizontal transfers affect the phylogenies of the transferred genes in specific ways. Among the most advanced phylogenetic methods for detecting transfers and identifying corresponding donors and recipients are those based on *Duplication-Transfer-Loss (DTL)* reconciliation [1–3, 6, 8, 9, 11, 13, 16, 19–27]. DTL reconciliation uses the patterns of discordance between a given gene tree (i.e., phylogenetic tree for genes from the same gene family) and a given species tree to infer individual transfer events along with their donor and recipient species (or edges) in the species tree, and is widely used in practice.

When a gene is horizontally transferred, it can either add itself as a new gene in the recipient genome, resulting in an *additive* transfer, or replace an existing homologous gene, resulting in a *replacing* transfer [7, 14, 15]. Despite being a fundamental property of transfer events, there do not currently exist any reliable approaches for inferring if a given transfer event is additive or replacing. In fact, existing phylogenetic approaches for transfer inference implicitly assume that all transfers are either additive or that

all transfers are replacing. To address this limitation, a new phylogenetic reconciliation framework, referred to as *DTRL reconciliation*, was recently proposed to simultaneously model both additive and replacing transfers, but inferring additive and replacing transfers under this framework has been shown to be NP-hard [12, 17]. Thus, the classification of transfer events as being either additive or replacing remains a largely unsolved problem, making it difficult to understand the relative frequencies, biological implications, and evolutionary impacts of these two categories of transfer events [7].

In an important recent development, it has been demonstrated that the efficiently solvable DTL reconciliation framework, which models all transfers as additive, is equally effective at inferring both additive and replacing transfers [17]. Thus, while it *is* possible to use DTL reconciliation to reasonably infer most transfers irrespective of whether they are additive or replacing, it *is not* possible to directly figure out which of the inferred transfers are additive and which are replacing.

In this work, we address this gap by proposing a novel supervised machine learning approach for classifying transfers inferred through DTL reconciliation as being either additive or replacing. Our classifier, which we call *ARTra*, uses as features the classifications provided by several simple reconciliation-based classification rules, along with topological information from the gene tree, and ensembles them to produce a more accurate classification. We demonstrate the accuracy of ARTra by applying it to a wide range of simulated datasets as well as to a large biological dataset with of over 4500 gene trees from 100 distantly related species. Our results show that ARTra is efficient and robust and performs well over the entire range of tested conditions. The method also performs well on the real dataset, yielding classifications that are consistent with expected distribution of additive and replacing transfers on that dataset. ARTra also significantly improves upon the classification accuracy of the only other existing computational approach for this problem, proposed recently by Kordi et al. [17]. Crucially, our results show that ARTra performs well over a broad range of evolutionary conditions and tree sizes, even when it is trained only on a narrow range of such conditions and only using simulated data.

An open-source implementation of our method, *ARTra*, short for Additive and Replacing Transfers, is freely available from https://compbio.engr.uconn.edu/software/ARTra/.

2 DEFINITIONS AND PRELIMINARIES

We follow basic definitions and notation from [17]. Given a rooted tree T, we denote its node, edge, and leaf sets by $V(T)$, $E(T)$, and $Le(T)$ respectively. The root node of T is denoted by $rt(T)$, the parent of a node $v \in V(T)$ by $pa_T(v)$, its set of children by $Ch_T(v)$, and the (maximal) subtree of T rooted at v by $T(v)$. The set of *internal nodes* of T, denoted $I(T)$, is defined to be $V(T) \setminus Le(T)$. We define $\leq_T$ to be the partial order on $V(T)$ where $x \leq_T y$ if y is a node on the path between $rt(T)$ and x. The partial order $\geq_T$ is defined analogously, i.e., $x \geq_T y$ if x is a node on the path between $rt(T)$ and y. We say that y is an *ancestor* of x, or that x is a *descendant* of y, if $x \leq_T y$ (note that, under this definition, every node is a descendant as well as ancestor of itself). We say that x and y are *incomparable* if neither $x \leq_T y$ nor $y \leq_T x$. A tree is *binary* if all of its internal nodes have exactly two children. Given a non-empty subset $L \subseteq Le(T)$, we denote by $lca_T(L)$ the last common ancestor (LCA) of all the leaves in L in tree T; that is, $lca_T(L)$ is the unique smallest upper bound of L under $\leq_T$. Throughout this work, the *term* tree refers to rooted binary trees.

A *species tree* is a tree that depicts the evolutionary relationships of a set of species. Given a gene family from a set of species, a *gene tree* is a tree that depicts the evolutionary relationships among the sequences encoding only that gene family in the given set of species. Thus, the nodes in a gene tree represent genes. Throughout this work, we denote the gene tree and species tree under consideration by G and S, respectively. We assume that each leaf of the gene tree is labeled with the species from which that gene (sequence) was obtained. This labeling defines a *leaf-mapping* $\mathcal{L}_{G,S}\colon Le(G) \to Le(S)$ that maps a leaf node $g \in Le(G)$ to that unique leaf node $s \in Le(S)$ which has the same label as g. Note that gene trees may have more than one gene from the same species. The species tree contains at least all the species represented in the gene tree.

2.1 Additive and replacing transfers

Transfers can be either *additive* or *replacing* depending on whether the transferred gene adds itself as a new gene in the recipient genome or replaces an existing homologous gene. More formally:

Definition 2.1 (Additive transfer). An *additive transfer* is a horizontal gene transfer that inserts itself into the recipient genome through the addition of a new gene locus.

Definition 2.2 (Replacing transfer). A *replacing transfer* is a horizontal gene transfer that inserts itself into the recipient genome by replacing a homologous gene at an existing gene locus. Replacing transfers are sometimes also referred to as *xenologous gene displacement* [15].

Note that (i) additive transfers increase in the total number of genes in the recipient genome but replacing transfers do not, and (ii) replacing transfers can only occur if the recipient genome already contains a homologous copy of the transferred gene. Figure 1 illustrates how additive and replacing transfer events impact the resulting gene tree topology.

2.2 Background on DTL-reconciliation

Our approach relies on the use of Duplication-Transfer-Loss (DTL) Reconciliation to infer and classify transfers. A DTL reconciliation shows the evolution of a given gene tree inside the corresponding species tree through speciation, gene duplication, horizontal gene transfer, and gene loss. It does so by mapping each gene tree node to a unique species tree node and designating each internal gene tree node as representing either a speciation, duplication, or transfer event. More formally:

Definition 2.3 (DTL Reconciliation). A *Duplication-Transfer-Loss (DTL) reconciliation* for G and S is a seven-tuple $\langle \mathcal{L}, \mathcal{M}, \Sigma, \Delta, \Theta, \Xi, \tau \rangle$, where $\mathcal{L}\colon Le(G) \to Le(S)$ represents the leaf-mapping from G to S, $\mathcal{M}\colon V(G) \to V(S)$ maps each node of G to a node of S, the sets Σ, Δ, and Θ partition $I(G)$ into speciation, duplication, and transfer nodes respectively, Ξ is a subset of gene tree edges that represent transfer edges, and $\tau\colon \Theta \to V(S)$ specifies the recipient species for each transfer event, subject to the following constraints:

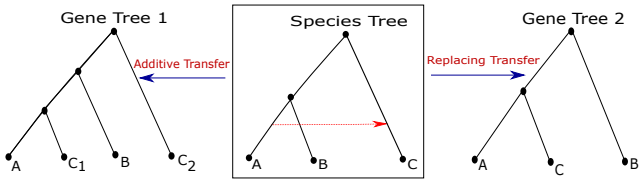

Figure 1: Additive and Replacing Transfers. The figure shows the evolution of two gene trees inside the same species tree. In both cases, there is a horizontal gene transfer (shown by the red arrow on the species tree) from the parent edge of species A to the parent edge of the species C. If this transfer was an additive transfer then the resulting gene tree would look like the tree shown on the left (Gene Tree 1). If this transfer was a replacing transfer then the resulting gene tree would look like the tree shown on the right (Gene Tree 2).

