TRACTION: Fast Non-Parametric Improvement of Estimated Gene Trees

Sarah Christensen
University of Illinois at Urbana-Champaign, USA
sac2@illinois.edu

Erin K. Molloy
University of Illinois at Urbana-Champaign, USA
emolloy2@illinois.edu

Pranjal Vachaspati
University of Illinois at Urbana-Champaign, USA
vachasp2@illinois.edu

Tandy Warnow[1]
University of Illinois at Urbana-Champaign, USA
http://tandy.cs.illinois.edu
warnow@illinois.edu

Abstract

Gene tree correction aims to improve the accuracy of a gene tree by using computational techniques along with a reference tree (and in some cases available sequence data). It is an active area of research when dealing with gene tree heterogeneity due to duplication and loss (GDL). Here, we study the problem of gene tree correction where gene tree heterogeneity is instead due to incomplete lineage sorting (ILS, a common problem in eukaryotic phylogenetics) and horizontal gene transfer (HGT, a common problem in bacterial phylogenetics). We introduce TRACTION, a simple polynomial time method that provably finds an optimal solution to the RF-Optimal Tree Refinement and Completion Problem, which seeks a refinement and completion of an input tree t with respect to a given binary tree T so as to minimize the Robinson-Foulds (RF) distance. We present the results of an extensive simulation study evaluating TRACTION within gene tree correction pipelines on 68,000 estimated gene trees, using estimated species trees as reference trees. We explore accuracy under conditions with varying levels of gene tree heterogeneity due to ILS and HGT. We show that TRACTION matches or improves the accuracy of well-established methods from the GDL literature under conditions with HGT and ILS, and ties for best under the ILS-only conditions. Furthermore, TRACTION ties for fastest on these datasets. TRACTION is available at `https://github.com/pranjalv123/TRACTION-RF` and the study datasets are available at `https://doi.org/10.13012/B2IDB-1747658_V1`.

2012 ACM Subject Classification Applied computing → Molecular evolution; Applied computing → Population genetics

Keywords and phrases Gene tree correction, horizontal gene transfer, incomplete lineage sorting

Digital Object Identifier 10.4230/LIPIcs.WABI.2019.4

Funding *Sarah Christensen*: Ira & Debra Cohen Fellowship
Erin K. Molloy: NSF Graduate Research Fellowship Grant Number DGE-1144245 and Ira & Debra Cohen Fellowship
Pranjal Vachaspati: NSF Graduate Research Fellowship Grant Number DGE-1144245
Tandy Warnow: NSF CCF-1535977

Acknowledgements We thank Mike Steel for encouragement and the members of the Warnow lab for valuable feedback. This study was performed on the Illinois Campus Cluster and Blue Waters, a computing resource that is operated and financially supported by UIUC in conjunction with the National Center for Supercomputing Applications.

[1] Corresponding author

1 Introduction

Reconstructing the evolutionary history of a gene is a core task in phylogenetics, and our ability to infer these evolutionary relationships accurately can have important implications for a variety of downstream analyses. For example, estimated gene trees are used in the inference of adaptation, evolutionary event detection (such as gene loss, gene duplication, and horizontal gene transfer), ortholog identification, analysis of functional trait evolution, and species tree estimation. However, unlike species tree estimation techniques that leverage information encoded across the entire genome, gene tree estimation based on a single locus may not contain enough signal to determine the correct gene tree topology with high confidence [27]. Indeed, many phylogenomic datasets have gene trees with average branch support well below 75%, which is a common lower bound for branches to be considered reliable. For example, the Avian Phylogenomic Project [17] reported average branch support values below 30%, and many other studies (surveyed in [25]) have had similar challenges. Estimating gene and species trees is further complicated by biological processes such as gene duplication/loss (GDL), incomplete lineage sorting (ILS), and horizontal gene transfer (HGT), that create heterogeneous tree topologies across the genome [21]. HGT has long been known to cause problems for bacterial phylogenetics, and ILS by itself has emerged as a major issue in phylogenomics, affecting most, if not all, genome-scale datasets [10].

Because gene trees often have low accuracy, a natural problem is to try to improve gene tree estimation using an estimated or known species tree. An approach from the gene duplication and loss literature is to modify estimated gene trees with respect to a reference species tree, which may either be an established tree from prior literature or an estimated species tree (e.g., based on an assembled multi-locus dataset). Some of these methods use the available sequence data as well as the estimated gene tree and species tree, and are referred to as "integrative methods"; examples include ProfileNJ [27], TreeFix [33], and TreeFix-DTL [2]. Other methods, called "gene tree correction methods", use just the topologies of the gene tree and species tree, and are typically based on parametric models of gene evolution; Notung [6, 9] and ecceTERA [16] are two of the well known methods of this type. Integrative methods are generally expected to be more accurate than gene tree correction methods when gene tree heterogeneity is due to GDL, but as a result of using likelihood calculations they are also more computationally intensive. See [4, 26, 31, 18, 15, 16, 34] for an entry into the vast literature on this subject.

Here, we examine the problem of gene tree correction for the case where gene tree heterogeneity is due to ILS or HGT, and where each gene tree has at most one copy of each species. We present a new approach to gene tree correction that is based on a very simple *non-parametric* polynomial-time method, TRACTION, that is *agnostic* to the cause of gene tree heterogeneity. In addition to correcting gene trees, TRACTION is also capable of completing gene trees that do not contain all the species present in the reference species tree, a condition that may occur in a multi-locus study as a result of taxon sampling strategies or unavailable data (such as when not all genomes have been sequenced and assembled). The input to TRACTION is a pair (t, T) of unrooted phylogenetic trees. The leaf set of t is a subset of the leaf set of T, tree T is binary, and tree t will generally be non-binary. We seek a tree T' created by refining t and adding any missing leaves so that T' has the minimum Robinson-Foulds (RF) [28] distance to T. We call this optimization problem the "RF-Optimal Tree Refinement and Completion Problem" (RF-OTRC). We show that TRACTION finds an optimal solution to this problem in $O(n^{1.5} \log n)$ time, where n is the number of leaves in the species tree T. To use TRACTION for gene tree correction, we

assume we are given an estimated gene tree with branch support values and an estimated (or known) binary species tree, which may have additional species. The low support branches in the gene tree are collapsed, forming the (unresolved) tree t. TRACTION has two steps: first it refines the input gene tree t into a binary tree t', and then it adds the missing species to t'. Although the algorithm is quite simple, the proof of correctness is non-trivial. We present the results of an extensive simulation study (on 68,000 gene trees, each with up to 51 species) in which gene tree heterogeneity is either due to only ILS or to both ILS and HGT. We explore TRACTION for gene tree correction with estimated species trees in comparison to Notung, ecceTERA, ProfileNJ, TreeFix, and TreeFix-DTL, evaluating the accuracy of the corrected gene trees by computing the RF distance to the true gene tree. Overall, while many methods (including TRACTION) tie for best on the ILS-only data, TRACTION dominates the other gene tree correction methods with respect to topological accuracy on the HGT+ILS data, and ties for fastest. Importantly, TRACTION provides good accuracy even when the estimated species tree is far from the true gene tree. The simplicity of the approach and its good accuracy under a range of different model conditions indicates that non-parametric approaches to gene tree correction may be promising, and encourages future research.

