Optimal Tree Metric Matching Enables Phylogenomic Branch Length Reconciliation

Shayesteh Arasti,[1†] Puoya Tabaghi,[2†] Yasamin Tabatabaee[3] and Siavash Mirarab[4*]

[1] *Department of Computer Science and Engineering, UC San Diego, CA 92093, USA*
[2] *Halıcıoğlu Data Science Institute, UC San Diego, CA 92093, USA*
[3] *Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*
[4] *Department of Electrical and Computer Engineering, UC San Diego, CA 92093, USA*
[*] *Corresponding author: smirarab@ucsd.edu*

November 13, 2023

Abstract

The abundant discordance between evolutionary relationships across the genome has rekindled interest in ways of comparing and averaging trees on a shared leaf set. However, most attempts at reconciling trees have focused on tree topology, producing metrics for comparing topologies and methods for computing median tree topologies. Using branch lengths, however, has been more elusive, due to several challenges. Species tree branch lengths can be measured in many units, often different from gene trees. Moreover, rates of evolution change across the genome, the species tree, and specific branches of gene trees. These factors compound the stochasticity of coalescence times. Thus, branch lengths are highly heterogeneous across both the genome and the tree. For many downstream applications in phylogenomic analyses, branch lengths are as important as the topology, and yet, existing tools to compare and combine weighted trees are limited. In this paper, we make progress on the question of mapping one tree to another, incorporating both topology and branch length. We define a series of computational problems to formalize finding the best transformation of one tree to another while maintaining its topology and other constraints. We show that all these problems can be solved in quadratic time and memory using a linear algebraic formulation coupled with dynamic programming preprocessing. Our formulations lead to convex optimization problems, with efficient and theoretically optimal solutions. While many applications can be imagined for this framework, we apply it to measure species tree branch lengths in the unit of the expected number of substitutions per site while allowing divergence from ultrametricity across the tree. In these applications, our method matches or surpasses other methods designed directly for solving those problems. Thus, our approach provides a versatile toolkit that finds applications in similar evolutionary questions.

[†] These authors contributed equally to this work.

1 Introduction

Discordance among evolutionary histories across the genome [20] is now understood to be pervasive and has motivated the study of region-specific histories (often called gene trees) and the speciation history (the species tree). This availability of alternative trees leads to mathematical questions seeking to compare and combine trees. We need metrics to quantify tree dissimilarity, algorithms for mapping gene trees to the species tree to reconcile them [10], and median tree methods to combine gene trees into a single species tree [2]. Solutions to these problems have been developed and are widely adopted. Measures of tree difference such as Robinson and Foulds [30] (RF), quartet distance [11], and matching split (MS) [4] are commonly used to quantify gene tree differences. Biologists often use reconciliation methods in comparative genomics to understand the evolution of gene function. Finally, inferring the species tree by finding the tree with the minimum total quartet distance to gene trees (quartet median tree) has been widely adopted due to its statistical consistency under several models of gene tree discordance [23].

The rich literature on comparing and combining trees, however, has mostly focused on tree topology. A typical phylogenomic pipeline infers a large number of gene trees with branch lengths in the unit of the expected number of substitutions per site (SU), but ignores those branch lengths altogether in downstream tasks such as species tree inference, reconciliation, and discordance quantification. And discordance is clearly not just limited to topology. One of the main causes of discordance is incomplete lineage sorting (ILS) as modeled by the multi-species coalescent (MSC) model [9]. Under MSC, internal nodes of a gene tree correspond to coalescence events that have to predate species divergences; thus, gene tree branch lengths are guaranteed to differ from the species tree (when both are in the unit of time) even if there is no topological discordance. Gene duplication times similarly do not match each other or the speciation times. Inference challenges compound the problem. From sequences alone, we can only measure SU lengths, which are a function of mutation rates. Assuming a fixed rate (i.e., strict clock model [41]) is often insufficient as accumulated evidence shows that rates can change across the tree, across the genome, or both [15, 29, 19]. Furthermore, gene trees are often inferred from short sequences (to avoid recombination within a gene), which makes branch length estimates very noisy, especially for shortest branches with no substitutions.

Methods for using branch lengths in phylogenomics do exist but are not predominant. Some of the first statistically consistent species tree inference methods [e.g., GLASS and STEM; 25, 14] used gene tree branch lengths alone. However, these methods have proved far less accurate than topology-focused methods [8, 37]. Bayesian co-estimation methods [e.g., 17, 13, 5] also consider branch length by jointly modeling gene trees and species trees with branch lengths and parameters for mutation rate, population size, and (potentially) rate heterogeneity. These methods have been highly accurate [37] and readily produce a reconciliation of species trees and gene trees (since all lengths are in the same unit). However, they do not scale to large datasets [26, 40]. Also, since complex rate models further reduce scalability, these methods often model only limited rate heterogeneity. Scalable and accurate methods such as ASTRAL [38] and NJst/ASTRID [18, 34] use gene tree topologies as input and produce only species tree topology. We can use *post hoc* calculations to infer coalescent units (CU) branch lengths for internal branches of the species tree using gene tree topologies

alone, but this approach can lead to systematic under-estimation [31, 12].

There have been attempts to reconcile gene tree branch lengths with an inferred species tree. It seems natural to somehow transfer the SU branch lengths from gene trees onto the species tree with some form of summarization. One proposed approach, called ERaBLE, uses a least-squared optimization applied to patristic distances (pairwise path lengths) of the gene trees to estimate species tree lengths, effectively taking a weighted average. ERaBLE does allow for a limited form of rate heterogeneity across loci. As Tabatabaee et al. [33] have shown, ERaBLE tends to overestimate terminal branches of the species tree because it ignores deep coalescence. They propose an alternative coalescent-based method called CASTLES that specifically corrects for the fact that under MSC, gene tree nodes predate species tree nodes. While CASTLES is more accurate than ERaBLE in MSC simulations [33], ERaBLE has the advantage of having a simple objective function that does not depend on any model of gene tree discordance. This observation leads us to ask: can other model-free methods with more flexibility do a better job of transferring branch lengths across trees?

In this paper, we approach the reconciliation of branch lengths across two trees using a linear algebraic formulation. Similar to ERaBLE, we form an optimization problem based on patristic distances. However, unlike ERaBLE, which estimates only one scalar parameter per gene tree, we assign one parameter per each branch of each gene tree (we come back to this comparison in the discussion). The optimization problem we define, called Topology-Constrained Metric Matching (TCMM), assigns branch lengths to a query tree such that its patristic distances match a reference tree as much as possible without changing the topology. We show that an extra regularization term enables the user to balance the competing goals of matching the reference tree and keeping the original branch length patterns of the query tree. We also show that TCMM extends to multiple reference trees naturally. The introduced TCMM problems are all convex and can be solved efficiently with standard techniques. Moreover, the input to the optimization problems can be computed efficiently using dynamic programming preprocessing with quadratic running time. While the ability to match the metric space of one tree to another (the core of TCMM) provides a versatile tool, in this paper, we focus on a particular application: reconciling gene tree branch lengths (used as reference) against the species tree (used as the query). We show that TCMM can be paired with outlier detection and smart averaging methods to design a method that is as accurate as CASTLES in estimating species tree branch lengths, despite being agnostic to MSC. Moreover, the regularization technique allows us to pair TCMM with CASTLES to obtain a method that tends to be more accurate than either alone. We end by speculating about some of the future applications of TCMM to other questions in phylogenomics.

2 Methods

Notations: For an unrooted tree $T = (V_T, E_T)$, we let L_T denote the leaf set and $P_T(v, v')$ denote the path between any pairs of vertices $v, v' \in V_T$. Assigning the natural orientation to the edges of a rooted tree away from the root, the structure becomes a directed rooted tree. For a weighted tree, each edge $e \in E_T$ has length $w_T(e) \in \mathbb{R}^+$. We use $\mathcal{T}(T)$ to denote the set of all weighted trees with the same topology as tree T. A weighted tree T with N vertices indexed by $[N] = \{1, \ldots, N\}$ defines a metric $d_T : V_T \times V_T \to \mathbb{R}^+$ where $\forall v, v' \in V_T : d_T(v, v') = \sum_{e \in P_T(v,v')} w(e)$. With slight abuse of notation, we define the *weight*

vector w_T and the leaves' *pairwise distance* vector d_T as:

$$w_T = (w_T(e))_{e \in E_T} \in \mathbb{R}^{|E_T|} \quad \text{and} \quad d_T = (d_T(v, v'))_{v, v' \in L_T, v < v'} \in \mathbb{R}^{\binom{|L_T|}{2}} .$$

Definition 1 (Path matrix A_T)**.** *Given a tree T we define a binary matrix A_T of size $\binom{|L_T|}{2} \times |E_T|$ where rows correspond to pairs of vertices in L_T, columns represent the edge set. For two distinct vertices $v, v' \in L_T$ and edge $e \in E_T$, an element $((v, v'), e)$ of A_T is one if and only if that the edge e is on the path between v and $v' \in L_T$. Otherwise, it is equal to zero.*

2.1 Problem Statements

We start with a general tree matching problem, which we then successively refine.

