8-2-2018

An Improved Probabilistic Simulation Framework for Gene Family Evolution

Soumya Kundu
University of Connecticut - Storrs, soumya.kundu@uconn.edu

An Improved Probabilistic Simulation Framework for Gene Family Evolution

Soumya Kundu

B.S.E., University of Connecticut, 2018

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

At the

University of Connecticut

2018

APPROVAL PAGE

Master of Science Thesis

An Improved Probabilistic Simulation Framework for Gene Family Evolution

Presented by

Soumya Kundu, B.S.E.

Major Advisor _____

Mukul S. Bansal

Associate Advisor _____

Ion I. Mandoiu

Associate Advisor _____

Yufeng Wu

University of Connecticut

2018

Contents

Abstract

Phylogenetic trees are frequently used to represent the evolution of species, genes, and protein domains. Gene family evolution is usually represented in a framework where gene trees evolve inside a species tree. The recent Domain-Gene-Species model of evolution presents a framework where protein domains also evolve inside one or more gene trees, each of which evolves inside a species tree [1]. The Duplication-Transfer-Loss (DTL) reconciliation and the Domain-Gene-Species (DGS) reconciliation algorithms allow for a parsimony-based approach to inferring the evolutionary histories of a given set of species, genes, and, in the case of DGS reconciliation, protein domains. However, in the absence of biological data regarding the true evolutionary histories of these species, genes, and domains, we must rely on simulated data to validate the accuracy of such phylogenetic reconciliation methods. Although numerous probabilistic simulation frameworks exist for gene family evolution, such as PrIME-GenPhyloData and SimPhy, none of these existing frameworks account for certain aspects of gene family evolution, such as the presence of both additive and replacing horizontal gene transfers and the possibility that the gene family might not be present at the root of the species tree [2–5]. Furthermore, no existing simulation framework can simulate sub-gene level events such as partial gene transfers and the evolution of domain families.

In this work, we modify the PrIME-GenPhyloData simulation framework to simulate both replacing and additive horizontal gene transfers, account for phylogenetic distance bias in choosing transfer recipients, and randomly select the location of gene birth in the species tree. In addition, we introduce the ability to simulate sub-gene level events such as partial gene transfers through the simulated evolution of protein domains within gene families, creating the first probabilistic simulation framework of its kind.

To demonstrate the utility of our new simulation framework, we systematically evaluate the accuracy of DTL reconciliation on simulated datasets that contain both additive and replacing transfers. Our results from this simulation study indicate that DTL reconciliation, which assumes that all transfers are additive, is surprisingly robust to the presence of replacing transfers.

1 Background

Evolutionary trees are fundamental to the study of evolution. The main types of evolutionary trees are species trees, gene trees, and domain trees. A species tree shows the evolutionary history of a chosen collection of species, a gene tree shows the evolutionary history of a gene family, and a domain tree shows the evolutionary history of a domain family. The leaves of each of these trees represent extant entities, such as species, genes, or domains for species trees, gene trees, and domain trees, respectively. The internal nodes of the trees represent hypothetical ancestral entities such as species, genes, or domains.

Gene families evolve inside species trees to form gene trees that are produced through evolutionary events that include speciation, gene duplication, gene loss, and horizontal gene transfer. Similarly, domain families evolve inside one or more gene trees to form domain trees that are produced through speciation, domain duplication, domain loss, and domain transfer events. The Duplication-Transfer-Loss (DTL) reconciliation model attempts to find a reconciliation of the given gene tree with the species tree. The Domain-Gene-Species (DGS) reconciliation model attempts to find both a reconciliation of the given gene trees with the species tree and a reconciliation of the given domain tree with the gene trees. Both reconciliation models are based on the parsimony principle where each event is assigned a cost and a DTL reconciliation or a DGS reconciliation of minimum total cost is sought.

Since true biological data is not available for extant gene and domain families, we need simulated data to evaluate the accuracy of any method that infers evolutionary history. Currently, there are numerous probabilistic simulation frameworks for gene family evolution that produce phylogenetic trees; two main examples that we will look at include PrIME-GenPhyloData and SimPhy [2–5]. However, none of the existing simulation frameworks allow for the simulation of both additive and replacing horizontal gene transfers, the two types of horizontal gene transfer; they usually simulate one of the two types exclusively. However, in reality, both types of horizontal gene transfer are prevalent, as a transferred gene can either be simply added to the new species' genome or replace the existing homologous gene in the new genome. In addition, neither of the

two contemporary simulation frameworks discussed here allow for a random sampling of the location of gene birth on the species tree. As a result, the simulated gene tree always starts evolving at the root of the species tree, which does not always happen in reality, as a gene family might come into existence well past the time of the common ancestor of the group of species in a given species tree. Therefore, accounting for all of these factors in gene family evolution is critical to the accurate simulation of these evolutionary trees.

Furthermore, there are no existing simulation frameworks that can also simulate sub-gene level events such as partial gene transfer or domain family evolution guided by gene trees, which are in turn evolved inside species trees, following the scheme laid out by the DGS model. This makes it difficult to carry out a simulation study to validate the results generated by the DGS reconciliation algorithm.

1.1 Previous Work

One of the two contemporary tree-based simulation frameworks for gene family evolution is SimPhy. SimPhy uses birth-death processes to simulate the evolution of species trees, locus trees, and gene trees, where gene families evolve under incomplete lineage sorting, gene duplication, gene loss, horizontal gene transfer, and gene conversion [3]. However, SimPhy only implements replacing horizontal gene transfers, and additive transfers are not considered. Overall, SimPhy is one of the most robust and advanced simulation frameworks for gene family evolution available today.

The other contemporary simulation framework for gene family evolution is PrIME-GenPhyloData. GenPhyloData also uses birth-death processes to simulate the evolution of species trees and gene trees, where gene families evolve under gene duplication, gene loss, and horizontal gene transfer events [2]. However, GenPhyloData, unlike SimPhy, only implements additive horizontal gene transfers, and replacing transfers are not considered. Furthermore, GenPhyloData also does not support a biased sampling of gene transfer recipients that favors transfers to those lineages that have a shorter phylogenetic distance from the lineage that contains the origin

of the transfer, which SimPhy does.

