




Average-Tree Phylogenetic Diversity of Networks

Leo van Iersel   




TU Delft, The Netherlands

Mark Jones   

TU Delft, The Netherlands

Jannik Schestag   

TU Delft, The Netherlands

Celine Scornavacca   

ISEM, Université de Montpellier, CNRS, IRD, EPHE, Montpellier, France

Mathias Weller  

LIGM, CNRS, Université Gustave Eiffel, Marne-la-Vallée, France

Abstract

Phylogenetic diversity is a measure used to quantify the biodiversity of a set of species. Here, we introduce the “average-tree” phylogenetic diversity score in rooted binary phylogenetic networks and consider algorithms for computing and maximizing the score on a given network. Basically, the score is the weighted average of the phylogenetic diversity scores in all trees displayed by the network, with the weights determined by the inheritance probabilities on the reticulation edges used in the embeddings. We show that computing the score of a given set of taxa in a given network is #P-hard, directly implying #P-hardness of finding a subset of k taxa achieving maximum diversity score and, thereby, ruling out polynomial-time algorithms for these problems unless the polynomial hierarchy collapses. However, we show that both problems can be solved efficiently if the input network is close to being a tree in the sense that its reticulation number is small. More precisely, we prove that we can solve the optimization problem in networks with n leaves and r reticulations in $2^{\mathcal{O}(r)} \cdot n \cdot k$ time. Using experiments on data produced by simulating a reticulate-evolution process, we show that our algorithm runs efficiently on networks with hundreds of taxa and tens of reticulations.

2012 ACM Subject Classification Applied computing → Computational biology; Theory of computation → Fixed parameter tractability; Theory of computation → Problems, reductions and completeness

Keywords and phrases phylogenetic diversity, phylogenetic networks, network phylogenetic diversity, algorithms, computational complexity

Digital Object Identifier 10.4230/LIPIcs.WABI.2025.15

Funding *Leo van Iersel*: Partially funded by the Dutch Organisation for Scientific Research (NWO) grant OCENW.KLEIN.125 and OCENW.M.21.306.

Mark Jones: Partially funded by the Dutch Organisation for Scientific Research (NWO) grant OCENW.KLEIN.125 and OCENW.M.21.306.

Jannik Schestag: Partially funded by the Dutch Research Council (NWO), project “Optimization for and with Machine Learning (OPTIMAL)”, OCENW.GROOT.2019.015.

Celine Scornavacca: Partially funded by French Agence Nationale de la Recherche through the CoCoAlSeq Project (ANR-19-CE45-0012). This is the contribution ISEM 2025x-XXX of the Institut des Sciences de l’Evolution de Montpellier.

1 Introduction

Human-driven habitat degradation and environmental change have led to accelerated rates of species extinction, posing a significant threat to global biodiversity [18]. In response, conservation efforts face both logistical constraints – such as limited financial resources –



© Leo van Iersel, Mark Jones, Jannik Schestag, Celine Scornavacca, and Mathias Weller; licensed under Creative Commons License CC-BY 4.0

25th International Conference on Algorithms for Bioinformatics (WABI 2025).

Editors: Broňa Brejová and Rob Patro; Article No. 15; pp. 15:1–15:20

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and conceptual challenges, notably in identifying effective prioritization strategies. This has motivated the development of formal criteria and quantitative indicators to guide decision-making and assess biodiversity in a systematic and reproducible manner. Here, we study formal measures for quantifying the biodiversity of a given set of species and algorithms for computing and maximizing these measures.

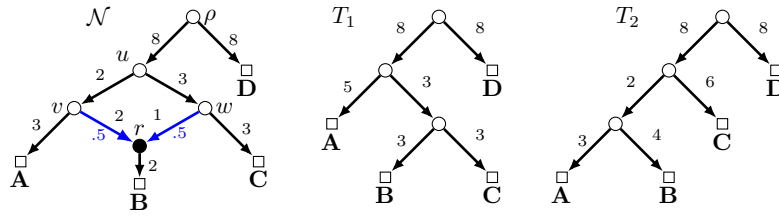
Phylogenetic Diversity (PD) is a well-studied measure used to quantify the biodiversity of a set of species using their evolutionary relationships [22]. When these relationships are described by a rooted phylogenetic tree (with edge lengths) on a set X of species, the phylogenetic diversity of a subset of the species $A \subseteq X$ is simply the total length of all edges that lie on at least one path from the root of the tree to a leaf in A [8]. The main intuition behind this is that species that are evolutionarily further apart together contribute more diversity than species that are evolutionarily closely related. Phylogenetic diversity can more formally be interpreted as follows in terms of features (e.g., genes or morphological features). Suppose that (1) the number of features introduced on each edge of the tree is proportional to the length of that edge, (2) features are never lost, and (3) no feature is introduced independently on multiple branches. Then, the number of features in A is proportional to its phylogenetic diversity. While the assumptions on the model may seem somewhat unrealistic, it has been used prominently in the literature [31, 3, 5, 24].

In many cases, however, describing evolutionary relationships by trees is too simplistic [7]. Rooted phylogenetic networks generalize rooted phylogenetic trees and are more suited to represent complex evolutionary scenarios involving, for example, hybridization events [12, 1, 26]. Roughly speaking, a rooted phylogenetic network is a directed acyclic graph with a single root and labeled leaves. Vertices with two (or more, in the nonbinary case) incoming arcs are called *reticulations* and can be used to model hybridization events, lateral gene transfer, or other types of horizontal evolutionary events. Generalizing the notion of phylogenetic diversity to networks is not obvious, and multiple variants have recently been proposed for rooted [32, 3, 5] and unrooted [20, 2, 4] networks.

The currently best-studied variants are ALLPATHSPD and NETWORKPD [3, 14, 27, 5, 24]. Both definitions generalize PD on trees, in the following ways. AllPathsPD(A) is the total length of all edges that lie on at least one path from the root of the network to a leaf in A . NETWORKPD generalizes ALLPATHSPD by allowing arcs to have inheritance probabilities, representing the probability that a given feature in the parent is inherited by the child. The idea of NetworkPD(A) is, roughly speaking, to compute the expected number of distinct features in A by using the inheritance probabilities to compute, for each arc uv , the probability γ_{uv} that at least one copy of a feature introduced on the arc uv is inherited by a leaf in A . More precisely, γ_{uv} is defined as γ_{vw} times the inheritance probability of uv if v is a reticulation with child w , as $1 - \prod_i (1 - \gamma_{vw_i})$ if v is a tree node with children w_i , and as 1 or 0 if v is a leaf, depending on whether or not $v \in A$.

Note that, in the NETWORKPD setting, features are inherited independently from different parents. For example, in the network in Figure 1, a feature introduced on the arc ρu is present in all parents of the reticulation above species B but it has only a 0.75 probability of being inherited by B . In addition, in NETWORKPD it is possible to inherit the same feature from both parents, resulting in the presence of multiple copies of that feature. While this can be realistic in certain scenarios (e.g. allopolyploidy), in other cases it is more realistic to assume that each feature is inherited from at most one parent.

A less studied alternative averages the phylogenetic diversity score over all trees displayed by the network [31]. Each tree has an assigned probability equal to the product of the inheritance probabilities of the reticulation edges used by its embedding, and the PD-score



■ **Figure 1** A rooted phylogenetic network \mathcal{N} on $X = \{A, B, C, D\}$ along with the two rooted phylogenetic trees T_1, T_2 that \mathcal{N} displays. Edge weights are indicated above the edges. Reticulation edges are in blue with their inheritance probability in blue below. The APD score of $\{B, D\}$ is 22 in T_1 and in T_2 and, hence, also in \mathcal{N} . The NetworkPD score of $\{B, D\}$ is 20 since $\gamma_{vA} = \gamma_{wC} = 0$, $\gamma_{rB} = \gamma_{\rho D} = 1$, $\gamma_{vr} = \gamma_{uv} = \gamma_{wr} = \gamma_{uw} = 0.5$, and $\gamma_{\rho u} = 0.75$, implying a score of $2 + 0.5 \cdot (2 + 2 + 1 + 3) + 0.75 \cdot 8 + 8 = 20$. However, the NetworkPD score of $\{C, D\}$ is $3 + 3 + 8 + 8 = 22$. It turns out that the only size-2 set maximizing the NetworkPD score is $\{C, D\}$, while $\{B, D\}$ is the only size-2 subset maximizing the APD score.

of each tree is weighted by this probability. Thus, the resulting measure can be interpreted as the expected diversity of any tree displayed by the network. In this model, features inherited from different edges incoming to the same reticulation vertex are not considered to be independent. The assumption is basically that each feature is inherited from exactly one parent (or introduced on that edge). Note however that Wicke and Fischer [31] use an uncommon definition of displayed trees, in which trees are only considered to be displayed if they contain all vertices of the network and all their leaves are in X . In particular, a network that is not tree-based [13] would have no displayed trees and an undefined phylogenetic diversity score. Since we see no reason to ignore certain displayed trees, we will take the (weighted) average over all displayed trees in this paper. We call the resulting phylogenetic diversity measure on phylogenetic networks the *average-tree phylogenetic diversity*, APD for short, see Figure 1 for an example.

While APD is a natural and biologically relevant generalization of phylogenetic diversity from trees to networks, we show that it comes with great challenges. Although for most PD measures on networks, such as ALLPATHSPD and NETWORKPD, the diversity score of a given set $A \subseteq X$ of species can be easily computed in polynomial time, we will show in this paper that for APD this is already computationally hard (more precisely, #P-hard; a polynomial-time algorithm for it would imply a collapse of the polynomial hierarchy.) Consequently, the maximization problem associated to APD is also computationally hard. This problem, called MAX-APD, aims to find a set $A \subseteq X$ of k species with maximum APD score.

On the positive side, we show that, although both these problems are #P-hard, they can both be solved exactly and rather efficiently in practice. We do this by presenting a parameterized algorithm finding a size- k set of leaves that maximizes the APD score in binary networks with r reticulations and n leaves in $\mathcal{O}(2^{6r} \cdot nk)$ time. This algorithm can easily be adapted to compute the score of a given set A . The running time of our algorithm scales linearly with the number of species but exponentially with the number of reticulations in the network. Using practical experiments on simulated data we show that the algorithm can solve instances with up to 500 species and 55 reticulations within 5 minutes on 16 GB RAM and 8 CPUs computer, with $k \leq 5$. In addition, the algorithm efficiently solves instances with k up to at least 40 if the number of reticulations is fixed as 12.

