MaxTiC: Fast ranking of a phylogenetic tree by Maximum Time Consistency with lateral gene Transfers

Cédric Chauve[1], Akbar Rafiey[1], Adrian A. Davin[3], Celine Scornavacca[4], Philippe Veber[3], Bastien Boussau[3], Gergely J Szöllősi[5,6], Vincent Daubin[3], and Eric Tannier[2,3]

[1]Department of Mathematics, Simon Fraser University, Burnaby (BC), Canada
[2]Inria Grenoble Rhône-Alpes, F-38334 Montbonnot, France
[3]Univ Lyon, Université Lyon 1, CNRS, Laboratoire de Biométrie et Biologie Évolutive UMR5558, F-69622 Villeurbanne, France
[4]Institut des Sciences de l'Evolution, Université de Montpellier, CNRS, IRD, EPHE 34095 Montpellier Cedex 5, France
[5]MTA-ELTE "Lendulet" Evolutionary Genomics Research Group, Budapest Hungary
[6]Deptartment of Biological Physics, Eötvos Loránd University, Budapest Hungary

April 14, 2017

Abstract

Lateral gene transfers (LGTs) between ancient species contain information about the relative timing of species diversification. Specifically, the ancestors of a donor species must have existed before the descendants of the recipient species. Hence, the detection of a LGT event can be translated into a time constraint between nodes of a phylogeny if donors and recipients can be identified. When a set of LGTs are detected by interpreting the phylogenetic discordance between gene trees and a species tree, the set of all deduced time constraints can be used to order totally the internal nodes and thus produce a ranked tree. Unfortunately LGT detection is still very challenging and all methods produce some proportion of false positives. As a result the set of time constraints is not always compatible with a ranked species tree. We propose an optimization method called MaxTiC (Maximum Time Consistency) for obtaining a ranked species tree that is compatible with a maximum number of time constraints. We give in particular an exact polynomial time method based on dynamic programming to compute an optimal ranked binary tree supposing that a ranked subtree is given and fixed below each of the two children. We turn this principle into a heuristic to solve the general problem and test it on simulated datasets. Under a wide range of conditions, the obtained ranked tree is very close to the real one, confirming the theoretical possibility of dating with transfers by maximizing time consistency.

1 Introduction

Telling the evolutionary time [3] is usually achieved by combining molecular clocks and the fossil record. It was pointed out by Gogarten [5] and demonstrated by Szöllősi *et al* [12] that there existed a third source of information about evolutionary time in ancient lateral gene transfers.

Indeed, suppose an ancient species A transfers a gene to another species B, and the latter has descendants $\mathcal{B}$ that are sampled in a phylogenetic study (a necessary condition for the transfer to be detected). It is not necessary to assume the same for A, that is, that A has descendants in the phylogeny [14]. If we call X the most recent common ancestor of A and sampled species, and Y the most recent common ancestor of the species in $\mathcal{B}$, then X has to be older than Y because a gene from a descendant of X has been transferred to an ancestor of Y (see Figure 1).

While a single transfer can provide a time constraint between two nodes of a phylogeny, many transfers combined can provide a multitude of time constraints that can be used to determine the time order of the internal nodes of a phylogeny and obtain a *ranked phylogeny* [10].

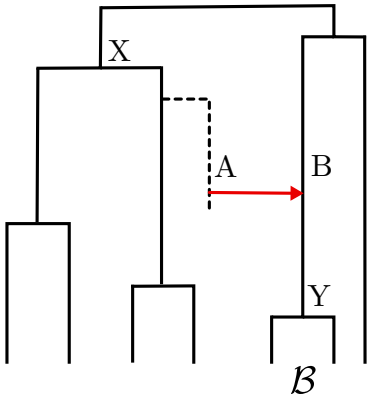

Figure 1: The dating information in transfers. A species tree is depicted, with a transfer from species A to contemporaneous species B. As it is likely, A belongs to a lineage with no sampled descendants (dotted line in the phylogeny), because it is unreasonable to assume that transfers happen between species that are in the phylogeny. The transfer from A to B informs that speciation X is older than speciation Y.

Such an approach, however, requires that the direction of lateral gene transfers events be specified, which can be challenging [9]. The method by Szöllősi *et al* [12] consisted in searching in the space of ranked trees the one that has the best likelihood according to a model of gene tree species tree reconciliation taking lateral gene transfers into account. Due to the size of the space, it does not scale up to more than a few dozen species.