2 TRACTION

2.1 Terminology and Basics

Each edge e in an unrooted phylogenetic tree defines a *bipartition* π_e (also referred to as a *split*) on the leaves of the tree induced by the deletion of e (but not its endpoints). Each bipartition of the leaf set into two non-empty disjoint parts, A and B, is denoted by $A|B$. The set of bipartitions of a tree T is given by $C(T) = \{\pi_e : e \in E(T)\}$, where $E(T)$ is the edge set for T. Tree T' is a *refinement* of T if T can be obtained from T' by contracting a set of edges in $E(T')$. A tree T is *fully resolved* (i.e., binary) if there is no tree that refines T other than itself.

A set Y of bipartitions on some leaf set S is *compatible* if there exists an unrooted tree T leaf-labelled by S such that $Y \subseteq C(T)$; furthermore, when a set of bipartitions is compatible then there is a *unique* tree such that $Y = C(T)$. In addition, pairwise compatibility of a set of bipartitions ensures setwise compatibility [11, 12]. A bipartition π_e of a set S is said to be compatible with a tree T with leaf set S if and only if there is a tree T' such that $C(T') = C(T) \cup \{\pi\}$ (i.e., T' is a refinement of T that includes the bipartition π). Similarly, two trees on the same leaf set are said to be compatible if they share a common refinement; it then follows that two trees are compatible if and only if the union of their sets of bipartitions is compatible.

Given a phylogenetic tree T on taxon set S, T *restricted to* $R \subseteq S$ is the minimal subgraph of T connecting elements of R and suppressing nodes of degree two. We denote this as $T|_R$. If T and T' are two trees with R as the intersection of their leaf sets, their *shared edges* are edges whose bipartitions restricted to R are in the set $C(T|_R) \cap C(T'|_R)$. Correspondingly, their *unique edges* are edges whose bipartitions restricted to R are not in the set $C(T|_R) \cap C(T'|_R)$. See Figure 1 for a pictorial depiction of unique and shared edges.

The *Robinson-Foulds* (RF) distance [28] between two trees T and T' on the same set of leaves is the number of bipartitions present in only one tree; equivalently, the RF distance is equal to the total number of unique edges in both trees. The normalized RF distance is the RF distance divided by $2n - 6$, where n is the number of leaves in each tree; this produces a value between 0 and 1 since the two trees can only disagree with respect to internal edges and $n - 3$ is the maximum number of internal edges in an unrooted tree with n leaves.

2.2 RF-Optimal Tree Refinement and Completion (RF-OTRC) Problem

Input: An unrooted binary tree T on S and an unrooted tree t on $R \subseteq S$

Output: An unrooted binary tree T' on S with two key properties: (1) T' contains all the leaves of S and is compatible with t (i.e., $T'|_R$ is a refinement of t) and (2) T' minimizes the RF distance to T among all binary trees satisfying condition (1).

If the trees t and T have the same set of taxa, then the RF-OTRC problem becomes the RF-OTR (RF-Optimal Tree Refinement) problem, while if t is already binary but can be missing taxa, then the RF-OTRC problem becomes the RF-OTC (RF-Optimal Tree Completion) problem. OCTAL, presented in [7], solves the RF-OTC problem in $O(n^2)$ time, and an improved approach presented by Bansal [1] solves the RF-OTC problem in linear time. We refer to this faster approach as **Bansal's algorithm**. In this paper we present an algorithm that solves the RF-OTR problem exactly in polynomial time and show that the combination of this algorithm with Bansal's algorithm solves the RF-OTRC problem exactly in $O(n^{1.5} \log n)$ time, where T has n leaves. We refer to the two steps together as TRACTION (Tree Refinement And CompleTION).

2.3 TRACTION Algorithm

The input to TRACTION is a pair of unrooted trees (t, T), where t is the estimated gene tree on set R of species and T is the binary reference tree on S, with $R \subseteq S$. We note that t may not be binary (e.g., if low support edges have already been collapsed) and may be missing species (i.e., $R \subset S$ is possible).

- Step 1: Refine t so as to produce a binary tree t^* with as many shared bipartitions with T as possible (using a polynomial time algorithm described below).
- Step 2: Add the missing species from T into t^*, minimizing the RF distance.

2.3.1 Step 1: Greedy Refinement of t

To compute t^*, we first refine t by adding all bipartitions from $T|_R$ that are compatible with t; this produces a unique tree t'. If t' is not fully resolved, then there are multiple optimal solutions to the RF-OTR problem, as we will later prove. The algorithm selects one of these optimal solutions as follows. First, we add edges from t that were previously collapsed (if such edges are available). Next, we randomly refine the tree until we obtain a fully resolved refinement, t^*. Note that if t' is not binary, then t^* is not unique. We now show that the first step of TRACTION solves the RF-OTR problem.

▶ **Theorem 1.** *Let T be an unrooted binary tree on leaf set S, and let t be an unrooted tree on leaf set $R \subseteq S$. A fully resolved (i.e. binary) refinement of t minimizes the RF distance to $T|_R$ if and only if it includes all compatible bipartitions from $T|_R$.*

Proof. Let C_0 denote the set of bipartitions in $T|_R$ that are compatible with t. By the theoretical properties of compatible bipartitions (Section 2.1), this means the set $C_0 \cup C(t)$ is a compatible set of bipartitions that define a unique tree t' where $C(t') = C_0 \cup C(t)$. We now prove that for any binary tree B that refines t, B minimizes the RF distance to $T|_R$ if and only if B refines t'.