(1) If $g \in Le(G)$, then $\mathcal{M}(g) = \mathcal{L}(g)$.
(2) If $g \in I(G)$ and g' and g'' denote the children of g, then,
   (a) $\mathcal{M}(g) \not<_S \mathcal{M}(g')$ and $\mathcal{M}(g) \not<_S \mathcal{M}(g'')$,
   (b) At least one of $\mathcal{M}(g')$ and $\mathcal{M}(g'')$ is a descendant of $\mathcal{M}(g)$.
(3) Given any edge $(g, g') \in E(G)$, $(g, g') \in \Xi$ if and only if $\mathcal{M}(g)$ and $\mathcal{M}(g')$ are incomparable.
(4) If $g \in I(G)$ and g' and g'' denote the children of g, then,
   (a) $g \in \Sigma$ only if $\mathcal{M}(g) = lca(\mathcal{M}(g'), \mathcal{M}(g''))$ and $\mathcal{M}(g')$ and $\mathcal{M}(g'')$ are incomparable,
   (b) $g \in \Delta$ only if $\mathcal{M}(g) \geq_S lca(\mathcal{M}(g'), \mathcal{M}(g''))$,
   (c) $g \in \Theta$ if and only if either $(g, g') \in \Xi$ or $(g, g'') \in \Xi$.
   (d) If $g \in \Theta$ and $(g, g') \in \Xi$, then $\mathcal{M}(g)$ and $\tau(g)$ must be incomparable, and $\mathcal{M}(g')$ must be a descendant of $\tau(g)$, i.e., $\mathcal{M}(g') \leq_S \tau(g)$.

An illustration of DTL reconciliation appears in Figure 2.

Given a DTL reconciliation α, one can directly count the minimum number of gene losses, $Loss_\alpha$, in the corresponding reconciliation [1]. Let P_Δ, P_Θ, and P_{loss} denote the non-negative costs associated with duplication, transfer, and loss events, respectively. The reconciliation cost of a DTL reconciliation is defined as follows.

Definition 2.4 (Reconciliation cost). Given a DTL reconciliation $\alpha = \langle \mathcal{L}, \mathcal{M}, \Sigma, \Delta, \Theta, \Xi, \tau \rangle$ for G and S, the *reconciliation cost* associated with α is given by $P_\Delta \cdot |\Delta| + P_\Theta \cdot |\Theta| + P_{loss} \cdot Loss_\alpha$.

A most parsimonious, or optimal, DTL reconciliation is one that has minimum reconciliation cost.

Definition 2.5 (Most Parsimonious DTL reconciliation). Given G and S, along with P_Δ, P_Θ, and P_{loss}, a *most parsimonious DTL reconciliation* for G and S is a DTL reconciliation with minimum reconciliation cost.

2.3 Transfer classification

Observe that DTL reconciliation models all transfer events as additive transfers. Nonetheless, most parsimonious DTL reconciliations

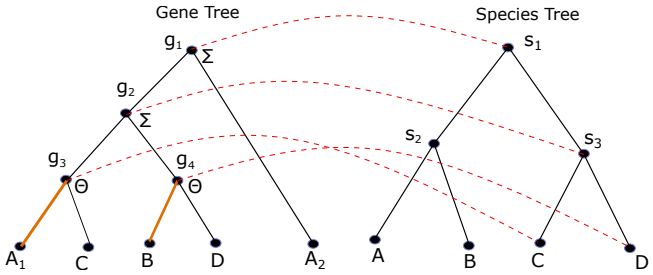

Figure 2: A DTL reconciliation. Each internal node of the gene tree is mapped to a node in the species tree (red arcs) and is labeled with an event type (Δ for duplication, Θ for transfer, and Σ for speciation). Bold edges (colored orange) on the gene tree represent transfer edges. This reconciliation invokes two transfer events, with the transfer event at node g_3 representing a transfer from the parent edge of species C to the parent edge of species A in the species tree, and the transfer event at node g_4 representing a transfer from the parent edge of species D to the parent edge of species B in the species tree. The depicted reconciliation also invokes a loss event along the edge (s_2, B) in the species tree.

has been shown to be almost equally effective at inferring both additive and replacing transfers [17]. This motivates the following problem formulation:

Problem 1 (Reconciliation Transfer Classification). *Let $\alpha = \langle \mathcal{L}, \mathcal{M}, \Sigma, \Delta, \Theta, \Xi, \tau \rangle$ be a most parsimonious DTL reconciliation for G and S. Given G, S, and α, the* Reconciliation Transfer Classification (RTC) *problem is to partition Θ into Θ_A and Θ_R, where Θ_A represents the set of additive transfers and Θ_R represents the set of replacing transfers, such that the number of transfers from Θ that are classified correctly is maximised.*

Note that, due to reconciliation uncertainty and error, the set Θ typically contains both false-negatives and false-positives. In other words, (i) Θ may not represent all true transfer events, and (ii) several of the transfers represented in Θ may not be true transfers. Thus, the RTC problem seeks only to correctly classify the subset of true transfers represented in Θ.

2.4 An existing heuristic for the RTC problem

A simple heuristic was recently proposed for the RTC problem [17]. We refer to this heuristic as the *gene-frequency* heuristic. Briefly, this heuristic works by initially classifying all inferred transfer events as additive and then greedily attempting to reclassify some of these transfer events as replacing. To determine if a transfer can be replacing, the heuristic uses a simple decision rule based on comparison of actual frequencies of genes (from the gene family corresponding to G) at each leaf of S with frequencies of genes implied by the computed reconciliation. We refer the reader to [17] for a more detailed description of this heuristic.

This *gene-frequency* heuristic has been shown to work well when rates of duplication, transfer, and loss events are low, but performance decreases drastically as evolutionary event rates increase, with classification accuracy for additive transfers falling to only

52% at higher rates of evolutionary events [17]. The heuristic is also known to substantially biased in favor of replacing transfers, which is not surprising since the heuristic attempts to greedily label as many of the transfers as replacing as possible. Despite these weaknesses, we use the classification generated by the *gene-frequency* heuristic as one of the features in our proposed machine learning approach.