Here, we describe a fast method to compute a ranked species tree from LGTs detected on an unranked tree. Several pieces of software are available to detect transfers using phylogenetic incongruence between species trees and gene trees without the need of a ranked species tree [1, 11, 15, 6]. We transform transfers into time constraints, and the set of time constraints into a total order.

If all detected transfers were real, this would be the end of the story. Indeed, all transfers would be compatible with the real chronology. Finding a ranked tree agreeing with a set of constraints is trivial if there is one order agreeing with *all* constraints. However, due to errors or uncertainties in the output of any method, the set of time constraints inferred from transfers is not necessarily entirely compatible with a total order of the species tree nodes. In practice it is never the case. Conflict may be due to errors in the species tree, imprecisions in gene tree reconstructions, phylogenetic artifacts, imprecisions in species tree gene tree reconciliations, or insufficiencies of reconciliation models often ignoring events like incomplete lineage sorting or transfers with replacement of an homologous gene. We then propose to compute a ranked tree which maximizes an agreement with a set of constraints, a particular case of the FeedBack Arc Set problem. We describe a method called MaxTiC, for Maximal Time Consistency, based on a divide and conquer principle. The divide step consists in solving the problem for subtrees of the species tree. The conquer step consists in exactly solving by dynamic programming the particular case in which a total order on nodes is given for each of the two children subtrees.

This conquer step can also be seen as a general method to mix two parts of a species tree which have been independently dated, provided transfers have been detected between the two clades.

We test the whole method (transfer detection by ALE [15] + MaxTiC) on a benchmark of simulated data generated by SimPhy [8]. We use a wide range of number of gene families, transfer rates and population sizes (which has an effect on the gene tree species tree incongruence through incomplete lineage sorting), to test the limits of the principle. We show that under most conditions, the ranked tree recovered by the method is very close to the true one (A normalized Kendall τ close to 0.95), but is never exactly the true one because of false transfers inferred by ALE.

We first describe the protocol, including simulations, transfer detection, conversion of each transfer to a time constraint. Then we describe our main algorithm, the exact dynamic programming procedure on a subproblem and how we use it as a heuristic for the general problem. We finally present the results on the simulated datasets.

2 Method

2.1 Generalities.

We consider that phylogenetic trees are binary and species trees are rooted. In a species tree a node x is the descendant of a node y, or equivalently y is an ancestor of x if y is on the path from the root to x. We note this relation $x \leq y$, and it defines a partial order on the nodes. A species tree is *ranked* if there is a total order of its internal nodes which generalizes the partial order given by the tree. Gene trees can be rooted or not, and each of their leaves maps to a leaf of a species tree. Reconciled gene trees are rooted and annotated gene trees, where every node maps to nodes or branches of the species tree and is annotated with a speciation, duplication or transfer event [16].

2.2 Simulation protocol

Simulation by SimPhy. We generated simulated dataset with an independent piece of software[1]. For all sets of parameters, we used Symphy [8] to generate a ranked species tree with 500 leaves. Along this species tree, we generate typically 1000 gene trees with a population size between 2 and 10^6, null rates of duplications and losses, and a rate of transfers from 10^{-9} to 10^{-5}.

Then we pruned each leaf of the species tree with a probability 0.8, so that the final species tree has approximately 100 leaves. Gene trees are pruned accordingly by removing leaves belonging to the removed species. This simulates a sampling of sequenced species, accounting for species extinction or species absence in the study.

Detection of transfers. Transfers are detected by ALEundated [15], which takes as input an unranked rooted species tree and an unrooted gene tree, and produces a sample of 100 reconciled gene trees, sampled according to their likelihood under a model of duplication, loss, and transfers. Transfers from unsampled lineages are handled [13]. We kept transfers found in at least 5% of the reconciliations, in order to reduce the noise from improbable transfers.

From transfers to constraints. Each transfer inferred by ALE has an ancestor of the donor species and descendants of the recipient species in the phylogeny. The most recent species in the phylogeny which is an ancestor of the donor is called X, the first descendant of the recipient is called Y, and a constraint is inferred as $X > Y$, which means X is older than Y. We assign to the constraint $X > Y$ the support of the transfer, which is the frequency at which the constraint $X > Y$ is found in the 100 reconciled gene trees, summed across all gene families.