Problem 1 (General Topology-Constrained Metric Matching)**.** *Given is a query tree T and a reference weighted tree R, both on a shared leafset L. For any tree T' with the same topology as T, let $\text{cost}(T'; R)$ measure the branch length discrepancy of T' to the reference tree R. Then, we seek $T^* \in \arg\min_{T' \in \mathcal{T}(T)} \text{cost}(T'; R)$.*

The path matrix A_T completely encodes the topology of T; we can completely identify a weighted tree $T = (V_T, E_T)$ using its topology (i.e., the matrix A_T) and its weight function w_T. Thus, we can parameterize the cost in Problem 1 with respect to the weight vector w_T — since the topology is fixed. We reparameterize the objective as $\text{cost}(w'; R)$ and reformulate Problem 1 as follows:

$$w^* \in \arg\min_{w'} \text{cost}(w'; R),$$

where $w' \in \mathbb{R}^{+\binom{|V_T|}{2}}$ are constrained to be nonnegative. A closest tree to reference tree R has the same topology as T and edge weights w^*, that is, $T^* = (V_T, E_T)$ and $w_{T^*} = w^*$.

One strategy in designing an appropriate cost function is to quantify the discrepancy of metric information between the two trees. To do so, we first find a separable form for pairwise distances in terms of two quantities: (1) topology, and (2) edge weights. For any distinct $v, v' \in L$, we have

$$d_T(v, v') = \sum_{e \in P_T(v, v')} w_T(e) = \langle a_{v,v'}, w_T \rangle$$

where $a_{v,v'} = \left(1_{e \in P_T(v,v')}\right)_{e \in E_T} \in \{0, 1\}^{|E_T|}$ and 1_x is the indicator function. By definition, the vector $a_{v,v'}$ is the row of the path matrix A_T, corresponding to vertex pair (v, v'). Thus, given a tree $T = (V_T, E_T)$, we have $d_T = A_T w_T$. This enables us to arrive at a separable form for pairwise distances in terms of A_T, representing topology, and the weight vector w_T. We next provide a specific form of Problem 1 with the goal of transforming the weights of a tree so that its metric information best matches to a reference tree.

Problem 2 (Topology-Constrained Metric Matching— under ℓ_2^2)**.** *Let T, R be weighted trees with edge weight functions w_T and w_R on the common leafset L with at least two elements.*

We define the following cost to measure the discrepancy of patristic distances of T to the reference tree R:

$$\text{cost}(w_T; R) = \sum_{v,v' \in L} |d_R(v,v') - d_T(v,v')|^2 = \|d_R - A_T w_T\|_2^2$$

The weights of the closest tree to R matching T in topology with edge weights lower bounded by $\varepsilon \geq 0$ is:

$$w^* \in \underset{w \in \mathbb{R}^{|E_T|}, w \geq \varepsilon}{\arg\min} \|d_R - A_T w\|_2^2.$$

Clearly, if both R and T have the same topology, then w_R is an optimal solution (i.e., $T^* = R$). More broadly, the cost function in Problem 2 is agnostic to the weights w_T, and thus, it can dramatically change the branch lengths of T. In particular, branches that do not match between T and R tend to become very short (as we will see in experiments). For some applications, it may be desirable to retain the original branch lengths to some extent, while making the tree similar to the other. We can accommodate that goal by adding a regularizing term. For an optimal edge weight vector w_{T^*}, we define the *adjustment rate vector* by elementwise dividing w_{T^*} and w_T, to get w_{T^*}/w_T. These adjustment rates do change with the initial weights w_T. To incorporate the information of w_T in the problem statement, we penalize the deviation of estimated w_{T^*} from the initial weight vector w_T, save for a total scaling factor shared by all branches. This is summarized in the following problem.

Problem 3 (Regularized Topology-Constrained Metric Matching — under ℓ_2^2)**.** *Let T, R be weighted trees with edge weight functions w_T and w_R on the common leafset L. We define the following regularized cost to measure the discrepancy of $T' \in \mathcal{T}(T)$ with weights w to the reference tree R:*

$$\text{cost}(w; R) = \|d_R - A_T w\|_2^2 + \lambda \sigma^2(w/w_T),$$

where λ is the regularization coefficient, $\sigma^2(w/w_R)$ computes the variance of the adjustment rates vector w/w_R. We seek the tree $T^* \in \mathcal{T}(T)$ with edge weights

$$w_{T^*} \in \underset{w \in \mathbb{R}^{|E_T|}, w \geq \varepsilon}{\arg\min} \text{cost}(w; R)\ .$$

The regularizing term, which is a quadratic function of the weight vector w, leads to lowered variance in the adjustment rate vector, w/w_T, and thus, lowered change in *relative* branch lengths compared to T. Clearly, if the variance of adjustment rate values is 0, the tree T has been simply scaled up or down, without any real change to its relative branch lengths. Higher λ values encourage less change to the relative branch lengths of T. As $\lambda \to \infty$, the optimal solution will simply scale branch lengths of T with a fixed global rate $r*$. That is, $w_{T^*} \parallel w_T$ where $\parallel$ denotes parallel vectors. More precisely, we have $r^* \in \arg\min_{r \cdot \min_e w_T(e) \geq \varepsilon} \|d_R - r A_T w_T\|_2^2$. Our regularization, however, does have a side effect. It penalizes scaling of the weights and thus prefers smaller values for w_{T^*}. Note that for any r, we have $\sigma^2((rw)/w_T) = r^2 \sigma^2(w/w_T)$. Thus, compared to no regularization, regularization is expected to produce shorter branch lengths.

Problem 4 (Topology-Constrained Consensus Metric Matching)**.** *Let* $\{R_k = (V_{T_k}, E_{T_k}) : k \in [K]\}$ *be a set of weighted reference trees such that* T_k *has weight function* w_k*, for all* $k \in [K]$*. Given a tree* $R = (V_R, E_R)$ *and the set of leaves common to all trees denoted by* L*, we seek to find the optimal (adjusted) tree* $T^* = (V_T, E_T)$ *with the same topology as tree* T *such that its weights minimizes the following (regularized) consensus cost:*

$$w_{T^*} \in \underset{\substack{w \in \mathbb{R}^{|E_R|} \\ w \geq \varepsilon}}{\arg\min} \sum_{k \in [K]} \alpha_k^2 \|d_{R_k} - A_T w\|_2^2 + \lambda \sigma^2(w/w_T),$$

where α_k *is a nonnegative weight controlling the contribution of reference tree* R_k *in the cost function,* $\lambda > 0$ *is a regularizing coefficient,* $\sigma^2(w/w_T)$ *penalizes the weights that result in a large variance for the adjustment rate vector* w/w_R*, and* $\varepsilon \geq 0$ *is a lower bound for the estimated edge weights.*

2.2 Solving TCMM Efficiently as Convex Optimization

2.2.1 TCMM problems are all convex

We can rewrite the cost functions in Problems 2–4 in quadratic forms. For ease of notation, we use $d = d_R$ and $A = A_T$ in Problems 2,3 and $d = d_{R_k}$, $A = A_T$ in Problem 4. Note that Problem 2 minimizes $l_1(w) \overset{\text{def.}}{=} \|d - Aw\|_2^2$. We convert l_1 to the following quadratic form

$$l_1(w) = \frac{1}{2} w^\top B w + c^\top w + \text{const}, \tag{1}$$

where $B = 2A^\top A$ is a positive semidefinite matrix, $c = -2A^\top d$, and const $= \|d\|_2^2$. Since $B \succeq 0$ by construction, l_1 is a convex function of the weight parameter w. Adding the regularization term $\sigma^2(r)$ — where $r = w/w_T$ — does not change the convexity as we can convert it to the following quadratic form:

$$l_2(w) \overset{\text{def.}}{=} \sigma^2(r) = \frac{1}{|E|}(r - 1\overline{r})^\top (r - 1\overline{r}) = \frac{1}{|E|} r^\top (P_1^\perp)^\top P_1^\perp r = \frac{1}{|E|} r^\top P_1^\perp r$$

where $1 \in \mathbb{R}^{|E|}$ is the vector of all ones, $\overline{r}$ is a scalar that is the average value of elements in r, $P_1^\perp = I - \frac{1}{|E|} 1 1^\top$. It is a known fact that $P_1^\perp$ is an orthogonal projection operator, $(P_1^\perp)^\top P_1^\perp = P_1^\perp$. Furthermore, it is a positive semidefinite matrix with $0 - 1$ eigenvalues. Therefore, the regularizing term $\sigma^2(w/w_T)$ is indeed a convex function of the weight parameter w. Thus, Problems 2,3, and 4 all admit globally optimal solutions.

