Classification of Phylogenetic Networks

Anastasios Stefanou

Abstract

By considering rooted Reeb graphs as a model for phylogenetic networks, using tools from category theory we construct an injection that assigns to each phylogenetic network with \( n \)-labelled leaves and \( s \) cycles a finite set of phylogenetic trees with \((n + s)\)-labelled leaves. In particular, we show this map is canonical, i.e. it classifies phylogenetic networks up to isomorphism. Finally we discuss some upper bounds for the isomorphism complexity of these networks.

Contents

1 Introduction 1
2 Categorical structures 2
2.1 Basic definitions 3
2.2 Coproducts 4
3 Rooted Reeb graphs 5
3.1 Reeb graphs versus directed acyclic graphs 5
3.2 Tree-decomposition of simple rooted Reeb graphs 9
4 Phylogenetic networks 12
4.1 Classification of phylogenetic networks 13
4.2 Isomorphism complexity 16
5 Concluding remarks 17

1 Introduction

The study of evolution originated from the study of the origin of species by Charles, Darwin [13] and continued with the development of the research field of phylogenetics which studies and analyze the evolutionary relationships among biological organisms. Among a plethora of applications include DNA sequences, viral and cancer evolution. The mathematical tool that is often used to model evolutionary relationships among objects is a rooted tree. One often assigns weights on the vertices of these rooted trees and obtains a diagram called dendogram or merge tree [25, 20]. These dendrograms usually are constructed from datasets via hierarchical clustering methods [24, 22, 7]. In biology when we study dendrograms we also require that the leaves of the tree are bijectively labelled by a set of ‘taxa’ which we think of as a set of integers \( 1, \ldots, n \). The resulting structure it is called a phylogenetic tree. Phylogenetic trees have been extensively studied both on their structure [22, 16, 18] as well as on their metric structure [14]. However, in biology when we study time evolving data we often encounter recombinations among objects known as reticulations. Hence we are interested in studying general non tree-like evolutionary relationships among objects. The resulting mathematical structure that we obtain is called a rooted directed acyclic graph. The reticulations corresponds precisely to the (undirected) cycles of the graph. Again, just like the tree-case we require that the leaves of these graphs are bijectively labelled with a set of ‘taxa’ often represented by the integers \( 1, \ldots, n \). This mathematical structure generalizes the notion of a phylogenetic tree and it is known in the literature as a phylogenetic network [17]. In practice one studies weighted variants of these networks. In this paper, however we think of a weighted phylogenetic network as a rooted Reeb graph where its leaves are bijectively labelled by the integers \( 1, \ldots, n \) and argue why this is an appropriate model. We are particularly motivated by the following problems related to phylogenetic networks:
• The Classification Problem: Find a mathematical model to represent the isomorphic classes of phylogenetic networks with \( n \)-labelled leaves.

• The Metric Problem: Find a metric for comparison of the isomorphic classes of phylogenetic networks with \( n \)-labelled leaves.

• The Isomorphism Complexity Problem: Given a pair of phylogenetic networks with \( n \)-labelled leaves, decide if they are isomorphic.

Related work The Classification problem was solved for the special case of binary phylogenetic trees already in 1857 by Arthur Cayley by considering a correspondence between trees and nested parentheses. This correspondence was later named as the Newick format of a phylogenetic tree. For the general case of a phylogenetic tree with weighted nodes, in 1962 Sokal and Rohlf in \cite{25} constructed an injection of leaves of the tree. This injective map which is known in the literature as the cophenetic map \cite{4}. Since every node in a phylogenetic tree \( T \) with \( n \)-leaves is the join of two leaves one can think of this map as a particular type of an \( n \)-motif in the general setting of networks \cite{9,11}. Using the cophenetic map we can pull back the \( \ell^p \)-norms to define metrics for comparison of phylogenetic trees \cite{4}. In particular the \( \ell^\infty \)-distance can be realized as an interleaving distance—in the setting of topological data analysis—on phylogenetic trees \cite{21}.