1.1 Related Literature

There is a rich body of literature on phylogenetic diversity on (rooted and unrooted) phylogenetic trees. Most relevant to our work, the optimization problem of finding k species with maximum phylogenetic diversity can be solved in polynomial time on (rooted and unrooted) trees [21, 25]. Further research found fast algorithms on trees, where biological dependencies were considered also [19, 23, 16]. However, only a few papers have studied phylogenetic diversity on phylogenetic networks [32, 3, 15, 27].

Wicke and Fischer [31] introduced several generalizations of phylogenetic diversity of rooted trees to rooted networks, including ALLPATHSPD and a variant of APD (as discussed above). The question of efficient computability is not addressed, nor are the corresponding maximization problems. Instead, the authors define several biodiversity indices on phylogenetic networks, which can be used to rank species for conservation. The associated maximization problems are first addressed by Bordewich et al. [3], showing that the decision versions of the maximization problems of ALLPATHSPD and NETWORKPD are NP-hard, even for tree-child networks but, on level-1 networks,¹ a solution maximizing the ALLPATHSPD score can be found in polynomial time. They also introduce two more variants, MAXWEIGHTTREE-PD and MINWEIGHTTREE-PD. In these, the $PD_{\mathcal{N}}(A)$ of taxa is the maximum and minimum weight of the phylogenetic diversity of A in any display tree. While for the former, the maximization problem is polynomial-time solvable, for the latter, computing the score of a given subset of the species is already NP-hard.

The ALLPATHSPD problem was shown to also be susceptible to greedy approaches, as long as the input network is semi-binary (maximum in-degree two) and has *level*¹ at most two [5].

Jones and Schestag [14] studied the maximization problem for ALLPATHSPD in more detail. They showed that it is W[2]-hard when the parameter is the number of species to save, but fixed-parameter tractable (FPT) when the parameter is the optimal phylogenetic diversity, the acceptable loss of phylogenetic diversity, the number of reticulations in the network, or the treewidth of the underlying graph. Recently, this measure has been considered in semidirected networks [11].

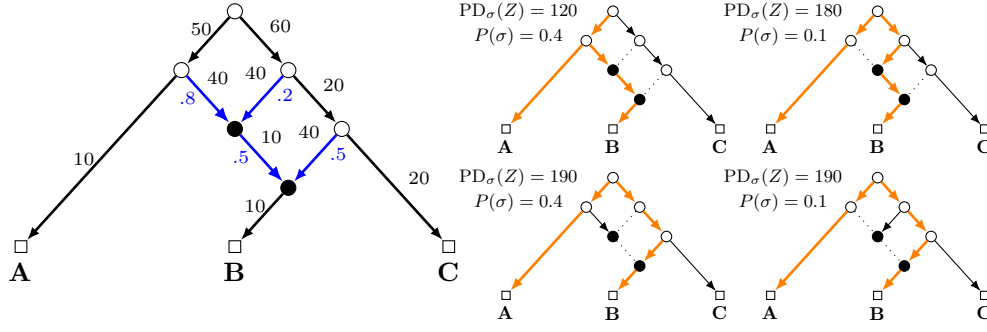
In a recent study [27], the maximization problem for NETWORKPD has been shown to be FPT with respect to the number of reticulations of the network. Unfortunately, this result cannot be strengthened by using the level as a parameter, since NETWORKPD remains NP-hard on level-1 networks [27].

2 Preliminaries

Phylogenetic Networks. In this paper, we consider binary, rooted phylogenetic networks with integer weights on arcs, in which each reticulation has an associated probability distribution on its incoming arcs. Formally, for a set X of taxa, a *binary, rooted phylogenetic network on X* , later only called *phylogenetic network*, is a tuple $\mathcal{N} = (X, V, A, \ell, \omega, \lambda)$ satisfying the following conditions:

- (V, A) forms a directed acyclic graph with vertices V and arcs A , in which $L(\mathcal{N})$ is the set of *leaves* (vertices with in-degree 1 and out-degree 0), and for which
 - There is a single vertex of in-degree 0 and out-degree ≤ 2 , called the *root* $\rho(\mathcal{N})$;
 - Vertices of in-degree 2 and out-degree 1 are called the *reticulations* and denoted $R(\mathcal{N})$;
 - The remaining vertices have in-degree 1 and out-degree 2 and are called the *tree nodes*.

¹ The level of a network is the maximum reticulation number of any biconnected component. It may be arbitrarily smaller than the reticulation number of the network.



■ **Figure 2 Left:** A network with edge-weights (above each edge) and reticulation edges in blue with their inheritance proportions in blue below. Reticulation vertices are filled black. **Right:** The four possible switching-trees with edges marked in orange if they are considered when $Z = \{A, B\}$ is saved, yielding $\text{APD}_{\mathcal{N}}(Z) = .4 \cdot 120 + .1 \cdot 180 + .1 \cdot 190 + .4 \cdot 190 = 161$.

An arc $uv \in A$ is a *reticulation arc* if $v \in R(\mathcal{N})$ and $A_R(\mathcal{N})$ is the set of reticulation arcs.

- $\ell : L(\mathcal{N}) \rightarrow X$ is a bijective mapping of leaves of \mathcal{N} to taxa in X and is called *labelling* of \mathcal{N} . We refer to a leaf of \mathcal{N} interchangeably with the taxa it is labeled with.
- $\omega : A \rightarrow \mathbb{Z}_{\geq 0}$ assigns each arc uv a non-negative integer, called the (*arc*-)weight of uv .
- $\lambda : A_R(\mathcal{N}) \rightarrow [0, 1]$ is a function that assigns each reticulation arc a real value between 0 and 1, subject to the constraint that $\lambda(u_1v) + \lambda(u_2v) = 1$ for any reticulation v with incoming arcs u_1v and u_2v . We call $\lambda(uv)$ the *inheritance probability* of uv . Informally, $\lambda(uv)$ represents the probability that a randomly-selected gene in v is inherited from u .²

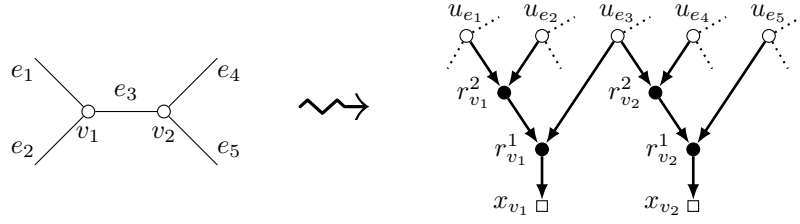
We often refer to the graph (V, A) as \mathcal{N} and we refer to $v \in V$ as a vertex of \mathcal{N} and $uv \in A$ as an arc in \mathcal{N} . For such an arc uv , we call u a *parent* of v , and v a *child* of u . For a vertex v of a phylogenetic network, the offspring of v is the set of leaves reachable from v with a direct path. We will permit ourselves to write $V(\mathcal{N})$ for V and $A(\mathcal{N})$ for A . A *phylogenetic tree* is a phylogenetic network with no reticulations.

Phylogenetic Diversity. For a phylogenetic tree $\mathcal{T} = (V, A)$ on X , the *phylogenetic diversity* $\text{PD}_{\mathcal{T}}(Z)$ of a set $Z \subseteq X$ is the total weight of arcs $e \in A$ that are on a path from the root to a leaf in Z [8]. In this section, we generalize this measure to phylogenetic networks. For an example consider Figure 2.

A *switching* $\sigma : R(\mathcal{N}) \rightarrow A(\mathcal{N})$ is a function that maps each reticulation to one of its incoming arcs. That is, a switching corresponds to a decision about which incoming arc each reticulation will inherit from. The set of switchings in \mathcal{N} is denoted by $\mathcal{S}(\mathcal{N})$. The *probability* $P(\sigma)$ of a switching σ is $\prod_{r \in R(\mathcal{N})} \lambda(\sigma(r))$. That is, we assume that for each reticulation, one of its incoming arcs is chosen (with probability given by λ) independently at random.

The *switching-tree* T_{σ} for a switching σ is the subgraph of \mathcal{N} derived by deleting all reticulation arcs except for those in $\{\sigma(r) \mid r \in R(\mathcal{N})\}$. Observe that T_{σ} is a directed tree, possibly with some “dead” (unlabelled) leaves and vertices of degree 2. Note that X is a subset of the leaves of T_{σ} , for each switching σ . Given a switching σ and a set of taxa $Z \subseteq X$, we define $\text{PD}_{\sigma}(Z)$ as $\text{PD}_{T_{\sigma}}(Z)$ under the usual definition of phylogenetic diversity on trees.

² We explicitly point out the difference to NETWORKPD, where $\lambda(uv)$ represents the proportion of genes in u that are inherited by v [3, 27]. Here, $\lambda(uv)$ is the probability that a randomly chosen switching σ has $\sigma(v) = e$. Consequently, $\lambda(e)$ gives the probability that anything is inherited along e .



■ **Figure 3 Left:** part of graph G . **Right:** the corresponding part of \mathcal{N} . Observe that u_e has a directed path to x_v in \mathcal{N} if and only if e is incident with v in G .

Now, we can define the measure we study in this paper. For a phylogenetic network \mathcal{N} and a set $Z \subseteq X$ of taxa, the *average phylogenetic diversity of Z with respect to \mathcal{N}* is

$$\text{APD}_{\mathcal{N}}(Z) := \sum_{\sigma \in \mathcal{S}(\mathcal{N})} P(\sigma) \cdot \text{PD}_{\sigma}(Z)$$

Informally, $\text{APD}_{\mathcal{N}}(Z)$ is the amount of phylogenetic diversity that we expect to preserve by saving Z , under the assumption that each reticulation independently chooses one of its parents to inherit from. Note that this is equivalent to the definition given in the introduction based on displayed trees since suppressing degree-2 vertices and deleting unlabelled leaves does not affect the phylogenetic diversity score of a tree. The associated decision problem is as follows.

MAX-APD

Input: A phylogenetic network \mathcal{N} and integers k and D .

Question: Is there a set $Z \subseteq X$ of taxa which has a size of k and with $\text{APD}_{\mathcal{N}}(Z) \geq D$?

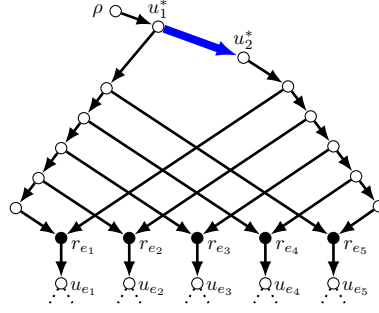
3 #P-Hardness of computing $\text{APD}_{\mathcal{N}}$ when saving all taxa

In the following, we prove that computing $\text{APD}_{\mathcal{N}}$ is #P-hard, even if all taxa are being saved, in contrast to other phylogenetic diversity measures on networks like NETWORKPD [3, 27] or ALLPATHSPD [3, 14, 16], that can be computed (not optimized) in polynomial time.