Proposition 1. *The cost functions in Problems 2, 3, and 4 admit unique minimums but not necessarily unique minimizers. We can arrive at an optimal solution using a projected gradient descent method.*

Proof. The gradient of the cost function in equation (1) with respect to the weights $w \in \mathbb{R}^{|E|}$ is:

$$\nabla_w l_1(w) = Bw + c,$$

where $B \in \mathbb{R}^{|E| \times |E|}$ and $c \in \mathbb{R}^{|E|}$. The Hessian $\nabla_w^2 l_1(w) = B$ is a positive semidefinite matrix; thus, $l_1(w)$ is a convex function everywhere in its domain. We can also compute the gradient

of $l_2(w)$ as follows:

$$\nabla_w l_2(w) = \frac{2}{|E|}(P_1^{\perp} r)(1/w_T) = \frac{2}{|E|}P_1^{\perp}(w/w_T^2).$$

Furthermore, its Hessian $\nabla_w^2 l_2(w) = \frac{2}{|E|}\text{diag}(1/w_T)P_1^{\perp}\text{diag}(1/w_T)$ is a positive semidefinite matrix, making $l_2(w)$ a convex function of w as well. Since both $l_1(w)$ and $l_2(w)$ are convex functions of weight parameter w and cost functions in Problems 2, 3, and 4 are linear functions of l_1 and l_2 with *positive coefficients* (λ and α_k^2 terms), each cost function is also a convex function of w. In Problem 2, the constraint set is $w \geq 0$ which is nonempty, closed, and convex. In Problems 3 and 4, the constraint set is $w \geq \varepsilon$ which is also nonempty, closed, and convex. Therefore, all three problems involve convex function minimizations over nonempty, closed, and convex sets. Since convex functions over nonempty, bounded, convex sets have a minimum, the projected gradient descent method converges to the global minimum of the cost functions [35]. □

By Proposition 1, we can use the simple projected gradient descent to solve Problems 2 – 4 using the gradient terms shown above. We implemented and optimized the quadratic functions l_1 and l_2 using built-in functions in the Python package `CVXPY`. The only remaining challenge is scalability, which we next address.

2.2.2 Dynamic programming enables calculating B and c in quadratic time

Luckily, our formulation does not need to ever compute A_T and d_R — whose dimensions grow cubically with N. Rather, it needs the matrix B, which grows quadratically, and vector c, which grows linearly with N. Naively computing $A_T^\top A_T$ from its building blocks requires $O(N^4)$ time and $O(N^3)$ memory. Similarly, computing $A_T^\top d_R$ naively requires $O(N^3)$ time and $O(N^3)$ memory. However, we can compute B and c directly in quadratic time and memory using dynamic programming without ever computing A_T and d_R.

Computing $A_T^\top d_R \in \mathbb{R}^{|E_T|}$. We use a coloring scheme for the nodes of the trees T and R. Each edge $e \in E_T$ corresponds to a bipartition of taxa, dividing the leaves of T into two sets. For an arbitrary rooting of T, we define $C_T(e)$ as the set of leaves of T below e and $C'_T(e)$ as $L_T \setminus C_T(e)$. By Definition 1, the column of A_T corresponding to $e \in E_T$ has a 1 only for rows (v, v') where e is on the path $P_T(v, v')$. Therefore, The element of $c = A_T^\top d_R$ that corresponds to $e \in E_T$ is

$$c_e = \sum_{v,v' \in L_T, v<v'} d_R(v, v') 1_{e \in P_T(v,v')} = \sum_{v \in C_T(e)} \sum_{v' \in C'_T(e)} d_R(v, v') \tag{2}$$

where the second equality follows from the fact that each path in the first equation is counted only if it includes e. We color each leaf $v \in L_R$ in the reference tree R as blue if $v \in C_T(e)$ and red if $v \in C'_T(e)$ (See Figure 1). Both the query tree T and the reference tree R are induced down already to the common leaf-set L. Using this coloring scheme and a single bottom-up traversal of the reference tree (Algorithm 1), c_e can be computed in linear time for each edge $e \in E_T$ and thus quadratic time and memory for all edges.

Computing $B = A_T^\top A_T$. Given the Definition 1 of the path matrix A_T, it can be easily deduced that for two edges e and e', $B_{e,e'}$ is the number of pairs of leaves $v, v' \in L$ where

Algorithm 1 Computing $A_T^\top A_T$ and $A_T^\top d_R$

procedure COMPUTE_$A^\top A$(Tree T)
  `pre-compute` $|C_T|$ `and` $|C'_T|$ `for each edge of` T `in a bottom-up traversal`
  for e `in` E_T **do**
    for e' `in` E_T **do**
      if $e == e'$ **then**
        $A^\top A[e, e'] \leftarrow |C_T(e)| \cdot |C'_T(e)|$
      else
        if e `is a descendant of` e' **then**
          $A^\top A[e, e'] \leftarrow |C_T(e)| \cdot |C'_T(e')|$
        else if e' `is a descendant of` e **then**
          $A^\top A[e, e'] \leftarrow |C_T(e')| \cdot |C'_T(e)|$
        else
          $A^\top A[e, e'] \leftarrow |C_T(e)| \cdot |C_T(e')|$
  return $A^\top A$
procedure COMPUTE_$A^\top d$(Tree T, Tree R)
  for $e \in E_T$ **do**
    for $n \in$ bottom-up traversal of V_R **do**
      $e' \leftarrow$ the edge above n
      if n `is leaf` **then**
        $C_b(n), C_r(n) = 1_{n \in C_T(e)}, 1_{n \notin C_T(e)}$ ▷ Number of blue and red leaves below n
        $L_b(n), L_r(n) = w_{e'} \cdot C_b(n), w_{e'} \cdot C_r(n)$ ▷ Sum of the distances of blue and red leaves up to the parent of n
      else
        $c_l, c_r \leftarrow$ children of n
        $A^\top d[e] = C_b(c_l) \cdot L_r(c_r) + C_r(c_l) \cdot L_b(c_r) + C_b(c_r) \cdot L_r(c_l) + C_r(c_r) \cdot L_b(c_l)$
        $C_b(n), C_r(n) = C_b(c_l) + C_b(c_r), C_r(c_l) + C_r(c_r)$
        $L_b(n), L_r(n) = L_b(c_l) + L_b(c_r) + w_{e'} \cdot C_b(n), L_r(c_l) + L_r(c_r) + w_{e'} \cdot C_r(n)$
  return $A^\top d$

$e, e' \in P_T(v, v')$. We easily pre-compute $|C_T(e)|$ and $|C'_T(e)|$ for each edge $e \in E_T$ in linear time and the least common ancestor (LCA) of all pairs of nodes of T in quadratic time. With these, we can easily compute $B_{e,e'}$ for all $e, e' \in E_T$ in quadratic time using simple calculations given in Algorithm 1.