2.3 Finding a maximum consistent set of constraints

Definition. We have as input an unranked rooted species tree, which is the ranked simulated species tree from which we delete the total order information, and a large set of constraints with weights $\mathcal{C}$ inferred from transfers. Some constraints might be conflicting, for example like in Figure 2. Y is found to be older than X, and Z is found to be older than T, but T is an ancestor of Y and X is an ancestor of Z.

A subset of $\mathcal{C}$ is said to be *time consistent* if there exists a ranked species tree for which all constraints are compatible with the total order. We search for a maximum weight time consistent subset of $\mathcal{C}$.

Relation with the Feedback Arc Set. If we see the branches of the unranked species tree as arcs of a directed graph with infinite weight, and the constraints as weighted arcs in this graph, then this problem translates exactly in an instance of the Feedback Arc Set problem. This classical problem is NP-complete [4], and cannot be approximated with a constant ratio. The best algorithms to solve it in practice are local search heuristics. It is equivalent to finding a total order of the nodes of a directed graph, which maximizes the total sum of the arcs xy such that $x > y$ in this order.

[1] Independent meaning that it was developed by an independent team, with other purposes than to test our method. However it has been developed to validate inference methods in general, which is a kind of dependency [2].

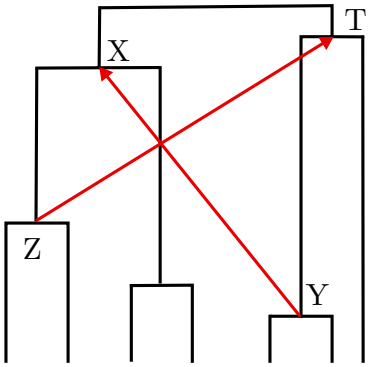

Figure 2: A set of two conflicting constraints. Each of the constraints $Y > X$ and $Z > T$ can be fulfilled by some ranked version of the species tree, but not both.

NP-hardness. Note that as we have a species tree with infinite weight arcs, we are not in the general case of the Feedback Arc Set problem, so the NP-completeness of our variant is not immediate. However it is easy to reduce the Feedback Arc Set to our problem, leading to the NP-hardness property.

Theorem 1. *The maximum time consistency problem is NP-hard.*

Proof. Let us take any instance of the Feedback Arc Set in the form of a weighted graph with n vertices. Construct a species tree with $2n$ leaves, connected by n cherry nodes (i.e. nodes having two leaves as children), and complete the rest of the tree by a comb. The cherry nodes are identified with the nodes of the graph, so that any arc can be assimilated to a constraint, and a ranked species tree maximizing the set of compatible constraints yields a total order of the vertices of the initial graph maximizing the consistency with the arcs. Any algorithm finding a maximum time consistent set of contraints, applied on the comb with cherries, would find the solution to the feedback arc set. This proves NP-hardness of the maximum time consistency problem. □

A heuristic principle based on divide-and-conquer approximations. Specificities of our problem compared to the Feedback Arc Set can be harnessed to design specific heuristics. Feedback Arc Set is approximable within a factor of $\log n$ where n is the size of the graph. The approximation ratio is obtained by a divide and conquer strategy, first cutting the graph into two balanced parts, solving recursively the two parts and then mixing the two subsolutions [7]. The presence of an underlying tree for the graph (the species tree) provides a "natural" way to recursively cut the graph into two. Indeed, let r be the root of the species tree. It is always the highest node in any ranked tree. Then define c_1 and c_2 the two children of r (descendants separated by only one edge), and t_1 and t_2 the two subtrees rooted at c_1 and c_2. Define three sets of constraints: those having two extremities in t_1, those having two extremities in t_2, and those having one extremity in t_1 and one in t_2. t_1 and the first set on constraints, as well as t_2 and the second, define new instances of the problem. So the divide step is to solve independently and recursively the problem on these two instances, providing ranked trees for t_1 and t_2, that is, two independent total orders of the internal nodes of t_1 and t_2. Providing an order of all the internal nodes, that is, containing r, the internal nodes of t_1 and the internal nodes of t_2, according to the third set of constraints, is the mixing (conquer) step.

