Deciphering Microbial Gene Family Evolution Using Duplication-Transfer-Loss Reconciliation and RANGER-DTL

Mukul S. Bansal
Department of Computer Science & Engineering, University of Connecticut,
Storrs, CT 06269, USA
mukul.bansal@uconn.edu

Abstract

Phylogenetic reconciliation has emerged as a principled, highly effective technique for investigating the origin, spread, and evolutionary history of microbial gene families. Proper application of phylogenetic reconciliation requires a clear understanding of potential pitfalls and sources of error, and knowledge of the most effective reconciliation-based tools and protocols to use to maximize accuracy. In this book chapter, we provide a brief overview of Duplication-Transfer-Loss (DTL) reconciliation, the standard reconciliation model used to study microbial gene families, and provide a step-by-step computational protocol to maximize the accuracy of DTL reconciliation and minimize false-positive evolutionary inferences.

Keywords: Gene family evolution, phylogenetic reconciliation, horizontal gene transfer, RANGER-DTL

1 Introduction

Microbes, i.e., prokaryotes and single-celled eukaryotes, are the most ancient and most abundant life forms on earth and play a fundamental role in shaping and sustaining the biosphere. Understanding microbial evolutionary processes and inferring the evolutionary histories of microbial genes and genomes is therefore crucial to understanding life on our planet. Microbial gene family evolution is driven by complex evolutionary processes such as speciation, gene duplication, gene loss, and, perhaps most importantly, horizontal gene transfer (transfer for short). Figure 1 illustrates how these processes shape the evolution and phylogenetic distribution of microbial gene families. *Duplication-Transfer-Loss (DTL) reconciliation* is a well-developed and widely-used computational technique for inferring the presence and impact of such evolutionary processes in the evolutionary histories of microbial gene families [2, 3, 5, 8, 11, 12, 15, 18, 21, 24, 27, 35, 47, 48, 50, 52, 54, 56–58]. The detailed knowledge of gene family evolution enabled by DTL reconciliation has many important biological applications in evolutionary biology and comparative genomics; these include inference of transfer events, dating gene birth, inference of orthologs, paralogs, and xenologs, reconstruction of ancestral microbial gene content, gene tree error-correction, and species tree selection and dating.

DTL reconciliation involves the systematic comparison of a *gene tree* (i.e, a phylogeny showing the evolutionary relationships between genes of the chosen gene family) with a corresponding *species tree* (i.e., a phylogeny showing the evolutionary relationships between the microbial species or strains included in the analysis). Algorithms for DTL reconciliation work by reconciling any topological differences between the two trees by invoking gene duplication, transfer, gene loss, and speciation. The end result of DTL reconciliation is a mapping of the nodes of the gene tree to nodes (or edges/branches) of the species tree, showing how the branches of the gene tree are embedded within the branches of the species tree, as well as a labeling of each internal

node of the gene tree as representing either a speciation, duplication, or transfer event. This is illustrated in Figure 2.

To correctly apply DTL reconciliation, one requires a clear understanding of its potential pitfalls and sources of error as well as knowledge of how to correctly interpret its results. The purpose of this book chapter is to equip readers with knowledge about some of the most effective tools and techniques for DTL reconciliation and to provide a step-by-step computational protocol to maximize the accuracy of DTL reconciliation and minimize false-positive evolutionary inferences.

The remainder of this book chapter is organized as follows. In the next section, we review the three key software tools used in our proposed computational protocol. We provide a step-by-step description of the computational protocol in Section 3. In Section 4, we identify potential difficulties in applying the proposed protocol and discuss further enhancements, best-practices, alternatives, and expected accuracy. Concluding remarks appear in Section 5.

2 Description of software tools

Our proposed computational protocol for reconciling microbial gene families use three key software tools, TreeFix-DTL [6], MAD [59], and RANGER-DTL 2.0 [5]. Next, we briefly describe these tools and discuss how they work.

TreeFix-DTL: The accuracy of any reconciliation-based evolutionary inference depends critically on the accuracy of the gene tree being reconciled. Since reconstructed gene trees often have considerable topological uncertainty, e.g., due to insufficient information in gene family sequence alignments or short or long branches in the tree, it is often necessary to error-correct gene trees using additional information prior to reconciliation. This has led to the development of several species-tree-aware methods for microbial gene tree reconstruction or error-correction [6,11,24,28,40,42,50]. TreeFix-DTL [6] is one of the most effective such methods and can be used to error-correct a

given maximum likelihood gene tree when a reasonable estimate of the species tree is available. Specifically, TreeFix-DTL takes as input a maximum likelihood gene tree (such as one constructed using RAxML [51]), the corresponding gene family alignment, and a rooted species tree, and uses tree search heuristics and statistical tests to find an alternative, error-corrected gene tree whose likelihood is statistically equivalent to that of the input gene tree but which has a lower DTL reconciliation cost against the species tree. This tool has been rigorously tested and shown to be highly effective at improving gene tree accuracy and the accuracy of downstream evolutionary inferences [6, 28]. TreeFix-DTL is open source and can be freely downloaded from: `https://www.cs.hmc.edu/~yjw/software/treefix-dtl/`

MAD: DTL reconciliation requires that both the gene tree and species tree being reconciled are rooted. Since nearly all standard gene tree inference methods, including RAxML and TreeFix-DTL, result in unrooted gene trees, it is necessary to root gene trees prior to reconciliation. Minimal Ancestor Deviation (MAD) [59] is a phylogenetic rooting method that has been shown to work well for rooting microbial gene trees [60]. MAD rooting works by calculating the mean relative deviation from the molecular clock implied by each possible rooting of the unrooted gene tree, and rooting the gene tree at the edge that minimizes this relative deviation [59]. A software implementation of MAD rooting can be freely downloaded from:
`https://www.mikrobio.uni-kiel.de/de/ag-dagan/ressourcen`

RANGER-DTL 2.0: Short for "Rapid ANalysis of Gene family Evolution using Reconciliation – DTL" RANGER-DTL is a software package that implements various algorithms related to DTL reconciliation [5]. RANGER-DTL is based on a parsimony-based model of DTL reconciliation (see Section 4 for a discussion of other models) and has been shown to be both highly accurate and highly efficient/scalable [2, 3, 5, 6, 27, 29, 30]. The RANGER-DTL 2.0 software package consists of several programs out of which our suggested computational protocol makes use of the following two: *Ranger-*