2.3 TCMM for Reconciling Species Tree and Gene Tree Branch Lengths

Phylogenomic analyses often produce a species tree topology S and a set of gene trees with estimated SU branch lengths $\mathcal{G} = \{G_1, \ldots, G_K\}$. To be useful for downstream analyses (e.g., dating), the species tree needs SU branch lengths, which methods such as ASTRAL do not produce. TCMM can be used to draw SU branch lengths on the species tree, reconciling those of gene trees. We propose two primary approaches to do so, both of which counterintuitively use gene trees as the reference and the species tree as the query.

Per-gene TCMM + Summarizing. We use Problem 2 to match the species tree (as query T) to each gene tree (as reference R). When S is already furnished with some branch lengths (e.g., results of running other methods), we can instead use Problem 3. The solution provides us with a set of estimated branch lengths $\tilde{w}_{e,k}$ where $e \in E_S$ and $G_k \in \mathcal{G}$. These lengths can be further summarized into a single value $\hat{w}_e$ for each branch using common

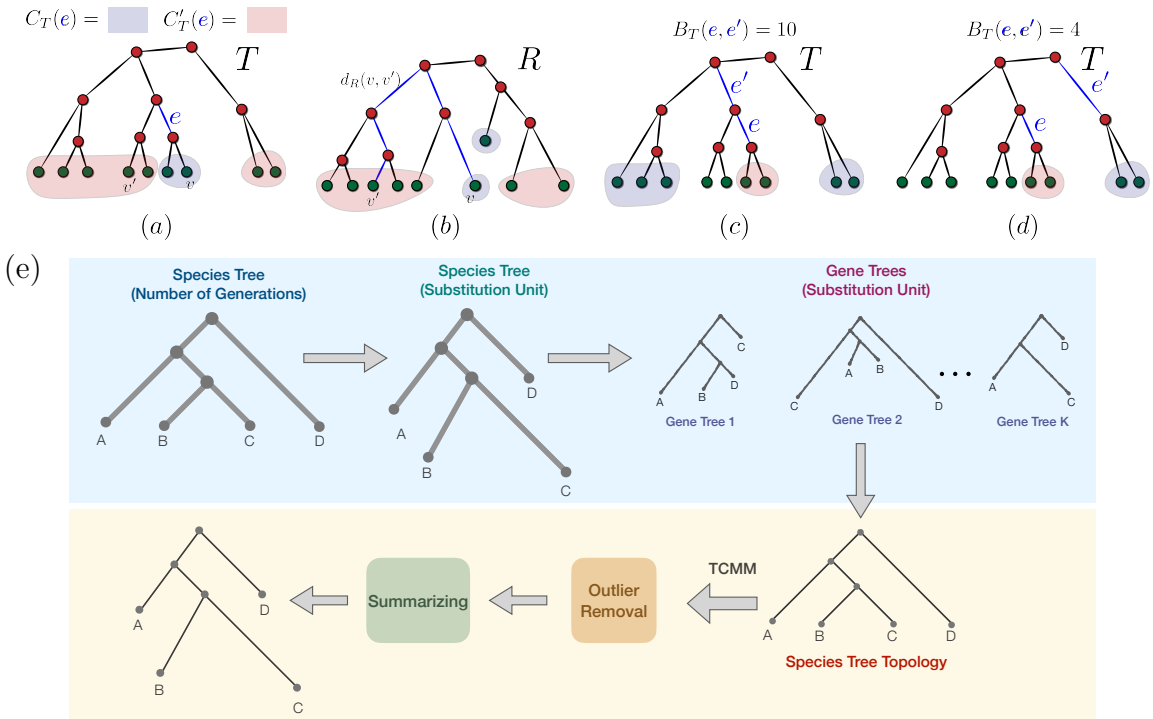

Figure 1: (a) Tree T with $C_T(e)$ and $C'_T(e)$ highlighted in blue and red for an edge e. (b) Tree R; Recall $c_e = \sum_{v \in C_T(e)} \sum_{v' \in C'_T(e)} d_R(v, v')$. An Example path $P_T(v, v')$ corresponding to $d_R(v, v')$ is shown in blue. (c, d) For edges e, e', $B_T(e, e')$ is the number of pairs of leaves v, v' where $e, e' \in P_T(v, v')$. c) If e is a descendant of e', $B_T(e, e') = C_T(e)C'_T(e')$. d) Otherwise, $B_T(e, e') = C_T(e)C_T(e')$. (e) Assumed data generation model (blue) and branch length estimation (yellow) processes. The second panel shows the default TCMM pipeline where the input is a set of gene trees as well as the species tree topology and the output is the species tree with SU branches.

averaging methods — arithmetic mean, harmonic mean, and geometric mean. By default, we will use geometric mean, which works best empirically (as shown in Section 3) and has a theoretical justification. Assume estimated $\tilde{w}_{e,k}$ values are noisy estimates of the correct length with multiplicative (as opposed to additive) noise. Thus, they follow a log-normal distribution, with the median equal to the true length. This model has worked well for dating [21] and is more reasonable than the standard additive error (i.e., a normal distribution) given that estimated branch lengths can never be negative. Under the log-normal model, the geometric mean is the appropriate estimator of the median parameter. Using the per-gene approach has the added benefit of enabling outlier removal. TCMM tends to assign exceedingly small values to *some* branches that are in the query but not the reference tree (as we will see), creating outliers. To detect these near-zero outliers, assume that $\tilde{w}_{e,k}$ values follow an exponential distribution with a median equal to that of the observed data (using median reduces the impact of outliers). Exponential distribution is a natural choice for outlier detection because 1) its domain is positive, 2) it requires only specifying one parameter (in contrast to log-normal which needs two parameters, making parameter setting more sensitive to outliers), and 3) is leptokurtic, making outlier detection conservative. We calculate the empirical cumulative distribution function (ECDF) of the $\tilde{w}_{e,k}$ values and for each point measure the difference between ECDF and CDF of the exponential with the same median.

We then identify the point p with the largest positive difference between the observed ECDF and the model CDF where ECDF(p) < 0.5. We eliminate all data points smaller p and compute the geometric mean of the remaining points.

Consensus TCMM. Here, we use Problem 4 specifically tailored to match a single query tree to multiple reference trees. The solution to Problem 4 simply requires computing weighted means of patristic distances across input gene trees used as input to the same optimization problem. Note that once again, instead of means, we can use geometric mean. The weights $\{\alpha_k : k \in [K]\}$ for the gene trees can be carefully selected to avoid overemphasizing the gene trees with overall longer branches. Here, we set the gene tree weights to the inverse of the sum of pairwise distances:

$$\forall k \in [K] : \alpha_k^{-1} = \sum_{v,v' \in L_{G_k}} d_{G_k}(v, v'). \tag{3}$$

2.4 Experimental design

We evaluated our proposed approach, TCMM, for reconciling species tree and gene tree branch lengths using both simulated and real datasets. On simulated data, the model species tree branches have true SU lengths (w) obtained by multiplying its length (number of generations) with varying mutation rates, creating non-ultrametric trees. We compared the true w values to estimated values $\hat{w}$ to quantify the absolute error ($|w - \hat{w}|$) and bias ($w - \hat{w}$), both averaged across all species tree branches.

We compare TCMM to four leading methods from a prior study [33], all run on the true topology. We include the coalescent-based method CASTLES, which accounts for coalescence times predating species divergence times. This method does allow rate variation across the species tree, but assumes gene trees follow the same patterns of rate variation (i.e., no heterotachy). CASTLES was developed as an alternative to what biologists have often used in practice: After inferring the species tree using ASTRAL, they use concatenation (here using RAxML [32]) to estimate branch lengths of the species tree. This approach does not account for gene tree differences, unlike the other models tested in this paper. We also include two methods that like TCMM use patristic distances. A simple alternative is to average patristic distance matrices across gene trees and use a distance-based method (FastME [16]) to draw branch lengths on S. A more advanced version of this idea is ERaBLE [3], which uses a proportional model. It estimates a single scalar rate per gene tree by matching them simultaneously to the species tree. In comparison, TCMM estimates one scalar per branch of the species tree (in the consensus mode) or per gene $\times$ species (in the per-gene mode). Thus TCMM has far more parameters than the ERaBLE approach. We use ERaBLE in its default mode.

In addition, we compare TCMM in the per-gene and consensus modes mentioned earlier. In the per-gene mode, we test it with and without outlier removal and with both arithmetic and geometric means. In the consensus mode, we compare arithmetic and geometric means as well as weighting input trees using Eq. (3). The default version is the per-gene with outlier detection and geometric mean summarization.