► **Theorem 3.1.** *Computing $\text{APD}_{\mathcal{N}}(X)$ for a phylogenetic network \mathcal{N} on X is #P-hard.*

We prove the theorem by reduction from COUNTING PERFECT MATCHINGS. A *perfect matching* of an undirected graph $G = (V, E)$ is a set of $|V|/2$ edges $E' \subseteq E$ such that each vertex of G is incident with exactly one edge of E' . Counting the number of perfect matchings in a cubic graph is #P-hard [6, Theorem 6.2].

We first observe that the number of perfect matchings in a graph $G = (V, E)$ is the same as the number of mappings $\phi : V \rightarrow E$ that are perfect assignments, as defined next. A mapping $\phi : V \rightarrow E$ is an *assignment* if $\phi(v)$ is an incident edge of v for all $v \in V$. An assignment ϕ is *perfect* if $|\phi(V)| = |V|/2$. As $|\phi^{-1}(e)| \leq 2$ for each edge e , an assignment ϕ is perfect if and only if $|\phi^{-1}(e)| = 2$ for each $e \in \phi(V)$. That is, for each $e \in E$, a perfect assignment ϕ assigns either both or none of the incident vertices to e . Thus, mapping each perfect assignment ϕ to $\phi(V)$ constitutes an injection from perfect assignments to perfect matchings in G . On the other hand, if $E' \subseteq E$ is a perfect matching in G , then a perfect assignment $\phi : V \rightarrow E$ can be defined from E' by setting $\phi(v)$ to be the unique edge in E' incident to v for all $v \in V$. It is easy to see that this mapping is an injection from the set of



■ **Figure 4** Example of the top part of the network \mathcal{N} , given a graph G with 5 edges. The heavy arc $u_1^*u_2^*$ is in bold and blue. For each of the reticulations r_{e_1} to r_{e_5} , the inheritance probability of the left incoming arc is $(1 - \delta)$, and the inheritance probability of the right incoming arc is δ .

all perfect matchings in G to the set of perfect assignments. Thus, the perfect matchings in G are in bijection with its perfect assignments, so their number is the same. In the following, we let M_G denote this number.

Given an arbitrary cubic graph $G = (V, E)$, we construct a network \mathcal{N} on a set of taxa X in such a way that M_G can be determined from $\text{APD}_{\mathcal{N}}(X)$. See Figure 3 for an example.

Let $X := \{x_v \mid v \in V\}$, that is, X has one element for each vertex in G . Now, for each $v \in V$, add two vertices r_v^1 and r_v^2 , where r_v^1 is a parent of r_v^2 , and r_v^1 is a parent of x_v . For each $e \in E$, add a vertex u_e to \mathcal{N} . For any vertex $v \in V$, choose one of its three incident edges e arbitrarily (recall that G is cubic), and add an arc $u_e r_v^1$. Then, for the remaining two edges e' and e'' incident with v , add the arcs $u_{e'} r_v^2$ and $u_{e''} r_v^2$. Observe that, so far, u_e has out-degree 2 and in-degree 0 for each $e \in E$, and r_v^1 and r_v^2 are reticulations for all $v \in V$. Furthermore, u_e has a directed path to x_v in \mathcal{N} if and only if e is incident to v in G .

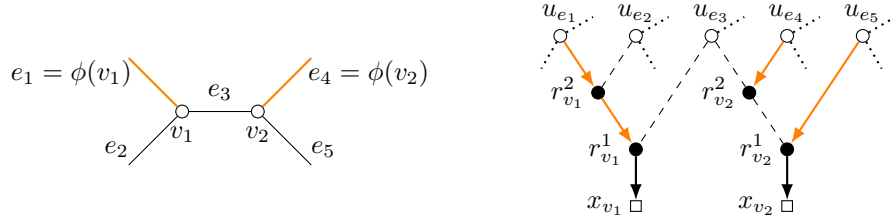
Create two new special vertices u_1^* and u_2^* and add an arc $u_1^*u_2^*$ (this will be the only arc that gets a large weight). Add a root ρ as the parent of u_1^* . Now, for each $e \in E$, add a new vertex r_e as the parent of u_e . For each vertex r_e , add two parents p_e and q_e . Add arcs such that $\{u_1^*\} \cup \{q_e \mid e \in E\}$ and $\{u_2^*\} \cup \{p_e \mid e \in E\}$ form paths. We now have that each vertex r_e is a reticulation, and exactly one of the incoming arcs of r_e is on a directed path from the root that passes through $u_1^*u_2^*$. In Figure 4 an illustration is given.

We define the arc weights by setting $\omega(u_1^*u_2^*) := H$ for H a large value to be defined later, and setting $\omega(a) := 1$ for all other arcs $a \in A(\mathcal{N})$.

It remains to define the inheritance probabilities on the reticulation arcs.

First, consider r_e for some $e \in E$. Then, set $\lambda(p_e r_e) := 1 - \delta$ and $\lambda(q_e r_e) := \delta$, where $\delta \in (0, 1)$ is a small constant to be defined later. Note that, when we choose \mathcal{T}_σ with probability $P(\sigma)$ then, for each $e \in E$, the probability that u_2^* has a path to some r_e is exactly $1 - \delta$.

Now, consider r_v^1 and r_v^2 for some $v \in V$. Then, set $\lambda(r_v^2 r_v^1) := 2/3$, $\lambda(a_1) = 1/3$, and set $\lambda(a_2) := \lambda(a_3) = 1/2$, where a_1 is the other arc incoming at r_v^1 and a_2 and a_3 are the two arcs incoming at r_v^2 . The idea here is that when we choose a switching σ and corresponding tree \mathcal{T}_σ with probability $P(\sigma)$, each vertex u_e for $e \in E$ adjacent to v has probability $1/3$ of having a path to x_v in \mathcal{T}_σ . As exactly one such x_e will have a path to x_v , we can think of the choice of σ as corresponding to a choice of assignment (where v is mapped to e in the assignment if u_e has a path to x_v in \mathcal{T}_σ). Our choice of inheritance probabilities ensures that each of the three edges incident to v is chosen with probability $1/3$, independently for each v , and so each possible assignment is chosen with equal probability. This completes the reduction.



■ **Figure 5** **Left:** part of a graph and an assignment ϕ on v_1 and v_2 . **Right:** partial switching σ_V on this segment. σ_V and ϕ agree on v_1 , but not on v_2 .

Given a switching σ , let $\chi_\sigma := 1$ if u_2^* has a path to some leaf x_v in \mathcal{T}_σ , and $\chi_\sigma := 0$ otherwise. Thus, the expected contribution of $u_1^* u_2^*$ to $\text{APD}_N(X)$ is $H \cdot \sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_\sigma$. We first show that this value depends entirely on the possible assignments of G .

► **Lemma 3.2.** $\sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_\sigma = (1/3)^{|V|} \sum_{\phi} (1 - \delta^{|\phi(V)|})$.

Proof. We first fix some notation. A *partial switching on $R' \subseteq R$* is the restriction of a switching to the subset R' of reticulations. Let $\mathcal{S}(R')$ denote the set of all partial switchings on R' . For a partial switching $\sigma' \in \mathcal{S}(R')$, let $P(\sigma') := \prod_{r \in R'} \lambda(\sigma'(r))$.

Let R_V be the set of reticulations $\{r_v^i \mid v \in V, i \in \{1, 2\}\}$, and let R_E be the set of reticulations $\{u_e \mid e \in E\}$. If σ is a switching and σ_V and σ_E are the restrictions of σ to R_V and R_E respectively, then we write $\sigma = \sigma_V \cup \sigma_E$. Observe that $P(\sigma) = P(\sigma_V) \cdot P(\sigma_E)$ in this case and that $R = R_V \cup R_E$ and therefore $\mathcal{S}(R) = \{\sigma_V \cup \sigma_E \mid \sigma_V \in \mathcal{S}(R_V), \sigma_E \in \mathcal{S}(R_E)\}$.

For a partial switching σ_V on R_V , we can associate an assignment ϕ on G , as follows. For any $v \in V$, let $e, e',$ and e'' be the incident edges of v in G , and without loss of generality assume that u_e is a parent of r_v^1 and the parents of r_v^2 are $u_{e'}$ and $u_{e''}$. We say a partial switching $\sigma' \in \mathcal{S}(R_V)$ and assignment ϕ *agree on v* if one of the following holds: 1. $\phi(v) = e$ and $\sigma'(r_v^1) = u_e r_v^1$; 2. $\phi(v) = e'$ and $\sigma'(r_v^1) = r_v^2 r_v^1, \sigma'(r_v^2) = u_{e'} r_v^2$; or 3. $\phi(v) = e''$ and $\sigma'(r_v^1) = r_v^2 r_v^1, \sigma'(r_v^2) = u_{e''} r_v^2$. That is, $\phi(v)$ is the unique edge incident to v for which the switching σ_V chooses a set of reticulation arcs that connect u_e to x_v . See Figure 5. An assignment ϕ and a partial switching σ' are *associated* if ϕ and σ' agree on all $v \in V$.

For assignments ϕ on G , let \mathcal{S}_ϕ denote the set of all partial switchings $\sigma' \in \mathcal{S}(R_V)$ associated with ϕ . We use these sets \mathcal{S}_ϕ to decompose the sum $\sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_\sigma$, as follows:

$$\begin{aligned} \sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_\sigma &= \sum_{\sigma' \in \mathcal{S}(R_V)} \sum_{\sigma'' \in \mathcal{S}(R_E)} P(\sigma' \cup \sigma'') \cdot \chi_{\sigma' \cup \sigma''} \\ &= \sum_{\sigma' \in \mathcal{S}(R_V)} \sum_{\sigma'' \in \mathcal{S}(R_E)} P(\sigma') \cdot P(\sigma'') \cdot \chi_{\sigma' \cup \sigma''} \\ &= \sum_{\phi} \sum_{\sigma' \in \mathcal{S}_\phi} P(\sigma') \cdot \sum_{\sigma'' \in \mathcal{S}(R_E)} P(\sigma'') \cdot \chi_{\sigma' \cup \sigma''} \end{aligned}$$

The next two claims allow us to express the above in terms of $|\phi(V)|$ for all assignments ϕ .

► **Claim 3.3.** $\sum_{\sigma'' \in \mathcal{S}(R_E)} P(\sigma'') \cdot \chi_{\sigma' \cup \sigma''} = 1 - \delta^{|\phi(V)|}$, for each assignment ϕ on G and $\sigma' \in \mathcal{S}_\phi$.