DTL and *AggregateRanger*. Among these, the program *Ranger-DTL* implements the main, parsimony-based DTL reconciliation algorithm; it takes as input a rooted gene tree, rooted species tree, and event costs for duplications, transfers, and losses, and computes a most parsimonious reconciliation, i.e., one with minimum total cost for all duplications, transfers, and losses invoked by the reconciliation. (Speciations are treated as null events and have a cost of 0.) Since there can be multiple most parsimonious reconciliations, each execution of *Ranger-DTL* on the same input may output a different reconciliation, sampled uniformly at random from the space of all most parsimonious reconciliations [3]. The program *AggregateRanger* is designed to merge multiple reconciliations into a single, aggregate reconciliation. Specifically, *AggregateRanger* takes as input the output files created by executing *Ranger-DTL* multiple times on the same gene tree and species tree and outputs a single combined reconciliation with support values for the inferred events and mappings. Thus, *AggregateRanger* can be used to compute support values for individual events and mappings by accounting for reconciliation uncertainty due to the existence of multiple most parsimonious reconciliations and use of alternative event cost assignments. RANGER-DTL 2.0 is open source and can be freely downloaded from:

`https://compbio.engr.uconn.edu/software/ranger-dtl/`

Note that our protocol also makes use of some other, broadly used phylogenetic tools such as RAxML [51] and ModelTest-NG [10].

3 Suggested computational protocol

The computational protocol consists of five sequential steps. For several of these steps, we describe two variants of the protocol for that step: *Variant-1* is suitable for large-scale aggregate analyses over hundreds or thousands of gene families, while *variant-2* is suitable for fine-grained evolutionary analysis of individual gene families. Thus, variant-1 for any step is more scalable and computationally efficient but slightly less

rigorous, while variant-2 is more computationally intensive but also more rigorous.

We assume that the specific species/strains to be included in the analysis have already been chosen and that genes from these chosen species/strains have already been clustered into homologous gene families. We also assume that the gene/protein sequences in each gene family have been aligned using a high-quality multiple sequence aligner [43].

We provide sample program execution commands for some of the steps below. Note that these sample commands are for illustrative purposes only and researchers applying the suggested protocol will need to appropriately modify these commands to apply them to their own datasets.

3.1 Step 1: Species tree estimation

Meaningful application of DTL reconciliation depends on the ability to reconstruct one or more credible and well-supported species trees. Despite the presence of extensive horizontal gene transfer, it is widely accepted that microbial phylogenies representing the evolution of microbial species do exist, even if individual genes from those species do not share the same vertical history [19, 32, 39, 45]. As such, there are two main approaches that are currently used for reconstructing microbial phylogenies. The first approach is to use small-subunit ribosomal RNA (rRNA) genes, e.g., [44, 64], and the second is to use a concatenated alignment of some core genes from the genomes of interest, e.g., [9, 31, 38]. While both these approaches have their drawbacks [13, 14, 17, 22, 23, 33, 39] they are widely used to produce credible estimates of microbial species trees. Some other genome-scale methods have also been proposed for microbial species tree construction, e.g., [7, 49, 54, 63], but these have not been as widely used or tested.

Variant-1. This variant requires the careful reconstruction of only a single species phylogeny. We suggest using either an rRNA based approach or core-gene concatenated-alignment based approach, as described above, to construct a single, credible, well-

supported species tree using established tools, such as RAxML [51], IQ-TREE [41] and PhyML [20], and established best-practices. The resulting tree can be rooted using standard approaches such as outgroup rooting [26, 36, 62] or MAD [59]. If the species tree shows poor branch support values, e.g., bootstrap support scores below 70% for a majority of internal nodes in the tree, then an alternative species tree reconstruction approach may be needed.

Variant-2. For a more conservative analysis that takes into account inference error due to species tree uncertainty, we suggest constructing two or more alternative species trees using both an rRNA based approach and a core-gene concatenated-alignment based approach. Other newer approaches for microbial species tree reconstruction, some of which were cited above, may also be employed. As with variant-1, the goal is to compute credible, well-supported species trees.

3.2 Step 2: Gene tree construction and error-correction

We suggest a two-step approach to constructing the gene trees to be used for DTL reconciliation. Th first step is to infer gene trees using RAxML and the second is to error-correct the RAxML gene trees using TreeFix-DTL. TreeFix-DTL is among the most effective tools for microbial gene tree error-correction [6, 28] and has been extensively evaluated and tested. TreeFix-DTL is also designed to work seamlessly with RAxML; it uses RAxML under the hood to perform likelihood calculations and therefore provides access to all sequence evolution models implemented in RAxML.

Variant-1. In this variant, we carefully construct a single gene tree representative per gene family. We suggest using thorough search settings to first compute a good RAxML tree. A sample RAxML command follows:

```
raxmlHPC -f a -x 12345 -p 12345 -s inputAlignment.fasta
    -m PROTGAMMAJTT -# 100 -n raxmlTree
```

In this command, the model of evolution, as specified using the `-m` option, would need to be adjusted based on the type of sequence data used (nucleotide or amino acid) and on the best fitting nucleotide or amino acid evolutionary model as suggested by a tool such as ModelTest-NG [10]. In general, the `PROTGAMMAJTT` model often works well for amino acid sequences and `GTRGAMMAI` model often works well for nucleotide sequences. The `-f a` option specifies the main algorithm that RaxML will execute, `-x` and `-p` are used to specify arbitrary random seeds to be used during the search, `-s` is used to specify the input sequence alignment file, `-#` controls the thoroughness of the tree search heuristic, and `-n` specifies the suffix to be appended to the name of each output file. Once the RAxML tree has been constructed, it must be further error-corrected using TreeFix-DTL. The RAxML tree provided to TreeFix-DTL for error-correction must be rooted. This rooting can be arbitrary and does not influence TreeFix-DTL output. RaxML itself can be used to perform this rooting as follows:

```
raxmlHPC -f I -t geneFamily1.raxmlTree -m PROTGAMMAJTT
   -n rooted
```

In the command above, the `-f I` option specifies that RaxML should root the unrooted tree passed using the `-t` option.

The resulting tree can then be error-corrected using TreeFix-DTL using the following sample command:

```
treefixDTL -s speciesTree.newick -S smap.txt -A .fasta
   -o .raxmlTree.rooted -n .treefixDTL.tree
   -e "-m PROTGAMMAJTT" -V1 -l treefixDTL.log
    geneFamily1.raxmlTree.rooted
```

In this command, the `-s` option specifies the species tree, `-S` option specifies the mapping from the leaves of the gene tree to the leaves of the species tree, `-A` is used to specify the suffix of the sequence alignment file used to construct the RAxML gene tree, `-o` is used to specify the suffix of the input rooted RAxML gene tree (i.e., the file

name suffix of the tree to be error-corrected), `-e` is used to specify, within quotes, the RAxML evolutionary model under which the input RAxML tree was constructed, `-n` specifies the suffix to be appended to the name of the output file, `-l` species the name of the log file, and the final argument specifies the input rooted RAxML gene tree to be error-corrected. The output of this command is an error-corrected unrooted gene tree in Newick format. A step-by-step tutorial that shows how to install TreeFix-DTL and explains in greater detail how to use it is available from:
`http://compbio.mit.edu/treefix/tutorial.html`

See Note 1 for further discussion on controlling the thoroughness of TreeFix-DTL's tree search.