3 RULE-BASED HEURISTICS FOR TRANSFER CLASSIFICATION

Our machine learning based classifier, ARTra, uses as features the classifications obtained from several simple rule-based heuristics for the RTC problem. In addition to the existing *gene-frequency* heuristic described in the previous section, we developed two novel rule-based heuristics for the RTC problem for use with ARTra. We refer to these as *lost-gene* heuristic and *mapping-count* heuristic.

3.1 *Lost-gene* heuristic

Observe that, to DTL reconciliation, a replacing transfer looks like an additive transfer with a concurrent loss at the recipient edge. In other words, if DTL reconciliation correctly infers the donor and recipient edges for a replacing transfer, then the corresponding reconciliation must invoke a loss event at the recipient edge in the species tree. We denote the number of losses implied by the reconciliation α along a specific edge $(pa(s), s) \in E(S)$ by $Loss_\alpha(s)$. Note that, given α and any $(pa(s), s) \in E(S)$, $loss_\alpha(s)$ can be easily computed based on the minimum number of losses implied by reconciliation α [1].

The *lost-gene* heuristic is based on the simple observation above and makes use of the computed loss values at the edges of S. Specifically, given G, S, and α, the *lost-gene* heuristic works as follows: It considers each transfer event in the given DTL reconciliation (in any arbitrary order) and checks if the given reconciliation invokes at least one gene loss at the recipient edge for that transfer. If it does, then the transfer event under consideration is labeled as a replacing transfer. If the recipient edge did not have any gene losses, then the transfer is labeled as an additive transfer. In practice, we found that it helps to consider losses not only at the specific recipient edge, but also up to two edges above the parent edge. We therefore parameterize the *lost-gene* heuristic by the number, h, of edges (recipient edge and any ancestral edges) considered for each transfer event. A formal description of the *lost-gene* heuristic appears below.

Algorithm *Lost-Gene-Heuristic*(G, S, α, h)

for each $s \in V(S) \setminus \{rt(S)\}$ **do**
  Compute $Loss_\alpha(s)$
Initialize Θ_A and Θ_R to be empty sets
for each $g \in \Theta$ **do**
  Let $r = \tau(g)$ and initialize $LossSum = 0$ and $counter = h$
  while $counter > 0$ and $r \neq rt(S)$ **do**
    $LossSum = LossSum + Loss_\alpha(r)$
    $r = pa(r)$
    $counter = counter - 1$
  if $LossSum > 0$ **then**
    $\Theta_R = \Theta_R \cup g$
  else
    $\Theta_A = \Theta_A \cup g$
return Θ_A and Θ_R

In our experiments we observed that this heuristic performed best with $h = 2$. The heuristic also performed reasonably well with $h = 1$ and $h = 3$, and we use all three versions of this heuristic (i.e., with $h \in \{1, 2, 3\}$) to generate classifications for ARTra. An illustration of this heuristic with $h = 1$ appears in Figure 3.

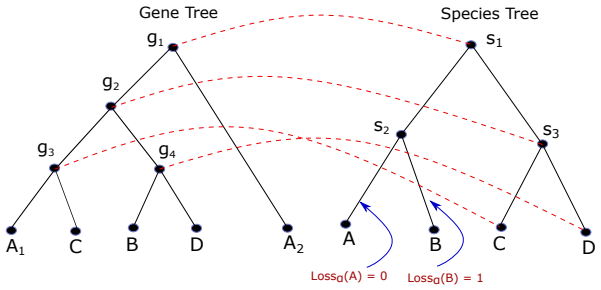

Figure 3: *Lost-gene* Heuristic. This figure shows a DTL reconciliation, α, of the gene tree on the left with the species tree on the right. This reconciliation invokes two transfer events, one at node g_3 and another at node g_4. If we apply the *lost-gene* heuristic with $h = 1$ to this scenario, g_3 will be classified as an additive transfer and g_4 as a replacing transfer. The heuristic classifies g_3 as additive because the corresponding recipient edge, (S_2, A), does not have any loss events associated with it according to α, i.e., $Loss_\alpha(A) = 0$. The heuristic classifies g_4 as replacing because the corresponding recipient edge, (S_2, B), has an associated loss event with $Loss_\alpha(B) = 1$.

3.2 *Mapping-Count* Heuristic

The *mapping-count* heuristic is based on a different observation regarding the effect of additive and replacing transfers on gene tree topology. Suppose a transfer is additive and $(pa(r), r)$ denotes its recipient edge, then, we expect to see at least two copies of the species tree subtree $S(r)$ in the gene tree. Note that these copies need not be identical to $S(r)$, since subsequent duplication, transfers and losses can change their topologies; nonetheless there should be more than one subtree in the gene tree evolving inside $S(r)$. On the other hand, if that transfer was replacing, then we would expect to see only one copy of the subtree $S(r)$ in the gene tree (assuming other events such as duplications or other transfers have not created additional copies of $S(r)$).

The *mapping-count* heuristic utilises this insight and builds upon it. Specifically, given G, S, and α, this heuristic works as follows: It considers each transfer event $g \in \Theta$ (in any arbitrary order), identifies its recipient edge $(pa(r), r)$ (i.e., $r = \tau(g)$), in the species tree, and checks if there are other nodes of the gene tree that map to node r in the species tree. In checking the mappings of the other nodes of the gene tree, we exclude all those gene tree nodes that are comparable to the corresponding transfer edge, i.e., we do not consider mappings from nodes that are ancestors or descendants of the corresponding transfer edge in the gene tree. If there is at least one such node in the gene tree that also maps to r, then the transfer event g is classified as an additive transfer. Otherwise, g is classified

as a replacing transfer. In practice, we found it highly beneficial to consider not only the specific gene tree node mappings assigned by the reconciliation α, but also to check if a gene tree node *could* potentially map to r without increasing its local reconciliation cost. This is described more precisely (see Step 9) in the formal description of the *mapping-count* heuristic below.

Algorithm *Mapping-Count-Heuristic*(G, S, α)

1: Initialize Θ_A and Θ_R to be empty sets
2: **for** each $g \in \Theta$ **do**
3: Let $r = \tau(g)$ and $\{g', g''\} = Ch(g)$, and initialize *transferChild* = *NULL* and *MappingSum* = 0
4: **if** $(g, g') \in \Xi$ **then**
5: *transferChild* = g'
6: **else**
7: *transferChild* = g''
8: **for** each $v \in V(G)$ that is incomparable to *transferChild* **do**
9: **if** $\mathcal{M}(v) = r$ **or** $cost(v, \mathcal{M}(v)) = cost(v, r)$ **then**
10: *MappingSum* = *MappingSum* + 1
11: **if** *MappingSum* = 0 **then**
12: $\Theta_R = \Theta_R \cup g$
13: **else**
14: $\Theta_A = \Theta_A \cup g$
15: **return** Θ_A and Θ_R

In the pseudocode above, $cost(v, r)$ is defined as follows: Given any $v \in V(G)$ and $r \in V(S)$, $cost(v, r)$ is the cost of an optimal reconciliation of $G(v)$ with S such that v maps to r. These $cost(\cdot, \cdot)$ values serve as sub-problems in the dynamic programming algorithm for computing most parsimonious DTL reconciliations and can be efficiently computed [1]. An illustration of the *mapping-count* heuristic using the same example as in Figure 3 appears in Figure 4.