Consider a sequence of trees $t = t_0, t_1, t_2, \ldots, t_k$, each on leaf set R, where t_i is obtained from t_{i-1} by adding one edge to t_{i-1}, and thus adds one bipartition to $C(t_{i-1})$. Let $\delta_i = RF(t_i, T|_R) - RF(t_{i-1}, T|_R)$, so that δ_i indicates the change in RF distance produced by adding a specific edge to t_{i-1} to get t_i. Hence,

$$RF(t_i, T|_R) = RF(t_0, T|_R) + \sum_{j \leq i} \delta_j.$$

Since T is fully resolved, a new bipartition π_i added to $C(t_{i-1})$ is in $C(T|_R)$ if and only if $\pi_i \in C_0$. If this is the case, then the RF distance will decrease by one (i.e., $\delta_i = -1$). Otherwise, $\pi_i \notin C_0$, and the RF distance to $T|_R$ will increase by one (i.e., $\delta_i = 1$).

Now suppose B is a binary refinement of t. We split the bipartitions in $C(B) \setminus C(t)$ into two sets, X and Y, where X are bipartitions in C_0 and Y are bipartitions not in C_0. By the argument just provided, it follows that $RF(B, T|_R) = RF(t, T|_R) - |X| + |Y|$. Note that $|X \cup Y|$ must be the same for all binary refinements of t, because all binary refinements of t have the same number of edges. Thus, $RF(B, T|_R)$ is minimized when $|X|$ is maximized, so B minimizes the RF distance to $T|_R$ if and only if $C(B)$ contains all the bipartitions in C_0. In other words, $RF(B, T|_R)$ is minimized if and only if B refines t'. ◀

▶ **Corollary 2.** *TRACTION finds an optimal solution to the RF-OTR problem.*

Proof. Given input gene tree t and reference tree T, both on the same leaf set, TRACTION produces a tree t'' that refines t and contains every bipartition in T that is compatible with t; hence by Theorem 1, TRACTION solves the RF-OTR problem. ◀

2.3.2 Step 2: Adding in missing species

The second step of TRACTION can be performed using OCTAL or Bansal's algorithm, each of which finds an optimal solution to the RF-OTC problem in polynomial time. More generally, we show that any method that optimally solves the RF-OTC problem can be used as an intermediate step to solve the RF-OTRC problem.

To prove this, we first restate several prior theoretical results. In the proof of correctness for OCTAL, [7] showed the minimum achievable RF distance between T and T' is given by:

$$RF(T, T') = RF(T|_R, t) + 2m \tag{1}$$

where m is the number of Type II superleaves in T relative to t, which we now define.

▶ **Definition 3.** *Let T be a binary tree on leaf set S and t be a tree on leaf set $R \subseteq S$. The superleaves of T with respect to t are defined as follows (see Figure 1). The set of edges in T that are on a path between two leaves in R define the backbone; when this backbone is removed, the remainder of T breaks into pieces. The components of this graph that contain vertices from $S \setminus R$ are the superleaves. Each superleaf is rooted at the node that was incident to one of the edges in the backbone, and is one of two types:*

- *Type I superleaves: the edge e in the backbone to which the superleaf was attached is a shared edge in $T|_R$ and t*
- *Type II superleaves: the edge e in the backbone to which the superleaf was attached is a unique edge in $T|_R$ and t*

▶ **Theorem 4** (Restatement of Theorem in [7]). *Given unrooted binary trees t and T with the leaf set of t a subset of the leaf set S of T, OCTAL(T, t) solves the RF-OTC problem and runs in $O(n^2)$ time, where T has n leaves.*

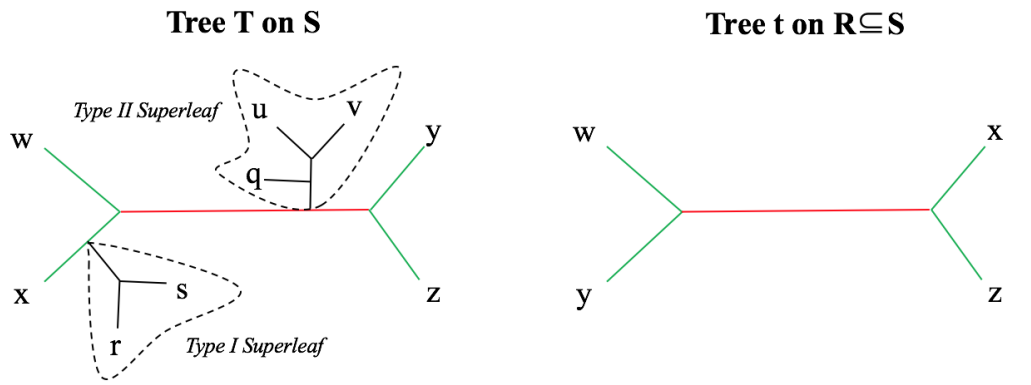

Figure 1 Type I and Type II superleaves of a tree T with respect to t. Edges in the backbone (defined to be the edges on paths between nodes in the common leaf set) are colored green for shared and red for unique; all other edges are colored black. The deletion of the backbone edges in T defines the superleaves; one is a Type I superleaf because it is attached to a shared (green) edge and the other is a Type II superleaf because it is attached to a unique (red) edge. This figure is taken from [7].

2.4 Proof of correctness for TRACTION

▶ **Theorem 5.** *Let T be an unrooted binary tree on leaf set S with $|S| = n$, and let t be an unrooted tree on leaf set $R \subseteq S$. TRACTION returns a binary unrooted tree T' on leaf set S such that $RF(T', T)$ is minimized subject to $T'|_R$ refines t. The first stage (refining t) takes $O(|S| + |R|^{1.5} \log(|R|))$ time. Hence, TRACTION runs in $O(n^{1.5} \log n)$ time if used with Bansal's algorithm and $O(n^2)$ time if used with OCTAL.*

Proof. By construction *TRACTION* outputs a tree T' that, when restricted to the leaf set of t, is a refinement of t. Hence, it is clear that $T'|_R$ refines t. Now, it is only necessary to prove that RF(T', T) is minimized by *TRACTION*. Since the intermediate tree t^* produced in the first step of TRACTION is binary, Theorem 4 gives that TRACTION using OCTAL (or any method exactly solving the RF-OTC problem) will add leaves to t^* in such a way as to minimize the RF distance to T.

As given in Equation 1, the optimal RF distance between T' and T is the sum of two terms: 1) RF(t^*, $T|_R$) and 2) the number of Type II superleaves in T relative to t^*. Theorem 1 shows that TRACTION produces a refinement t^* that minimizes the first term. All that remains to be shown is that t^* is a binary refinement of t minimizing the number of Type II superleaves in T relative to t^*.