Proof. Let $\sigma = \sigma' \cup \sigma''$ for some $\sigma'' \in \mathcal{S}(R_E)$ and recall that $\chi_\sigma = 1$ if and only if u_2^* has a path to some leaf x_v in \mathcal{T}_σ . Any such path must pass through some u_e , and u_e only has a path in \mathcal{T}_σ to some x_v if e is assigned to v by ϕ i.e. $\phi(v) = e$. Thus, $\chi_\sigma = 1$ if and only if u_2^* has a path in \mathcal{T}_σ to u_e for some $e \in \phi(V)$. This occurs if and only if $\sigma''(r_e) = p_e r_e$ for some $e \in \phi(V)$. Or, equivalently, $\chi_\sigma = 0$ if and only if $\sigma''(r_e) = q_e r_e$ for all $e \in \phi(V)$. It follows

$$\sum_{\sigma'' \in \mathcal{S}(R_E)} P(\sigma'') \cdot \chi_{\sigma' \cup \sigma''} = 1 - \sum_{\substack{\sigma'' \in \mathcal{S}(R_E) \\ \forall e \in \phi(V): \sigma''(r_e) = q_e r_e}} P(\sigma'')$$

as, roughly speaking, the sum of the probabilities of partial switchings that choose the $q_e r_e$ -arcs equals the probability that a random switching makes all these choices, this equals

$$= 1 - \prod_{e \in \phi(V)} \lambda(q_e r_e) = 1 - \prod_{e \in \phi(V)} \delta = 1 - \delta^{|\phi(V)|}. \quad \triangleleft$$

▷ **Claim 3.4.** For any assignment ϕ on G , it holds that $\sum_{\sigma' \in \mathcal{S}_\phi} P(\sigma') = (1/3)^{|V|}$.

Proof. Recall that, for any $\sigma' \in \mathcal{S}(R_V)$ and any assignment ϕ , we have $\sigma' \in \mathcal{S}_\phi$ if and only if σ' agrees with ϕ on all $v \in V$. Recall also that for any $v \in V$, whether σ' and ϕ agree on v depends only on $\phi(v)$, $\sigma(r_v^1)$ and $\sigma(r_v^2)$. So suppose we choose a partial switching $\sigma^* \in \mathcal{S}(R_v)$ with probability $P(\sigma^*)$. Note that the probability that $\sigma^* \in \mathcal{S}_\phi$ is $\sum_{\sigma' \in \mathcal{S}_\phi} P(\sigma')$.

Now consider the probability that σ^* agrees with ϕ on some $v \in V$. If $\phi(v) = e$ for the edge e such that $u_e r_v^1 \in A(\mathcal{N})$, then σ^* agrees with ϕ on v if and only if $\sigma^*(r_v^1) = u_e r_v^1$, and this occurs with probability $\lambda(u_e r_v^1) = 1/3$. Now, if $\phi(v) = e'$ where $u_{e'}$ is one of the parents of r_v^2 , then σ^* agrees with ϕ on v if and only if $\sigma^*(r_v^1) = r_v^2 r_v^1$ and $\sigma^*(r_v^2) = u_{e'} r_v^2$, and this occurs with probability $\lambda(r_v^2 r_v^1) \lambda(u_{e'} r_v^2) = 2/3 \cdot 1/2 = 1/3$. In either case, σ^* and ϕ agree on v with probability $1/3$. As these agreements are independent for each $v \in V$, the probability that σ^* agrees with ϕ on all $v \in V$ is $(1/3)^{|V|}$. But this is exactly the probability that $\sigma^* \in \mathcal{S}_\phi$, which is $\sum_{\sigma' \in \mathcal{S}_\phi} P(\sigma')$. \triangleleft

By Claim 3.3 and Claim 3.4, we have

$$\begin{aligned} \sum_{\phi} \sum_{\sigma' \in \mathcal{S}_\phi} P(\sigma') \cdot \sum_{\sigma'' \in \mathcal{S}(R_E)} P(\sigma'') \cdot \chi_{\sigma' \cup \sigma''} &= \sum_{\phi} \sum_{\sigma' \in \mathcal{S}_\phi} P(\sigma') \cdot (1 - \delta^{|\phi(V)|}) \\ &= \sum_{\phi} (1/3)^{|V|} \cdot (1 - \delta^{|\phi(V)|}) \\ &= (1/3)^{|V|} \cdot \sum_{\phi} (1 - \delta^{|\phi(V)|}) \end{aligned} \quad \blacktriangleleft$$

Lemma 3.2 gives us an expression for the expected contribution $H \cdot \sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_\sigma$, in terms of the values $|\phi(V)|$ for all assignments ϕ on G . We next show that this value can be restricted to a certain range depending on the number M_G of perfect assignments. For this next step, we now fix the value of δ such that $\delta \leq (1/3)^{|V|} \cdot 1/2$. Let $\alpha := (1/3)^{|V|} \cdot \delta^{|V|/2}$.

► **Lemma 3.5.** $1 - \alpha \cdot M_G \geq \sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_\sigma \geq 1 - \alpha \cdot (M_G + 1/2)$.

Proof. From Lemma 3.2 we have that $\sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_\sigma = (1/3)^{|V|} \cdot \sum_{\phi} (1 - \delta^{|\phi(V)|})$. As there are exactly $3^{|V|}$ assignments ϕ on G , this is equivalent to

$$\sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_\sigma = (1/3)^{|V|} \cdot 3^{|V|} - (1/3)^{|V|} \cdot \sum_{\phi} \delta^{|\phi(V)|} = 1 - (1/3)^{|V|} \cdot \sum_{\phi} \delta^{|\phi(V)|}$$

Recall that $|\phi(V)| \geq |V|/2$ for each assignment ϕ , with equality if and only if ϕ is a perfect assignment. Then, counting only the perfect assignments,

$$1 - (1/3)^{|V|} \cdot \sum_{\phi \text{ perfect}} \delta^{|\phi(V)|} = 1 - (1/3)^{|V|} \cdot M_G \cdot \delta^{|V|/2} = 1 - \alpha \cdot M_G$$

proving the first inequality.

For the second inequality, observe that $|\phi(V)| \geq |V|/2 + 1$ for any imperfect assignment ϕ . (As G is cubic, $|V|$ is even). Then, counting perfect and imperfect assignments separately, and using the fact that there are at most $3^{|V|}$ imperfect assignments, we have

$$\begin{aligned} 1 - (1/3)^{|V|} \cdot \sum_{\phi} \delta^{|\phi(V)|} &\geq 1 - (1/3)^{|V|} \cdot M_G \cdot \delta^{|V|/2} - (1/3)^{|V|} \cdot 3^{|V|} \cdot \delta^{|V|/2+1} \\ &\geq 1 - \alpha \cdot M_G - \delta^{|V|/2} \cdot \delta \\ &\geq 1 - \alpha \cdot M_G - \delta^{|V|/2} \cdot (1/3)^{|V|} \cdot 1/2 \\ &= 1 - \alpha \cdot (M_G + 1/2) \end{aligned} \quad \blacktriangleleft$$

By what we have seen so far, the expected contribution of arc $u_1^* u_2^*$ to $\text{APD}_{\mathcal{N}}(X)$ is between $H \cdot (1 - \alpha \cdot (M_G + 1/2))$ and $H \cdot (1 - \alpha \cdot M_G)$, where M_G is the number of perfect assignments (and therefore the number of perfect matchings) on G . It remains to show that this term dominates the other terms in $\text{APD}_{\mathcal{N}}(X)$, such that $\text{APD}_{\mathcal{N}}(X)$ is also bounded by a range dependent on M_G . For this, we fix H such that $H > 2|A(\mathcal{N})|/\alpha$. That is, the weight of $u_1^* u_2^*$ is more than $2/\alpha$ times the total weight of all the other arcs in \mathcal{N} .

► **Lemma 3.6.** *We have*

$$H \cdot (1 - \alpha \cdot (M_G + 1/2)) \leq \text{APD}_{\mathcal{N}}(X) < H \cdot (1 - \alpha \cdot (M_G - 1/2))$$

or equivalently, $M_G - 1/2 < (H - \text{APD}_{\mathcal{N}}(X))/(H \cdot \alpha) \leq M_G + 1/2$.

Proof. First, consider the tree \mathcal{T}_{σ} associated with an arbitrary switching σ . If u_2^* has a path to a leaf x_v in \mathcal{T}_{σ} (i.e. if $\chi_{\sigma} = 1$) then $u_1^* u_2^*$ has an offspring in X , and so $\text{PD}_{\sigma}(X) \geq \omega(u_1^* u_2^*) = H$. Thus $\text{PD}_{\sigma}(X) \geq H \cdot \chi_{\sigma}$. As the weight of all other arcs is 1, we also have that $\text{PD}_{\sigma}(X) < H \cdot \chi_{\sigma} + |A(\mathcal{N})| \leq H \cdot \chi_{\sigma} + H \cdot \alpha/2$. Then, by Lemma 3.5,

$$\text{APD}_{\mathcal{N}}(X) = \sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \text{PD}_{\sigma}(X) \geq H \cdot \sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \chi_{\sigma} \geq H \cdot (1 - \alpha \cdot (M_G + 1/2))$$

and

$$\begin{aligned} \text{APD}_{\mathcal{N}}(X) &= \sum_{\sigma \in \mathcal{S}(R)} P(\sigma) \cdot \text{PD}_{\sigma}(Z) \\ &< H \cdot \sum_{\sigma \in \mathcal{S}(R)} (P(\sigma) \cdot \chi_{\sigma}(Z)) + H \cdot \sum_{\sigma \in \mathcal{S}(R)} (P(\sigma) \cdot \alpha/2) \\ &\leq H \cdot (1 - \alpha \cdot M_G) + H \cdot \alpha/2 = H \cdot (1 - \alpha \cdot (M_G - 1/2)) \end{aligned} \quad \blacktriangleleft$$

Given Lemma 3.6, we can count the number of perfect matchings in a cubic graph G as follows. Construct the network \mathcal{N} on X as described above in polynomial time, then compute $M' := (H - \text{APD}_{\mathcal{N}}(X))/(H \cdot \alpha)$. Determine the unique integer M for which we have $M - 1/2 < M' \leq M + 1/2$. By Lemma 3.6, M is the number of perfect matchings in G .

As counting perfect matchings is #P-hard, it follows that computing $\text{APD}_{\mathcal{N}}(X)$ is also #P-hard. This completes the proof of Theorem 3.1.