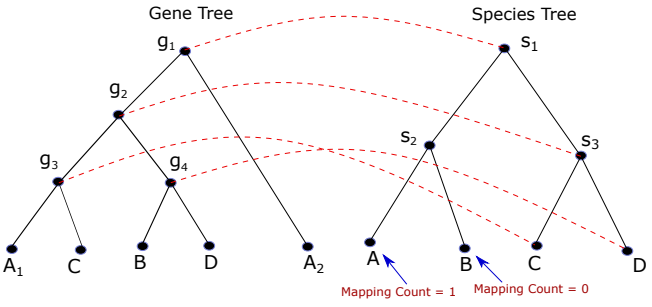

Figure 4: *Mapping-count* heuristic. This figure shows a DTL reconciliation, α, that invokes two transfer events, one at node g_3 and another at node g_4. If we apply the *mapping-count* heuristic to this scenario, g_3 will be classified as an additive transfer and g_4 as a replacing transfer. The heuristic classifies g_3 as additive because the corresponding recipient edge is (S_2, A) and there is at least one other node in the gene tree, A_2, that is incomparable to A_1 and also maps to species node A. The heuristic classifies g_4 as replacing because the corresponding recipient edge is (S_2, B), but there is no other node in the gene tree, incomparable to B, that either already maps to species node B or *can* map to species node B with the same local reconciliation cost.

4 SUPERVISED MACHINE LEARNING MODEL

We observed in our experimental results, shown in the next section, that the *lost-gene* heuristic provided significantly more accurate classifications than the existing *gene-frequency* heuristic. We also observed that the *gene-frequency* heuristic was biased towards inferring replacing transfer events and that the *lost-gene* heuristic was biased towards inferring additive transfer events. We therefore developed a supervised machine learning approach, implemented in ARTra, to ensemble these biased heuristics into a single, less biased and more accurate, classification framework for transfers. This binary classification problem lends itself to a supervised machine learning framework since a large amount of labeled training data can be generated using recently developed simulation frameworks for gene family evolution that can simulate both additive and replacing transfers [18]. We describe key aspects of the machine learning model implemented in ARTra below.

4.1 Feature Selection and Normalization

We used 6 features in the machine learning model. Five of these features corresponded to the classifications provided by the rule-based heuristics and variants described in previous sections; specifically, classifications provided by the *gene-frequency*, *lost-gene* with $h = 1$, *lost-gene* with $h = 2$, *lost-gene* with $h = 3$, and *mapping-count* heuristics. The sixth feature is a structural feature corresponding to the *height* of each transfer node (from Θ) in the gene tree, where the height of a node is defined to be the number of nodes on the path from that node to its furthest leaf descendant. We initially included several other structural features, such as transfer node depth, reconciliation cost, number of transfers, duplications, and losses, and subtree sizes of transferred nodes, but found that including these additional features either did not improve classification accuracy or made it worse.

To normalize the heights of transfer nodes across different gene trees, we performed a preprocessing step where we used the standard *min-max* criteria to normalize node heights. Specifically, if $height_{min}$ and $height_{max}$ denote the minimum and maximum heights observed in the input, then each height was normalized by first subtracting out $height_{min}$ and then dividing the result by $(height_{max} - height_{min})$.

We also calculated feature importance under our final machine learning model (using the *scikit-learn* library [5]) and results are shown in Figure 5. As the figure shows, the three most important features, by far, are the *lost-gene* heuristic with $h = 2$, the *gene-frequency* heuristic, and the *mapping-count* heuristic.

4.2 Model Selection

We considered several popular machine learning algorithms like Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Random Forest (RF), K-Nearest Neighbor (KNN), and Decision Trees (DT) for our classfication problem. In initial comparisons using preliminary simulated data, we found that SVM, MLP and RF performed best, with RF showing slightly better accuracy than the other two. Furthermore, RF had far better running time than SVM and MLP. Based on this initial analysis, we chose RF as our classifier for this problem.

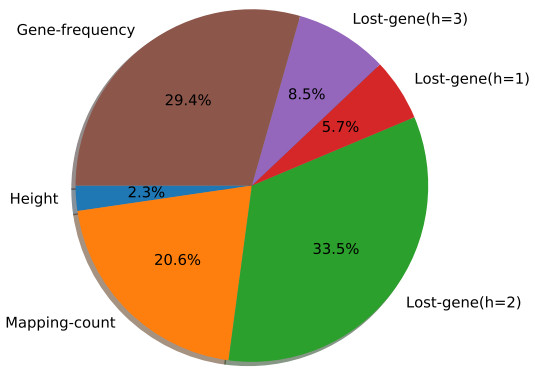

Figure 5: Feature Importance. This pie chart shows the relative importance of the six features used in the final machine learning model implemented in ARTra.

We used Python's scikit-learn library [5] to implement our random forest model. RF is a commonly used ensemble learning technique for classification and is based on constructing several decision tree during training and then typically using a majority vote from these trees to classify new samples. Each of the decision trees is trained/built on a randomly selected subset of features. We performed fine tuning of three RF-related parameters: number of decision trees, number of features to be randomly selected for building each decision tree, and the maximum depth of each decision tree. We fixed the number of decision trees in the model at 10 based on initial experiments studying number of decision trees versus classification error. We followed the standard practice of selecting $\sqrt{p}$ features randomly for each decision tree, where p is the total number of features used. Finally, we did not cap decision tree depth, allowing each tree to grow fully.

Model evaluation was performed using 10-fold cross validation.

5 EXPERIMENTAL ANALYSIS

We trained ARTra on a collection of simulated data and evaluated its classification accuracy using 10-fold cross validation. We also compared the accuracy of our trained classifier against the rule-based heuristics (*gene-frequency*, *lost-gene*, and *mapping-count* heuristics), on additional simulated datasets with different evolutionary characteristics as well as on a large real dataset. We first describe the datasets used in the analysis and then the results of our analysis.

5.1 Datasets

Baseline simulated datasets used for training and model evaluation. For training the supervised learning model and evaluating performance using 10-fold cross validation, we created 10 sets of simulated gene trees and species trees using the simulation tool SaGePhy [18]. SaGePhy uses a probabilistic birth-death process to simulate species trees along with gene trees that stochastically evolve inside those species trees through speciations, duplications, additive transfers, replacing transfers, and losses. Each of the 10 sets of trees was created as follows: We first simulated 100 birth-death species trees, each with 100 taxa (leaves), and a height of 1. For each of these 100 species trees, we then simulated 9 gene trees capturing different evolutionary conditions. Specifically, the 9 gene trees correspond to (i) a low, medium, or high rate of duplication, transfer, and loss, referred to as *low DTL*, medium DTL, and high DTL gene trees, and (ii) a transfer content type of mixed, additive, or replacing, with mixed gene trees containing a roughly equal number of additive and replacing transfers, additive-only gene trees containing only additive and no replacing transfers, and replacing-only gene trees containing only replacing and no additive transfers. Thus, for each species tree, the 9 associated gene trees correspond to Mixed-Low-DTL, Mixed-Medium-DTL, Mixed-High-DTL, Additive-Low-DTL, Additive-Medium-DTL, Additive-High-DTL, Replacing-Low-DTL, Replacing-Medium-DTL, and Replacing-High-DTL.