The mixing principle In Leighton and Rao [7], the mixing step was achieved by concatenating the two obtained orders obtained from the solutions to the two subproblems. We propose here a better (optimal) way to achieve this mixing by dynamic programming. We solve exactly the mixing problem by an algorithm which is valid for ranking species trees as well as solving the Feedback Arc Set, which improves on the approximation solutions to the general Feedback Arc Set problem (the approximation ratio however is not improved).

Recall that as a result of the divide step the node set of the species tree is split into three parts, the root r, and the two children subtrees t_1 and t_2. Suppose that a total order of the nodes is given in t_1,

$a_1, \ldots, a_k$ and t_2, $b_1, \ldots, b_l$, as the result of a recursive application of the algorithm. Suppose also that the set of constraints C is only composed of constraints with one extremity in t_1, and one in t_2 (all other constraints do not affect the solution if we suppose a total order in t_1 and t_2).

A total order on the whole tree, respecting the orders inside t_1 and t_2, and given this constraint, maximizing the total weight of the subset of compatible constraints of $\mathcal{C}$ can be obtained in polynomial time thanks to a recursive formula. Given a subset S of the internal nodes of the species tree, note $\mathcal{C}_S$ the set of constraints which have both their extremities in S. If $S_{ij} = \{a_i, \ldots, a_k, b_j, \ldots, b_l\}$, note $s(i,j)$ the size of the maximum set of compatible time constraints in $\mathcal{C}_{S_{ij}}$, also compatible with the orders $a_i, \ldots, a_k$ and $b_j, \ldots, b_l$. We are interested in the value of $s(1,1)$, but we can compute it recursively with:

- $s(k+1, j) = s(i, l+1) = 0$ for all i, j
- $s(i,j) = min(s(i+1,j) + incoming(a_i), s(i, j+1) + incoming(b_j))$, if $i \leq k$ and $j \leq l$

where $incoming(x)$ is the total weight of the constraints ending on x.

This translates into a dynamic programming scheme. Backtracking along the matrix of $s(i,j)$ gives the optimal mixing of the two orders $a_1, \ldots, a_k$ and $b_1, \ldots, b_l$. Putting r before the mixed order gives the final solution.

Implementation. In our piece of software MaxTiC, we implemented in Python the heuristic recursive principle just described, plus a greedy heuristic and a local search. The greedy heuristic consists in progressively adding to the species tree the constraints in decreasing order of their weight, provided that they do not create conflict with has already been added. The local search consists in proposing a move to the total order of the species tree nodes, by taking one node at random and changing its position in the total order to a randomly chosen alternative one, and accepting the move if it is compatible with the partial order given by the species tree and if it increases the value of the solution.

We tested this program on simulated data, taking the best solution out of the greedy one and the heuristic one, and applying on it the local search during three minutes.

3 Results

For each experiment we computed the best ranked trees according to the constraints computed from transfers. Most of the time the mixing heuristic was giving a better solution than the greedy heuristic, and the local search could improve the solution by a few percents.

First, to give an idea of the value of the optimal solution and the amount of conflicting constraints in a typical set of constraints computed from transfers we plot in Figure 3 the fraction of constraints that have to be removed in order to get a compatible set, as a function of the transfer rate (black points). We compare this value to the fraction of the constraints conflicting with the true (simulated) ranked species tree (red points). We see that the values on reconstructed node orders are close and always a bit under the true values. This justifies the minimizing approach: the true conflict is close to the minimum. However as the optimum is always lower than the true value, it also shows that discrepancies to the truth are not due to limitations in the optimization algorithm but in the model itself. The small difference is probably due to overfitting of artefactual constraints.

We measure the accuracy of the method empirically by comparing the true (simulated) ranked tree with the obtained ranked tree and computing the Kendall τ distance. The Kendall τ distance between two orders is the number of pairs i, j of elements of the two orders such that i is before j in one order, and j is before i in another. We normalize this number by the maximum possible Kendall distance given that the two orders have to be compatible with the species tree, to get a number between 0 and 1 (0 for the maximum distance between orders given a species tree, 1 for two equal orders). To compute the maximum Kendall distance between two linear extensions of a partial order determined by a tree we use the following property.