Also, it is important to note that none of the simulation frameworks that are currently available support both additive and replacing transfers. Furthermore, all of them assume that the evolution of the gene family starts at the root of the species tree. Finally, none of these simulation frameworks have support for sub-gene level events such as partial gene transfer and domain family evolution.

1.2 Our Contributions

In this work, we modify the PrIME-GenPhyloData simulation framework to simulate both replacing and additive horizontal gene transfers. We also add the ability to select transfer recipients with a distance bias and the ability to randomly sample the location of gene birth on the species tree. Finally, we extend the simulation framework to also simulate the evolution of domain trees guided by one or more gene trees and a species tree, which makes possible the simulation of sub-gene level events such as partial gene transfers. These simulated domain trees evolve using the events of domain duplication, loss, speciation, and transfer, where the transfer events can be sampled with user-defined probabilities of replacing or additive transfer, transfers within the same gene tree or across gene trees, and transfers within the same species or across species. Our software will be made publicly available once it is completely ready for release.

Specifically, we chose to modify GenPhyloData instead of a more sophisticated or advanced simulation framework such as SimPhy because of a few reasons. First, the basic two-tree model supported by GenPhyloData matched the DTL models that we use in our work, making it easy to produce simulated data that can be seamlessly fed into our DTL reconciliation programs for a simulation study. Next, since both SimPhy and GenPhyloData lacked most of the key features that we wanted to implement, such as simulating both replacing and additive transfers, randomly sampling the location of gene birth on the species tree, and domain family evolution, we decided to modify the software that had the simpler design and would be easily malleable, which was definitely GenPhyloData. Finally, GenPhyloData uses only the evolutionary events that we also account for in our reconciliation algorithms, namely duplication, transfer, loss, and speciation.

On the other hand, SimPhy simulates more events such as incomplete lineage sorting and gene conversion, which we are not immediately concerned with. Therefore, GenPhyloData was a better fit for the goals of this project.

To demonstrate the utility of our new simulation framework, we systematically evaluate the accuracy of the DTL reconciliation algorithm RANGER-DTL on simulated datasets that contain both additive and replacing transfers. Specifically, we look at how often RANGER-DTL correctly predicts the event types of the nodes on the gene tree and their mappings to nodes on the species tree under different rates of evolution used to generate the gene trees. An important goal is to characterize this accuracy for the case when replacing gene transfers are simulated, as currently, there exists no algorithm that explicitly handles DTL reconciliation with both replacing and additive transfers. Our results from this simulation study indicate that DTL reconciliation, which assumes that all transfers are additive, is surprisingly robust to the presence of replacing transfers, and suggest that it should be possible to design effective heuristics for the DTL reconciliation problem with replacing transfers based on standard DTL reconciliation. Since both replacing and additive transfers are recovered, this might be done through a heuristic that infers which of the inferred transfer events could be replacing transfers, leaving the rest as inferred additive transfers.

The structure of this thesis is as follows. In the methods section, we will introduce the mechanics of the GenPhyloData simulation framework that we modify, followed by a discussion of our modifications. Next, in the application section, we will cover the simulation study of the accuracy of RANGER-DTL conducted using our simulation framework, and in the conclusion section, we will present a summary of our work, followed by potential directions for future work.

2 Methods

The original GenPhyloData simulation framework consists of independent tools that are used to generate species trees and gene trees, among other functions. The two primary tools that we use in our work are *HostTreeGen* and *GuestTreeGen*. Both of these tools make use of the birth-death process for generating trees.

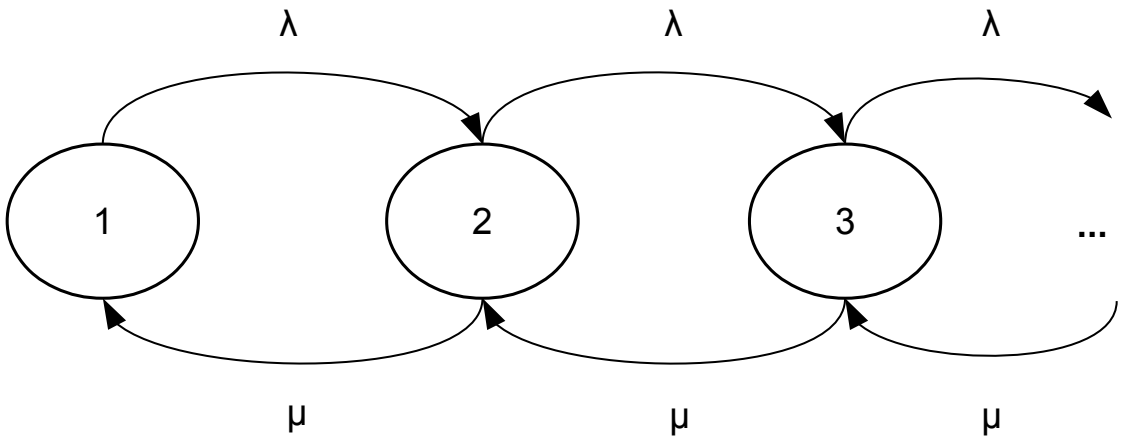

Figure 1: The circles represent states in a birth-death process that correspond to the number of extant lineages that are being evolved. The number of lineages grows with a birth rate of λ and shrinks with a death rate of μ.

2.1 Birth-Death Process

The Birth-Death Process used by the tools in GenPhyloData occurs over a defined time interval. At the beginning of the time interval, the process starts with a single lineage. After that, until the end of the time interval, any extant lineage evolves independently of all other lineages and can undergo births or deaths that occur at rates λ and μ, respectively. Figure 1 illustrates a basic Birth-Death process. A birth creates two independent child lineages that replace the parent lineage while a death stops the evolution of the lineage. At the end of the time interval, all lineages stop evolving; those lineages that do not either reach the end or have a child that reaches the end are usually pruned away. This results in a bifurcating tree where the lineages are analogous to the edges, the births are analogous to the vertices, and the tips of the lineages reaching the end of the time interval are analogous to the leaves.