Variant-2. For a more thorough analysis of individual gene families, we suggest using at least 10 error-corrected gene trees per gene family. Using multiple gene trees helps capture the uncertainty of gene tree construction and, depending on how reconciliation results on these trees are interpreted, can help minimize both false-positive and false-negative evolutionary inferences resulting from gene tree error/uncertainty.

These 10 (or more) error-corrected gene trees can be constructed using the procedure given for Variant-1, but repeated 10 (or more) times using different random seeds when using RAxML (`-x` and `-p` options). Such repetition can be easily automated using simple scripting. For example, Bash scripting can be used to compute 10 initial RAxML trees as follows:

```
for i in {1..1000}; do
raxmlHPC -f a -x $i -p $i -s inputAlignment.fasta
   -m PROTGAMMAJTT -# 100 -n "$i".raxmlTree
done
```

3.3 Step 3: Gene tree rooting

DTL reconciliation requires gene trees to be rooted. Since the result of the previous step is one (or more) *unrooted* error-corrected gene trees, they must first be rooted.

MAD rooting [59] has been shown to be among the most accurate methods for rooting microbial gene trees [60]. The unrooted error-corrected gene tree resulting from the previous step (after applying TreeFix-DTL) does not have branch lengths. Thus, prior to using MAD to root the error-corrected gene trees, one must recompute/reoptimize branch lengths on each gene tree using its original gene family alignment and the same evolutionary model used with RAxML and TreeFix-DTL. This can be easily done using RAxML. A sample command follows:

```
raxmlHPC -f e -t geneFamily1.treefixDTL.tree -s
   geneFamily1.fasta -n branchLengths -m PROTGAMMAJTT
```

MAD can now be easily used to root the resulting gene trees with branch lengths. A sample command follows:

```
mad geneFamily1.treefixDTL.tree.branchLengths -n
```

See Note 2 for a discussion on an alternative rooting method, DTL rooting, that has been shown to work even better than MAD under certain conditions, and Note 3 for a discussion on gene tree rooting versus species tree rooting.

3.4 Step 4: DTL reconciliation

Once the final, rooted gene tree(s) and species tree(s) are available, it is easy to perform DTL reconciliation. As explained in Section 2, each execution of *Ranger-DTL* computes one optimal DTL reconciliation, sampled uniformly at random from the space of all optimal DTL reconciliations for the given gene tree species tree pair. To adequately sample the diversity of possible optimal reconciliations, we therefore recommend executing *Ranger-DTL* 100 times per gene tree/species tree pair and aggregating over the resulting 100 reconciliations.

Variant-1. For each distinct gene tree species tree pair, the first step is to use *Ranger-DTL* to compute the 100 reconciliations and save the resulting output files in a separate directory, and the second step is to use *AggregateRanger* to create a single reconciliation output aggregating the 100 individual reconciliations. It is important to make sure that the leaf labels on the input gene tree and species tree are in accordance with the format required by *Ranger-DTL* (see Note 4) and that the species tree and gene tree to be reconciled are placed in the same input file (species tree on line 1, gene tree on line 2). Once the input file is in the correct format, the following sample Bash script can be used to automatically run *Ranger-DTL* 100 times and then run *AggrerateRanger*. We assume that a directory named "geneFamily1_reconciliation" has been created to save the resulting output files.

```
for i in {1..100}; do
Ranger-DTL --seed $i -i inputFile.newick -o
   geneFamily1_recon/rangerOutput$i
done
AggregateRanger geneFamily1_recon/rangerOutput >>
   geneFamily1_AggregateOutput.txt
```

For *Ranger-DTL*, the `--seed` command is used to specify a starting seed for the random number generator (see Note 5), `-i` specifies the input file containing the species tree and rooted gene tree, and `-o` specifies where the resulting reconciliation output should be saved. For *AggregateRanger*, the first argument specifies the path to and prefix of the 100 reconciliation files to be aggregated, and the >> is used to redirect the output to the specified file.

This script can be easily extended to compute individual "AggregateOutput.txt" files for multiple gene trees per gene family, multiple gene families, and/or multiple species trees through the use of additional nested for loops and variables. The RANGER-DTL 2.0 software package includes a sample Bash script file showing how

to automate the analysis of multiple gene families.

Variant-2. In the above analysis, we use default event costs of 2, 3, and 1 for duplication, transfer, and loss events, respectively. While these costs have been shown to work well in practice, for a more thorough analysis it may be desirable to use two or more different event cost settings and aggregate over the results. We suggest using default cost values for duplications and losses, but two different costs, 3 and 4, for transfer events. To perform such an analysis, the Bash script from variant-1 above can be slightly modified as follows:

```
for i in {1..50}; do
Ranger-DTL --seed $i -i inputFile.newick -o
   geneFamily1_recon/rangerOutput$i
done
for i in {51..100}; do
Ranger-DTL --seed $i -T 4 -i inputFile.newick -o
   geneFamily1_recon/rangerOutput$i
done
AggregateRanger geneFamily1_recon/rangerOutput >>
   geneFamily1_AggregateOutput.txt
```

Note that, since the gene tree species tree pair remains the same, a single run of *AggregateRanger* is able to aggregate over the reconciliations computed with different event costs. See Note 6 for further discussion on customizing event costs and on possible use of distance dependant transfer costs and Note 7 for discussion on a variant of *Ranger-DTL* that can account for unsampled or extinct lineages during reconciliation.

3.5 Step 5: Interpreting reconciliation output

Each run of *Ranger-DTL* outputs a text file showing an optimal reconciliation of the input gene tree and species tree. This reconciliation shows the mapping and event type

for each node of the gene tree to a node of the species tree. Figure 3 shows the output of running *Ranger-DTL* on the gene tree and species tree shown in Figure 1. Observe that the reconciliation in Figure 3, as output by *Ranger-DTL*, corresponds exactly to the reconciliation shown in Figure 2. We also note that, even though each gene tree is mapped to node of the species tree, the interpretation for gene tree nodes labeled as duplications or transfers is that they map to the parent edge of the mapped species node (i.e., those evolutionary events occurred somewhere along the parent edge).