For the general case of classifying phylogenetic networks the situation is more complicated. A naive approach to the classification problem is to consider the set of all spanning trees of a phylogenetic network. However these invariants do not completely characterize phylogenetic networks up to isomorphism. That is, one can find a pair of non-isomorphic phylogenetic networks with the same set of spanning trees. A partial answer to the problem was proposed from some biologists by considering a variant of the Newick format known as the extended Newick format \cite{5}. The idea is to consider for a given phylogenetic network with \( n \)-labelled leaves and \( s \) cycles a phylogenetic tree with \((n+s)\)-labelled leaves where some of the leaves are ‘allowed to have repeated nodes’ \cite{5}. However, the extended Newick format—in contrast with the tree case—it is also dependent also on a fixed choice of ordering on the reticulation nodes that is taken independent from the ordering of the leaves of the network. Thus the extended Newick format is not able to classify—up to isomorphism—phylogenetic networks where only their leaves are ordered or labelled. In particular, that means that there is no clear way how to use this format to define a metric on the isomorphism classes of phylogenetic networks with \( n \)-labelled leaves.

For the metric problem, there are several metrics that have been proposed for special types of networks. The major part of the problem lies on how to properly model the isomorphism classes of these networks that comes back to the classification problem.

For the isomorphism complexity problem the complexity of phylogenetic networks remains unknown. However, it has recently showed that the phylogenetic network isomorphism is GI-complete, namely it is as difficult as solving the graph isomorphism problem (whose complexity is also unknown) \cite{3}.

Our contribution In this paper we focus primarily on the classification problem but discuss potential interactions with the other problems. We propose an alternative approach to the extended Newick format by considering a tree-decomposition for phylogenetic networks. We show that this isomorphism invariant of phylogenetic networks is complete. First, in Sec. 2 and 3 we mention the basic definitions and tools from category theory \cite{14} and the setting of Reeb graphs as studied in \cite{14} which we need in order to formulate properly the classification of phylogenetic networks. In Sec 3 we show that the category of simple rooted Reeb graphs with a fixed edge-labelling forms a thin category. In particular we show that inside this category each simple rooted Reeb graph decomposes as a coproduct of merge trees. In Sec. 4 we provide a classification of phylogenetic networks with \( n \)-labelled leaves up to isomorphism by applying carefully the tree-decomposition theorem of simple rooted Reeb graphs of Sec. 3. Finally, we prove two upper bounds to the isomorphism complexity of phylogenetic networks.

2 Categorical structures Category theory is a general theory of functions. A general notion of a function is called a morphism and the notion of a set is replaced by an object. An object can be any mathematical construction and its not necessary to be a set. In contrast with set theory the focus is concentrated in the study of morphisms between objects rather than just study the objects themselves. In particular, we require that morphisms
between objects to have a composition operation that is associative and unital. The structure we obtain is said to be a category [10]. Category theory is fundamentally a language that formalizes mathematical structure having the capability of bridging together different mathematical constructions or theories. In this section we give the basic definitions and tools from category theory that we need.

2.1 Basic definitions

First we define the notion of a category. Here by a class we mean a collection of sets that is unambiguously defined by property that all these sets share in common. A class might not be a set and if that is the case is called a proper class.

Definition 2.1. A category $\mathcal{C}$ consists of

- a class $\text{Ob}\mathcal{C}$ whose elements $X, Y, \ldots$ are called objects, together with
- for each pair of objects $X, Y$ in $\mathcal{C}$ a set $\text{Mor}_\mathcal{C}(X, Y)$, whose elements are called morphisms and denoted by $f : X \to Y$, and each having a unique source $X$ and a unique target $Y$,
- for each object $X$ in $\mathcal{C}$ an identity morphism $I_X : X \to X$,
- a binary operation $\circ : \text{Mor}_\mathcal{C}(X, Y) \times \text{Mor}_\mathcal{C}(Y, Z) \to \text{Mor}_\mathcal{C}(X, Z)$, $(f, g) \mapsto g \circ f$ called composition which is associative and unital, i.e.

\[
\begin{align*}
  h \circ (g \circ f) &= (h \circ g) \circ f \\
  f \circ I_Y &= I_Y \circ f
\end{align*}
\]

for any triple of morphisms $f : X \to Y$, $g : Y \to Z$ and $h : Z \to W$ in $\mathcal{C}$.

Definition 2.2. A morphism $f : X \to Y$ is said to be an isomorphism from $X$ to $Y$ if there exists a morphism $g : Y \to X$ (often called the inverse) such that $g \circ f = I_X$ and $f \circ g = I_Y$. Two objects are said to be isomorphic if there exists an isomorphism from $X$ to $Y$.

If the class of object $\text{Ob}\mathcal{C}$ is a set then $\mathcal{C}$ is said to be small. Else, if $\text{Ob}\mathcal{C}$ is a proper class (not a set) then $\mathcal{C}$ is said to be large.