4 Maximizing Average-Tree Phylogenetic Diversity

In this section, we show that MAX-APD is fixed-parameter tractable (FPT) with respect to the number of reticulations. For the sake of brevity, we restrict our theoretical proof to simple networks – that is, \mathcal{N} has no cut-arcs except for those incident to degree-1 vertices. However, we see no reason why this result could not be extended to non-simple networks. Indeed, our implementation, see Section 5, can deal with arbitrary networks.

We make use of the notion of generators, as introduced in [28]. This gives us a certain useful partition of the leaves of a network into *sides*. At the high level, our algorithm “guesses” the optimal solution Z by guessing which sides of the network \mathcal{N} contain a leaf from Z . As the number of sides is $\mathcal{O}(|R(\mathcal{N})|)$, there are at most $2^{\mathcal{O}(|R(\mathcal{N})|)}$ such guesses to consider. Once it has been decided which sides contain a leaf from the solution, the problem reduces to a problem on forests, which can be solved in polynomial time.

In what follows, let \mathcal{N} be a network and let $r := |R(\mathcal{N})|$, $n := |V(\mathcal{N})|$ and $m := |A(\mathcal{N})|$. That is, r , n , and m denote the number of reticulations, vertices, and arcs, respectively.

4.1 Characterizing solutions in terms of generator sides

We recall the notion of generators, introduced in [28]. Our definitions are taken from [29].

► **Definition 4.1.** *Let \mathcal{N} be a simple network. The generator of \mathcal{N} is the directed multigraph G obtained from \mathcal{N} by (1) deleting nodes with in-degree one and out-degree zero (leaves) and (2) suppressing all nodes with in- and out-degree one. The arcs and in-degree-2 out-degree-0 vertices of G are called sides. The arcs are also called arc sides, and the in-degree-2 out-degree-0 vertices are also reticulation sides. We say that a leaf ℓ is on side S (or that side S contains ℓ) if either*

- *S is a reticulation side of G and the parent of ℓ in \mathcal{N} , or*
- *S is an arc side of G , obtained by suppressing in-degree-1 out-degree-1 vertices of a path P in \mathcal{N} and the parent of ℓ in \mathcal{N} lies on P .*

In addition, if S is an arc side of G , obtained by suppressing in-degree-1 out-degree-1 vertices of a path P in \mathcal{N} , then we say that every arc in P is a path-arc of side S . We call the lowest arc in P the lowest path-arc of S . Observe that every arc in \mathcal{N} is either a leaf-arc, or a path-arc of some side.

For an arc side S , let T_S denote the subgraph of \mathcal{N} whose vertices are the leaves on side S together with the vertices incident to any path-arcs of S , and whose arcs are the leaf-arcs incident to the leaves of S together with the path-arcs of S . Note that T_S is a tree (though not necessarily a phylogenetic tree). We call T_S the side-tree of S .

The generator of a simple binary network with r reticulations has at most r reticulation sides and at most $4r - 2$ arc sides [9].

In order to find solutions for MAX-APD on simple networks, we characterize possible solutions in terms of their relation to the generator of \mathcal{N} .

► **Definition 4.2.** *For a simple phylogenetic network \mathcal{N} on X , let $\mathcal{S}(\mathcal{N})$ denote the sides of the generator of \mathcal{N} . For $Z \subseteq X$, the signature SIG_Z of $Z \subseteq X$ is the set of all sides that contain at least one leaf of Z .*

Now, to find a solution for MAX-APD, it is enough to compute, for each signature SIG' , the maximum value of $\text{APD}_{\mathcal{N}}(Z)$ over all sets $Z \subseteq X$ with signature SIG' and $|Z| \leq k$.

4.2 Contribution of lowest path-arcs

For an arc e in \mathcal{N} , a set of taxa $Z \subseteq X$ and a switching $\sigma : R(\mathcal{N}) \rightarrow A(\mathcal{N})$, define $\chi(e, Z, \sigma)$ to be 1 if e has an offspring from Z in T_σ , and 0 otherwise (if e is not an arc in T_σ then define $\chi(e, Z, \sigma)$ to be 0). Then define $\psi_{\mathcal{N}}(e, Z) := \sum_{\text{switching } \sigma} P(\sigma) \cdot \chi(e, Z, \sigma)$

Intuitively, $\psi(e, Z)$ corresponds to the expected probability that e has an offspring in Z under a randomly selected switching. In the next lemma, we show that $\omega(e) \cdot \psi_{\mathcal{N}}(e, Z)$ gives the contribution of arc e to $\text{APD}_{\mathcal{N}}(z)$.

► **Lemma 4.3.** $APD_{\mathcal{N}}(Z) = \sum_{e \in A(\mathcal{N})} \omega(e) \cdot \psi_{\mathcal{N}}(e, Z)$.

Proof. By definition, $PD_{\sigma}(Z) = \sum_{e \in A(T_{\sigma})} \omega(e) \cdot \chi(e, Z, \sigma) = \sum_{e \in A(\mathcal{N})} \omega(e) \cdot \chi(e, Z, \sigma)$. Thus,

$$\begin{aligned} APD_{\mathcal{N}}(Z) &= \sum_{\text{switching } \sigma} P(\sigma) \cdot PD_{\sigma}(Z) \\ &= \sum_{\text{switching } \sigma} P(\sigma) \cdot \sum_{e \in A(\mathcal{N})} \omega(e) \cdot \chi(e, Z, \sigma) \\ &= \sum_{e \in A(\mathcal{N})} \omega(e) \cdot \sum_{\text{switching } \sigma} P(\sigma) \cdot \chi(e, Z, \sigma) = \sum_{e \in A(\mathcal{N})} \omega(e) \cdot \psi_{\mathcal{N}}(e, Z). \quad \blacktriangleleft \end{aligned}$$

Thus, to compute $APD_{\mathcal{N}}(Z)$, it is enough to compute $\psi_{\mathcal{N}}(e, Z)$ for each arc e . The next lemma shows that for certain arcs, the value of $\psi_{\mathcal{N}}(e, Z)$ depends only on the signature of Z . This will be important for the construction that follows.

► **Lemma 4.4.** *Let e be the lowest path-arc of a path side S and let $Z, Z' \subseteq X$ be two subsets of X with the same signature. Then $\psi_{\mathcal{N}}(e, Z) = \psi_{\mathcal{N}}(e, Z')$.*

Proof. By the definition of $\psi_{\mathcal{N}}(e, Z)$, it suffices to show that $\chi(e, Z, \sigma) = \chi(e, Z', \sigma)$ for each switching σ . So suppose that e has an offspring $z \in Z$ in T_{σ} such that $\chi(e, Z, \sigma) = 1$. Let S' be the side containing z . As Z and Z' have the same signature, S' also contains some leaf $z' \in Z'$. Since z and z' are on the same side and z is an offspring of the arc e in T_{σ} , also z' is an offspring of e in T_{σ} . Thus z' is an offspring of e in T_{σ} , and so $\chi(e, Z', \sigma) = 1$. By a symmetric argument, from $\chi(e, Z', \sigma) = 1$ we conclude $\chi(e, Z, \sigma) = 1$. Thus $\chi(e, Z', \sigma) = \chi(e, Z, \sigma)$ for all σ and we can conclude that $\psi_{\mathcal{N}}(e, Z) = \psi_{\mathcal{N}}(e, Z')$. \blacktriangleleft

For a lowest path-arc e and signature $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{N})$, define $\phi(e, \mathcal{S}')$ to be $\psi_{\mathcal{N}}(e, Z)$ for any $Z \subseteq X$ with signature $\text{SIG}_Z = \mathcal{S}'$. By Lemma 4.4, this is well-defined.

► **Lemma 4.5.** *For any lowest path-arc e and any $Z \subseteq X$, the value of $\psi_{\mathcal{N}}(e, Z)$ can be computed in $\mathcal{O}(2^r \cdot r)$ time. Consequently, $\psi_{\mathcal{N}}(e, \mathcal{S}')$ can be computed in $\mathcal{O}(2^r \cdot r)$ time for each lowest path-arc e and signature \mathcal{S}' .*

Proof. Observe that as each reticulation has two incoming arcs and a switching maps each reticulation to one of its incoming arcs, there are exactly 2^r switchings. For each switching σ , the value of $P(\sigma) \cdot \chi(e, Z, \sigma)$ for all e that are lowest arcs of a generator arc side, can be computed in $\mathcal{O}(r)$ time by a bottom-up traversal in the generator of \mathcal{N} , registering whether each such e has an offspring from Z in T_{σ} . Adding up all values of $P(\sigma) \cdot \chi(e, Z, \sigma)$ (to compute $\psi_{\mathcal{N}}(e, Z)$) can therefore be done in $\mathcal{O}(2^r \cdot r)$ time.

To compute $\psi_{\mathcal{N}}(e, \mathcal{S}')$, it is sufficient to construct a set $Z \subseteq X$ with signature \mathcal{S}' by picking an arbitrary leaf from each side $S \in \mathcal{S}'$. As $\psi_{\mathcal{N}}(e, Z) = \psi_{\mathcal{N}}(e, \mathcal{S}')$, we can compute this value in $\mathcal{O}(2^r \cdot r)$ time. \blacktriangleleft

4.3 Reduction to Forests

Let (\mathcal{N}, k, D) be an instance of MAX-APD. Fix a signature $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{N})$ and assume there exists a solution Z with $\text{SIG}_Z = \mathcal{S}'$. We now show how to reduce MAX-APD under this assumption to a related problem on forests.

Let X_1, \dots, X_s be non-empty and pairwise disjoint subsets of X , and let F be a forest of X_i -trees T_i . Define X^* to be the union $X_1 \cup \dots \cup X_s$. For technical reasons, we allow T_i to have non-integer arc weights and degree-2 vertices. A set $Z \subseteq X^*$ *respects* F if Z contains a

taxon of each X_i for $i \in \{1, \dots, s\}$. For a set $Z \subseteq X$, define $PD_F(Z) := \sum_{i=1}^s PD_{T_i}(Z \cap X_i)$. For a forest F and two integers $k, D \in \mathbb{N}$, we define a decision problem MAX-PD in which we are asked whether a set $Z \subseteq X^*$ exists such that Z has a size of k , $PD_F(Z) \geq D$, and respects F . This definition generalizes the classic definition of MAX-PD on trees [8].