We study two simulated datasets and a biological dataset.

S100 dataset. We reused a published [33] 101-taxon (100 ingroup and one outgroup) 1000-gene simulated dataset with species trees that have both SU and CU length. Ultrametric

species trees with generation times are simulated using a birth-death model; each branch length is multiplied by a rate multiplier drawn from a Gamma distribution with mean 1 (variance ranging between 0.0153 and 2.6027) and a mutation rate to get SU length and divided by a constant population size to get CU. The 50 replicates of this dataset are highly heterogeneous in terms of ILS, with the mean RF distance between the model species tree and the true gene trees (AD) at 46%. In addition to true gene trees, we have FastTree-2 [28] estimated gene trees from 200bp, 400bp, 800bp, and 1600bp alignments, leading to average RF distance between true and estimated gene trees of 23%, 31%, 42% and 55%, respectively.

HGT dataset. We generated a new 51-taxon (50 ingroups + outgroup) simulated dataset with gene tree discordance due to both ILS and Horizontal Gene Transfer (HGT). We used the parameters of Davidson et al. [6] but generated species trees with branch lengths in SU and CU units, using a modified version of the simulation software SimPhy [22] created by Tabatabaee et al. [33]. This dataset has six model conditions that are characterized by different rates of HGT; the base ILS level is moderate (30% AD) and the same across all model conditions. The HGT rates for the six model conditions are 0, 2×10^{-9}, 5×10^{-9}, 2×10^{-8}, 2×10^{-7}, and 5×10^{-7} corresponding to the expected number of HGT events per gene of 0, 0.08, 0.2, 0.8, 8 and 20 respectively. Total discordance is a product of both ILS and HGT. The AD level of the six conditions are 30%, 30%, 31%, 34%, 53%, and 68%, respectively, with the increase being completely due to HGT. This dataset has 50 replicates with 1000 gene trees in each of them. We use only true gene trees for this dataset.

Biological dataset. We analyzed a prokaryotic dataset studied by Moody et al. [24] to address a debate about a long-standing hypothesis that the domains bacteria and archaea are separated by a long branch. The authors claimed that lower divergences obtained by some recent studies [39, 27] were the results of highly abundant HGT in microbial datasets [1]. Moody et al. [24] suggest that those studies that had used concatenation to infer branch lengths on the species tree have likely underestimated the length of the branch separating domains archaea and bacteria (AB branch), potentially because of high levels of inter-domain gene transfer. Here, we reanalyzed two smaller dataset analyzed by Moody et al. [24]: A 72-taxon tree [36] with 49 core genes including ribosomal proteins and other conserved genes, and a 108-taxon dataset [27] with 38 genes that only includes non-ribosomal proteins. We compare the length of the AB branch inferred using TCMM, CASTLES and concatenation, all drawn on ASTRAL topologies built from the two gene sets.

3 Results

Versions of TCMM. Overall, the most accurate version of TCMM is the per-gene mode, used with the geometric mean, and outlier removal (Fig. 2a). All choices of the default method contribute to accuracy and are needed. Outlier removal has minimal impact on terminal branch lengths, but substantially reduces error (42%) and bias (65%) for internal branches. When comparing the branch lengths from CASTLES to those from TCMM, we observe a long tail of branches that are made much smaller (median: 2.4×10^{-19}) by TCMM (Figs. 2 and S7b). These outliers (1.7% of the per-gene estimates) are among branches that do not match between the species tree and the gene trees. Consequently, the removal of these outliers predominantly impacts the accuracy of internal branches, as validated by our results. Using the geometric mean in the default version instead of the arithmetic mean

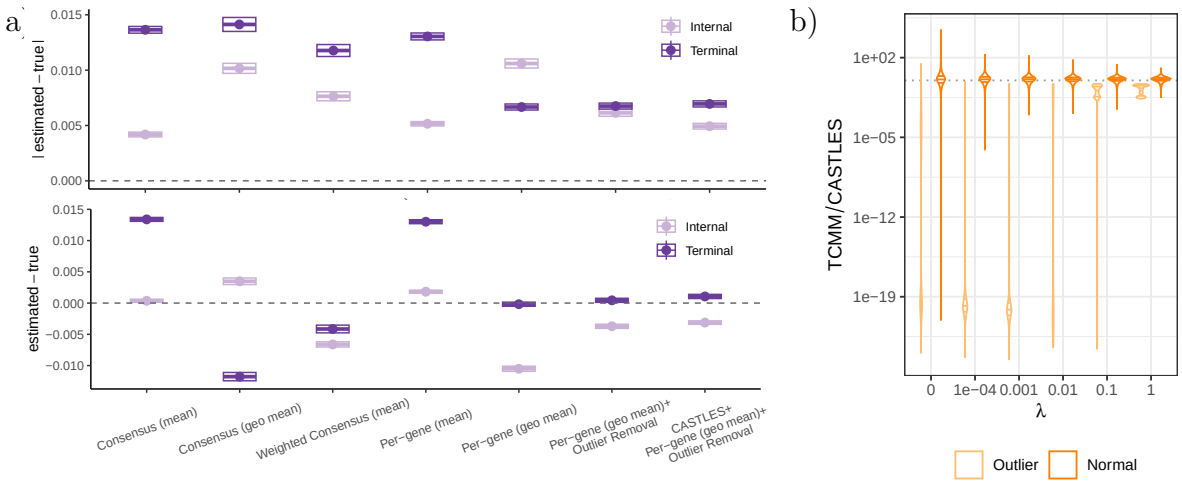

Figure 2: (S100 Dataset) (a) Comparison of different versions of TCMM in terms of absolute error (top) and bias (bottom), showing mean (dot) and 0.95% normal CI (box). Summarization methods are arithmetic mean (*mean*) or geometric mean (*geo mean*). The weighted consensus method used Equ. (3). The regularization coefficient (λ) was set to zero, except for the CASTLES+per-gene method where $\lambda = 10^{-3}$. (b) The effect of the regularization coefficient (λ) on the estimated branch lengths. TCMM output length per gene (before summarization) over the input length (CASTLES) is shown; the variance of this ratio (spread of distribution) indicates the change in relative length of branches between input and output of TCMM. Outliers are those identified by our per-gene method.

leads to an error reduction of 5.7% in the per-gene method. As expected, the geometric mean results in shorter branch estimates. In particular, the arithmetic mean tends to introduce an overestimation bias, averaging 0.0075 SU, perhaps due to ignoring the impact of coalescence. In contrast, the geometric mean introduces an underestimation bias, averaging at −0.0053 SU.

Compared to the default per-gene version, the consensus method using a weighted mean of gene tree patristic distances has a higher under-estimation bias and higher mean absolute error (Fig. 2a). Removing the weights has an opposite impact on terminal and internal branches: it increases terminal branch length error and bias slightly, but dramatically decreases the internal length error and bias. In fact, the unweighted consensus method has the lowest bias and highest accuracy for internal lengths among all the methods. Thus, while the default method has 28% less mean error than the consensus (unweighted mean), the latter should be considered a viable alternative when the internal branch lengths are the focus. Using geometric mean instead of the arithmetic mean does not lead to an overall reduction in error for the consensus method.