Consider a superleaf X in T with respect to t. If t were already binary, then every superleaf X is either a Type I or a Type II superleaf. Also, note that every Type I superleaf in T with respect to t will be a Type I superleaf for any refinement of t. However, when t is not binary, it is possible for a superleaf X in T to be a Type II superleaf with respect to t but a Type I superleaf with respect to a refinement of t. This happens when the refinement of t introduces a new shared edge with T to which the superleaf X is attached in T. Notice that since the set of all possible shared edges that could be created by refining t is compatible, any refinement that maximizes the number of shared edges with T also minimizes the number of Type II superleaves. Theorem 1 shows that *TRACTION* produces such a refinement t^* of t. Thus, *TRACTION* finds a binary unrooted tree T' on leaf set S such that RF(T', T) is minimized subject to the requirement that $T'|_R$ refine t.

We now analyze the running time, focusing on the first stage. Constructing $T|_R$ takes $O(|S|)$ time. Checking compatibility of a single bipartition with a tree on K leaves, and then adding the bipartition to the tree if compatible, can be performed in only $O(|K|^{0.5}\log(|K|))$ after a fast preprocessing step (see Lemmas 3 and 4 from [14]). Hence, determining the set of edges of $T|_R$ that are compatible with t takes only $O(|S|+|R|^{1.5}\log(|R|))$ time. Therefore, the first stage of TRACTION takes $O(|S|+|R|^{1.5}\log(|R|))$ time. Hence, if used with OCTAL, TRACTION takes $O(|S|^2)$ time and if used with Bansal's algorithm TRACTION takes $O(|S|^{1.5}\log|S|)$ time. ◀

3 Evaluation

Overview. We evaluated TRACTION in comparison to Notung, ecceTERA, profileNJ, TreeFix, and TreeFix-DTL on estimated gene trees under two different model conditions (ILS-only and ILS+HGT), using estimated species trees. In total, we analyzed 68,000 genes: 8,000 with 26 species under ILS-only models and 60,000 with 51 species under ILS+HGT models. All estimated gene trees we correct in these experiments were complete (i.e., have all the species). The motivation for this is two-fold. First, the methods we benchmark against do not provide an option for completing gene trees with missing data. This is understandable since these methods were developed for GDL, where missing species in a gene tree are interpreted as true loss events rather than incomplete sampling. Second, an experimental evaluation of OCTAL, the algorithm that performs the completion step of TRACTION, was previously performed in [7].

Datasets. We briefly describe the datasets used in this study (all are from prior studies [8, 7] and available online). The datasets include single copy genes with 26 or 51 species (each with a known outgroup), and were generated under model conditions where true gene trees and true species trees differed due to only ILS (two levels of ILS) or to both ILS and HGT (one ILS level but two levels of HGT). The true gene tree heterogeneity (**GT-HET**, the topological distance between true species trees and true gene trees) ranged from 10% (for the ILS-only condition with moderate ILS) to as high as 68% (for the ILS+HGT condition with high HGT). Each model condition has 200 genes, and we explore multiple replicates per model condition with different sequence lengths per gene. See Table 1 for details.

Estimated gene trees and estimated reference species trees. For each gene, we used RAxML v8.2.11 [29] under the GTRGAMMA model to produce maximum likelihood gene trees, with branch support computed using bootstrapping. Because sequence lengths varied, this produced estimated gene trees with different levels of gene tree estimation error **(GTEE)** (defined to be the average RF distance between the true gene tree and the estimated gene tree), ranging from 32% to 63% as defined by the missing branch rate (see Table 1). We estimated a species tree using ASTRID v1.4 [32] on the RAxML gene trees. Since we know the true outgroup for all species trees and gene trees, we used this in our performance study to root the input species tree and the estimated gene trees, given as input to all methods.

The gene trees given as input to the different methods were computed as follows. Each edge in each RAxML gene tree we computed was annotated with its bootstrap branch support, and we identified all the branches with bootstrap support less than 75% (the standard threshold for "low support"). Low support branches were collapsed in the gene trees before being given to TRACTION, Notung, and ProfileNJ. When we ran ecceTERA, we gave the binary gene trees with the threshold value (i.e., minimum required bootstrap support

Table 1 Empirical properties of the simulated datasets used in this study: GT-HET (gene tree heterogeneity, the average normalized RF distance between true gene trees and true species trees); GTEE (average gene tree estimation error); and the average distance of the ASTRID reference tree to the true gene trees. The publications from which the simulated datasets are taken are also indicated. In total we analyzed 68,000 genes with varying levels and causes of true gene tree heterogeneity (to the true species tree) and gene tree estimation error. The ILS-only conditions each had 20 replicates, and the ILS+HGT conditions each had 50 replicates.

	GT-HET	GTEE	Distance ASTRID to true gene trees
			ILS-only, Low ILS, 26 species [7]
# sites varies	0.10	0.32	0.08
			ILS-only, High ILS, 26 species [7]
# sites varies	0.36	0.40	0.33
			ILS+HGT, Moderate HGT (m5), 51 species [8]
100 sites	0.54	0.63	0.55
250 sites	0.54	0.47	0.55
500 sites	0.54	0.47	0.54
			ILS+HGT, High HGT (m6), 51 species [8]
100 sites	0.68	0.62	0.68
250 sites	0.68	0.46	0.68
500 sites	0.68	0.38	0.68

value) of 75%; ecceTERA then collapses all branches that have support less than 75%, and explores the set of refinements. Thus, the protocol we followed ensures that ecceTERA, ProfileNJ, Notung, and TRACTION all used the same set of collapsed gene trees. TreeFix and Treefix-DTL used the uncollapsed gene trees.

Gene tree correction and integrative methods. The RAxML gene trees were corrected using TRACTION v1.0, Notung v2.9, ecceTERA v1.2.4, ProfileNJ (as retrieved from GitHub after the March 20, 2018 commit with ID 560b8b2) [27], TreeFix v1.1.10 (for the ILS-only datasets), and TreeFix-DTL v1.0.2 (for the HGT+ILS datasets), each with a species tree estimated using ASTRID v1.4 [32] as the reference tree rooted at the outgroup. The integrative methods (TreeFix, TreeFix-DTL, and ProfileNJ) also required additional input data related to the gene alignments, which we detail in the appendix. All estimated gene trees are complete (i.e., there are no missing taxa), so TRACTION only refines the estimated gene tree and does not add any taxa.