Property 1. *Given a rooted tree T inducing a partial order P on its internal nodes, two depth first search of T, ordering the children of any node in, respectively lexicographical and anti-lexicographical order, output two linear extensions of P such that their Kendall distance is maximum, among all pairs of linear extensions of P.*

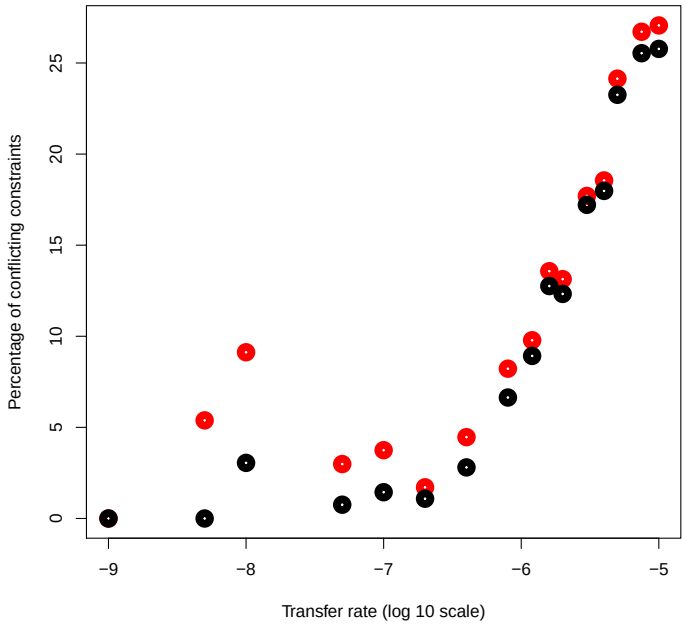

Figure 3: Fraction of constraints that have to be removed in order to get a time compatible set, as a function of the transfer rate ($\log_{10}$ scale). Red dots are for the fraction of constraints in conflict with the true (simulated) tree, and black dots are for the fraction of constraints in conflict with the reconstructed tree, minimizing the conflicts.

This property is easy to demonstrate: take any pair i, j of internal nodes of a rooted tree. Either one is the ancestor of the other and they appear in the same order in any pair of linear extensions. Or they are incomparable, with a last common ancestor a, having children a_1, the ancestor or i, and a_2, the ancestor of j. In one depth first search a_1 and its descendants, including i, appear before a_2 and its descendants, including j, and in the other it is the opposite. So all incomparable pairs appear in a different order, contributing to the Kendall distance. This obviously gives the maximum possible Kendall distance.

We give an idea of how many gene trees (how many transfers) are necessary to get the dating information. In Figure 4 (right), we plot the Kendall similarity between the true tree and the obtained tree, as a function of the number of gene trees, for a constant transfer rate of 1.6×10^{-6}.

We see that the method starts with a very low similarity if there are not enough gene trees, which is expected as in the absence of transfers there is no information to infer the ranked tree. Then the similarity is quickly going up, almost reaching a plateau from about 400 families, then slowly increasing up to 5000. This means that the more gene trees are available, the best the result will be, but with little gain after 1000 gene trees.

We then investigated the effect of the transfer rate on the accuracy of the result. We measured the normalized Kendall similarity as a function of the average number of transfers per gene family (this number is computed on the 500 leaves species tree, so the number of inferred transfers is much smaller). The results are shown on Figure 5. As expected, too few transfers give a low quality result, because of a lack of signal, and too many transfers make the similarity to the true node order decrease. However the slopes are very different : whereas a few transfers are sufficient to give a good ranked tree, the ranked tree stays reasonably good even with a huge number of transfers (several dozens per family).

Note however that in any conditions, the normalized Kendall similarity to true trees stays bounded at 95%, and under almost all conditions, it is between 90% and 95%. So it is possible, with ALE to detect transfers, to get a result close to the real order of speciations in a wide range of conditions, but