Also, it is important to note that the simulation process proceeds discretely with lineages being processed one at a time. When each lineage is processed, the next event time that will mark the end of that lineage is sampled from an exponential distribution with a rate equal to that of the sum

Figure 2: This figure from [6] shows an overview of the HostTreeGen tool, which uses a Birth-Death process to evolve lineages over a given time period.

of the input event rates, and unless the new event is either a loss or a leaf, the two resulting child lineages replacing the parent lineage are added to a queue to be processed in the same way. As a result, some lineages might be far ahead in time compared to other lineages, since events times are sampled from a distribution and some lineages might, by chance, have successively long intervals between events, while others might have successively shorter ones.

2.2 HostTreeGen

HostTreeGen uses the birth-death process to generate a bifurcating species tree. The user provides the time interval and the speciation and loss rates, which are used to sample the evolutionary events; here, the speciation and loss rates are analogous to the birth and death rates, respectively. The sum of the speciation and loss rates is used as the rate parameter for the exponential distribution from which the times for successive events are sampled. Once an event is sampled, it is assigned to be either a speciation or loss event according to a weighted random sampling based on the relative rates of speciation and loss. Furthermore, a minimum and maximum number of leaves in the final pruned tree can be specified, where the simulation is repeated for a number of maximum attempts until the size restrictions are met. Figure 2 from [6] illustrates the process of generating a species tree using this tool.

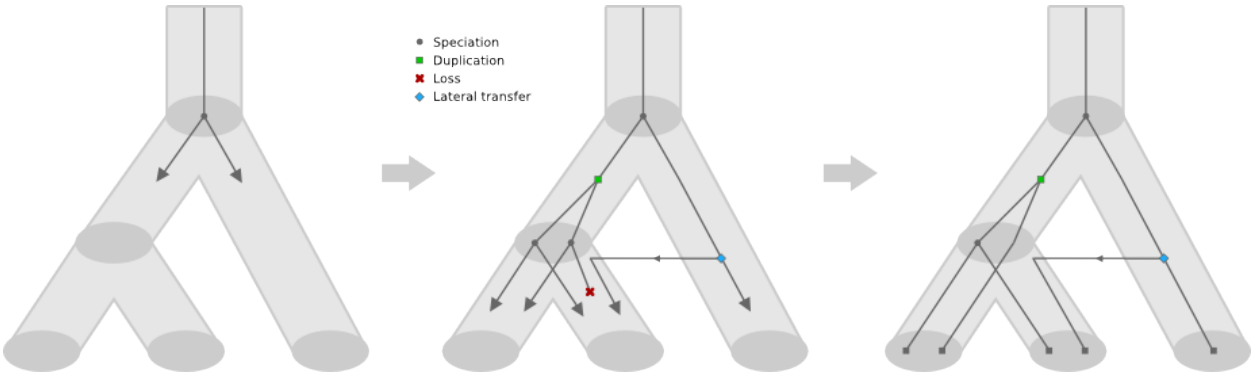

Figure 3: This figure from [6] shows an overview of the GuestTreeGen tool that evolves gene tree lineages guided by the topology of an underlying species tree using a Birth-Death process.

2.3 GuestTreeGen

GuestTreeGen also uses the birth-death process to generate a bifurcating gene tree. The user provides a species tree, the time interval, and the gene duplication, loss, and transfer rates, which are used to sample the evolutionary events; here, the duplication and loss rates are analogous to the birth and death rates, respectively, while the transfer rate has an effect similar to that of the birth rate as it too expands the size of the gene family. The sum of the duplication, loss, and transfer rates is used as the rate parameter for the exponential distribution from which the times for successive events are sampled.

The birth-death process for the gene tree starts with a single lineage at the root of the species tree. From there, each lineage evolves independently through the different evolutionary events. If the time for the sampled event is at or past the time of speciation of the species tree lineage within which the gene tree lineage has been evolving, then the sampled event is assigned to be a speciation event. Otherwise, if the lineage reaches a leaf of the species tree, then the event is labeled as a leaf. If neither is the case, then the event is assigned to be either a duplication, loss, or transfer event according to a weighted random sampling based on the relative rates of duplication, loss, and transfer.

A duplication replaces the parent gene lineage with two independent child lineages that continue evolving within the same species lineage, while a loss terminates the evolution of that lineage.

A gene transfer replaces the parent gene lineage with two independent child lineages, among which one, at random, continues evolving within the same species lineage, while the other starts evolving within a contemporary species tree lineage that is chosen uniformly at random from a collection of all contemporary species lineages for the given transfer event. Thus, the transfer event simulates an additive gene transfer. Figure 3 from [6] on page 7 illustrates the process of generating a gene tree using this tool. Also, transfers are not allowed at the root of the species tree as there are no other species lineages that can receive a transferred gene lineage in that case.

2.4 Replacing Transfers

Most of the modifications to the GenPhyloData simulation framework, other than the ones made for implementing domain family evolution, were made to the GuestTreeGen tool. The first such modification allowed for the simulation of replacing transfers alongside the existing additive transfers in an effort to more realistically capture the true nature of gene family evolution.

2.4.1 Challenges.

A key property of the birth-death process used so far has been the fact that lineages evolve independent of each other. However, including replacing transfers leads to this property no longer holding true. In a replacing transfer event, a transferred gene replaces a homologous gene in another species and continues evolving in the new species. When simulating this event, the transfer lineage replacing the existing gene lineage in the new species will violate the assumption of independent evolution of the gene lineages. As a result, since the simulation process proceeds discretely one lineage at a time, we must handle complications that can arise due to this interdependence of lineages, such as the case where a replacing transfer might replace a lineage which already has a child that has replaced another lineage. In that case, the latest replacing transfer will invalidate the previous replacing transfer, leading to a complication. Figure 4 (a) on page 9 illustrates this challenge.