Evaluation criteria. We use RF tree error (the standard criterion in performance studies evaluating phylogeny estimation methods) to quantify error in estimated and corrected gene trees as compared to the known true gene tree (as defined in the simulation protocol) and the impact of TRACTION, Notung, ecceTERA, and TreeFix-DTL, on these errors. Note that although we use the RF distance within the OTR optimization criterion, there it refers to the distance between the corrected gene tree and the reference tree (which is an *estimated species tree*); in contrast, when we use the RF error rate in the evaluation criterion, in that context it refers to the distance between the corrected gene tree and the true *gene tree*. Since

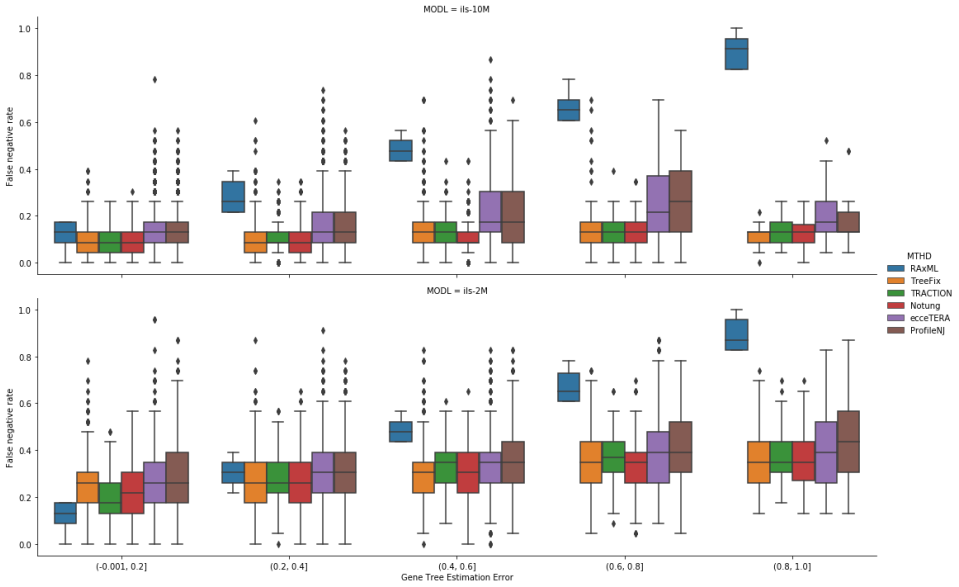

Figure 2 Comparison of methods on the ILS-only datasets with respect to average RF error rate, as a function of GTEE. Results are only shown for those datasets on which all methods complete. Each model condition (characterized by ILS level) has 20 replicate datasets, each with 200 genes.

the reference trees used in our experiments are typically very topologically different from the true gene tree (8% RF distance for the moderate ILS condition, 33% for the high ILS condition, 54% to 68% for the ILS+HGT conditions, see Table 1), optimizing the RF distance to the reference tree is quite different from optimizing the RF distance to the true gene tree.

Experiments. We performed two main experiments: one in which we explored performance on ILS-only datasets and the other in which we explored performance on datasets with HGT and ILS. In each case, we directly explored how the GTEE level impacted absolute and relative accuracy of the gene tree correction methods. We also indirectly explored how GT-HET affects relative and absolute accuracy. The heterogeneity is higher on the HGT+ILS datasets than on the ILS-only datasets, as HGT adds to the heterogeneity between gene trees and species trees (see Table 1).

4 Results and Discussion

Experiment 1: Comparison of methods on ILS-only datasets. Not all methods completed on all datasets: ecceTERA failed to complete on 67 datasets, profileNJ failed to complete on two datasets, and all other methods completed on all datasets. Results shown in Figure 2 are restricted to those datasets on which all methods completed; see the Appendix for additional results. For the moderate ILS condition (Figure 2 (top)), all methods are able to improve on RAxML, and the degree of improvement increases with GTEE. For the high ILS condition (Figure 2 (bottom)), methods improve on RAxML only when GTEE is at least 20%. Thus, GTEE and ILS level both impact whether methods can improve on RAxML. Furthermore, the methods group into two sets: TRACTION, Notung, and TreeFix performing very similarly and ProfileNJ and ecceTERA having somewhat higher error.

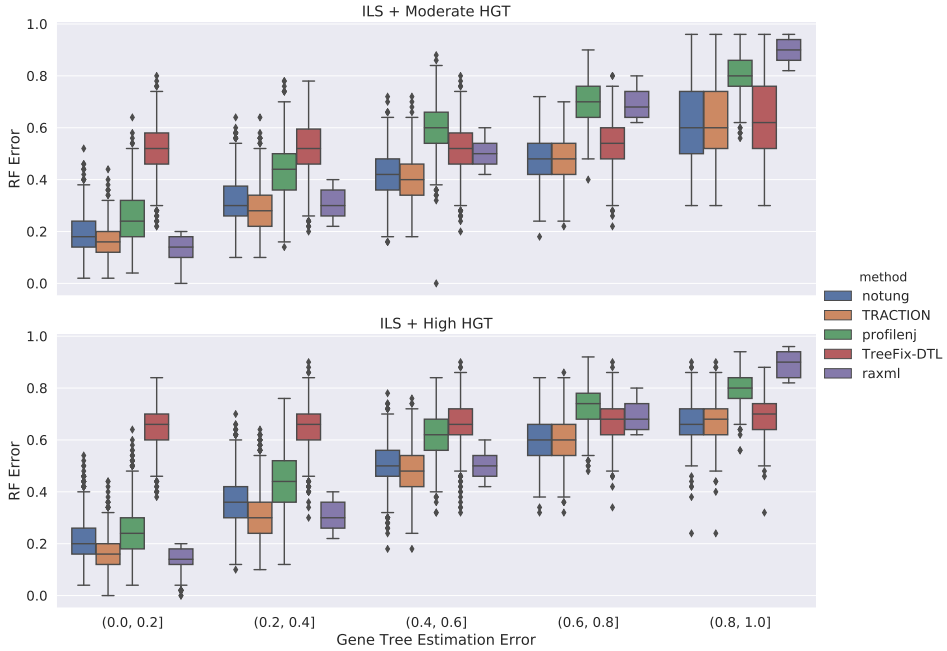

Figure 3 Comparison of methods on ILS+HGT datasets with respect to average RF error rate (max is 1.0), as a function of GTEE (gene tree estimation error of the RAxML gene trees); ecceTERA failed on many datasets (with increasing failure rate as GTEE increases), and so those results are not shown. Results shown here are for only those datasets on which all methods completed.