There are often a very large number of optimal reconciliations, each slightly different, for any given gene tree species tree pair. The aggregate reconciliation computed via *AggregateRanger* can be very useful whenever multiple optimal reconciliations exist. In such cases, the aggregate reconciliation can be used to easily identify those aspects of the reconciliation that are either fully or largely conserved across all optimal reconciliations. For example, for the gene tree and species tree of Figure 1, there exists one other equally optimal reconciliation (invoking only two transfer events) when using default event costs. Figure 4 shows the output of running *AggregateRanger* on 100 optimal reconciliation samples for that gene tree species tree pair. This output can be easily used to infer that 5 out of the 6 internal nodes of the gene tree are always mapped consistently across all (only two in this case) optimal reconciliations and that 4 out of the 6 nodes consistently have the same event assignment. In general, it has been observed that, on biological datasets, over 90% of events and over 70% of mappings are inferred consistently (i.e., identically) across all sampled reconciliations [4, 48]. See Note 8 for a discussion on the accuracy of events and mappings inferred by DTL reconciliation.

When using multiple gene trees per gene family, it may be appropriate to trust only those evolutionary inferences that are fully supported by reconciliations for all or most of the gene trees for that gene family.

If the goal of the reconciliation analysis is to infer transfer events on the species

tree, then the aggregate reconciliation files are not useful. Instead, users should write a simple script to extract all those lines in individual *Ranger-DTL* output files that correspond to transfer events. Once all such transfer events have been extracted from all 100 reconciliation samples for each of the gene trees for that gene family, simple post-processing based on the "Mapping" and "Recipient" for each transfer event will yield all distinct transfers of that gene family on the species tree, along with their support values. Transfer events with high support values (i.e., those that are present in all or most of the individual reconciliation files for that gene family) are likely to be correct. Since inference of transfers is highly sensitive to error in the species tree topology, it is beneficial to use multiple species tree candidates and compare the transfers inferred for each.

In some cases it may be useful to visually see the embedding of the gene tree inside the species tree as implied by a computed reconciliation. The reconciliation output from *Ranger-DTL* can be converted into the standard RecPhyloXML format [16] for DTL reconciliations using the *RangerToXML* tool available from `https://compbio.engr.uconn.edu/software/ranger-dtl/`. The converted output can then be visualised using the tool available from `http://phylariane.univ-lyon1.fr/recphyloxml/recphylovisu`.

4 Notes

1. *Adjusting Treefix-DTL thoroughness.* TreeFix-DTL uses a local-search based heuristic approach to search through the space of candidate gene tree topologies. By default, TreeFix-DTL executes 1000 iterations/steps of this local search heuristic. This has been shown to provide a good trade-off between speed and accuracy for gene trees with up to a few hundred leaves [6]. If even greater accuracy is needed, or if the gene tree has more than a few hundred leaves (say greater than 400), then it may be desirable to increase the number of search iterations

to, say, 3000 to 5000 using the `--niter` option. In general, a run of TreeFix-DTL with default parameters takes about 3 times as long as the corresponding RAxML run, and increasing the number of search iteration will correspondingly increase running time.

2. *Alternative gene tree rooting approaches.* DTL rooting, implemented in the *OptRoot* program of RANGER-DTL 2.0 [5], is a technique for rooting gene trees that has been shown to be more accurate than MAD rooting under certain conditions [60]. DTL rooting works by considering all possible rootings of the given unrooted gene tree and finding those rootings that have minimum total DTL reconciliation cost. DTL rooting works very well (especially when a higher-than-default transfer cost is used) when the rate of transfers is low and the gene tree is relatively error-free, but accuracy degrades rapidly as the prevalence of transfers increases and/or gene trees become more error prone [60]. Thus, DTL rooting can be used in place of MAD rooting whenever gene trees are of high quality and the number of transfer events on the gene tree being rooted is, roughly speaking, no more than one-tenth the number of leaves in that gene tree. A sample command to perform DTL rooting appears below:

   ```
   OptRoot -T 5 -r -i geneFamily1.treefixDTL.tree
   ```

   Another recently developed phylogenetic tree rooting method, Minimum Variance (MV) rooting [37], has also been shown to have rooting accuracy almost equivalent to that of MAD on microbial gene families [60].

   MAD rooting and MV rooting both use estimated branch lengths on the input unrooted tree to estimate the most likely root position. However, branch lengths are affected by substitution rate variation along tree edges, which can mislead methods like MAD and MV rooting that depend on branch lengths to estimate root positions. Branch lengths can also be difficult to infer accurately, further affecting the accuracy of these methods. DTL rooting ignores branch lengths and

only use the topology of the input gene tree, along with a rooted species tree, for root identification. Thus, DTL rooting is not misled by substitution rate variation or errors in branch length estimation. However, as described above, DTL reconciliation has been shown to be sensitive to topological errors in the gene tree and to increasing rates of transfer events. MAD rooting and MV rooting, on the other hand, are sensitive to substitution rate variation but have been shown to be quite robust to gene tree reconstruction error and to increasing rates of evolutionary events [60].

3. *Gene tree rooting versus species tree rooting.* Rooting methods that are based on finding the "middle point" or "center" of a phylogenetic tree based on its branch lengths, such as MAD and MV rooting, can be applied to both gene trees and species trees. In contrast, some rooting methods are primarily or exclusively applicable either to gene trees or to species trees. Such methods include DTL rooting, which is designed exclusively for gene tree rooting (see Note 2), and the widely-used outgroup rooting approach [26, 36, 62], which is primarily intended for species tree rooting.

4. *Ranger-DTL input format.* The species tree and gene tree must both be rooted and binary, and leaf labels in the gene and species tree must follow the specific format specified in the user manual for RANGER-DTL 2.0. Internal nodes of the gene and species trees should not be labeled. Branch lengths are allowed, but are not used. Most unexpected errors related to the input are caused by the presence of non-numeric characters (such as 'e') that can sometimes be present in branch lengths; we therefore recommend stripping species trees and gene trees of branch lengths prior to using them with *Ranger-DTL*.

5. *Random seeds for Ranger-DTL. Ranger-DTL* uses system time at the resolution of 1 second as seed for the random number generator. Thus, if multiple instances of *Ranger-DTL* are executed within the same second then they will re-

sult in identical output reconciliations. Thus, we suggest explicitly specifying a different random seed, as shown in the sample command, for each execution of *Ranger-DTL* on the same gene tree species tree pair.