► **Construction 1.** Initialize $D^* = 0$, $k' = k$, and let F be an empty set. Then for each side S in $\mathcal{S}(\mathcal{N})$ do the following:

- If S is a reticulation side, then let x be the single leaf on this side and let rx be its incoming arc. If $S \in \mathcal{S}'$, then increase D^* by $\omega(rx)$, and add T'_S to F , where T'_S is a tree with a single leaf x and single arc rx of weight $\omega(rx)$.
- If S is an arc side, then let $e_S = vr$ denote the lowest path-arc of S . For each path-arc e of S (including e_S), add $\omega(e) \cdot \psi_{\mathcal{N}}(e, \mathcal{S}')$ to D^* . Then if $S \in \mathcal{S}'$, add T'_S to F , where T'_S is derived from the side-tree T_S as follows: Multiply the weight of each path-arc by $(1 - \psi_{\mathcal{N}}(e_S, \mathcal{S}'))$, then delete the lowest path-arc e_S .

► **Lemma 4.6.** Let $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{N})$ be a subset of sides and let (D^*, F) be derived from $(\mathcal{N}, \mathcal{S}')$ according to Construction 1. Then for $Z \subseteq X$, the signature of Z is $SIG_Z = \mathcal{S}'$ if and only if Z respects F . Moreover, if $SIG_Z = \mathcal{S}'$, then $APD_{\mathcal{N}}(Z) = PD_F(Z) + D^*$.

Proof. Observe that by construction, each tree in F corresponds to a side $S \in \mathcal{S}'$, and as such Z respects F if and only if $SIG_Z = \mathcal{S}'$.

So now assume that $SIG_Z = \mathcal{S}'$. We prove that $APD_{\mathcal{N}}(Z) = PD_F(Z) + D^*$.

Note that every arc in F has a corresponding arc in \mathcal{N} . For ease of notation, we speak of the same arc e as existing in F and \mathcal{N} . Let $\omega_{\mathcal{N}}(e)$ denote the weight of an arc e in \mathcal{N} , and let ω_F denote the weight of e in F . For any arc e that is not a lowest path-arc in \mathcal{N} , let $\chi_F(e, Z) := 1$ if e has an offspring from Z in F , and let $\chi_F(e, Z) := 0$ otherwise. Observe that $PD_F(Z) = \sum_{e \in A(F)} \omega_F(e) \cdot \chi_F(e, Z)$.

Recall that $\psi_{\mathcal{N}}(e, Z)$ denotes the probability that, taken some switching σ , the arc e has an offspring from Z in T_{σ} . By Lemma 4.3 we know $APD_{\mathcal{N}}(Z) = \sum_{e \in A(\mathcal{N})} \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z)$.

▷ **Claim 4.7.** For an arc side S with lowest path-arc e_S , and e a path-arc of S that is not e_S , if $S \in \mathcal{S}'$ then $\omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) = \omega_F(e) \cdot \chi_F(e, Z) + \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}')$ and, otherwise, $\omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) = \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}')$.

▷ **Claim 4.8.** For a leaf-arc $e = vx$ with x on side S , it holds that $\omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) = \omega_F(e) \cdot \chi_F(e, Z)$ if $S \in \mathcal{S}'$, and $\psi_{\mathcal{N}}(e, Z) = 0$, otherwise.

Due to space constraints, Claim 4.7 and 4.8 are proven in the appendix.

Recall that, for any lowest path-arc e_S on side S , it holds that $\psi_{\mathcal{N}}(e_S, Z) = \psi_{\mathcal{N}}(e_S, \mathcal{S}')$ by definition. Let $\mathcal{AS}(\mathcal{N})$ denote the arc sides on \mathcal{N} and let $\mathcal{RS}(\mathcal{N})$ denote the reticulation sides. Note that $\mathcal{S}(\mathcal{N}) = \mathcal{AS}(\mathcal{N}) \cup \mathcal{RS}(\mathcal{N})$. By construction it is $D^* = \sum_{S \in \mathcal{AS}(\mathcal{N})} \omega_{\mathcal{N}}(e_S) \cdot \psi(e_S, \mathcal{S}')$.

To prove the main claim, let E_1 denote the leaf-arcs of \mathcal{N} , let E_2 denote the path-arcs which are not the lowest path-arc on their side, and let E_3 denote the arcs which are the lowest path-arc of their side. For each $i \in \{1, 2, 3\}$, let E_i^1 denote the arcs in E_i belonging to a side in \mathcal{S}' , and let $E_i^0 = E_i \setminus E_i^1$. Note that the arc set of F is $E_1^1 \cup E_2^1$. For each arc $e \in E_2 \cup E_3$, let e_S denote the lowest path-arc on the side containing e (note that if $e \in E_3$ then $e_S = e$). Due to space constraints, $APD_{\mathcal{N}}(Z) = PD_F(Z) + D^*$ is proven in the appendix. ◀

Finally, we show that MAX-PD can be solved efficiently on forests. For this, we use Faith's famous greedy algorithm on trees [8, 21, 25].

► **Lemma 4.9.** *MAX-PD can be solved in $\mathcal{O}(nk)$ time on forests.*

Due to space constraints, Lemma 4.9 is proven in the appendix.

4.4 Putting everything together

For any signature $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{N})$, let $(F_{\mathcal{S}'}, D_{\mathcal{S}'}^*)$ denote the pair (F, D^*) derived for \mathcal{S}' according to Construction 1. Lemma 4.6 implies that $\text{APD}_{\mathcal{N}}(Z) = \text{PD}_{F_{\mathcal{S}'}}(Z) + D_{\mathcal{S}'}^*$, for any $Z \subseteq X$ with signature \mathcal{S}' . As every $Z \subseteq X$ has a signature, it follows that (\mathcal{N}, k, D) is a YES-instance of MAX-APD if and only if $(F_{\mathcal{S}'}, k, D - D_{\mathcal{S}'}^*)$ is a YES-instance of MAX-PD for some $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{N})$. We can therefore solve MAX-APD on (\mathcal{N}, k, D) using the following algorithm: For each $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{N})$ in turn, construct $(F_{\mathcal{S}'}, D_{\mathcal{S}'}^*)$ and solve MAX-PD on $(F_{\mathcal{S}'}, k, D - D_{\mathcal{S}'}^*)$. If the answer is YES for any MAX-PD instance, return YES. Otherwise, return NO.

As the generator of \mathcal{N} has at most $5r - 2$ sides [9], there are at most 2^{5r-2} signatures to consider. For each signature \mathcal{S}' and each lowest path-arc e_S , the value of $\psi_{\mathcal{N}}(e_S, \mathcal{S}')$ can be computed in $\mathcal{O}(2^r \cdot r)$ time by Lemma 4.5. After these values are computed, $(F_{\mathcal{S}'}, D_{\mathcal{S}'}^*)$ can be constructed in $\mathcal{O}(m) = \mathcal{O}(n + r)$ time. Finally, solving MAX-PD on $(F_{\mathcal{S}'}, k, D - D_{\mathcal{S}'}^*)$ can be done in $\mathcal{O}(nk)$ time by Lemma 4.9. Thus, the total running time of the algorithm is in $\mathcal{O}(2^{5r-2} \cdot (2^r \cdot r + n + r + nk)) \subseteq \mathcal{O}(2^{6r-2} \cdot nk)$.

5 Experiments

5.1 Data

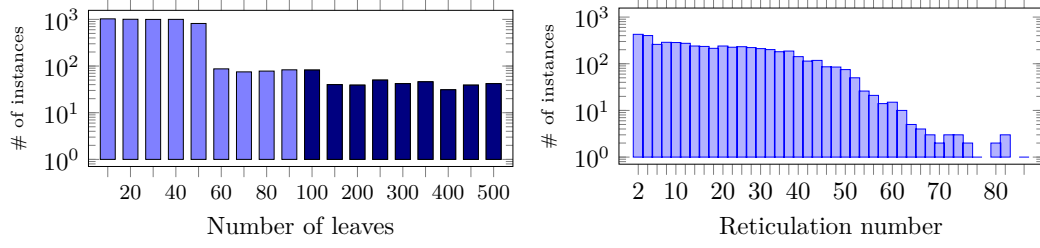
Phylogenetic networks have been simulated³ via the package **SiPhyNetwork** [17]. This package allows simulations of phylogenetic networks as well as traits developing along the networks, giving several customization options. In the following, we describe the choices made for our simulations, and we refer to the **SiPhyNetwork** paper for more details [17]. We simulated 5324 phylogenetic networks with 1, 20, ..., 100, 150, ... 500 leaves using the generalized sampling approach [10] implemented in **SiPhyNetwork**, with 100 simulation replicates, speciation rate equal to 1, extinction rate equal to 0.6, and hybridization rate varying from 0.1 to 0.001 for getting a reasonable reticulation number, even for large instances (see distributions of instances in Figure 6).

In our simulations, we aimed at simulating hybridizations between plants and, for doing so, we opted for the “lineage generative hybridizations” implemented in **SiPhyNetwork**, with inheritance probability fixed to $1/2$ and a trait model where hybridization occurs only among species with equal ploidy. The rate of autopolyploidy has been fixed to 0.05. In this simulation setting, the number of leaves and the number of reticulations are correlated, so it is unlikely to produce large networks with small reticulation number or small networks with large reticulation number.

5.2 Implementation Details

The algorithm presented in Section 4 was implemented in C++ in a serial manner (no multithreading) in the framework of the **phylo-tools** library [30] (both published under the open source CeCILL Free Software License 2.1) with a few heuristic improvements that we discuss in the following.

³ All data is publicly available at <https://sdrive.cnrs.fr/s/aEFNN3iRp7goQBT>.



■ **Figure 6** Distribution of instances for the number n of leaves, and the reticulation number r .

Guesses on the Generator. Instead of “guessing” which of the $\mathcal{O}(r)$ sides of the generator have tree-paths to selected leaves as stated in Construction 1, it is sufficient to guess which *nodes* of the generator have tree-paths to selected leaves. The nodes are pre-filtered, so the guess is done only on nodes that actually have tree-paths to leaves in the input network. Then, all $\sum_{i \leq k} \binom{\mathcal{O}(r)}{i}$ subsets of size at most k of such nodes are enumerated. Note that those are polynomially many if k is constant (with the degree depending on k).