To generate the low DTL gene trees, we used duplication and (additive plus replacing) transfer rates of 0.133 and 0.266, respectively; for the medium DTL gene trees we used rates of 0.3 and 0.6, respectively; and for the high DTL gene trees we used rates of 0.6 and 1.2, respectively. Thus, the total transfer rate was twice the duplication rate, (split different between additive and replacing depending on whether the gene tree is mixed, additive, or replacing). In each case, the loss rate was assigned to be equal to the sum of the duplication and additive transfer rates; so, for Mixed-Low-DTL the loss rate was 0.266, while for Additive-Low-DTL the loss rate was 0.4. These duplication, transfer, and loss rates are based on rates observed in real data and capture datasets with both lower and higher rates of these events [4].

Thus, overall, each of the 10 sets of simulated gene trees and species trees consists of exactly 900 gene tree/species tree pairs.

Additional simulated datasets for evaluation. To assess how the classifier trained on the simulated datasets described above would perform on datasets with different evolutionary characteristics, we simulated 6 other sets of gene tree sand species trees. Each of these sets consisted of 100 species trees and 900 gene trees generated using the same overall process as described above, except for the following changes: (i) double duplication rates, (ii) zero duplication rates, (iii) double transfer rates, (iv) smaller species tree sizes of 25 taxa, (v) smaller species tree sizes of 50 taxa, and (vi) larger species tree sizes of 200 taxa. Each of these 6 additional sets of gene trees and species trees corresponded to one of these variations.

Real biological dataset used for evaluation. We also used a large biological dataset to evaluate ARTra and compare against the rule-based heuristics. This dataset is composed of 4547 gene trees from 100 species, predominantly prokaryotic, broadly sampled from across the tree of life [8]. A key feature of this dataset is that the species represented in it are all evolutionarily distant to each other, making it reasonable to assume that most transfers on this dataset should be additive.

5.2 Results

Accuracy of ARTra. We used 10-fold cross validation to evaluate the performance of ARTra using the baseline simulated datasets described above. Recall that the goal of the RTC problem is to correctly

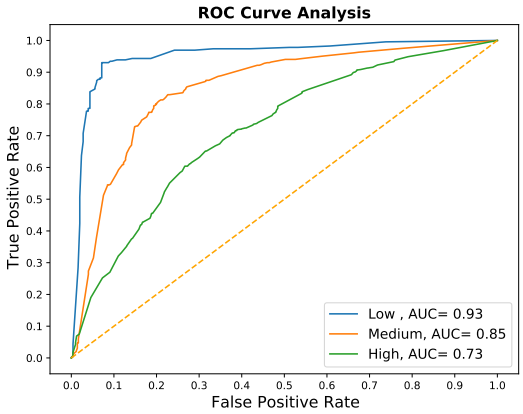

Figure 6: ROC curves for different rates of evolutionary events on the mixed baseline datasets.

classify the subset of true transfers inferred through DTL reconciliation. Accordingly, we calculated the subset of transfer nodes (i.e., Θ) inferred through DTL reconciliation that corresponded to true transfer nodes (known through the simulation framework). We then separately calculated the fractions of additive transfers and replacing transfers in this true transfer set that were classified correctly by ARTra. Table 1 shows the results of this analysis, 10-fold cross validated, on all 9 categories of the baseline simulated datasets. We find that ARTra shows excellent classification accuracy when rates of evolutionary events (duplications, transfers, and losses) are low to medium, and that, as expected, classification accuracy decreases as the rates of these events increase. Specifically, on the mixed baseline datasets, we observe roughly a 90% classification accuracy for both additive and replacing transfer inference on the Low-DTL datasets, a roughly 80% classification accuracy for both additive and replacing transfer inference on the Medium-DTL datasets, and additive transfer classification accuracy of 73.8% and replacing transfer classification accuracy of 64% on the High-DTL datasets. Results follow the same trend for the additive-only and replacing-only baseline datasets as well, with classification accuracies approaching or exceeding 90% for Low-DTL datasets, 80% for Medium-DTL datasets, and 70% for High-DTL datasets (Table 1).

We also plotted a receiver operating characteristics (ROC) curve for ARTra. This analysis measures the robustness of a binary classifier as its discrimination threshold is varied. We used 10-fold cross validated results on the mixed baseline datasets to plot the ROC curve, shown in Figure 6. As the figure shows, we obtain AUCs of 0.93 for the Low-DTL dataset, 0.85 for the Medium-DTL dataset, and 0.73 for the High-DTL dataset.

Accuracy of rule-based heuristics. The classification accuracies of the *lost-gene* heuristic with $h = 2$, *gene-frequency* heuristic, and *mapping-count* heuristic on the baseline simulated datasets are shown in Table 1. Note that results are not shown for the two other variants of the *lost-gene* heuristic because we found their classification accuracies to be clearly worse than that of *lost-gene* heuristic with $h = 2$. The results in Table 1 reveal many interesting insights. We find that (i) the *mapping-count* shows overall worse classification accuracy than the other two rule-based heuristics, (ii) the *gene-frequency* heuristic is biased towards replacing transfers, showing the best classification accuracy among all methods on the replacing-only datasets but the worst accuracy among all methods on the additive-only datasets, and (iii) the *lost-gene* heuristic with $h = 2$ is biased towards inferring additive transfers, showing the best classification accuracy among all methods on the additive-only datasets but worst overall accuracy on the replacing-only datasets. These results also show that, overall, the *lost-gene* heuristic with $h = 2$ clearly outperforms the *gene-frequency* heuristic, with lower accuracy in classifying replacing transfers but much higher accuracy in classifying additive transfers, and achieves a better balance in the accurately classifying both additive and replacing transfers. This is also easy to see in Table 2, which shows that the overall classification accuracy of *lost-gene* heuristic with $h = 2$ is significantly higher than for the *gene-frequency* heuristic for each of the mixed baseline datasets.

Comparison of ARTra against rule-based heuristics. As Tables 1 and 2 show, on the mixed baseline datasets, the classification accuracy of ARTra slightly exceeds the classification accuracy of the *lost-gene* heuristic with $h = 2$ and significantly exceeds the classification accuracies of the other rule-based heuristics. The key advantage of using ARTra over the *lost-gene* heuristic with $h = 2$ becomes apparent when one also considers the additive-only and replacing-only datasets. As Table 1 shows, ARTra is slightly worse than the *lost-gene* heuristic in additive-only datasets but much better than it in the replacing-only datasets. Likewise, ARTra is slightly worse than the *gene-frequency* heuristic in replacing-only datasets but much better than it in the additive-only datasets. As a result, the average classification accuracy of ARTra across the additive-only and replacing-only datasets is significantly higher than the average accuracies for the *lost-gene* heuristic with $h = 2$ and the *gene-frequency* heuristic. For instance, for high-DTL, the average classification accuracies for ARTra, *lost-gene* heuristic with $h = 2$, and *gene-frequency* heuristic are 71.9%, 69.5, and 67.8, respectively. In other words, ARTra is more accurate and more robust than any of the rule-based heuristics.