6. *Ranger-DTL event costs and distance dependant transfer costs.* Event costs can be adjusted to customize the reconciliation analysis to the specific characteristics of the dataset being analyzed. We suggest keeping loss and duplication costs at their default values of 1 and 2, respectively. The default transfer cost of 3 has been shown to work well for microbial gene families, but this cost can be increased to 4 or even greater if gene duplication is expected to play a significant role in the evolutionary history of the gene family being analyzed. In essence, higher transfer costs lead to greater utilization of gene duplication and loss as a mechanism to explain gene tree species tree discordance. *Ranger-DTL* also allows for the use of distance dependant transfer costs. This is meant to capture the reality that transfer generally occurs more easily and frequently between more closely related species than between more distantly related ones. Thus, it may make sense to assign a lower cost to tansfers between closely related species and higher costs to transfers between distantly related ones. *Ranger-DTL* provides two different schemes for using such distance-based transfer costs and we refer interested users to the user manual for RANGER-DTL 2.0. We point out, however, that the impact of using distance-based transfer costs has not been systematically studied.

7. *Accounting for extinct and unsampled lineages.* A species tree represents the evolutionary history of a collection of sampled extant species/strains and does not capture the evolutionary history of those lineages, present within that clade, that were either not sampled or that have gone extinct. It is reasonable to expect that such extinct and unsampled species/strains may have engaged in horizontal gene transfer with the species lineages represented on the species tree. In other

words, extinct and unsampled species lineages may have affected the evolutionary history of the gene family under consideration. However, most existing models and implementations of DTL reconciliation, including RANGER-DTL 2.0, do not consider the potential impact of unsampled or extinct species lineages on gene family evolution. There have been efforts to address this limitation by explicitly accounting for unsampled/extinct lineages during DTL reconciliation [24, 56, 61]. Such approaches are based on augmenting the species tree with one or more branches representing unsampled and extinct species lineages, and allowing these lineages to engage in transfer events. While promising, these approaches have not yet been thoroughly tested and a preliminary study using simulated data suggests that accounting for unsampled/extinct lineages may not lead to an overall improvement in reconciliation accuracy. Nonetheless, a prototype version of *Ranger-DTL* that accounts for extinct/unsampled lineages [61], called *Ranger-DTLx* is available from `https://compbio.engr.uconn.edu/ranger-dtlx/`.

8. *Accuracy of DTL reconciliation.* The DTL reconciliation model implemented in *Ranger-DTL* has been shown to be highly accurate, even under high rates of evolutionary events, when error-free (i.e., topologically correct) gene trees and species trees are used [5, 6, 29]. Recall that DTL reconciliation labels each internal node of the gene tree with a mapping to a node of the species tree and an event type (speciation, duplication, or transfer). When considering event type accuracy, these previous studies have found that speciations and transfers are identified with well over 90% accuracy, and duplications with about 85% accuracy, even with high rates of DTL events. When considering mapping accuracy, these studies have found that gene tree nodes labeled as speciation or duplication have a mapping accuracy of well over 90%, even at high rates of DTL events. Mappings for gene tree nodes labeled as transfer events have been found to be

more sensitive to DTL event rates, with mapping accuracy decreasing from about 90% at low DTL rates to about 75% for high DTL rates. As expected, accuracy decreases when the gene trees being reconciled have reconstruction error [6], with both precision and sensitivity decreasing by about 20% for transfers and by smaller amounts for speciations and duplications [6].

5 Conclusion

A detailed understanding of the evolutionary histories of gene families, their spread through horizontal transfer and vertical inheritance, and their relationship to species evolution has many applications in evolutionary biology and comparative genomics. This book chapter provides a step-by-step computational protocol for investigating such questions through the proper application of DTL reconciliation. A defining feature of the suggested computational protocol is its focus on controlling false-positive evolutionary inferences through proper reconstruction of species trees and gene trees and explicit accounting of different sources of inference error and uncertainty. While this protocol is designed around the use of some specific tools, such as RAxML and TreeFix-DTL for gene tree construction and error-correction, MAD for gene tree rooting, and RANGER-DTL 2.0 for DTL reconciliation, there are also other tools that may be appropriate to use as replacements for one or more steps of the overall protocol. For example, Minimum Variance Rooting [37] could be used in place of MAD rooting for rooting gene trees [37, 60] and other popular maximum likelihood based phylogeny inference tools, such as PhyML [20] and IQTree [41] could be used in place of RAxML. Several alternative approaches also exist for gene tree error-correction, including ALE [55], ecceTERA [24, 47], TreeSolve [28], GeneRax [40], and some of these may be appropriate to use instead of TreeFix-DTL, especially if the scalability of TreeFix-DTL becomes a bottleneck. Some probabilistic models of gene family evolution, e.g., [50], can also simultaneously reconstruct gene trees and their reconciliations,

though such models are highly computationally intensive and can only be applied to small datasets. Many different models and software packages also exist for performing DTL reconciliation, with some based on parsimony [2, 3, 5, 8, 11, 12, 15, 18, 21, 24, 27, 35, 47, 48, 52, 57, 58] and some based on probabilistic models of gene family evolution [50, 54–56]. Among these, models implemented in parsimony-based DTL reconciliation software packages, such as NOTUNG [52], ecceTERA [24], and eMPRess [46], may be most compatible with the overall computational protocol described in this chapter.

Existing models of DTL reconciliation have several limitations worth understanding. Perhaps the most important limitation is that DTL reconciliation only models a subset of the evolutionary events/phenomena that may have played a role in the evolution of the chosen gene family, likely leading to the inference of false-positive duplication, transfer, or loss events. While there has been some effort towards incorporating further events in DTL reconciliation, e.g., [1, 29, 52], further modeling, development, and testing may be needed before such models are mature enough for widespread use. Another important limitation is that DTL reconciliation models consider gene families as the "unit" of evolution and do not account for the scale of individual evolutionary events. More advanced reconciliation models that can consider both sub-gene and multi-gene events, e.g., [25,34,53] will likely lead to improved reconciliation accuracy.

Funding: This work was funded in part by US National Science Foundation grants IIS 1553421, MCB 1616514, and IES 1615573 to MSB.

References

[1] Y. ban Chan, V. Ranwez, and C. Scornavacca. Inferring incomplete lineage sorting, duplications, transfers and losses with reconciliations. *Journal of Theoretical Biology*, 432:1–13, 2017.

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

[3] M. S. Bansal, E. J. Alm, and M. Kellis. Reconciliation revisited: Handling multiple optima when reconciling with duplication, transfer, and loss. *Journal of Computational Biology*, 20(10):738–754, 2013.

[4] M. S. Bansal, G. Banay, T. J. Harlow, J. P. Gogarten, and R. Shamir. Systematic inference of highways of horizontal gene transfer in prokaryotes. *Bioinformatics*, 29(5):571–579, 2013.