Another problem that arises from simulating replacing transfers is the possibility that when a

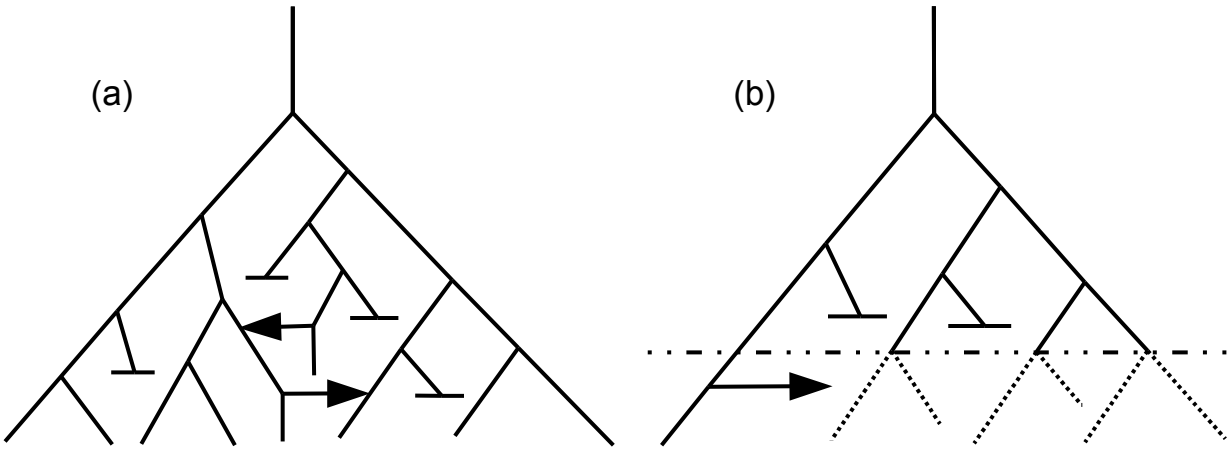

Figure 4: These diagrams represent gene trees and the arrows represent replacing transfers. Part (a) illustrates the challenge of new replacing transfers invalidating previous ones. Part (b) illustrates the challenge of finding transfer recipients for replacement if the transfer lineage is far ahead in the simulation process.

replacing transfer is sampled, if the lineage that is being transferred is far ahead in time in the simulation compared to the other lineages in the tree, then there might not be any gene lineages to replace at that time, even though replaceable lineages might be processed in the future as the other lineages catch up. Although we can re-sample the event type in these scenarios, ideally, we would like to do that as seldom as possible in order to produce a tree where the relative proportion of events matches closely the input parameters provided by the user. Figure 4 (b) illustrates this challenge.

2.4.2 Implementation.

In the modified version of GuestTreeGen, we implement a simulation of both replacing and additive transfers as follows. First, we add an input parameter for the probability of any transfer event to be a replacing transfer, where the default value is set to 0.5. Next, in the GuestTreeGen simulation process, whenever a transfer event is sampled, it is assigned to be either an additive or a replacing transfer using a weighted random sampling based on the probability of each event; the probability of an additive transfer, as expected, is the difference of the probability of a replacing transfer from 1.

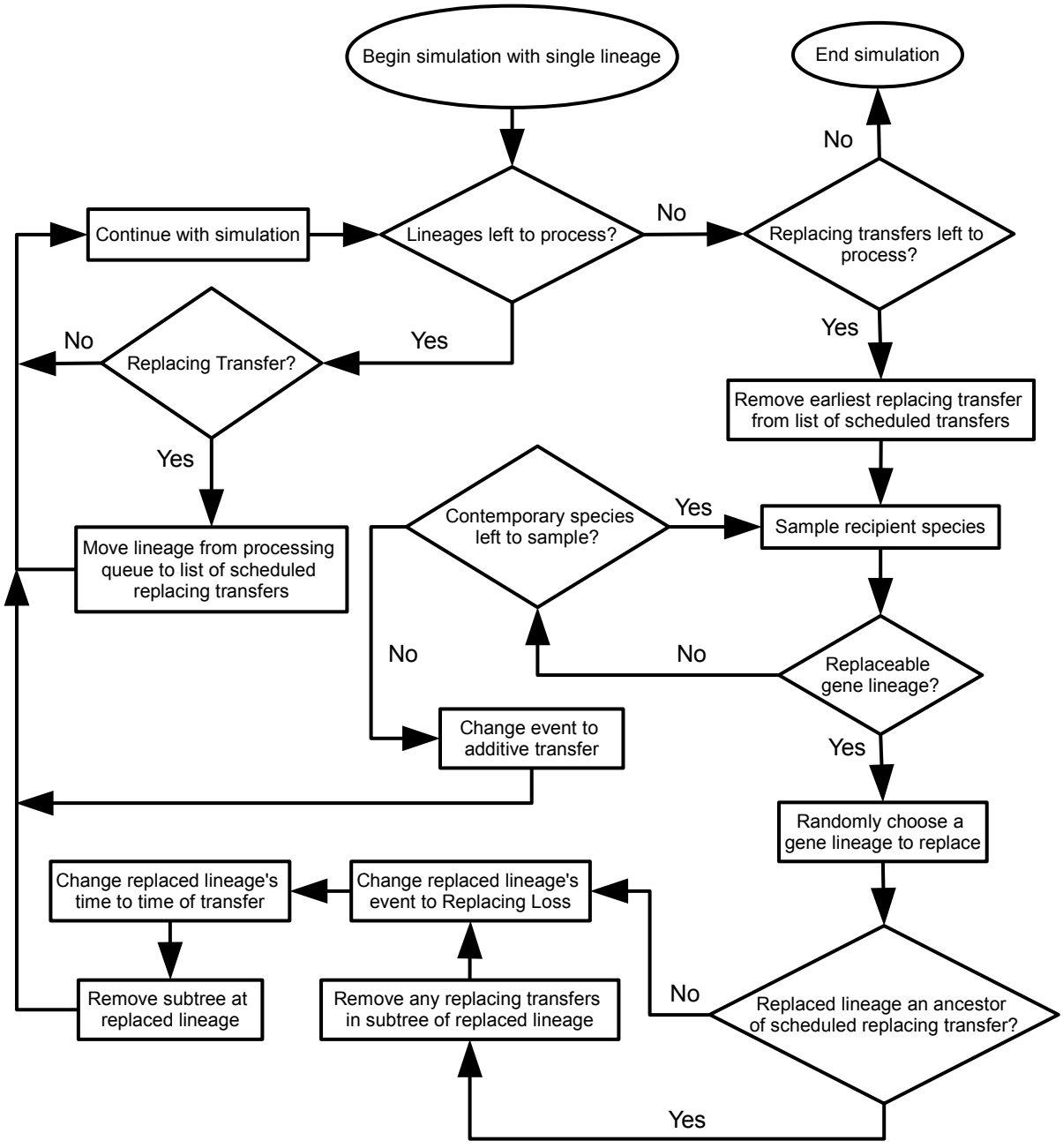

Figure 5: This flowchart summarizes the strategy that we implement in order to handle replacing transfers during the simulation process.