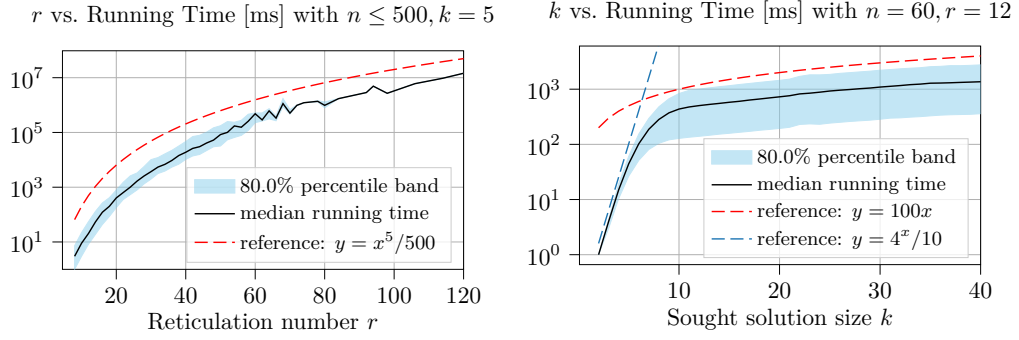
Enumerating Switchings. Given the information which generator nodes have tree-paths to selected leaves, we can group the possible switchings in order to avoid enumerating all 2^r of them. Indeed, if Z contains no leaf below some reticulation v , then the switching of its reticulation arcs has no bearing on the value of $\chi(e, Z, \sigma)$ for any e . This allows us to restrict enumeration to switchings involving reticulations above leaves in Z . If the input network is mostly “flat”, that is, most paths contain only constantly many reticulations, then we expect the enumeration-step to take $\mathcal{O}(\binom{r}{k})$, rather than $\mathcal{O}(2^r)$ steps. For fixed k , we would therefore expect polynomial running time (with the degree depending on k).

Keeping Multiple Solutions. In order to assess the quality of our diversity measure and compare it to other phylogenetic diversity measures, as well as the actual feature diversity, it would be helpful to output not only one, but all best-scoring solutions, and maybe even some sub-optimal solutions. To accommodate this, we store solutions in a sorted array that retains a given number of solutions and automatically discards solutions with a lower score.

Implementation of Greedy Selection. Once the values $\psi_{\mathcal{N}}(e, Z)$ are computed for all e , we set up a structure mapping each node u of the network to a list of its children v sorted by diversity attainable by saving a leaf with a tree-path from v that avoids generator nodes. Then, the best entry is selected, the pointers are followed down to the best leaf ℓ , and the structure is updated from ℓ upwards until a generator node is encountered. While we incur a factor of $\mathcal{O}(\log \Delta_{\mathcal{N}})$ (where $\Delta_{\mathcal{N}}$ is the maximum out-degree in the network), this degenerates to $\mathcal{O}(1)$ since our input networks are binary. Note that we avoid explicitly constructing the forest on which we run the greedy algorithm.

5.3 Experimental Results

While the theoretical running time of $\mathcal{O}(2^{6r-2} \cdot nk)$ may not seem convincing, the implementation performs satisfactorily in practice, possibly due to the heuristic improvements discussed in Section 5.2.



■ **Figure 7** Plots of the observed running time in dependence of the reticulation number r (with $k = 5$ fixed), and of the solution size k (with $n = 60$ and $r = 12$ fixed).

Reticulation number r . Figure 7 (left) shows that the observed dependence on the reticulation number r for $k = 5$ behaves very similarly to the reference function $x^5/500$. For $k = 5$, this seems to confirm our hypothesis that the running time can be bounded in $\mathcal{O}^*(r^k)$ on most networks. While the real-time setting (up to 17ms per run) ends rather early at $r \approx 12$, we can cover up to $r \approx 55$ in under 5 minutes, even for networks with many leaves ($n = 500$).

Solution size k . Figure 7 (right) shows a clear partition of growth scenarios for the running time dependent on k with $n = 60$ and $r = 12$ fixed: For $k \lesssim r$, the running time grows exponentially, roughly proportional to $4^k \approx r^{0.56k}$ but for $k \gtrsim r$, its growth behaves as a linear function in k . Again, this is characteristic for functions in $\mathcal{O}(\min\{r^k, 2^{6r}\} \cdot nk)$.

Each of the 18 instances with $n = 60$, $r = 12$ finished within 8 seconds for each $k \in [1, 40]$.

6 Discussion

Multiple formulations of the concept of phylogenetic diversity on phylogenetic networks have been proposed recently, and it is not yet clear which variants are most relevant in practice, for example for conservation biology. In our opinion, the average-tree phylogenetic diversity score (APD) appears to be a very natural and biologically relevant generalization of phylogenetic diversity on trees. In addition, in the small toy example in Figure 1, APD seems to make a more reasonable choice than NETWORKPD by selecting the hybrid species. However, to properly compare the different variants of phylogenetic diversity and to compare them to the feature diversity [32], extensive simulations and multiple case studies are needed. We believe that this is a very important topic for further research.

From a computational point of view, an interesting open problem that remains is whether our FPT algorithm for the maximization version of APD can be improved to a parameterized algorithm with respect to the level of the network. This would be interesting since the level can be much smaller than the current parameter, the number of reticulations. We conjecture that, in contrast to NETWORKPD (which is NP-hard for level-1 networks [27]), this is possible for APD. Another interesting direction would be to extend the algorithm to nonbinary networks.

Finally, while the theoretical running time of our algorithm is $\mathcal{O}(2^{6r} \cdot kn)$, experiments suggest that the practical running time behaves like $\mathcal{O}(\min\{r^k, 4^r\} \cdot kn)$ on networks generated by evolutionary processes. It could be interesting to study this further.

References

- 1 Eric Bapteste, Leo van Iersel, Axel Janke, Scot Kelchner, Steven Kelk, James O. McInerney, David A. Morrison, Luay Nakhleh, Mike Steel, Leen Stougie, and James Whitfield. Networks: expanding evolutionary thinking. *Trends in Genetics*, 29(8):439–441, 2013.
- 2 Magnus Bordewich, Charles Semple, and Andreas Spillner. Optimizing phylogenetic diversity across two trees. *Applied Mathematics Letters*, 22(5):638–641, 2009. doi:10.1016/j.aml.2008.05.004.
- 3 Magnus Bordewich, Charles Semple, and Kristina Wicke. On the complexity of optimising variants of phylogenetic diversity on phylogenetic networks. *Theoretical Computer Science*, 917:66–80, 2022. doi:10.1016/J.TCS.2022.03.012.
- 4 Olga Chernomor, Steffen Klaere, Arndt von Haeseler, and Bui Quang Minh. *Split Diversity: Measuring and Optimizing Biodiversity Using Phylogenetic Split Networks*, volume 14, pages 173–195. Springer Cham, 2016.
- 5 Tomás M. Coronado, Gabriel Riera, and Francesc Rosselló. An interchange property for the rooted phylogenetic subnet diversity on phylogenetic networks. *Journal of Mathematical Biology*, 89(5):48, 2024.
- 6 Paul Dagum and Michael Luby. Approximating the permanent of graphs with large factors. *Theoretical Computer Science*, 102(2):283–305, 1992. doi:10.1016/0304-3975(92)90234-7.
- 7 W. Ford Doolittle. Phylogenetic Classification and the Universal Tree. *Science*, 284(5423):2124–2128, 1999. doi:10.1126/science.284.5423.2124.
- 8 Daniel P. Faith. Conservation evaluation and phylogenetic diversity. *Biological Conservation*, 61(1):1–10, 1992.
- 9 Philippe Gambette, Vincent Berry, and Christophe Paul. The Structure of Level-k Phylogenetic Networks. In *Proceedings of the 20th Annual Symposium on Combinatorial Pattern Matching (CPM 2009)*, pages 289–300. Springer, 2009. doi:10.1007/978-3-642-02441-2_26.
- 10 Klaas Hartmann, Dennis Wong, and Tanja Stadler. Sampling Trees from Evolutionary Models. *Systematic Biology*, 59(4):465–476, 2010.
- 11 Niels Holtgreffe, Leo van Iersel, Ruben Meuwese, Yuki Murakami, and Jannik Schestag. PANDA: Maximizing Phylogenetic Diversity in Network. Manuscript in preparation, 2025.
- 12 Daniel H. Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: Concepts, Algorithms and Applications*. Cambridge University Press, 2010.
- 13 Laura Jetten and Leo van Iersel. Nonbinary Tree-Based Phylogenetic Networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(1):205–217, 2016.
- 14 Mark Jones and Jannik Schestag. How Can We Maximize Phylogenetic Diversity? Parameterized Approaches for Networks. In *Proceedings of the 18th International Symposium on Parameterized and Exact Computation (IPEC 2023)*, pages 30:1–30:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.IPEC.2023.30.
- 15 Mark Jones and Jannik Schestag. Maximizing Phylogenetic Diversity under Time Pressure: Planning with Extinctions Ahead. *arXiv preprint arXiv:2403.14217*, 2024. doi:10.48550/arXiv.2403.14217.
- 16 Mark Jones and Jannik Schestag. Parameterized Algorithms for Diversity of Networks with Ecological Dependencies. Manuscript in preparation, 2025.
- 17 Joshua A. Justison, Claudia Solis-Lemus, and Tracy A. Heath. SiPhyNetwork: An R package for simulating phylogenetic networks. *Methods in Ecology and Evolution*, 14(7):1687–1698, 2023.
- 18 Elizabeth Kolbert. *The sixth extinction: an unnatural history*. Henry Holt and Company, New York, 2014.
- 19 Christian Komusiewicz and Jannik Schestag. Maximizing Phylogenetic Diversity under Ecological Constraints: A Parameterized Complexity Study. In *Proceedings of the 44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2024)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2024.

- 20 Binh T. Nguyen, Andreas Spillner, and Vincent Moulton. Computing Phylogenetic Diversity for Split Systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(02):235–244, 2008. doi:10.1109/TCBB.2007.70260.
- 21 Fabio Pardi and Nick Goldman. Species Choice for Comparative Genomics: Being Greedy Works. *PLoS Genetics*, 1(6):e71, 2005. doi:10.1371/journal.pgen.0010071.
- 22 Roseli Pellens and Philippe Grandcolas. *Biodiversity Conservation and Phylogenetic Systematics: preserving our evolutionary heritage in an extinction crisis*. Springer Nature, 2016.
- 23 Jannik Schestag. Weighted Food Webs Make Computing Phylogenetic Diversity So Much Harder. Submitted for publication, 2025.
- 24 Jannik Schestag. *Who Should Have a Place on the Ark? Parameterized Algorithms for the Maximization of the Phylogenetic Diversity*. Doctoral dissertation, Friedrich Schiller University Jena, 2025. URL: https://www.fmi.uni-jena.de/fmi_femedia/33737/dissertation-schestag-25-pdf.pdf.
- 25 Mike Steel. Phylogenetic Diversity and the Greedy Algorithm. *Systematic Biology*, 54(4):527–529, 2005.
- 26 Scott A. Taylor and Erica L. Larson. Insights from genomes into the evolutionary importance and prevalence of hybridization in nature. *nature ecology & evolution*, 3(2):170–177, 2019.
- 27 Leo van Iersel, Mark Jones, Jannik Schestag, Celine Scornavacca, and Mathias Weller. Phylogenetic Network Diversity Parameterized by Reticulation Number and Beyond. In *Proceedings of the 23rd RECOMB International Workshop on Comparative Genomics (RECOMB-CG 2025)*, Seoul, Republic of Korea, 2025.
- 28 Leo van Iersel, Judith Keijsper, Steven Kelk, Leen Stougie, Ferry Hagen, and Teun Boekhout. Constructing Level-2 Phylogenetic Networks from Triplets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(4):667–681, 2009. doi:10.1109/TCBB.2009.22.
- 29 Leo van Iersel, Sjors Kole, Vincent Moulton, and Leonie Nipius. An algorithm for reconstructing level-2 phylogenetic networks from trinets. *Information Processing Letters*, 178:106300, 2022. doi:10.1016/j.ipl.2022.106300.
- 30 Mathias Weller. phylo_tools (c++20 branch). https://github.com/igel-kun/phylo_tools/tree/C%2B%2B20, 2025. Accessed: 2025-05-10.
- 31 Kristina Wicke and Mareike Fischer. Phylogenetic diversity and biodiversity indices on phylogenetic networks. *Mathematical Biosciences*, 298:80–90, 2018.
- 32 Kristina Wicke, Arne Mooers, and Mike Steel. Formal Links between Feature Diversity and Phylogenetic Diversity. *Systematic Biology*, 70(3):480–490, 2021.