[5] M. S. Bansal, M. Kellis, M. Kordi, and S. Kundu. RANGER-DTL 2.0: rigorous reconstruction of gene-family evolution by duplication, transfer and loss. *Bioinformatics*, 34(18):3214–3216, 2018.

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

[7] G. Bernard, C. X. Chan, and M. A. Ragan. Alignment-free microbial phylogenomics under scenarios of sequence divergence, genome rearrangement and lateral genetic transfer. *Scientific Reports*, 6:28970, 2016.

[8] Z.-Z. Chen, F. Deng, and L. Wang. Simultaneous identification of duplications, losses, and lateral gene transfers. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 9(5):1515–1528, 2012.

[9] F. D. Ciccarelli, T. Doerks, C. von Mering, C. J. Creevey, B. Snel, and P. Bork. Toward automatic reconstruction of a highly resolved tree of life. *Science*, 311(5765):1283–1287, 2006.

[10] D. Darriba, D. Posada, A. M. Kozlov, A. Stamatakis, B. Morel, and T. Flouri. ModelTest-NG: A New and Scalable Tool for the Selection of DNA and Protein Evolutionary Models. *Molecular Biology and Evolution*, 37(1):291–294, 08 2019.

[11] L. A. David and E. J. Alm. Rapid evolutionary innovation during an archaean genetic expansion. *Nature*, 469:93–96, 2011.

[12] B. Donati, C. Baudet, B. Sinaimeri, P. Crescenzi, and M.-F. Sagot. Eucalypt: efficient tree reconciliation enumerator. *Algorithms for Molecular Biology*, 10(3), 2015.

[13] W. F. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–2128, 1999.

[14] W. F. Doolittle, Y. Boucher, C. L. Nesbo, C. J. Douady, J. O. Andersson, and A. J. Roger. How big is the iceberg of which organellar genes in nuclear genomes are but the tip? *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1429):39–58, 2003.

[15] J.-P. Doyon, C. Scornavacca, K. Y. Gorbunov, G. J. Szöllosi, V. Ranwez, and V. Berry. An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In E. Tannier, editor, *RECOMB-CG*, volume 6398 of *Lecture Notes in Computer Science*, pages 93–108. Springer, 2010.

[16] W. Duchemin, G. Gence, A.-M. Arigon Chifolleau, L. Arvestad, M. S. Bansal, V. Berry, B. Boussau, F. Chevenet, N. Comte, A. A. Davin, C. Dessimoz, D. Dylus, D. Hasic, D. Mallo, R. Planel, D. Posada, C. Scornavacca, G. Szollosi, L. Zhang, E. Tannier, and V. Daubin. RecPhyloXML: a format for reconciled gene trees. *Bioinformatics*, 34(21):3646–3652, 05 2018.

[17] S. R. Gadagkar, M. S. Rosenberg, and S. Kumar. Inferring species phylogenies from multiple genes: Concatenated sequence tree versus consensus gene tree. *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution*, 304B(1):64–74, 2005.

[18] K. Y. Gorbunov and V. A. Liubetskii. Reconstructing genes evolution along a species tree. *Molekuliarnaia Biologiia*, 43(5):946–958, Oct. 2009.

[19] S. Gribaldo and C. Brochier. Phylogeny of prokaryotes: does it exist and why should we care? *Research in Microbiology*, 160(7):513 – 521, 2009.

[20] S. Guindon, J. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel. New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. *Syst. Biol.*, 59(3):307–321, 2010.

[21] J. Haack, E. Zupke, A. Ramirez, Y.-C. Wu, and R. Libeskind-Hadas. Computing the diameter of the space of maximum parsimony reconciliations in the duplication-transfer-loss model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(1):14–22, 2019.

[22] E. Hilario and J. P. Gogarten. Horizontal transfer of {ATPase} genes – the tree of life becomes a net of life. *Biosystems*, 31(2-3):111 – 119, 1993.

[23] R. P. Hirt, J. M. Logsdon, B. Healy, M. W. Dorey, W. F. Doolittle, and T. M. Embley. Microsporidia are related to fungi: Evidence from the largest subunit of rna polymerase ii and other proteins. *Proceedings of the National Academy of Sciences*, 96(2):580–585, 1999.

[24] E. Jacox, C. Chauve, G. J. Szollosi, Y. Ponty, and C. Scornavacca. eccetera: comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics*, 32(13):2056, 2016.

[25] L. Kloub, S. Gosselin, M. Fullmer, J. Graf, J. P. Gogarten, and M. S. Bansal. Systematic Detection of Large-Scale Multigene Horizontal Transfer in Prokaryotes. *Molecular Biology and Evolution*, 38(6):2639–2659, 2021.

[26] A. G. Kluge and J. S. Farris. Quantitative phyletics and the evolution of anurans. *Systematic Biology*, 18(1):1–32, 1969.

[27] M. Kordi and M. S. Bansal. Exact algorithms for duplication-transfer-loss reconciliation with non-binary gene trees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(4):1077–1090, 2019.

[28] M. Kordi and M. S. Bansal. Treesolve: Rapid error-correction of microbial gene trees. In C. Martín-Vide, M. A. Vega-Rodríguez, and T. Wheeler, editors, *Algorithms for Computational Biology*, pages 125–139, Cham, 2020. Springer International Publishing.

[29] M. Kordi, S. Kundu, and M. S. Bansal. 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*, BCB '19, page 514–523, New York, NY, USA, 2019. Association for Computing Machinery.

[30] S. Kundu and M. S. Bansal. On the impact of uncertain gene tree rooting on duplication-transfer-loss reconciliation. *BMC Bioinformatics*, 19(9):290, Aug 2018.

[31] J. M. Lang, A. E. Darling, and J. A. Eisen. Phylogeny of bacterial and archaeal genomes using conserved genes: Supertrees and supermatrices. *PLoS ONE*, 8(4):e62510, 04 2013.

[32] E. Lerat, V. Daubin, and N. A. Moran. From gene trees to organismal phylogeny in prokaryotes:the case of the γ-proteobacteria. *PLoS Biol*, 1(1):e19, 09 2003.

[33] P. O. Lewis, M.-H. Chen, L. Kuo, L. A. Lewis, K. Fucikova, S. Neupane, Y.-B. Wang, and D. Shi. Estimating bayesian phylogenetic information content. *Systematic Biology*, 65(6):1009–1023, 2016.

[34] L. Li and M. S. Bansal. An integrated reconciliation framework for domain, gene, and species level evolution. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(1):63–76, 2019.

[35] R. Libeskind-Hadas, Y.-C. Wu, M. S. Bansal, and M. Kellis. Pareto-optimal phylogenetic tree reconciliation. *Bioinformatics*, 30(12):i87–i95, 2014.