If an additive transfer is sampled, then the simulation proceeds according to the original framework. However, if a replacing transfer is chosen, then the evolution of the replacing lineage is temporarily halted while the rest of the tree continues to evolve. After all lineages, except those whose evolution has been temporarily halted, have finished evolving through the normal birth-death process, the simulation continues in a similar manner with the replacing transfer lineages, one lineage at a time, in a decreasing order of their height, measured by simulation time; the simulation proceeds to the next replacing lineage only when all of the lineages that have descended from the previous replacing lineage have finished evolving.

Furthermore, each replacing transfer is handled as follows. A contemporary species lineage is randomly chosen among the contemporary lineages available at the time of the transfer event. Next, the presence of a contemporary gene lineage within the chosen species lineage at the time of the transfer is verified. If such a replaceable gene lineage does not exist, then another contemporary species lineage is sampled from among the lineages that have not yet been found to lack a replaceable gene lineage. A replaceable lineage must be a gene lineage that is extant at the time of the transfer event and is not a sibling of the transferring gene lineage.

Once a gene lineage that is suitable for replacement is found, then the event for the lineage that is being replaced is changed to a replacing loss event, its event time is changed to match that of the replacing transfer event, and references to its children are nullified. Following that, all of the other replacing transfers that have not been processed yet are checked to ensure that they are not in the subtree of the new replacing loss lineage; if any are indeed found to be in its subtree, then those replacing transfers are discarded, as they are now in the subtree of a lost lineage. Finally, the replacing transfer lineage continues to evolve in the newly sampled species lineage, and the simulation proceeds until all lineages in the subtree of the replacing transfer have finished evolving before proceeding to any remaining replacing transfers that need to be processed. However, in the rare case where no replaceable gene lineage is found in all of the contemporary species lineages, the replacing transfer's event is changed to additive transfer and the transferred lineage continues evolving within the species tree lineage that was first sampled as the transfer recipient.

The simulation process continues in the manner as described until all lineages, including all of the replacing transfer lineages and their descendants, have been evolved to completion. Figure 5 on page 10 provides a visual guide to the strategy outlined here to handle the simulation of replacing transfers. In reference to the challenges mentioned before, the implementation tackles the second one, the availability of replaceable gene lineages, by handling replacing transfers after all other lineages have evolved in order to maximize the number of gene lineages that can be replaced. This implementation also tackles the first challenge by processing replacing transfers by the order of their simulation time or height on the gene tree so that no replacing transfer can replace an ancestor of another replacing transfer and any replacement affecting an ancestor of a scheduled replacing transfer can be simply addressed by discarding the scheduled replacing transfer.

2.5 Distance-Biased Transfers

The standard GenPhyloData simulation framework samples species lineage recipients for gene transfers uniformly at random among all contemporary lineages present at the time of the transfer event. However, gene transfer happens more frequently between species that are closely related than those that are more distantly related. In a simulation framework, this aspect of gene family evolution can be captured by assigning greater probabilities for choosing those species as transfer recipients that have shorter phylogenetic distances to the species from which the transfer is originating.

In the modified simulation framework, we implement this idea as follows. First, we add an input parameter that takes as an argument a string specifying the type of distance bias to be used. The three options are none, simple, and exponential. If the none option is chosen, then all sampling for gene transfer recipients happens uniformly at random with no changes from the original implementation. If the simple option is chosen, then each contemporary species lineage that can be chosen as a recipient for the transfer is assigned a probability that, compared to the other lineages, is proportional to the inverse of its phylogenetic distance, as measured in simulation time, to the origin of the transfer. This is implemented by assigning each candidate for transfer recipient a

weight that is equal to the inverse of its phylogenetic distance from the source of the transfer and then using weighted random sampling to choose a transfer recipient. Finally, if the exponential option is chosen, then each candidate for transfer is assigned a probability that is proportional to the value of the probability density function for the exponential distribution, $\lambda e^{-\lambda x}$, where x is the phylogenetic distance of the candidate from the origin of the transfer and λ is the rate parameter. The rate parameter has a default value of 1.0, but the user has the option of specifying a value for it as an input. Again, this is implemented by assigning each transfer candidate a weight that is equal to $\lambda e^{-\lambda x}$ and then using weighted random sampling to choose a transfer recipient. In both of the last two cases, those species lineages which have smaller distances to the origin of the transfer have larger weights and therefore have greater probabilities of being chosen as recipients, which creates the intended distance-biased sampling of transfer recipients.

2.6 Location of Gene Birth

The original GuestTreeGen program starts the evolution of the gene tree from the root of the species tree. However, this practice does not necessarily reflect the reality of gene family evolution, as for any group of species whose genomes contain members of a certain gene family, the common ancestor of all of those species might not have had any members of that gene family in its genome. This is quite possible when a gene appears or takes birth within a species that is a descendant of the common ancestor and then spreads to other species through evolutionary events such as horizontal gene transfer.