A Appendix

A.1 Proof of Lemma 4.6

Even though only parts of the proof of Lemma 4.6 have been deferred to the appendix, we present the entire proof here.

► **Lemma 4.6.** *Let $\mathcal{S}' \subseteq \mathcal{S}(\mathcal{N})$ be a subset of sides and let (D^*, F) be derived from $(\mathcal{N}, \mathcal{S}')$ according to Construction 1. Then for $Z \subseteq X$, the signature of Z is $\text{SIG}_Z = \mathcal{S}'$ if and only if Z respects F . Moreover, if $\text{SIG}_Z = \mathcal{S}'$, then $\text{APD}_{\mathcal{N}}(Z) = \text{PD}_F(Z) + D^*$.*

Proof. Observe that by construction, each tree in F corresponds to a side $S \in \mathcal{S}'$, and as such Z respects F if and only if $\text{SIG}_Z = \mathcal{S}'$.

So now assume that $\text{SIG}_Z = \mathcal{S}'$. We prove that $\text{APD}_{\mathcal{N}}(Z) = \text{PD}_F(Z) + D^*$.

Note that every arc in F has a corresponding arc in \mathcal{N} . For ease of notation, we speak of the same arc e as existing in F and \mathcal{N} . Let $\omega_{\mathcal{N}}(e)$ denote the weight of an arc e in \mathcal{N} , and let ω_F denote the weight of e in F . For any arc e that is not a lowest path-arc in \mathcal{N} , let $\chi_F(e, Z) := 1$ if e has an offspring from Z in F , and let $\chi_F(e, Z) := 0$ otherwise. Observe that $\text{PD}_F(Z) = \sum_{e \in A(F)} \omega_F(e) \cdot \chi_F(e, Z)$.

Recall that $\psi_{\mathcal{N}}(e, Z)$ denotes the probability that, taken some switching σ , the arc e has an offspring from Z in T_{σ} . By Lemma 4.3 we know $\text{APD}_{\mathcal{N}}(Z) = \sum_{e \in A(\mathcal{N})} \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z)$.

▷ **Claim 4.7.** For an arc side S with lowest path-arc e_S , and e a path-arc of S that is not e_S , if $S \in \mathcal{S}'$ then $\omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) = \omega_F(e) \cdot \chi_F(e, Z) + \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}')$ and, otherwise, $\omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) = \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}')$.

Proof. First suppose that $S \in \mathcal{S}'$ and $\chi_F(e, Z) = 1$. Then e has an offspring $z \in Z$ which is a leaf of side S . As there are no reticulations on the path between e and z in \mathcal{N} , it holds that $\psi_{\mathcal{N}}(e, Z) = 1$ and so $\omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) = \omega_{\mathcal{N}}(e)$. On the other hand,

$$\begin{aligned} \omega_F(e) \cdot \chi_F(e, Z) + \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') &= \omega_F(e) + \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') \\ &= (1 - \psi_{\mathcal{N}}(e_S, \mathcal{S}')) \cdot \omega_{\mathcal{N}}(e) + \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') \\ &= \omega_{\mathcal{N}}(e) \end{aligned}$$

and so the claim holds.

Next suppose that $\chi_F(e, Z) = 0$. Then e has no offspring from side S in \mathcal{N} . It follows that e has an offspring from Z in T_{σ} if and only if e_S has an offspring from Z in T_{σ} , for any switching σ . Thus $\psi_{\mathcal{N}}(e, Z) = \psi_{\mathcal{N}}(e_S, Z) = \psi_{\mathcal{N}}(e_S, \mathcal{S}')$. But then $\omega_F(e) \cdot \chi_F(e, Z) + \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') = \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') = \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z)$, and so the claim holds. A similar argument holds for the case that $S \notin \mathcal{S}'$. ◁

▷ **Claim 4.8.** For a leaf-arc $e = vx$ with x on side S , it holds that $\omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) = \omega_F(e) \cdot \chi_F(e, Z)$ if $S \in \mathcal{S}'$, and $\psi_{\mathcal{N}}(e, Z) = 0$, otherwise.

Proof. Observe that $\psi_{\mathcal{N}}(e, Z) = 1$ if and only if $x \in Z$, and $\psi_{\mathcal{N}}(e, Z) = 0$ otherwise. Since Z has signature \mathcal{S}' , it follows that $\psi_{\mathcal{N}}(e, Z) = 0$ if $S \notin \mathcal{S}'$. On the other hand if $S \in \mathcal{S}'$, then $\chi_F(e, Z) = 1$ if and only if $x \in Z$ and so $\psi_{\mathcal{N}}(e, Z) = \chi_F(e, Z)$. The claim then follows from the fact that $\omega_{\mathcal{N}}(e) = \omega_F(e)$ when e is a leaf-arc. ◁

Recall that, for any lowest path-arc e_S on side S , it holds that $\psi_{\mathcal{N}}(e_S, Z) = \psi_{\mathcal{N}}(e_S, \mathcal{S}')$ by definition. Let $\mathcal{AS}(\mathcal{N})$ denote the arc sides on \mathcal{N} and let $\mathcal{RS}(\mathcal{N})$ denote the reticulation sides. Note that $\mathcal{S}(\mathcal{N}) = \mathcal{AS}(\mathcal{N}) \cup \mathcal{RS}(\mathcal{N})$. By construction it is $D^* = \sum_{S \in \mathcal{AS}(\mathcal{N})} \omega_{\mathcal{N}}(e_S) \cdot \psi(e_S, \mathcal{S}')$.

To prove the main claim, let E_1 denote the leaf-arcs of \mathcal{N} , let E_2 denote the path-arcs which are not the lowest path-arc on their side, and let E_3 denote the arcs which are the lowest path-arc of their side. For each $i \in \{1, 2, 3\}$, let E_i^1 denote the arcs in E_i belonging to a side in \mathcal{S}' , and let $E_i^0 = E_i \setminus E_i^1$. Note that the arc set of F is $E_1^1 \cup E_2^1$. For each arc $e \in E_2 \cup E_3$, let e_S denote the lowest path-arc on the side containing e (note that if $e \in E_3$ then $e_S = e$).

$$\begin{aligned} \text{APD}_{\mathcal{N}}(Z) &= \sum_{e \in A(\mathcal{N})} \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) \\ &= \sum_{e \in E_1^1 \cup E_1^0 \cup E_2^1 \cup E_2^0 \cup E_3} \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) \\ &= \sum_{e \in E_1^1} \omega_F(e) \cdot \chi_F(e, Z) + 0 + \sum_{E_2^1 \cup E_2^0 \cup E_3} \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) \\ &= \sum_{e \in E_1^1} \omega_F(e) \cdot \chi_F(e, Z) + \sum_{e \in E_2^1} (\omega_F(e) \cdot \chi_F(e, Z) + \omega_{\mathcal{N}} \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}')) \\ &\quad + \sum_{E_2^0 \cup E_3} \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e, Z) \end{aligned}$$

$$\begin{aligned}
&= \sum_{e \in E_1^1} \omega_F(e) \cdot \chi_F(e, Z) + \sum_{e \in E_2^1} \omega_F(e) \cdot \chi_F(e, Z) + \sum_{e \in E_2^1} \omega_{\mathcal{N}} \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') \\
&\quad + \sum_{E_2^0} \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') + \sum_{E_3} \omega_{\mathcal{N}}(e) \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') \\
&= \sum_{e \in E_1^1 \cup E_2^1} \omega_F(e) \cdot \chi_F(e, Z) + \sum_{e \in E_2^1 \cup E_2^0 \cup E_3} \omega_{\mathcal{N}} \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') \\
&= \sum_{e \in A(F)} \omega_F(e) \cdot \chi_F(e, Z) + \sum_{e \in E_2^1 \cup E_2^0 \cup E_3} \omega_{\mathcal{N}} \cdot \psi_{\mathcal{N}}(e_S, \mathcal{S}') = \text{PD}_F(Z) + D^* \quad \blacktriangleleft
\end{aligned}$$

A.2 Proof of Lemma 4.9

► **Lemma 4.9.** *MAX-PD can be solved in $\mathcal{O}(nk)$ time on forests.*

Proof. Let $\mathcal{I} := (F, k, D)$ be an instance of MAX-PD with forest $F = \{T_1, \dots, T_s\}$ where ρ_i is the root of T_i . Let M be an integer bigger than $\omega(A(F))$.

We define a tree T with root ρ , as the union of T_1, \dots, T_s with additional arcs $\rho\rho_i$ of weight M for each $i \in \{1, \dots, s\}$. Solve the instance $\mathcal{I}' := (T, k, D + s \cdot M)$ of MAX-PD with Faith's greedy algorithm and return the result.

Let $Z \subseteq X^*$ be a solution for \mathcal{I} . Because Z respects F , there is a taxon $x_i \in Z$ beneath ρ_i . Consequently, $\rho\rho_i$ is on the path from ρ to x_i and so $\text{PD}_T(Z) = sM + \text{PD}_F(Z) \geq sM + D$.

Conversely, let $Z \subseteq X^*$ be a solution for \mathcal{I}' . As $\text{PD}_T \geq D + sM$, we conclude that Z contains a taxon of each tree and therefore respects F . Therefore, Z is a solution for \mathcal{I} .

\mathcal{I}' is constructed in linear time. On trees, MAX-PD is solved in $\mathcal{O}(nk)$ time [25, 21]. ◀