Experiment 2: Comparison of methods on the HGT+ILS datasets. The HGT+ILS datasets have heterogeneity due to both HGT and ILS, with the degree of HGT varying from moderate (m5) to high (m6). On these data, ecceTERA failed on 1,318 datasets with the failure rates increasing as the gene tree estimation error of the initial RAxML gene tree (GTEE) increased: it failed 0% of the time when GTEE is less than 40%, 0.4% of the time when GTEE is 40-60%, 23.6% of the time when GTEE is 60-80%, and 90.8% of the time when GTEE is at least 80%. Because of the high failure rate, we do not report results for ecceTERA under these conditions. Figure 3 shows the impact of the remaining methods on RAxML gene trees as a function of GTEE. The relative performance between the remaining methods show that TRACTION and Notung are more accurate than profileNJ and TreeFix-DTL, with the gap between the two groups increasing with GTEE. We also see that TRACTION has an advantage over Notung for the low GTEE condition and matches the accuracy on the higher GTEE conditions. Finally, for the lowest GTEE bin, no method improves the RAxML gene tree, some methods make the gene trees much less accurate (e.g., profileNJ), and only TRACTION maintains the accuracy of the RAxML gene tree. Overall, on the HGT+ILS datasets, TRACTION consistently does well and has a clear advantage over the other methods in terms of accuracy.

Running Times. We selected a random sample of the 51-taxon HGT+ILS datasets to evaluate the running time (see Table 2). From fastest to slowest, the average running times were 0.5 seconds for TRACTION, 0.8 seconds for Notung, 1.7 seconds for ProfileNJ, 3.8 seconds for TreeFix-DTL, and 29 seconds for ecceTERA. Furthermore, most of the methods had consistent running times from one gene to another, but ecceTERA had high variability,

depending on the size of the largest polytomy. When the largest polytomy was relatively small, it completed in just a few seconds, but it took close to a minute when the largest polytomy had a size at the limit of 12. Results on other HGT+ILS replicates and model conditions gave very similar results.

Table 2 Total time (in seconds) for each method to correct 50 gene trees with 51 species on one replicate (label 01) of the HGT+ILS dataset with moderate HGT and sequences of length 100bp.

Method	Time (s)
EcceTERA	1470
NOTUNG	43
TRACTION	30
ProfileNJ	87
TreeFix-DTL	188

Overall comments. This study shows that the better methods for gene tree correction (TRACTION, Notung, and TreeFix) reliably produce more accurate gene trees than the initial RAxML gene trees for the ILS-only conditions (except for cases where the initial gene tree is already very accurate), and the improvement can be very large when the initial gene trees are poorly estimated. However, the impact of gene tree correction is reduced for the HGT+ILS scenarios, where improvement over the initial gene tree is only obtained when GTEE is fairly high. As shown in Table 1, the average normalized RF distance between the reference tree (ASTRID) and the true gene trees is never more than 33% for the ILS-only scenarios but very high for the HGT+ILS scenarios (54% for moderate HGT and 68% for high HGT). Since the reference tree is the basis for the correction of the gene trees, it is not surprising that improvements in accuracy are difficult to obtain for the HGT+ILS scenario. On the other hand, given the large distance between the reference tree and the true gene tree, the fact that improvements are obtained for several methods (TRACTION, Notung, and TreeFix-DTL) is encouraging.

5 Conclusions

We presented TRACTION, a method that solves the RF-OTRC problem exactly in $O(n^{1.5} \log n)$ time, where n is the number of species in the species tree; the algorithm itself is very simple, but the proof of optimality is non-trivial. TRACTION performs well, matching or improving on the accuracy of competing methods on the ILS-only datasets and dominating the other methods on the HGT+ILS datasets. Furthermore, although all the methods are reasonably fast on these datasets, TRACTION is the fastest on the 51-taxon gene trees, with Notung a close second.

The observation that TRACTION performs as well (or better) than the competing methods (ecceTERA, ProfileNJ, Notung, TreeFix, and TreeFix-DTL) is encouraging. However, the competing methods are all based on stochastic models of gene evolution that are inherently derived from gene duplication and loss (GDL) scenarios (and in one case also allowing for HGT), and thus it is not surprising that GDL-based methods do not provide the best accuracy on the ILS-only or HGT+ILS model conditions we explore (and to our knowledge, all the current methods for gene tree correction are based on GDL models). Yet, TRACTION has good accuracy under a wide range of scenarios. We conjecture that this generally good performance is the result of its non-parametric criterion which can help it to be robust to model mis-specification (of which gene tree estimation error is one aspect).

This study showed that when the reference tree is very far from the true gene trees (as for our HGT+ILS data), gene tree correction typically fail to improve the initial gene tree (and even here, some methods can make the gene tree worse). This brings into question why the species tree (whether true or estimated) is used as a reference tree. We note that while the GDL-based methods may benefit from the use of a species tree as a reference tree (since the correction is based on GDL scenarios), this type of reference tree may not be optimal for TRACTION, which has no such dependency. Thus, part of our future work will be to explore techniques (such as statistical binning [3, 23]) that might enable the estimation of a better reference tree for TRACTION in the context of a multi-locus phylogenomic analysis.

This study suggests several other directions for future research. The GDL-based methods have variants that may enable them to provide better accuracy (e.g., alternative techniques for rooting the gene trees, selecting duploss parameter values, etc.), and future work should explore these variants. Most gene tree correction methods have been developed specifically to address the case where genes have multiple copies of species as a result of gene duplication events. TRACTION is currently restricted to gene trees with at most one copy of each species. In future work, we will explore extensions of TRACTION to handle multi-copy genes by using a generalization of the RF distance, such as proposed in [5]. In particular, one could construct the extended species tree to use as a reference along with a full differentiation of the gene tree as described in [5]. Recent work has shown how Notung could be extended to address HGT [19]; a comparison between TRACTION and a new version of Notung that addresses HGT will need to be made when Notung is modified to handle HGT (that capability is not yet available). Finally, the effect of gene tree correction on downstream analyses should be evaluated carefully.

References

1 M.S. Bansal. Linear-Time Algorithms for Some Phylogenetic Tree Completion Problems Under Robinson-Foulds Distance. In M. Blanchette and A. Ouangraoua, editors, *Comparative Genomics*. RECOMB-CG 2018. Lecture Notes in Computer Science, vol 11183. Springer, 2018.

2 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.