In our modified simulation framework, we introduce two input parameters for sampling the location of gene birth on the species tree; one is a simple flag which lets the user decide whether or not they wish to randomly sample the location of gene birth on the species tree while the other lets the user input a real value that they wish to use as the rate parameter for the exponential distribution that is used to assign weights to each of the lineages of the species tree for a weighted random sampling to pick the lineage within which the gene tree will start evolving; the default value for the rate parameter is 1.0. If the user decides to not use random sampling of the location

of gene birth, then the gene tree starts evolving from the root of the species tree. Otherwise, each species tree lineage is assigned a weight equal to the value of the probability density function for the exponential distribution, $\lambda e^{-\lambda x}$, where x is the difference, in simulation time, between the time at the end or bottom vertex of the lineage and that at the first speciation at the root of the species tree and λ is the rate parameter. Then, weighted random sampling is used to pick a species tree lineage within which the gene tree evolution will start. This method of sampling the location of gene birth on the species tree is specifically designed to bias the choice of the location of gene birth toward the root of the species tree, as otherwise, a more uniform sampling approach would lead to very small trees being generated on average. However, the rate parameter allows the user to either sharpen or relax this bias according to their needs.

2.7 DomainTreeGen

A major component of our new simulation framework is the ability to simulate the evolution of a protein domain family guided by gene trees. We accomplished this by extending the modified GenPhyloData simulation framework with the addition of a new tool called DomainTreeGen, which borrows its general structure from GuestTreeGen and retains the modifications we made to it, including the mechanisms for replacing transfers, distance-biased transfers, and sampling the location of gene birth. However, while GuestTreeGen evolves a gene tree within a species tree, DomainTreeGen evolves a domain tree within one or more gene trees, which themselves have been evolved within a species tree.

For input, DomainTreeGen requires a species tree, a directory of gene trees containing at least one gene tree, and rates for the evolutionary events of domain duplication, loss, and transfer. There are also a multitude of optional arguments that can be provided. These arguments allow the user to specify the proportion of domain transfers that should be replacing, the proportion of transfers that should occur across gene trees, the proportion of transfers that should happen across species, the type of distance bias to use for transfers, whether to randomly sample the location of domain birth on the initial gene tree, and the degree of the bias towards the root of the gene tree in the sampling

of the location of domain birth, among numerous other factors that are important to the simulation.

DomainTreeGen uses a birth-death process to generate a domain tree as follows. First, among the gene trees that are provided as input, one is selected uniformly at random to serve as the starting point for the domain to evolve. Next, if random sampling of domain birth location is not used, then the root domain lineage starts evolving at the root of the selected gene tree. Otherwise, a starting gene lineage is randomly chosen according to the given parameters and according to the process described previously for sampling the location of gene birth. From there, the domain tree evolves according to the birth-death process in a manner similar to the evolution of a simulated gene tree. A domain lineage undergoes speciation if it comes across a speciation, duplication, or transfer in the underlying gene tree, resulting in two independent child lineages that replace the parent lineage and continue evolving within the two sibling gene lineages that result from the event on the gene tree. A domain duplication also produces two independent child lineages that replace the parent lineage, but these children continue to evolve within the same gene lineage. Finally, a loss results in the termination of evolution of that lineage.

If a transfer event is sampled, then it's first categorized as replacing or additive according to a weighted random sampling based on the input parameters. Next, the transfer is categorized as an intra-gene or inter-gene transfer, and again as either an intra-species or inter-species transfer, all according to weighted random sampling based on the input parameters. When a specific type of transfer is finally chosen from the different possibilities, it is processed nearly identically to the way gene transfers are processed, except the added restrictions on limiting transfers to either happen strictly within the same or across different gene trees or species. Therefore, the underlying replacing and additive transfer processes are the same as the ones present in the modified version of GuestTreeGen. However, new mechanisms are implemented for DomainTreeGen in order to specifically sample domain transfer recipients that are either from the same or different gene tree or species. In addition, if a domain transfer lineage does not have any gene lineage to which it can transfer while meeting the restrictions imposed by its specific event type, then the lineage is re-sampled with a fresh event assignment. Also, the distance-bias feature for choosing gene transfer

recipients is extended to work for the domain transfer case as well. Finally, the user can choose to enforce the evolution of the domain tree across all input gene trees. In this case, DomainTreeGen will keep producing domain trees until one is produced that satisfies the requirements or a maximum number of attempts, which can be set by the user, is reached.

2.8 Partial Gene Transfers

The DomainTreeGen tool also makes available the possibility of implementing the evolutionary event of partial gene transfers. In a partial gene transfer, parts of genes are transferred between different species instead of the entire gene. Since the domain family evolution implemented in DomainTreeGen evolves genetic information at the sub-gene level, partial gene transfers can now be simulated as follows. First, we can evolve nucleotide sequences down the simulated gene and domain trees by starting with a base sequence at the root and then using a substitution model to sample changes in the sequences at each node in the tree. Once these sequences have evolved and reached the leaves of the trees, we can concatenate the domain leaf sequences to their corresponding genes. Since the domains are sub-gene level units and they undergo transfer events, we would effectively simulate partial gene transfers.

In order to evolve the required sequences, we use the tool Seq-Gen [7]. Seq-Gen is a program that evolves nucleotide sequences along a given phylogeny using common substitution models. Along with the modified simulation framework, this provides us with all of the necessary components required to simulate partial gene transfers. We design a Python script that simplifies and automates this simulation process for generating datasets with partial gene transfers. The steps required to simulate partial gene transfers are as follows.

First, using the modified simulation framework, we generate a species tree and a number of gene trees and domain trees. Next, we provide a directory of domain trees, a directory of gene trees, and a directory of domain to gene leaf mappings as input to our Python script and run it. The script then formats the input trees to ensure compliance with Seq-Gen and calls Seq-Gen on each of the given domain and gene trees. Using the general time reversible (GTR) model, we use

Seq-Gen to generate short sequences of 100 bases along the domain trees and longer sequences of 1000 bases along the gene trees, all of which we output directly to FASTA format files. However, while these options for the substitution model and sequence lengths are set as the default ones, the user can certainly use their own preferred settings if they are different from the default ones.

Finally, the script parses each given leaf mapping file, and for each sequence at a leaf of a domain tree, it inserts that domain sequence at a random position in the gene sequence that can be found at the gene tree leaf to which the domain leaf maps. The one restriction that is enforced with the random insertion is that the insertion is not allowed to occur in the middle of a domain sequence that has been previously inserted into the gene sequence through this process. The user can also choose to simply append the domain sequences to the ends of their respective gene sequences instead of using the default random domain insertion.