Incorporating the estimated branch lengths derived from CASTLES into the per-gene (geometric mean) method, with a regularization coefficient $\lambda = 10^{-3}$ (Problem 3) yields a further 20% reduction in error for internal branches, while maintaining accuracy for terminal branches (Fig. 2a). Examining varying λ reveals that the exact choice has a limited impact on results (Fig. S6). As expected, as λ increases, branch lengths estimated by the default TCMM method become successively more similar to initial branch lengths (from CASTLES) provided as input in terms of their relative values (Fig. 2b). Interestingly, using the regularization also reduces the extent to which results of per-gene TCMM include outliers that need to be

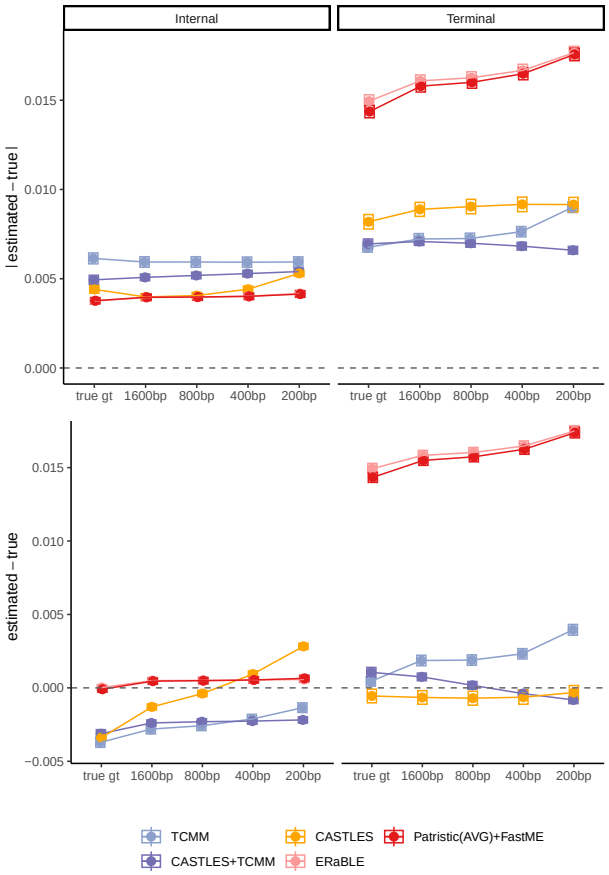

Figure 3: Comparing TCMM with the alternative methods on the S100 dataset in the presence of gene tree estimation error. TCMM is run with $\lambda = 0$ (TCMM) and $\lambda = 10^{-3}$ (CASTLES+TCMM).

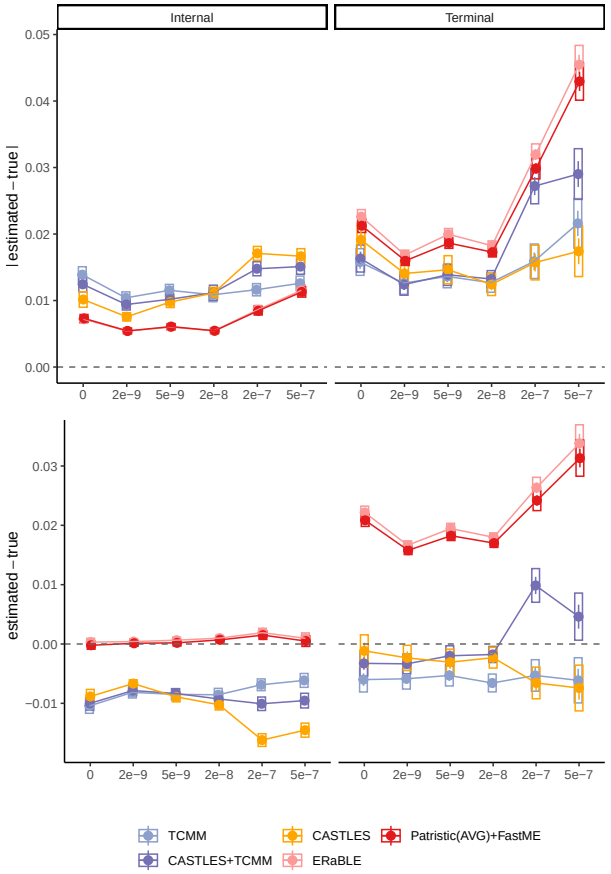

Figure 4: Comparing TCMM with the alternative methods on the HGT dataset with different rates of HGT (x-axis). TCMM is run with $\lambda = 0$ (TCMM) and $\lambda = 10^{-3}$ (CASTLES+TCMM).

eliminated.

Comparison to other methods. Considering both the S100 (Fig. 3) and HGT (Fig. 4) datasets, no single method was universally more accurate than others. The comparison depends on whether terminal or internal branch lengths are examined, the type of gene tree error, and whether bias or mean error was examined. The simple methods based on patristic distance (ERaBLE and AVG+FastME) exhibit the lowest error and bias on internal branch lengths on both datasets but have very high error and extreme over-estimation bias on terminal branch length. TCMM, which is also based on patristic distances, has a slightly higher error and under-estimation bias for internal branches; however, it has far less error and bias for terminal branches. Concatenation is consistently less accurate than both CASTLES and TCMM in terms of both error and bias, especially for terminal branches (Fig. S8).

TCMM and CASTLES have similar accuracy overall, with subtle differences across conditions (Figs. 3 and 4). Under conditions with ILS only, TCMM without regularization has 12.1–49.2% higher error for internal branches and 1.75–19.8% lower error for terminal ones. Increased gene tree error tends to make internal lengths longer for both CASTLES and TCMM, but it has a far greater impact on CASTLES, changing it from under-estimation

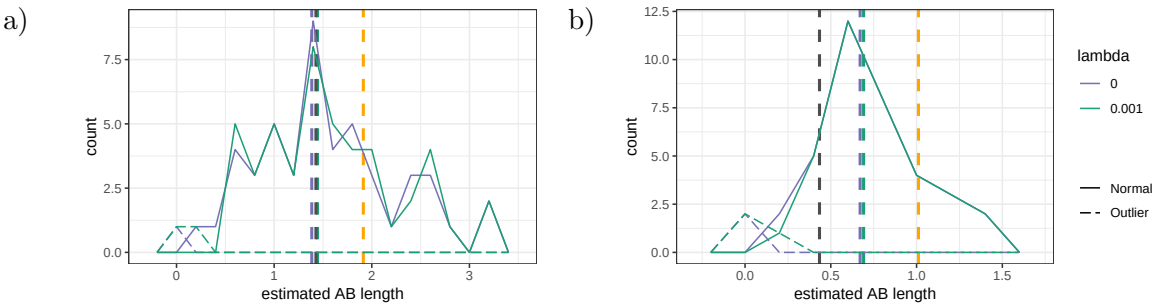

Figure 5: The distribution of the TCMM estimated length for the branch separating bacteria and archaea per gene tree using the 49 core gene (a) and the 38 non-ribosomal genes (b). Vertical dashed lines show the geometric mean estimate by TCMM, estimates of CASTLES (orange), concatenation (grey).

bias to over-estimation bias. The impact of HGT on error is erratic, but bias changes as expected (Fig. 4). The CASTLES internal branch lengths have an underestimation bias that, as expected, increases by 73.6% for the highest two levels of HGT. TCMM has a similar under-estimation bias for low HGT, but the bias does not increase with higher HGT. The absolute error also tends to increase with HGT for CASTLES while TCMM appears more robust. For terminal branches, CASTLES has a lower underestimation bias at low HGT levels but has a higher bias at the highest levels of HGT. Finally, over both datasets, TCMM +CASTLES (with regularization $\lambda = 10^{-3}$) tends to improve the accuracy compared to TCMM but not when the HGT levels are high.

AB branch length. On the biological dataset, ERaBLE fails to return an output and AVG+FastME returns an estimated tree with several long negative branches around the root; we exclude those methods and compare CASTLES, TCMM, and CASTLES+TCMM ($\lambda = 10^{-3}$). Using 49 core genes (including ribosomal genes) CASTLES estimates a length of 1.91 SU for the AB branch separating bacteria and archaea, while TCMM and CASTLES+TCMM estimate 1.39 and 1.44 SU, respectively. In contrast, on the 38 non-ribosomal genes (presumably with more HGT), the AB branch is much shorter, as [24] have found; TCMM estimates AB length to be 0.67 or 0.69 SU (depending on λ), which is higher than concatenation (0.43) but lower than CASTLES (1.01). The fact that both gene tree-based methods have higher length than concatenation for non-ribosomal genes is consistent with the observation by Moody et al. [24] that concatenation can under-estimate length in the face of HGT. However, the large difference between ribosomal and non-ribosomal genes even using methods that account for gene tree discordance supports the observation by Zhu et al. [39] that ribosomal genes have very different AB lengths than other proteins. In the per-gene mode, TCMM also estimates a separate branch length per each gene tree (Fig. 5). Examining the distribution shows that there is substantial diversity among genes, with some indicating an AB length of more than 3 SU, while others lead to a length below 0.5 SU. While our study alone is insufficient to settle this debate, we note that AB estimates based on ribosomal genes are distributed around those proposed by Moody et al. [24] while those based on non-ribosomal genes are closer to the Zhu et al. [39] estimate. We leave a more thorough exploration of this question with more marker genes and large trees for future work.