Classification accuracy on additional simulated datasets. We trained ARTra using only the baseline simulated datasets and then applied the trained ARTra classifier to the additional simulated datasets, described previously, representing different evolutionary conditions. We found that ARTra continued to outperform the other rule-based heuristics in all the additional datasets, showing similar trends as observed in Table 1. We also observed that overall classification accuracies varied across the different datasets, for example, classification was more accurate on the datasets with smaller species trees, as well as on the dataset with zero duplication rate. However, all methods appeared to be equally affected by such variation, with accuracies improving for all methods or worsening for all methods across the different datasets. Detailed results are shown in Tables 3 through 8. This analysis shows that ARTra is robust to evolutionary conditions and can be expected to perform well over a broad range of evolutionary conditions even when it is trained only on a narrow range of such conditions.

Method	Accuracy	Mixed Dataset			Additive-Only Dataset			Replacing-Only Dataset		
		Low	Medium	High	Low	Medium	High	Low	Medium	High
ARTra	Additive Accuracy	89.59	80.81	73.82	88.42	79.5	73.53	-	-	-
	Replacing Accuracy	91.51	80.97	64.02	-	-	-	95.23	87.78	70.32
Lost-gene (h=2) Heuristic	Additive Accuracy	89.76	81.95	74.82	90.64	81.49	74.60	-	-	-
	Replacing Accuracy	90.50	79.55	61.84	-	-	-	92.31	82.80	64.69
Gene-frequency Heuristic	Additive Accuracy	81.59	66.76	57.63	79.55	63.23	56.08	-	-	-
	Replacing Accuracy	91.69	84.51	72.09	-	-	-	95.67	91.5	79.58
Mapping-count Heuristic	Additive Accuracy	82.54	69.38	61.01	78.17	66.61	61.05	-	-	-
	Replacing Accuracy	83.15	71.60	58.20	-	-	-	92.18	83.11	70.72

Table 1: Classification accuracies of ARTra and the rule-based heuristics on baseline simulated datasets. This table shows the results of applying the different classification methods to the nine categories of baseline simulated datasets, averaged over the 100 gene tree/species tree pairs in each category of dataset. Mixed datasets refer to datasets with both additive and replacing transfers, Additive datasets refer to datasets with only additive transfers, and Replacing datasets refer to datasets with only replacing transfers. Low, Medium, and High, refer to rates of evolutionary events; specifically, they refer to Low-DTL, medium-DTL, and High-DTL datasets, respectively. For each method and each category of dataset we measure (i) the percentage of true additive transfers that are classified as additive transfers ("Additive Accuracy"), and (ii) the percentage of true replacing transfers that are classified as replacing transfers ("Replacing Accuracy"). Results for ARTra are 10-fold cross validated, and results for all other rule based heuristics are averaged over the 10 corresponding testing sets.

Method	Dataset Type		
	Low	Medium	High
ARTra	90.50	80.90	69.05
Lost-gene (h=2) Heuristic	90.18	80.74	68.34
Gene-frequency Heuristic	86.63	75.82	64.74
Mapping-count Heuristic	83.08	70.68	59.80

Table 2: Overall classification accuracy of methods on the mixed baseline simulated datasets. The table reports the percentage of true inferred transfers that are classified correctly (either as additive or replacing).

Results on real datasets. We applied the classification methods to the 100-species real dataset described previously. Since true labels are not known for this real dataset, we calculated only the percentages of additive and replacing transfers inferred by each method. Since the species in this dataset were broadly sampled from across the tree of life, there is considerable variation in the genomic content of these species. As a result, most transfers on this dataset are expected to be additive. Table 9 shows the results of this analysis. As the table shows, all methods except for the *mapping-count* heuristic classify a majority of inferred transfers as additive. While the *gene-frequency* heuristic and *lost-gene* heuristic with $h = 2$ each classify about 76% of the transfers as additive, ARTra classifies an even greater fraction, almost 79%, of the transfers as additive. While these results do not prove that ARTra yielded a more accurate classification, they do suggest that ARTra can be expected to work well on real data.

Running time and scalability. To calculate the running time and scalability of the different methods, we generated several large gene tree/species tree pairs and applied the methods to these trees. Specifically, we generated 100 simulated gene tree/species tree pairs where the species tree size (i.e., number of leaves) was 200, 100 pairs with species tree size 400, and 100 pairs with species tree size 800. In each case, gene tree sizes were roughly the same as corresponding species tree sizes. We found that all methods were equally efficient and scalable and that running times for ARTra were larger than those for the rule-based heuristics by only a small additive constant. For instance, on the size 400 datasets, we found that all rule-based heuristics required an average of 1 second, while ARTra required 2 seconds. On the size 800 datasets, these times increased to 12 seconds and 13 seconds, respectively. We also note that training the machine learning model implemented in ARTra requires only a few seconds using any of our baseline simulated datasets. These experiments show that the improved classification accuracy provided by ARTra does not come at the expense of longer run times or reduced scalability. All timed runs were executed using a single core on a commodity laptop computer with an Intel Core-i7 processor and 12 GB of RAM.

6 CONCLUSION

In this work, we introduced a novel supervised machine learning approach, ARTra, for classifying transfers inferred through DTL reconciliation as being either additive or replacing. ARTra uses as features the classifications provided by several simple reconciliation-based classification rules, along with topological information from the gene tree, and ensembles them to produce a more accurate classification. Our results show that ARTra is efficient and robust and performs well over a wide range of simulated evolutionary conditions as well as on real data. Remarkably, ARTra works well over a broad range of evolutionary conditions and tree sizes, even when it is trained only on a narrow range of such conditions and only using simulated data. This work makes it possible to infer additive and replacing transfers more reliably, and will enable a deeper and more fine-grained analysis of horizontal gene transfer in microbes. This work also demonstrates that, for some types of problems in phylogenetics and comparative genomics, it may be feasible to use simulated training data to train machine learning models that then perform well on real data.

Method	Accuracy	Mixed Dataset			Additive-Only Dataset			Replacing-Only Dataset		
		Low	Medium	High	Low	Medium	High	Low	Medium	High
ARTra	Additive Accuracy	84.94	75.59	68.03	81.59	76.95	66.77	-	-	-
	Replacing Accuracy	86.64	72.97	56.61	-	-	-	89.53	80.71	60.48
Lost-gene (h=2) Heuristic	Additive Accuracy	87.45	77.34	67.71	84.45	76.32	69.75	-	-	-
	Replacing Accuracy	83.62	71.60	56.61	-	-	-	87.31	74.35	57.74
Gene-frequency Heuristic	Additive Accuracy	77.41	60.35	49.95	67.68	61.73	50.25	-	-	-
	Replacing Accuracy	86.21	81.46	67.23	-	-	-	92.20	84.81	68.84
Mapping-Count Heuristic	Additive Accuracy	76.57	69.28	65.44	75.66	69.34	41.01	-	-	-
	Replacing Accuracy	74.56	56.61	60.58	-	-	-	83.29	61.31	45.46

Table 3: Classification accuracies for the different methods on datasets with double duplication rate.