3 Md Shamsuzzoha Bayzid, Siavash Mirarab, Bastien Boussau, and Tandy Warnow. Weighted statistical binning: enabling statistically consistent genome-scale phylogenetic analyses. *PLoS One*, 10(6):e0129183, 2015.

4 R. Chaudhary, J.G. Burleigh, and O. Eulenstein. Efficient error correction algorithms for gene tree reconciliation based on duplication, duplication and loss, and deep coalescence. *BMC Bioinformatics*, 13(10):S11, 2012.

5 Ruchi Chaudhary, John Gordon Burleigh, and David Fernández-Baca. Inferring species trees from incongruent multi-copy gene trees using the Robinson-Foulds distance. *Algorithms for Molecular Biology*, 8(1):28, 2013.

6 K. Chen, D. Durand, and M. Farach-Colton. NOTUNG: a program for dating gene duplications and optimizing gene family trees. *Journal of Computational Biology*, 7(3-4):429–447, 2000.

7 S. Christensen, E.K. Molloy, P. Vachaspati, and T. Warnow. OCTAL: optimal completion of gene trees in polynomial time. *Algorithms for Molecular Biology*, 13(1):6, March 2018.

8 R. Davidson, P. Vachaspati, S. Mirarab, and T. Warnow. Phylogenomic species tree estimation in the presence of incomplete lineage sorting and horizontal gene transfer. *BMC Genomics*, 16:S1, 2015.

9 D. Durand, B.V. Halldórsson, and B. Vernot. A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology*, 13(2):320–335, 2006.

10 S.V. Edwards. Is a new and general theory of molecular systematics emerging? *Evolution*, 63(1):1–19, 2009.

11 George F Estabrook, CS Johnson Jr, and Fred R Mc Morris. An idealized concept of the true cladistic character. *Mathematical Biosciences*, 23(3-4):263–272, 1975.

12 George F Estabrook, CS Johnson Jr, and FR McMorris. A mathematical foundation for the analysis of cladistic character compatibility. *Mathematical Biosciences*, 29(1-2):181–187, 1976.

13 W. Fletcher and Z. Yang. INDELible: A Flexible Simulator of Biological Sequence Evolution. *Molecular Biology and Evolution*, 26(8):1879–1888, 2009. 10.1093/molbev/msp098.

14 P. Gawrychowski, G.M. Landau, W.-K. Sung, and O. Weimann. A Faster Construction of Phylogenetic Consensus Trees. *arXiv preprint*, 2017. `arXiv:1705.10548`.

15 E. Jacox, C. Chauve, G.J. Szöllősi, Y. Ponty, and C. Scornavacca. ecceTERA: comprehensive gene tree-species tree reconciliation using parsimony. *Bioinf.*, 32(13):2056–2058, 2016.

16 E. Jacox, M. Weller, E. Tannier, and C. Scornavacca. Resolution and reconciliation of non-binary gene trees with transfers, duplications and losses. *Bioinf.*, 33(7):980–987, 2017.

17 E.D. Jarvis, S. Mirarab, A.J. Aberer, B. Li, P. Houde, C. Li, S. Ho, B.C. Faircloth, B. Nabholz, J.T. Howard, et al. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 2014.

18 M. Lafond, C. Chauve, N. El-Mabrouk, and A. Ouangraoua. Gene tree construction and correction using supertree and reconciliation. *IEEE/ACM Trans Comp Biol Bioinf (TCBB)*, 15(5):1560–1570, 2018.

19 H. Lai, M. Stolzer, and D. Durand. Fast Heuristics for Resolving Weakly Supported Branches Using Duplication, Transfers, and Losses. In J. Meidanis and L. Nakhleh, editors, *Comparative Genomics*, pages 298–320, Cham, 2017. Springer International Publishing.

20 Vincent Lefort, Richard Desper, and Olivier Gascuel. FastME 2.0: A Comprehensive, Accurate, and Fast Distance-Based Phylogeny Inference Program. *Molecular Biology and Evolution*, 32(10):2798–2800, 2015.

21 W. Maddison. Gene Trees in Species Trees. *Systematic Biology*, 46(3):523–536, 1997.

22 D. Mallo, L. Martins, and D. Posada. SimPhy: phylogenomic simulation of gene, locus, and species trees. *Systematic Biology*, 65(2):334–344, 2016.

23 S. Mirarab, M.S. Bayzid, B. Boussau, and T. Warnow. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science*, 346(6215), 2014. `doi:10.1126/science.1250463`.

24 S. Mirarab and T. Warnow. ASTRAL-II: Coalescent-based Species Tree Estimation with Many Hundreds of Taxa and Thousands of Genes. *Bioinformatics*, 31(12):i44, 2015.

25 E. Molloy and T. Warnow. To include or not to include: The impact of gene filtering on species tree estimation methods. *Systematic Biology*, 67(2):285–303, 2018.

26 T.H. Nguyen, V. Ranwez, S. Pointet, A.-M. Chifolleau, J.-P. Doyon, and V. Berry. Reconciliation and local gene tree rearrangement can be of mutual profit. *Algorithms for Molecular Biology*, 8(1):1, 2013.

27 E. Noutahi, M. Semeria, M. Lafond, J. Seguin, B. Boussau, L. Guéguen, N. El-Mabrouk, and E. Tannier. Efficient gene tree correction guided by genome evolution. *PLoS One*, 11(8):e0159559, 2016.

28 D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical biosciences*, 53(1-2):131–147, 1981.

29 A. Stamatakis. RAxML Version 8: A tool for Phylogenetic Analysis and Post-Analysis of Large Phylogenies. *Bioinformatics*, 30(9), 2014. 10.1093/bioinformatics/btu033.

30 J Sukumaran and M.T. Holder. DendroPy: a Python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–1571, 2010. 10.1093/bioinformatics/btq228.

31 G.J. Szöllősi, 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.

32 P. Vachaspati and T. Warnow. ASTRID: Accurate Species Trees from Internode Distances. *BMC Genomics*, 16(10):S3, 2015. 10.1186/1471-2164-16-S10-S3.

33 Y.-C. Wu, M.D. Rasmussen, M.S. Bansal, and M. Kellis. TreeFix: statistically informed gene tree error correction using species trees. *Systematic Biology*, 62(1):110–120, 2012.

34 Y. Zheng and L. Zhang. Reconciliation With Nonbinary Gene Trees Revisited. *Journal of the ACM (JACM)*, 64(4):24, 2017.