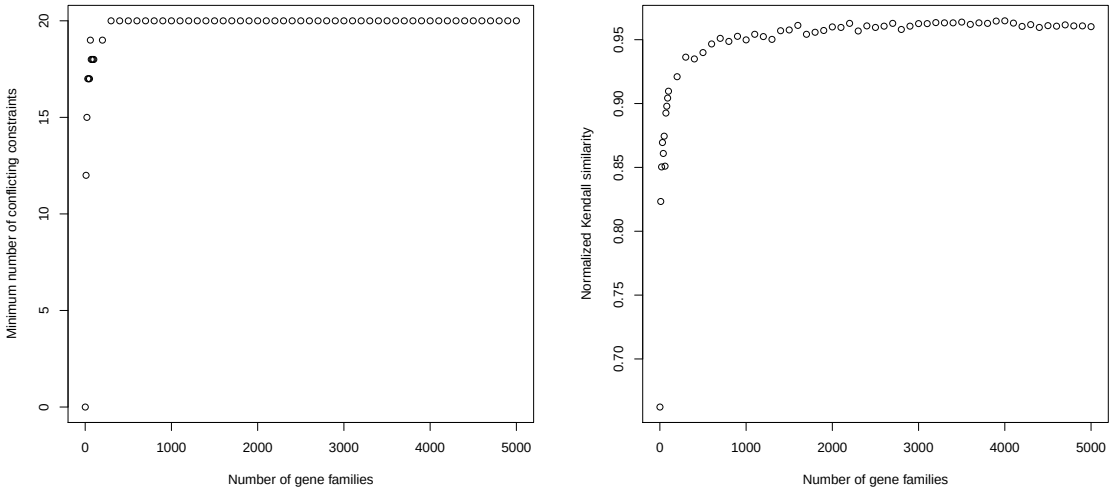

Figure 4: Left: Minimum fraction of the conflicting constraints to be removed as a function of the number of gene trees. Right: Normalized Kendall similarity of the true ranked tree and the obtained ranked tree, as a function of the number of gene trees in the experiment.

the real order seems never to be found.

Finally we examine the effect of gene tree uncertainties (Figure 6). In Simphy it is possible to vary the population size, and with the population size the probability of incomplete lineage sorting (ILS) increases. ALE does not handle ILS, so every supported ILS will be interpreted as duplications or transfers. So we use increasing population size as a general proxy for systematic errors in gene trees or processes not modeled by ALE. We see the expected tendency of the ability of MaxTic to infer the true tree decreasing with the increase in population size.

4 Conclusion

We give a proof of principle of a method to get a ranked species tree with the information of transfers. We present a method and a piece of software, called MaxTiC for Maximum Time Consistency, taking an unranked species tree as input, together with a set of possibly conflicting weighted time constraints, and outputting a ranked tree maximizing the total weight of a compatible subset of constraints. We validate this principle for dating on simulations from an independent (developed by an independent team, with different aims) genome simulator Simphy. The results confirm the principle of the possibility to date with transfers, thus introducing an additional source of information compared to dated fossils and the (relaxed) molecular clock. It is all the more important since the fossil record is poor or difficult to interpret precisely in clades where transfers are abundant.

References

[1] Mukul S. Bansal, Eric J. Alm, and Manolis Kellis. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics*, 28(12):i283–i291, Jun 2012.

[2] Priscila Biller, Carole Knibbe, Guillaume Beslon, and Eric Tannier. Comparative genomics on artificial life. In *Computability in Europe*, Lecture Notes in Computer Science, 2016.

[3] PCJ Donoghue and MP Smith, editors. *Telling the evolutionary time*. CRC press, 2003.

[4] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

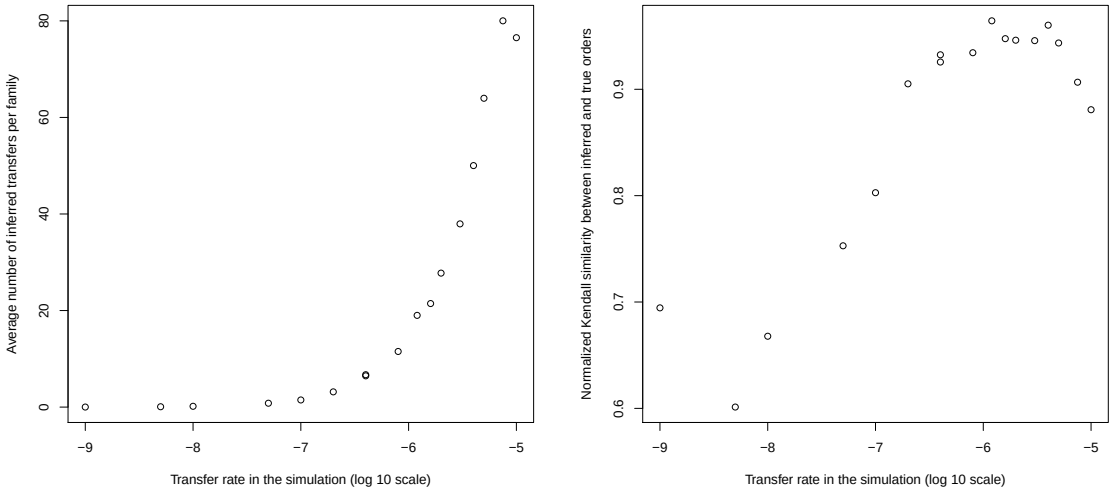

Figure 5: Left: Number of inferred transfers, as a function of the transfer rates in the simulations. Right: Normalized Kendall similarity of the true ranked tree and the obtained ranked tree, as a function of transfer rates ($\log_{10}$ scale).