3 An Application: Assessing Impact of Replacing Transfers on Transfer Inference

There do not currently exist any algorithms or heuristics to compute DTL reconciliations with both replacing and additive transfers, known as Duplication-additive-Transfer-Replacing-transfer-Loss (DTRL) reconciliations, and it is not even known how algorithms for computing optimal DTL reconciliations perform when confronted with gene trees that have been affected by both additive and replacing transfers. We therefore focused our experimental analysis on answering two fundamental questions: (i) How is the accuracy of DTL reconciliation affected by the presence of replacing horizontal gene transfers? (ii) How well does DTL reconciliation perform at inferring replacing transfer events?

To answer these questions, we used our new simulation framework to create a large number of species and gene trees with varying rates of evolutionary events, computed optimal DTL reconciliations for the gene/species tree pairs, and evaluated the accuracy of the inferred reconciliations by comparing them to the true evolutionary histories of those gene trees. To compute optimal DTL

reconciliations, we employed the widely-used RANGER-DTL [8, 9] software package.

3.1 Simulated Datasets

We used our new simulation framework to generate a large number of gene/species tree pairs, with varying rates of evolutionary events. Specifically, we generated 100 species trees, each containing 100 leaves and of height 1. Next, inside each of the species trees, we generated three different gene trees using low, medium, and high rates of duplication, additive transfer, replacing transfer, and loss events, resulting in three sets of 100 gene trees. To generate the low DTRL gene trees, we used duplication, additive transfer, replacing transfer, and loss rates of 0.133, 0.133, 0.133, and 0.266, respectively; for the medium DTRL gene trees we used rates of 0.3, 0.3, 0.3, and 0.6, respectively; and for the high DTRL gene trees we used rates of 0.6, 0.6, 0.6, and 1.2, respectively. Thus, the total transfer rate was twice the duplication rate, with an equal rate of additive and replacing transfers, and the loss rate was assigned to be equal to the sum of the duplication and additive transfer rates. These duplication, transfer, and loss rates are based on rates observed in real data and capture both datasets with lower rates of these events and datasets with a very high rate of these events [10].

For the low DTRL gene trees, the average gene tree leaf set size was 96.11, with an average of 2.37 additive transfers, 2.65 replacing transfers, and 2.19 duplication events per gene tree. For the medium DTRL gene trees, the average gene tree leaf set size was 94.75, with an average of 5.09 additive transfers, 5.01 replacing transfers, and 5.00 duplication events per gene tree. For the high DTRL gene trees, the average gene tree leaf set size was 110.22, with an average of 9.52 additive transfer events, 9.42 replacing transfer events, and 10.39 duplication events per gene tree.

3.2 Impact of Replacing Transfers on DTL Reconciliation

We evaluated the accuracy of DTL reconciliation in inferring the evolutionary event and species tree mapping for each internal node in the simulated gene trees. We computed a single optimal reconciliation for each gene tree using RANGER-DTL 2.0 [9] with default parameters and compared

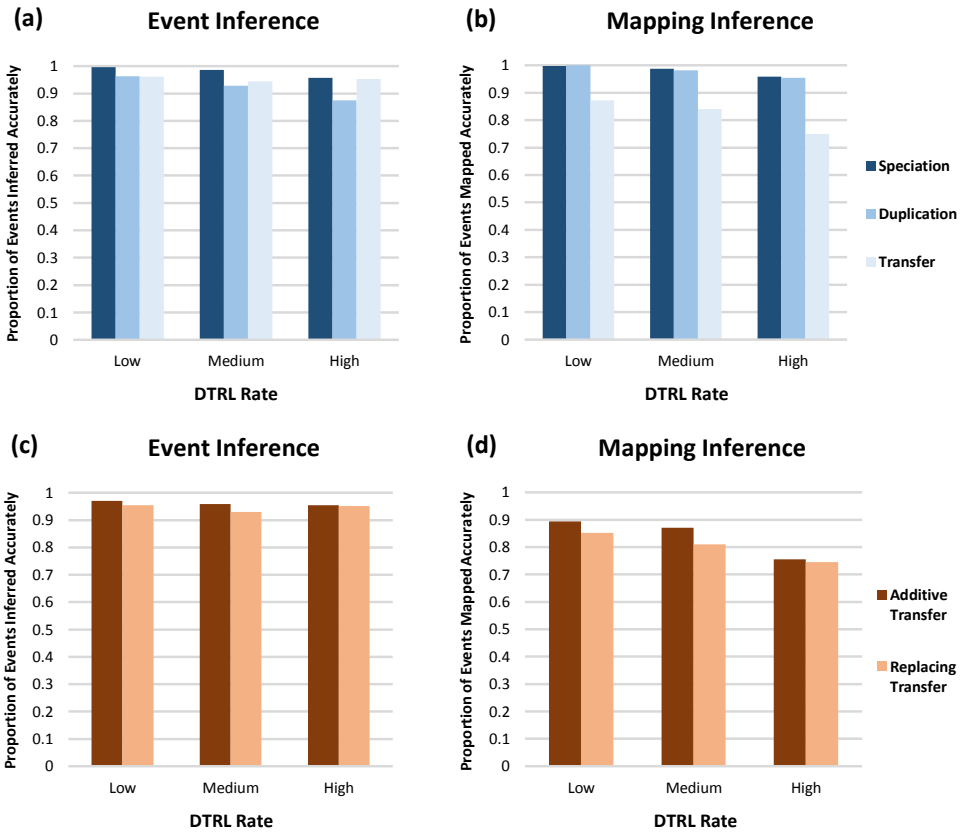

Figure 6: Accuracy of DTL reconciliation in the presence of replacing transfers. Part (a) shows the fraction of internal nodes across all low DTRL, medium, DTRL, and high DTRL gene trees, whose event types, speciation, duplication, or transfer, are inferred correctly through DTL reconciliation. Part (b) shows the fraction of internal nodes across all low DTRL, medium, DTRL, and high DTRL gene trees, whose mappings are inferred correctly through DTL reconciliation. Part (c) shows the fraction of additive transfer nodes and replacing transfer nodes across all low DTRL, medium, DTRL, and high DTRL gene trees, that are correctly inferred as transfer events by DTL reconciliation. Part (d) shows the fraction of additive transfer nodes and replacing transfer nodes across all low DTRL, medium, DTRL, and high DTRL gene trees, that are mapped correctly by DTL reconciliation. For each DTRL rate, results are averaged across 100 datasets.

the computed reconciliation against the true evolutionary history of that gene tree. We observed very high accuracy for inferring the correct event type (speciation, duplication, or transfer) at each gene tree node. For instance, for the low DTRL gene trees, 99.67%, 96.35% and 96.22% of the gene tree nodes labeled as speciation, duplication, and transfer, respectively, in the computed reconciliations were inferred correctly. Even for the high DTRL gene trees, these percentages remained very high at 95.69%, 87.49%, and 95.25%, respectively. These results are shown in Figure 6(a).