Method	Accuracy	Mixed Dataset			Additive-Only Dataset			Replacing-Only Dataset		
		Low	Medium	High	Low	Medium	High	Low	Medium	High
ARTra	Additive Accuracy	93.36	85.34	78.76	92.61	85.28	77.21	-	-	-
	Replacing Accuracy	93.85	90.50	72.68	-	-	-	97.47	95.57	88.85
Lost-gene (h=2) Heuristic	Additive Accuracy	96.09	84.52	75.77	93.61	86.38	75.85	-	-	-
	Replacing Accuracy	95.08	87.92	70.65	-	-	-	95.58	90.35	81.32
Gene-frequency Heuristic	Additive Accuracy	91.01	76.17	61.50	84.63	74.72	61.86	-	-	-
	Replacing Accuracy	93.85	90.49	78.78	-	-	-	97.68	97.44	95.76
Mapping-Count Heuristic	Additive Accuracy	89.84	80.44	73.45	85.83	79.07	70.48	-	-	-
	Replacing Accuracy	90.16	79.40	62.89	-	-	-	99.99	99.90	99.67

Table 4: Classification accuracies for the different methods on datasets with zero duplication rate.

Method	Accuracy	Mixed Dataset			Additive-Only Dataset			Replacing-Only Dataset		
		Low	Medium	High	Low	Medium	High	Low	Medium	High
ARTra	Additive Accuracy	84.79	76.18	72.04	85.91	72.69	73.49	-	-	-
	Replacing Accuracy	89.56	75.72	57.16	-	-	-	94.26	79.59	58.95
Lost-gene (h=2) Heuristic	Additive Accuracy	82.39	77.12	72.92	87.06	75.93	73.44	-	-	-
	Replacing Accuracy	89.99	72.83	57.02	-	-	-	89.22	77.41	54.75
Gene-frequency Heuristic	Additive Accuracy	72.39	59.35	53.14	72.28	56.54	53.32	-	-	-
	Replacing Accuracy	89.99	81.68	69.68	-	-	-	94.49	85.89	66.74
Mapping-Count Heuristic	Additive Accuracy	80.39	69.56	62.09	77.59	63.35	60.85	-	-	-
	Replacing Accuracy	69.13	56.99	47.49	-	-	-	82.11	67.77	53.58

Table 5: Classification accuracies for the different methods on datasets with double transfer rate.

Method	Accuracy	Mixed Dataset			Additive-Only Dataset			Replacing-Only Dataset		
		Low	Medium	High	Low	Medium	High	Low	Medium	High
ARTra	Additive Accuracy	92.50	83.06	80.18	88.89	82.28	75.23	-	-	-
	Replacing Accuracy	83.33	83.6	62.07	-	-	-	87.73	81.17	66.49
Lost-gene (h=2) Heuristic	Additive Accuracy	95.00	80.65	82.07	88.89	85.65	75.46	-	-	-
	Replacing Accuracy	83.33	78.69	60.09	-	-	-	86.79	76.15	60.47
Gene-frequency Heuristic	Additive Accuracy	95.00	74.19	80.66	85.47	78.48	66.67	-	-	-
	Replacing Accuracy	81.82	81.96	65.02	-	-	-	88.68	82.85	65.18
Mapping-Count Heuristic	Additive Accuracy	92.50	79.03	69.34	85.47	76.79	71.30	-	-	-
	Replacing Accuracy	83.33	68.85	55.17	-	-	-	90.56	86.61	67.28

Table 6: Classification accuracies for the different methods on datasets with species tree size 25.

Method	Accuracy	Mixed Dataset			Additive-Only Dataset			Replacing-Only Dataset		
		Low	Medium	High	Low	Medium	High	Low	Medium	High
ARTra	Additive Accuracy	88.89	79.69	76.83	88.72	81.56	74.13	-	-	-
	Replacing Accuracy	85.09	80.95	58.88	-	-	-	92.21	86.82	71.53
Lost-gene (h=2) Heuristic	Additive Accuracy	88.89	78.52	79.18	92.60	84.56	77.48	-	-	-
	Replacing Accuracy	86.84	75.76	56.07	-	-	-	90.16	77.41	64.57
Gene-frequency Heuristic	Additive Accuracy	86.11	70.31	60.12	82.88	69.82	61.66	-	-	-
	Replacing Accuracy	85.96	83.12	64.80	-	-	-	93.44	88.28	75.61
Mapping-count Heuristic	Additive Accuracy	87.03	77.34	60.12	80.16	68.20	60.32	-	-	-
	Replacing Accuracy	81.57	70.12	60.44	-	-	-	92.62	83.45	70.03

Table 7: Classification accuracies for the different methods on datasets with species tree size 50.

Method	Accuracy	Mixed Dataset			Additive-Only Dataset			Replacing-Only Dataset		
		Low	Medium	High	Low	Medium	High	Low	Medium	High
ARTra	Additive Accuracy	89.89	82.47	75.82	86.72	80.13	73.15	-	-	-
	Replacing Accuracy	91.75	81.49	62.53	-	-	-	95.53	88.28	69.50
Lost-gene (h=2) Heuristic	Additive Accuracy	90.32	82.38	74.57	86.62	80.67	71.02	-	-	-
	Replacing Accuracy	91.55	78.44	62.41	-	-	-	94.08	84.49	65.93
Gene-frequency Heuristic	Additive Accuracy	79.13	62.91	53.07	71.05	58.81	49.13	-	-	-
	Replacing Accuracy	93.40	86.33	76.17	-	-	-	97.30	92.94	83.33
Mapping-count Heuristic	Additive Accuracy	78.06	73.05	64.56	75.73	69.16	66.73	-	-	-
	Replacing Accuracy	80.62	64.88	55.83	-	-	-	85.36	77.62	63.71

Table 8: Classification accuracies for the different methods on datasets with species tree size 200.

Method	Additive Transfer Percentage	Replacing Transfer Percentage
ARTra	78.55	21.45
Lost-gene (h=2) Heuristic	75.76	24.24
Gene-frequency Heuristic	75.75	24.25
Mapping-count Heuristic	36.89	63.11

Table 9: Classification results on the real dataset. Percentage of inferred transfers classified as additive and as replacing by the different classification methods.

Several aspects of this work could be improved further. Our experimental results show that ARTra provides only a modest improvement over the *lost-gene* heuristic. This suggests that a more sophisticated machine learning framework may be able to improve classification accuracy even further. For example, it is possible that including additional structural features of the gene tree and species tree, which we have not yet tested, could help improve classification accuracy. It may also be beneficial to develop additional rule-based classification heuristics for the problem since their classifications could be used as features in the machine learning model.

Funding: This work was supported in part by NSF award MCB 1616514 to MSB.