[5] J P Gogarten, R D Murphey, and L Olendzenski. Horizontal gene transfer: pitfalls and promises. *The Biological bulletin*, 196:359–61; discussion 361–2, June 1999.

[6] Edwin Jacox, Cedric Chauve, Gergely J. Szöllősi, Yann Ponty, and Celine Scornavacca. eccetera: comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics*, Feb 2016.

[7] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proc. 1988] 29th Annual Symp. Foundations of Computer Science [*, pages 422–431, October 1988.

[8] Diego Mallo, Leonardo De Oliveira Martins, and David Posada. Simphy: Phylogenomic simulation of gene, locus, and species trees. *Syst Biol*, 65(2):334–344, Mar 2016.

[9] Matt Ravenhall, Nives Škunca, Florent Lassalle, and Christophe Dessimoz. Inferring horizontal gene transfer. *PLoS Comput Biol*, 11(5):e1004095, May 2015.

[10] C. Semple and M.A. Steel. *Phylogenetics*. Oxford lecture series in mathematics and its applications. Oxford University Press, 2003.

[11] Maureen Stolzer, Han Lai, Minli Xu, Deepa Sathaye, Benjamin Vernot, and Dannie Durand. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics*, 28(18):i409–i415, Sep 2012.

[12] Gergely J. Szöllosi, Bastien Boussau, Sophie S. Abby, Eric Tannier, and Vincent Daubin. Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. *Proc Natl Acad Sci U S A*, 109(43):17513–17518, Oct 2012.

[13] Gergely J. Szöllõsi, Wojciech Rosikiewicz, Bastien Boussau, Eric Tannier, and Vincent Daubin. Efficient exploration of the space of reconciled gene trees. *Syst Biol*, 62(6):901–912, Nov 2013.

[14] Gergely J. Szöllosi, Eric Tannier, Nicolas Lartillot, and Vincent Daubin. Lateral gene transfer from the dead. *Syst Biol*, 62(3):386–397, May 2013.

[15] Gergely J. Szöllősi, Adrián Arellano Davín, Eric Tannier, Vincent Daubin, and Bastien Boussau. Genome-scale phylogenetic analysis finds extensive gene transfer among fungi. *Philos Trans R Soc Lond B Biol Sci*, 370(1678):20140335, Sep 2015.

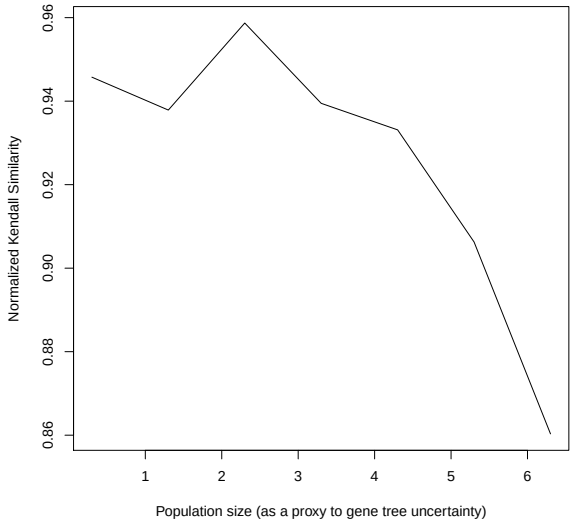

Figure 6: Normalized Kendall similarity of the true ranked tree and the obtained ranked tree, as a function of population size ($\log_{10}$ scale). Population size favors incomplete lineage sorting in SimPhy, so it is used here as a proxy for errors in phylogenetic reconstruction.

[16] Gergely J. Szöllősi, Eric Tannier, Vincent Daubin, and Bastien Boussau. The inference of gene trees with species trees. *Syst Biol*, 64(1):e42–e62, Jan 2015.