Looking at the accuracy of mapping inference, we found that 99.09%, 97.11%, and 92.15% of all internal nodes were assigned the correct species node mapping for the low, medium, and high DTRL gene trees, respectively. Detailed results are shown in Figure 6(b).

We compared these results for event and mapping accuracy with results obtained on gene trees simulated with the same overall rates of duplication, transfer, and loss events but in which all simulated transfers were additive transfers (no replacing transfers). We found that the numbers were nearly identical, showing that the presence of replacing transfers does not negatively affect the accuracy of DTL reconciliation itself. For example, for the high DTL gene trees, the percentage of speciation, duplication, and transfer nodes assigned the correct event type was 95%, 81%, and 95%, respectively, and 91% of all nodes were assigned the correct mapping. Note, however, that DTL reconciliation cannot distinguish between additive and replacing transfers, and both types of transfer events are simply inferred as "transfers".

3.3 Accuracy of Inferring Replacing Transfers

Next, we performed additional analysis to study if there was any discrepancy in the accuracies of inferring the correct event type (transfer) or mapping for additive transfers and those for replacing transfers. For the low DTRL gene trees, we found that additive transfers were assigned the correct event type 97.05% of the time and the correct mapping 89.45% of the time, while for replacing transfers these numbers were 95.47% and 85.28%, respectively. Likewise, for the medium DTRL gene trees, additive transfers were assigned the correct event type 95.87% of the time and the

correct mapping 87.03% of the time, while for replacing transfers these numbers were 93.01% and 81.04%, respectively. For high DTRL gene trees, these numbers were 95.38% and 75.53% for the additive transfers and 95.12% and 74.52% for the replacing transfers. Overall, this shows that replacing transfers are inferred and mapped with accuracy comparable to that of additive transfers. These results are shown in Parts (c) and (d) of Figure 6.

These results are highly significant and suggest that to design a good heuristic for the DTRL reconciliation problem, it would suffice to first use DTL reconciliation to identify transfer events and then classify that set of transfer events as being either replacing or additive. If this classification can be done accurately (and efficiently), then an accurate DTRL-reconciliation can be easily computed.

4 Conclusion

In this work, we develop a probabilistic simulation framework for domain and gene family evolution that can accurately generate gene trees, especially in the presence of horizontal gene transfer, and generate domain trees across multiple gene trees. Our simulation framework can also simulate both additive and replacing transfers, allow for gene and domain trees that do not always start evolving at the root of their host trees, support horizontal transfer events that are sensitive to phylogenetic distance, and make possible the simulation of partial gene transfers. We use this new simulation framework to study the accuracy of DTL reconciliation in recovering the evolutionary histories of gene families that have been shaped by both replacing and additive horizontal gene transfer and show that DTL reconciliation performs well at recovering both types of transfer events. This suggests that a good heuristic for the DTRL reconciliation problem can simply use DTL reconciliation to identify the transfer events and then separate the two types of transfers.

Going forward, the simulation framework can be improved further by adding support for features such as multi-gene transfers. Such new features can broaden the scope of this tool and allow for the simulation of an even more diverse range of evolutionary scenarios. In addition, this sim-

ulation framework can now be used to conduct simulation studies of DGS reconciliation. These studies can characterize the accuracy of those methods and shed light on how to make better evolutionary inferences by identifying the pitfalls of the current algorithms as demonstrated by the simulated data.

References

[1] Li, L., Bansal, M.S.: An integrated reconciliation framework for domain, gene, and species level evolution. (IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB); in press.)

[2] Sjostrand, J., Arvestad, J.L., Sennblad, B.: Genphylodata: realistic simulation of gene family evolution. BMC Bioinformatics **14**(209) (2013)

[3] Mallo, D., De Oliveira Martins, L., Posada, D.: Simphy : Phylogenomic simulation of gene, locus, and species trees. Systematic Biology **65**(2) (2016) 334–344

[4] Dalquen, D.A., Anisimova, M., Gonnet, G.H., Dessimoz, C.: Alfa simulation framework for genome evolution. Molecular biology and evolution. **29**(4) (2012) 1115–1123

[5] Beiko, R.G., Charlebois, R.L.: A simulation test bed for hypotheses of genome evolution. Computer applications in the biosciences : CABIOS. **23**(7) (2007) 825–831

[6] Sjostrand, J., Tofigh, A., Daubin, V., Arvestad, L., Sennblad, B., Lagergren, J.: A bayesian method for analyzing lateral gene transfer. Systematic Biology **63**(3) (2014) 409–420

[7] Rambaut, A., Grass, N.C.: Seq-gen: an application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. Bioinformatics **13**(3) (1997) 235–238

[8] Bansal, M.S., Alm, E.J., Kellis, M.: Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. Bioinformatics **28**(12) (2012) 283–291

[9] Bansal, M.S., Kellis, M., Kordi, M., Kundu, S.: Ranger-dtl 2.0: rigorous reconstruction of gene-family evolution by duplication, transfer and loss. (Bioinformatics; in press.)

[10] Bansal, M.S., Wu, Y.C., Alm, E.J., Kellis, M.: Improved gene tree error correction in the presence of horizontal gene transfer. Bioinformatics **31**(8) (2015) 1211–1218