[36] W. P. Maddison, M. J. Donoghue, and D. R. Maddison. Outgroup analysis and parsimony. *Systematic Biology*, 33(1):83–103, 1984.

[37] U. Mai, E. Sayyari, and S. Mirarab. Minimum variance rooting of phylogenetic trees and implications for species tree reconstruction. *PLOS ONE*, 12(8):1–19, 08 2017.

[38] V. M. Markowitz, I.-M. A. Chen, K. Palaniappan, K. Chu, E. Szeto, M. Pillay, A. Ratner, J. Huang, T. Woyke, M. Huntemann, I. Anderson, K. Billis, N. Varghese, K. Mavromatis, A. Pati, N. N. Ivanova, and N. C. Kyrpides. Img 4 version of the integrated microbial genomes comparative analysis system. *Nucleic Acids Research*, 42(D1):D560–D567, 2014.

[39] J. O. McInerney, J. A. Cotton, and D. Pisani. The prokaryotic tree of life: past, present... and future? *Trends in Ecology & Evolution*, 23(5):276 – 281, 2008.

[40] B. Morel, A. M. Kozlov, A. Stamatakis, and G. J. Szollosi. GeneRax: A Tool for Species-Tree-Aware Maximum Likelihood-Based Gene Family Tree Inference under Gene Duplication, Transfer, and Loss. *Molecular Biology and Evolution*, 37(9):2763–2774, 2020.

[41] L.-T. Nguyen, H. A. Schmidt, A. von Haeseler, and B. Q. Minh. Iq-tree: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32(1):268–274, 2015.

[42] T. H. Nguyen, J.-P. Doyon, S. Pointet, A.-M. A. Chifolleau, V. Ranwez, and V. Berry. Accounting for gene tree uncertainties improves gene trees and reconciliation inference. In B. J. Raphael and J. Tang, editors, *WABI*, volume 7534 of *LNCS*, pages 123–134. Springer, 2012.

[43] M. Nute, E. Saleh, and T. Warnow. Evaluating Statistical Multiple Sequence Alignment in Comparison to Other Alignment Methods on Protein Data Sets. *Systematic Biology*, 68(3):396–411, 10 2018.

[44] G. J. Olsen, C. R. Woese, and R. Overbeek. The winds of (evolutionary) change: breathing new life into microbiology. *Journal of Bacteriology*, 176(1):1–6, 1994.

[45] P. Puigbo, Y. I. Wolf, and E. V. Koonin. The tree and net components of prokaryote evolution. *Genome Biology and Evolution*, 2:745–756, 2010.

[46] S. Santichaivekin, Q. Yang, J. Liu, R. Mawhorter, J. Jiang, T. Wesley, Y.-C. Wu, and R. Libeskind-Hadas. eMPRess: a systematic cophylogeny reconciliation tool. *Bioinformatics*, 11 2020. btaa978.

[47] C. Scornavacca, E. Jacox, and G. J. Szollosi. Joint amalgamation of most parsimonious reconciled gene trees. *Bioinformatics*, 31(6):841–848, 2015.

[48] C. Scornavacca, W. Paprotny, V. Berry, and V. Ranwez. Representing a set of reconciliations in a compact way. *Journal of Bioinformatics and Computational Biology*, 11(02):1250025, 2013.

[49] A. Shifman, N. Ninyo, U. Gophna, and S. Snir. Phylo si: a new genome-wide approach for prokaryotic phylogeny. *Nucleic Acids Research*, 42(4):2391–2404, 2014.

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

[51] A. Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

[52] M. Stolzer, H. Lai, M. Xu, D. Sathaye, B. Vernot, and D. Durand. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics*, 28(18):409–415, 2012.

[53] M. Stolzer, K. Siewert, H. Lai, M. Xu, and D. Durand. Event inference in multidomain families with phylogenetic reconciliation. *BMC Bioinformatics*, 16(14):S8, 2015.

[54] G. J. Szollosi, B. Boussau, S. S. Abby, E. Tannier, and V. Daubin. Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. *Proceedings of the National Academy of Sciences*, 109(43):17513–17518, 2012.

[55] G. J. Szollosi, W. Rosikiewicz, B. Boussau, E. Tannier, and V. Daubin. Efficient exploration of the space of reconciled gene trees. *Systematic Biology*, 62(6):901–912, 2013.

[56] G. J. Szollosi, E. Tannier, N. Lartillot, and V. Daubin. Lateral gene transfer from the dead. *Systematic Biology*, 62(3):386–397, 2013.

[57] A. Tofigh. *Using Trees to Capture Reticulate Evolution : Lateral Gene Transfers and Cancer Progression*. PhD thesis, KTH Royal Institute of Technology, 2009.

[58] A. Tofigh, M. T. Hallett, and J. Lagergren. Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 8(2):517–535, 2011.

[59] F. Tria, G. Landan, and T. Dagan. Phylogenetic rooting using minimal ancestor deviation. *Nature Ecology and Evolution*, 1:0193, 2017.

[60] T. Wade, L. T. Rangel, S. Kundu, G. P. Fournier, and M. S. Bansal. Assessing the accuracy of phylogenetic rooting methods on prokaryotic gene families. *PLOS ONE*, 15(5):1–22, 05 2020.

[61] S. Weiner and M. S. Bansal. Improved duplication-transfer-loss reconciliation with extinct and unsampled lineages. *Algorithms*, 14(8), 2021.

[62] W. C. Wheeler. Nucleic acid sequence phylogeny and random outgroups. *Cladistics*, 6(4):363–367, 1990.

[63] C. Whidden, N. Zeh, and R. G. Beiko. Supertrees based on the subtree prune-and-regraft distance. *Systematic Biology*, 2014.

[64] C. R. Woese. Bacterial evolution. *Microbiological Reviews*, 51(2):221–271, 1987.

Figures

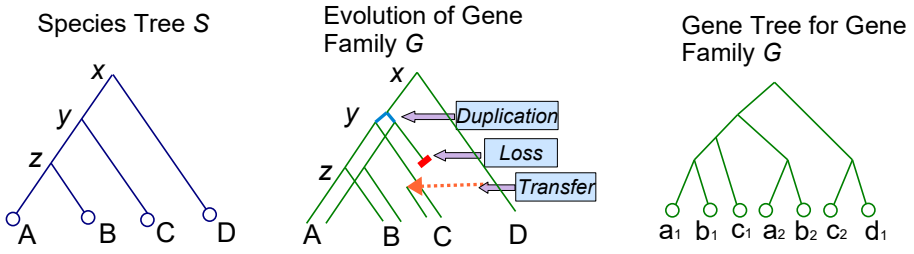

Figure 1: **Gene family evolution.** This figure shows a possible evolutionary history of some gene family G (middle) that evolves along the branches/lineages of the depicted species tree S (left). The gene family G starts as a single gene in the ancestral species represented by the root of S and evolves along the branches of the species tree where it is affected by duplication, transfer, and loss events, in addition to speciation. The tree on the right represents the gene tree topology for G that would result if one were to use all extant homologous gene sequences from gene family G (i.e., all gene from G present in species A, B, C, and D) and reconstruct a gene tree on those gene sequences. The leaves labeled with lower-case letters in the gene tree represent genes sampled from the corresponding upper-case species, e.g., genes a_1 and a_2 represent the two genes from G present in species A.