A Details about the Experimental Design

The datasets we used are from prior publications (which should be consulted for full details). The information for Steps 1-2 provided here describe the high-level process used to generate these datasets for those studies, Step 3 describes the process used to estimate gene trees, and Steps 4-6 describe the high-level process used to correct gene trees.

- **Step 1:** A model species tree and model gene trees (evolved within the model species tree) were generated using SimPhy [22]. This produced a set of gene trees that could differ from the species tree due to ILS alone or due to both ILS and HGT. Importantly, SimPhy modifies gene tree branch lengths to deviate from a strict molecular clock.
- **Step 2:** Molecular sequences were generated using INDELible [13] by evolving sequences down each true gene tree under the GTR+GAMMA model of sequence evolution without insertions or deletions. GTR+GAMMA model parameters and sequence lengths were drawn from distributions as described in [24]. Because the sequence length parameter was drawn from a distribution, sequences had different lengths. In some experiments, sequence lengths were truncated to 100, 250, or 500 sites before estimating gene trees in order to vary the degree of gene tree estimation error.
- **Step 3:** Binary gene trees were estimated on each gene sequence alignment using RAxML, a maximum likelihood (ML) heuristic, under the GTR+GAMMA model, with all numeric parameters estimated directly from the data. Branch support for each internal branch in the best ML tree was computed using the RAxML rapid bootstrapping procedure [29] with 100 bootstrap replicates for the ILS-only datasets and 50 bootstrap replicates for the ILS+HGT datasets.
- **Step 4:** Species trees were estimated on each multi-gene dataset using ASTRID [32] on the estimated (best ML) gene trees from Step 3.
- **Step 5**: For each estimated (best ML) gene tree, all edges with branch support below 75% were collapsed to produce a set of "collapsed gene trees".
- **Step 6**: Estimated gene trees were corrected using the ASTRID tree from Step 4 as the reference tree. The input reference tree was rooted at the outgroup for all gene tree correction methods except for TRACTION. TRACTION and Notung were given the collapsed gene trees as input, whereas TreeFix, TreeFix-DTL, and ecceTERA were given the best ML gene trees (without any edges collapsed) as input. To run ecceTERA, we specified the threshold value (i.e., minimum required bootstrap support value), with the default setting of 75%; ecceTERA then collapses all branches that have support less than that value and exhaustively evaluates all refinements. However, when the collapsed gene trees have polytomies of degree greater than 12, then ecceTERA lowers the threshold until all polytomies have degree at most 12. Finally, Notung, ProfileNJ, and ecceTERA required that the input gene trees be rooted, so we rooted these input gene trees at the outgroup.

B Commands

In the following commands, "resolved gene trees" refers to the gene trees estimated using RAxML, "unresolved gene trees" refers to these estimated gene trees with branches having bootstrap support less than 75% collapsed, and "reference species tree" refers to the species tree estimated using ASTRID. Rooted means the input tree was rooted at the outgroup.

RAxML v8.2.11 was run as

```
raxml -f a -m GTRGAMMA -p 12345 -x 12345 -N <# bootstrap replicates> \
      -s <alignment file> -n <output name>
```

ASTRID v1.4 was run as

```
ASTRID -i <resolved gene trees> -o <output>
```

Notung v2.9 was run as

```
java -jar Notung-2.9.jar --resolve -s <rooted reference species tree> \
     -g <rooted unresolved gene tree> --speciestag postfix \
     --treeoutput newick --nolosses
```

TRACTION v1.0 was run as

```
traction.py --refine -r -s 12345 -b <unrooted reference species tree> \
         -u <unrooted resolved gene trees> -i <unrooted unresolved gene trees> \
         -o <output>
```

ecceTERA v1.2.4 was run as

```
eccetera resolve.trees=0 \
         collapse.mode=1 \
         collapse.threshold=75 \
         dated=0 print.newick=true \
         species.file=<rooted reference species tree> \
         gene.file=<rooted resolved gene tree>
```

Since ecceTera enters an infinite loop on some gene trees, the "timeout" command was used to kill ecceTera if it took more than five minutes on a single gene tree.

ProfileNJ requires a distance matrix; to compute distance matrices (with K2P-corrected distances) for ProfileNJ, FastME v2.1.6.1 [20] was run as

```
fastme -i <input gene alignment> -O <output distance matrix> -dK
```

ProfileNJ was run as

```
profileNJ \
   -g  <rooted unresolved gene tree> \
   -s  <rooted reference species tree> \
   -d  <distance matrix> \
   -o  <output> \
   -S  <name map> \
   -r none \
   -c nj \
   --slimit 1 \
   --plimit 1 \
   --firstbest \
    --cost 1 0.99999
```

TreeFix v1.1.10 was run on the ILS-only datasets as

```
treefix -s <rooted reference species tree> \
        -S <name map> \
        -A <alignment file extension> \
        -o <old tree file extension> \
        -n <new tree file extension> \
        <resolved gene tree>
```

TreeFix-DTL v1.0.2 was run on the HGT+ILS datasets as

```
treefixDTL -s <rooted reference species tree> \
           -S <map file> \
           -A <alignment file extension> \
           -o <old gene tree file extension> \
           -n <new gene tree file extension> \
           <resolved gene tree>
```

Normalized Robinson-Foulds distances were computed using Dendropy v4.2.0 [30] as

```
n1 = len(t1.internal_edges(exclude_seed_edge=True))
n2 = len(t2.internal_edges(exclude_seed_edge=True))
[fp, fn] = false_positives_and_negatives(t1, t2)
rf = float(fp + fn) / (n1 + n2)
```

B.1 Details about failures

No method other than ecceTERA and profileNJ failed on any datasets.

B.1.1 ecceTERA failures

In our analyses for the ILS-only datasets, ecceTERA failed on 10/4000 genes (moderate ILS) and 57/4000 genes (high ILS). In our analyses for the ILS+HGT datasets, ecceTERA failed on 744/7500 genes (moderate HGT) and 574/7500 genes (high HGT). Notably, the number of datasets that ecceTERA failed on increased with gene trees estimation error; for example, for datasets with ILS and HGT, ecceTERA completed on 100% of datasets with GTEE in (0.0, 0.4], 99.6% of datasets with GTEE in (0.4, 0.6], 76.4% of datasets with GTEE in (0.6, 0.8], and 9.2% of datasets with GTEE in (0.8, 1.0].

B.1.2 profileNJ failures

ProfileNJ computes distances to construct the corrected gene tree; when used with FastME, it failed on 2/4000 genes for the ILS-only condition (moderate ILS).