4 Discussion and future work

We have introduced TCMM, a versatile framework for reconciling branch lengths of a query tree to one or more reference trees. Our paper focused on a particular application, species tree branch length estimation, where we showed TCMM performs comparably to specialized tools designed for this purpose. Notably, TCMM does not directly model coalescence, in contrast to methods like CASTLES. A useful feature of our framework is that, through regularization, it can be used as a complementary step alongside alternative tools such as CASTLES. However, our empirical results showed that even without that pairing, the default TCMM manages to estimate species tree branch lengths without the anticipated overestimation bias due to gene tree coalescence events predating speciation events. Since using the arithmetic mean across genes did result in an over-estimation bias (Fig. 2a), the lack of over-estimation bias in the default version is likely due to the use of the geometric mean, which is always lower than the arithmetic mean. We leave the further theoretical question of whether geometric mean used with TCMM is biased to future work.

More broadly, the per-gene mode of TCMM provides a length *distribution* for each species tree branch. Allowing one rate per branch per gene is effectively modeling across-genes heterotachy in the evolutionary process [19], a feature missing from other methods; e.g., CASTLES models change of rates across the species tree but assumes all genes have the same rate for each branch. Our biological results showed evidence (Fig. 5) for such heterotachy at the root of the Tree of Life. Part of the reason why TCMM is as accurate as alternatives despite the lack of direct modeling of coalescence can also be its modeling of heterotachy.

While we focused on species tree branch length estimation, the TCMM framework can have wide-ranging applications beyond what we explored. The TCMM formulation may be used in embedding tree leaves in unified spaces, as has been done previously for detecting outlier leaves of gene trees [7]. The linear algebraic approach used here is effectively comparing two trees by matching one to the other. Unlike many commonly used phylogenetic metrics, this matching incorporates both tree topology and branch lengths. It appears that the results of the matching can be translated to new metrics of tree distance and perhaps new median tree problems. Future work should explore those metrics and whether they provide additional insights.

References

[1] Arnold, B. J., Huang, I.-T., and Hanage, W. P. (2022). Horizontal gene transfer and adaptive evolution in bacteria. *Nature Reviews Microbiology*, 20(4):206–218.

[2] Barthélemy, J.-P. and McMorris, F. R. (1986). The median procedure for n-trees. *Journal of Classification*, 3(2):329–334. Publisher: Springer.

[3] Binet, M., Gascuel, O., Scornavacca, C., P. Douzery, E. J., and Pardi, F. (2016). Fast and accurate branch lengths estimation for phylogenomic trees. *BMC Bioinformatics*, 17(1):23.

[4] Bogdanowicz, D. and Giaro, K. (2012). Matching split distance for unrooted binary phylogenetic trees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

[5] Boussau, B., Szöllősi, G. J., and Duret, L. (2013). Genome-scale coestimation of species and gene trees. *Genome Research*, 23(2):323–330.

[6] Davidson, R., Vachaspati, P., Mirarab, S., and Warnow, T. (2015). Phylogenomic species tree estimation in the presence of incomplete lineage sorting and horizontal gene transfer. *BMC Genomics*, 16(Suppl 10):S1.

[7] de Vienne, D. M., Ollier, S., and Aguileta, G. (2012). Phylo-MCOA: A Fast and Efficient Method to Detect Outlier Genes and Species in Phylogenomics Using Multiple Co-inertia Analysis. *Molecular Biology and Evolution*, 29(6):1587–1598.

[8] DeGiorgio, M. and Degnan, J. H. (2014). Robustness to Divergence Time Underestimation When Inferring Species Trees from Estimated Gene Trees. *Systematic Biology*, 63(1):66–82.

[9] Degnan, J. H. and Rosenberg, N. A. (2009). Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends in Ecology and Evolution*, 24(6):332–340. ISBN: 0169-5347 Publisher: Department of Human Genetics, University of Michigan, Ann Arbor, MI 48109, USA. j.degnan@math.canterbury.ac.nz.

[10] Doyon, J.-P. J., Ranwez, V., Daubin, V., and Berry, V. (2011). Models, algorithms and programs for phylogeny reconciliation. *Briefings in Bioinformatics*, 12(5):392–400.

[11] Estabrook, G. F., McMorris, F. R., and Meacham, C. A. (1985). Comparison of Undirected Phylogenetic Trees Based on Subtrees of Four Evolutionary Units. *Systematic Biology*, 34(2):193–200.

[12] Forthman, M., Braun, E. L., and Kimball, R. T. (2022). Gene tree quality affects empirical coalescent branch length estimation. *Zoologica Scripta*, 51(1):1–13.

[13] Heled, J. and Drummond, A. J. (2010). Bayesian inference of species trees from multilocus data. *Molecular Biology and Evolution*, 27(3):570–580.

[14] Kubatko, L. S., Carstens, B. C., and Knowles, L. L. (2009). STEM: species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics*, 25(7):971–973.

[15] Kumar, S. (2005). Molecular clocks: four decades of evolution. *Nature Reviews Genetics*, 6(8):654–662.

[16] Lefort, V., Desper, R., and Gascuel, O. (2015). FastME 2.0: A comprehensive, accurate, and fast distance-based phylogeny inference program. *Molecular Biology and Evolution*, 32(10):2798–2800. ISBN: 1537-1719 (Electronic)\r0737-4038 (Linking).

[17] Liu, L. (2008). BEST: Bayesian estimation of species trees under the coalescent model.

Bioinformatics, 24(21):2542–2543. ISBN: 1367-4803 Publisher: Musuem of Comparative Zoology, Harvard University, Cambridge, MA, USA. lliu@oeb.harvard.edu.

[18] Liu, L. and Yu, L. (2011). Estimating Species Trees from Unrooted Gene Trees. *Systematic Biology*, 60(5):661–667. ISBN: 1063-5157\r1076-836X.

[19] Lopez, P., Casane, D., and Philippe, H. (2002). Heterotachy, an Important Process of Protein Evolution. *Molecular Biology and Evolution*, 19(1):1–7.

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

[21] Mai, U. and Mirarab, S. (2020). Log Transformation Improves Dating of Phylogenies. *Molecular Biology and Evolution*, 38(3):1151–1167.

[22] Mallo, D., De Oliveira Martins, L., and Posada, D. (2016). SimPhy : Phylogenomic Simulation of Gene, Locus, and Species Trees. *Systematic Biology*, 65(2):334–344.

[23] Mirarab, S., Nakhleh, L., and Warnow, T. (2021). Multispecies Coalescent: Theory and Applications in Phylogenetics. *Annual Review of Ecology, Evolution, and Systematics*, 52(1):247–268.

[24] Moody, E. R., Mahendrarajah, T. A., Dombrowski, N., Clark, J. W., Petitjean, C., Offre, P., Szöllősi, G. J., Spang, A., and Williams, T. A. (2022). An estimate of the deepest branches of the tree of life from ancient vertically evolving genes. *eLife*, 11:e66695.

[25] Mossel, E. and Roch, S. (2010). Incomplete lineage sorting: consistent phylogeny estimation from multiple loci. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(1):166–171.

[26] Ogilvie, H. A., Bouckaert, R. R., and Drummond, A. J. (2017). StarBEAST2 Brings Faster Species Tree Inference and Accurate Estimates of Substitution Rates. *Molecular Biology and Evolution*, 34(8):2101–2114. ISBN: 0737-4038.

[27] Petitjean, C., Deschamps, P., López-García, P., and Moreira, D. (2015). Rooting the domain archaea by phylogenomic analysis supports the foundation of the new kingdom proteoarchaeota. *Genome biology and evolution*, 7(1):191–204.

[28] Price, M. N., Dehal, P. S., and Arkin, A. P. (2010). FastTree-2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*, 5(3):e9490. Publisher: Public Library of Science.

[29] Rasmussen, M. D. and Kellis, M. (2007). Accurate gene-tree reconstruction by learning gene- and species-specific substitution rates across multiple complete genomes. *Genome Research*, 17(12):1932–1942. ISBN: 1088-9051.

[30] Robinson, D. and Foulds, L. (1981). Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147.

[31] Sayyari, E. and Mirarab, S. (2016). Fast Coalescent-Based Computation of Local Branch Support from Quartet Frequencies. *Molecular Biology and Evolution*, 33(7):1654–1668.