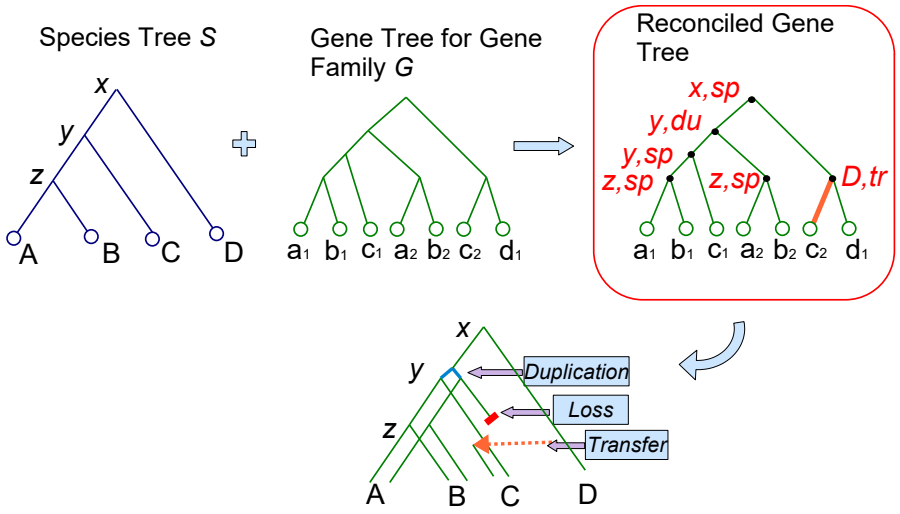

Figure 2: **DTL reconciliation output**. The two trees on the left are the rooted species tree and rooted gene tree being reconciled. The tree on the right, within the red box, depicts the result of applying DTL reconciliation to the trees on the left. Specifically, DTL reconciliation labels each internal node of the gene tree with a mapping to a node of the species tree and an event type (*sp*, *du*, and *tr* for speciation, duplication, and transfer, respectively). The reconciliation also specifies the edges of the gene tree, marked in bold orange, that represent transfer edges. These events and mappings show how the gene tree may have evolved inside the species tree. The reconciliation (i.e., labeled gene tree) shown in this figure implies that gene family G evolved inside the species tree as shown at the bottom.

```
Species Tree:
(((A,B)n3,C)n2,D)n1;

Gene Tree:
((((A_a1,B_b1)m4,C_c1)m3,(A_a2,B_b2)m8)m2,(C_c2,D_d1)m11)m1;

Reconciliation:
A_a1: Leaf Node
B_b1: Leaf Node
m4 = LCA[A_a1, B_b1]: Speciation, Mapping --> n3
C_c1: Leaf Node
m3 = LCA[A_a1, C_c1]: Speciation, Mapping --> n2
A_a2: Leaf Node
B_b2: Leaf Node
m8 = LCA[A_a2, B_b2]: Speciation, Mapping --> n3
m2 = LCA[A_a1, B_b2]: Duplication, Mapping --> n2
C_c2: Leaf Node
D_d1: Leaf Node
m11 = LCA[C_c2, D_d1]: Transfer, Mapping --> D, Recipient --> C
m1 = LCA[A_a1, D_d1]: Speciation, Mapping --> n1

The minimum reconciliation cost is: 6 (Duplications: 1, Transfers: 1, Losses: 1)
Total number of optimal solutions: 2
Total number of candidates for gene birth: 1
```

Figure 3: **A screenshot of *Ranger-DTL* output.** The screenshot shows the result of reconciling the gene tree and species tree shown in Figure 1 using *Ranger-DTL* with default event costs. The top 5 lines show the species tree and gene tree in Newick format, with internal nodes labeled. The reconciliation block shows the event type and mapping for each internal node of the gene tree. For transfer events, the mapping for the recipient species is also given. This reconciliation is identical to the reconciliation shown in Figure 2. The bottom 3 lines provide information about reconciliation cost, total number of optimal reconciliations, and number of possible optimal mappings for the root node of the gene tree.

```
Processed 100 files


Aggregate reconciliation:
A_a1: Leaf Node
B_b1: Leaf Node
m4 = LCA[A_a1, B_b1]: [Speciations = 100, Duplications = 0, Transfers = 0], [Most Frequent mapping --> n3, 100 times].
C_c1: Leaf Node
m3 = LCA[A_a1, C_c1]: [Speciations = 49, Duplications = 0, Transfers = 51], [Most Frequent mapping --> C, 51 times].
A_a2: Leaf Node
B_b2: Leaf Node
m8 = LCA[A_a2, B_b2]: [Speciations = 100, Duplications = 0, Transfers = 0], [Most Frequent mapping --> n3, 100 times].
m2 = LCA[A_a1, B_b2]: [Speciations = 51, Duplications = 49, Transfers = 0], [Most Frequent mapping --> n2, 100 times].
C_c2: Leaf Node
D_d1: Leaf Node
m11 = LCA[C_c2, D_d1]: [Speciations = 0, Duplications = 0, Transfers = 100], [Most Frequent mapping --> D, 100 times].
m1 = LCA[A_a1, D_d1]: [Speciations = 100, Duplications = 0, Transfers = 0], [Most Frequent mapping --> n1, 100 times].


Percentage of events with 100% consistency = 66.6667
Percentage of mappings with 100% consistency = 83.3333
Total number of non-leaf nodes in gene tree: 6
Total number of optimal solutions: 2
```

Figure 4: **Screenshot of *AggregateRanger* output.** The screenshot shows the result of running *AggregateRanger* on 100 optimal reconciliation samples for the gene tree species tree pair of Figure 1. The aggregated reconciliation output shows that 5 out of the 6 internal nodes of the gene tree are mapped consistently across all optimal reconciliations and that 4 out of the 6 internal nodes are assigned consistent event types. In other words, the mapping of one of the nodes, m3, is uncertain, and the event types of two of the nodes, m2 and m3, are uncertain.