REFERENCES

[1] Mukul S. Bansal, Eric J. Alm, and Manolis Kellis. 2012. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics* 28, 12 (2012), 283–291.
[2] Mukul S. Bansal, Eric J. Alm, and Manolis Kellis. 2013. Reconciliation Revisited: Handling Multiple Optima when Reconciling with Duplication, Transfer, and Loss. *Journal of Computational Biology* 20, 10 (2013), 738–754.
[3] Mukul S Bansal, Manolis Kellis, Misagh Kordi, and Soumya Kundu. 2018. RANGER-DTL 2.0: rigorous reconstruction of gene-family evolution by duplication, transfer and loss. *Bioinformatics* 34, 18 (2018), 3214–3216. https://doi.org/10.1093/bioinformatics/bty314
[4] Mukul S. Bansal, Yi-Chieh Wu, Eric J. Alm, and Manolis Kellis. 2015. Improved gene tree error correction in the presence of horizontal gene transfer. *Bioinformatics* 31, 8 (2015), 1211–1218. https://doi.org/10.1093/bioinformatics/btu806
[5] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *Proc. of ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 108–122.
[6] Zhi-Zhong Chen, Fei Deng, and Lusheng Wang. 2012. Simultaneous Identification of Duplications, Losses, and Lateral Gene Transfers. *IEEE/ACM Trans. Comput. Biology Bioinform.* 9, 5 (2012), 1515–1528.
[7] Sang Chul Choi, Matthew D. Rasmussen, Melissa J. Hubisz, Ilan Gronau, and Michael J. Stanhope. 2012. Replacing and Additive Horizontal Gene Transfer in Streptococcus. *Molecular Biology and Evolution* 29 (2012), 3309–3320. https://doi.org/10.1093/molbev/mss138
[8] Lawrence A. David and Eric J. Alm. 2011. Rapid evolutionary innovation during an Archaean genetic expansion. *Nature* 469 (2011), 93–96.
[9] Jean-Philippe Doyon, Celine Scornavacca, K. Yu. Gorbunov, Gergely J. Szöllosi, Vincent Ranwez, and Vincent Berry. 2010. An Efficient Algorithm for Gene/Species Trees Parsimonious Reconciliation with Losses, Duplications and Transfers. In *RECOMB-CG (Lecture Notes in Computer Science, Vol. 6398)*, Eric Tannier (Ed.). Springer, 93–108.
[10] Robert Friedman and Bert Ely. 2012. Codon Usage Methods for Horizontal Gene Transfer Detection Generate an Abundance of False Positive and False Negative Results. *Current Microbiology* 65, 5 (2012), 639–642. https://doi.org/10.1007/s00284-012-0205-5
[11] K. Y. Gorbunov and V. A. Liubetskii. 2009. Reconstructing genes evolution along a species tree. *Molekuliarnaia Biologiia* 43, 5 (Oct. 2009), 946–958.
[12] Damir Hasić and Eric Tannier. 2019. Gene tree reconciliation including transfers with replacement is NP-hard and FPT. *Journal of Combinatorial Optimization* (22 Feb 2019). https://doi.org/10.1007/s10878-019-00396-z
[13] Edwin Jacox, Cedric Chauve, Gergely J. Szollosi, Yann Ponty, and Celine Scornavacca. 2016. ecceTERA: comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics* 32, 13 (2016), 2056. https://doi.org/10.1093/bioinformatics/btw105

[14] Slimane Khayi, Pauline Blin, Jacques Pedron, Teik-Min Chong, Kok-Gan Chan, Mohieddine Moumni, Valerie Helias, Frederique Van Gijsegem, and Denis Faure. 2015. Population genomics reveals additive and replacing horizontal gene transfers in the emerging pathogen Dickeya solani. *BMC Genomics* 16, 1 (2015), 788.

[15] Eugene V. Koonin, Kira S. Makarova, and L. Aravind. 2001. Horizontal Gene Transfer in Prokaryotes: Quantification and Classification. *Annual Review of Microbiology* 55, 1 (2001), 709–742. https://doi.org/10.1146/annurev.micro.55.1.709 PMID: 11544372.

[16] Misagh Kordi and Mukul S. Bansal. 2019. Exact Algorithms for Duplication-Transfer-Loss Reconciliation with Non-Binary Gene Trees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16, 4 (2019), 1077–1090. https://doi.org/10.1109/TCBB.2017.2710342

[17] Misagh Kordi, Soumya Kundu, and Mukul S. Bansal. 2019. On Inferring Additive and Replacing Horizontal Gene Transfers Through Phylogenetic Reconciliation. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* (Niagara Falls, NY, USA) *(BCB '19)*. Association for Computing Machinery, New York, NY, USA, 514–523. https://doi.org/10.1145/3307339.3342168

[18] Soumya Kundu and Mukul S. Bansal. 2019. SaGePhy: An improved phylogenetic simulation framework for gene and subgene evolution. *Bioinformatics* 35, 18 (02 2019), 3496–3498. https://doi.org/10.1093/bioinformatics/btz081

[19] Ran Libeskind-Hadas, Yi-Chieh Wu, Mukul S. Bansal, and Manolis Kellis. 2014. Pareto-optimal phylogenetic tree reconciliation. *Bioinformatics* 30, 12 (2014), i87–i95.

[20] Celine Scornavacca, Edwin Jacox, and Gergely J. Szöllosi. 2015. Joint amalgamation of most parsimonious reconciled gene trees. *Bioinformatics* 31, 6 (2015), 841–848.

[21] Celine Scornavacca, Wojciech Paprotny, Vincent Berry, and Vincent Ranwez. 2013. REPRESENTING A SET OF RECONCILIATIONS IN A COMPACT WAY. *Journal of Bioinformatics and Computational Biology* 11, 02 (2013), 1250025. https://doi.org/10.1142/S0219720012500254

[22] Joel Sjostrand, Ali Tofigh, Vincent Daubin, Lars Arvestad, Bengt Sennblad, and Jens Lagergren. 2014. A Bayesian Method for Analyzing Lateral Gene Transfer. *Systematic Biology* 63, 3 (2014), 409–420. https://doi.org/10.1093/sysbio/syu007

[23] Maureen Stolzer, Han Lai, Minli Xu, Deepa Sathaye, Benjamin Vernot, and Dannie Durand. 2012. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics* 28, 18 (2012), 409–415.

[24] Gergely J. Szollosi, Bastien Boussau, Sophie S. Abby, Eric Tannier, and Vincent Daubin. 2012. Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. *Proceedings of the National Academy of Sciences* 109, 43 (2012), 17513–17518. https://doi.org/10.1073/pnas.1202997109

[25] Gergely J. Szollosi, Eric Tannier, Nicolas Lartillot, and Vincent Daubin. 2013. Lateral Gene Transfer from the Dead. *Systematic Biology* 62, 3 (2013), 386–397. https://doi.org/10.1093/sysbio/syt003

[26] Ali Tofigh. 2009. *Using Trees to Capture Reticulate Evolution : Lateral Gene Transfers and Cancer Progression.* Ph.D. Dissertation. KTH Royal Institute of Technology. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-10608

[27] Ali Tofigh, Michael T. Hallett, and Jens Lagergren. 2011. Simultaneous Identification of Duplications and Lateral Gene Transfers. *IEEE/ACM Trans. Comput. Biology Bioinform.* 8, 2 (2011), 517–535.

[28] Olga Zhaxybayeva. 2009. *Horizontal gene transfer: Genomes in flux.* Methods in Molecular Biology, Vol. 532. Humana Press, Chapter Detection and quantitative assessment of horizontal gene transfer, 195–213.