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

[33] Tabatabaee, Y., Zhang, C., Warnow, T., and Mirarab, S. (2023). Phylogenomic branch length estimation using quartets. *Bioinformatics*, 39(Supplement_1):i185–i193.

[34] Vachaspati, P. and Warnow, T. (2015). ASTRID: Accurate Species TRees from Internode Distances. *BMC Genomics*, 16(Suppl 10):S3. Publisher: BioMed Central Ltd.

[35] Wang, C. and Xiu, N. (2000). Convergence of the gradient projection method for generalized convex minimization. *Computational optimization and applications*, 16(2):111–120.

[36] Williams, T. A., Cox, C. J., Foster, P. G., Szöllősi, G. J., and Embley, T. M. (2020). Phylogenomics provides robust support for a two-domains tree of life. *Nature ecology & evolution*, 4(1):138–147.

[37] Yang, J. and Warnow, T. (2011). Fast and accurate methods for phylogenomic analyses. *BMC Bioinformatics*, 12(Suppl 9):S4. Publisher: BioMed Central Ltd.

[38] Zhang, C., Rabiee, M., Sayyari, E., and Mirarab, S. (2018). ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC Bioinformatics*, 19(S6):153.

[39] Zhu, Q., Mai, U., Pfeiffer, W., Janssen, S., Asnicar, F., Sanders, J. G., Belda-Ferre, P., Al-Ghalith, G. A., Kopylova, E., McDonald, D., Kosciolek, T., Yin, J. B., Huang, S., Salam, N., Jiao, J.-Y., Wu, Z., Xu, Z. Z., Cantrell, K., Yang, Y., Sayyari, E., Rabiee, M., Morton, J. T., Podell1, S., Knights, D., Li1, W.-J., Huttenhower, C., Segata, N., Smarr, L., Mirarab, S., and Knight, R. (2019). WoL: reference phylogeny for microbes (data pre-release).

[40] Zimmermann, T., Mirarab, S., and Warnow, T. (2014). BBCA: Improving the scalability of *BEAST using random binning. *BMC genomics*, 15(Suppl 6):S11.

[41] Zuckerkandl, E. and Pauling, L. (1962). Molecular disease, evolution, and genic heterogeneity. *Horizons in biochemistry*, pages 189–225.

Supplementary figures and tables

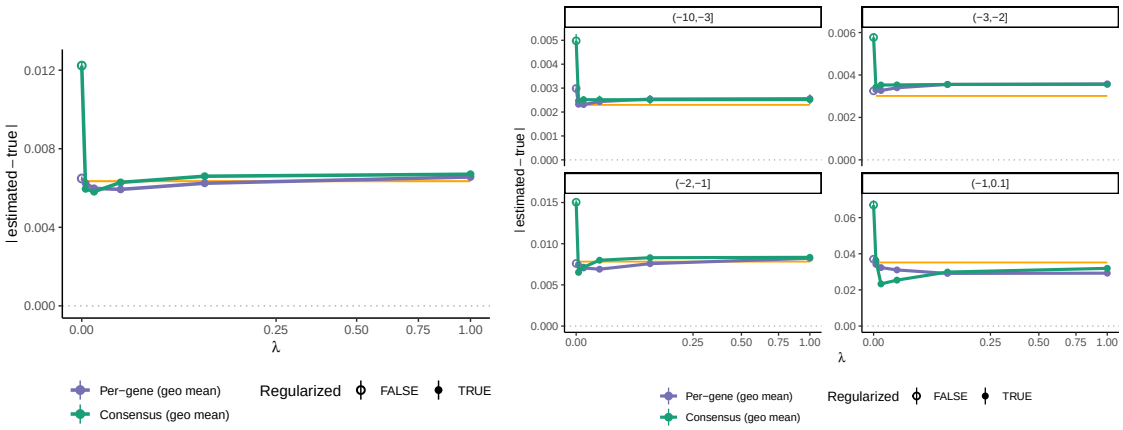

Figure S6: (S100 Dataset) Error of different lambda values for the two methods per gene and consensus. Orange line indicates CASTLES. For $\lambda \neq 0$ CASTLES branch lengths were used in the regularization term. Plots are partitioned based on $\log_{10}(true)$

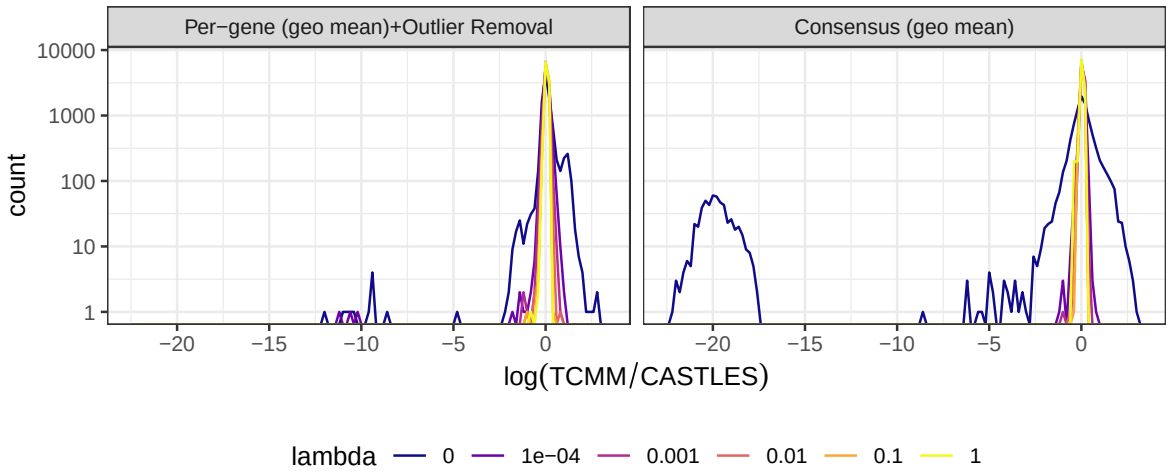

Figure S7: The distribution of the estimated branch lengths by two version of TCMM divided by the CASTLES branch lengths.

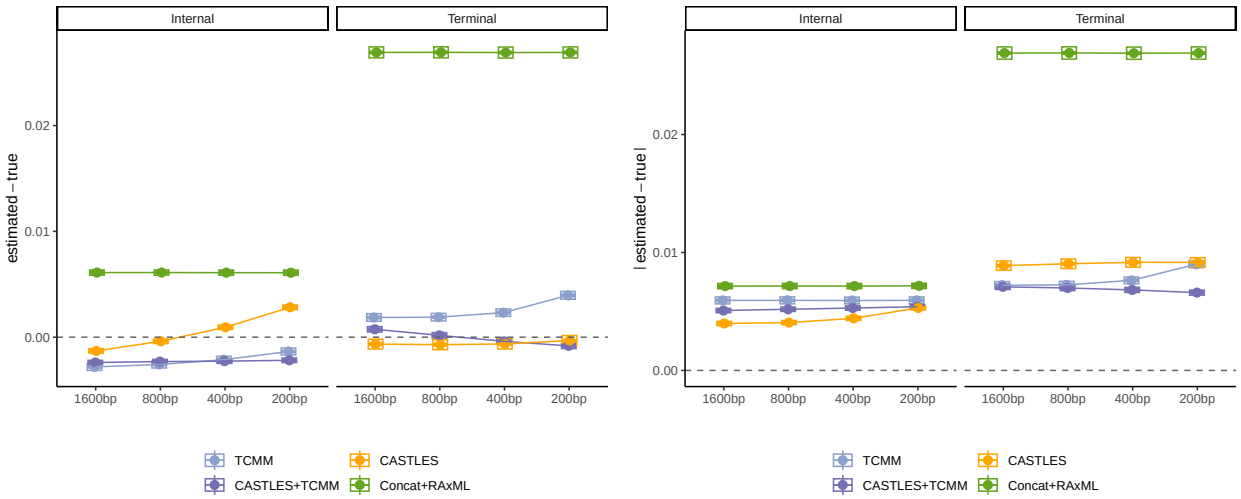

Figure S8: (S100 Dataset) Comparing TCMM and CASTLES with concatenation method. TCMM is run with $\lambda = 0$ (TCMM) and $\lambda = 10^{-3}$ (CASTLES+TCMM).