Example 2.3. Examples of large categories include:

- the category $\text{Set}$ whose objects are sets and morphisms are functions between sets
- the category $\text{Top}$ whose objects are topological spaces and morphisms are continuous maps.
- the category $\text{Groups}$ whose objects are groups and morphisms are group homomorphisms
- the category $\text{Ab}$ whose objects are abelian groups and morphisms are group homomorphisms

Definition 2.4. A category $\mathcal{D}$ whose objects and morphisms are in $\mathcal{C}$ and with the same identities and composition operation as of $\mathcal{C}$ is said to be a subcategory of $\mathcal{C}$.

Let $\mathcal{C}$ be a category and let $\mathcal{S}$ be any subset of $\text{Ob}\mathcal{C}$. Then we can consider the same sets $\text{Mor}_\mathcal{C}(X, Y)$ of morphisms between $X, Y \in \mathcal{S}$. That way we obtain a category with the same morphisms but fewer objects. We say that $\mathcal{S}$ forms a full subcategory of $\mathcal{C}$. In a full subcategory we only need to specify what are the objects so we often say $\mathcal{S}$ is the full subcategory of $\mathcal{C}$ whose objects are in $\mathcal{S}$. For example the category $\text{Ab}$ is the full subcategory of $\text{Groups}$ whose objects are abelian groups.

Example 2.5. The type of categories we work on are the following:

- slice categories: given a category $\mathcal{C}$ and an object $X$ we consider the slice category $\mathcal{C} \downarrow X$ whose objects are tuples $(Y, f)$ where $Y \in \text{Ob}\mathcal{C}$ and $f \in \text{Mor}_\mathcal{C}(Y, X)$, and morphisms $\varphi : (Y, f) \to (Z, g)$ are ordinary morphisms $\varphi : Y \to Z$ in $\mathcal{C}$ such that $g \circ \varphi = f$.
- thin categories: a category $\mathcal{C}$ is called thin if for every pair of objects $X, Y$ in $\mathcal{C}$ there exists at most one morphism $f : X \to Y$ in $\mathcal{C}$. When a morphism $f : X \to Y$ exists we write $X \leq Y$. A thin category coincides with the notion of a preorder.

Now, we define the notion of maps that preserve the structure of a category.

Definition 2.6. A functor $F : \mathcal{C} \to \mathcal{D}$ between categories consists of

- a function $F : \text{Ob}\mathcal{C} \to \text{Ob}\mathcal{D}$, $X \mapsto F(X)$, together with
Definition 2.10. Let \( \mathcal{C} \) be a category. We define the notion of a coproduct of objects in \( \mathcal{C} \) as follows:

\[
\coprod_{i \in I} X_i \quad \text{for each pair of objects } X_i, Y_i \in \mathcal{C}, \text{ a function }
\]

\[
\mathbf{F}_{X,Y} : \text{Mor}_\mathcal{C}(X, Y) \to \text{Mor}_\mathcal{D}(\mathbf{F}(X), \mathbf{F}(Y))
\]

\[
f \mapsto \mathbf{F}[f]
\]

such that for any object \( X \) and any morphisms \( f : X \to Y, g : Y \to Z \) in \( \mathcal{C} \):

\[
\mathbf{F}[g \circ f] = \mathbf{F}[g] \circ \mathbf{F}[f]
\]

\[
\mathbf{F}[\mathbf{1}_X] = \mathbf{1}_{\mathbf{F}(X)}.
\]

When \( \mathcal{C} = \mathcal{D} \), \( \mathbf{F} \) is called an endofunctor. A special case is the identity endofunctor \( \mathbf{1}_\mathcal{C} : \mathcal{C} \to \mathcal{C} \) that sends each object and morphism to itself.

The collection of all functors from a category \( \mathcal{C} \) to a category \( \mathcal{D} \) forms a category on its own called a functor category and it is denoted by \([\mathcal{C}, \mathcal{D}]\): the objects are functors \( \mathbf{F} : \mathcal{C} \to \mathcal{D} \) and the morphisms are natural transformations \( \eta : \mathbf{F} \Rightarrow \mathbf{G} \).

Definition 2.7. A natural transformation \( \eta : \mathbf{F} \Rightarrow \mathbf{G} \) consists of a family of morphisms \( \eta_X : \mathbf{F}(X) \to \mathbf{G}(X) \) in \( \mathcal{D} \) one for each object \( X \) in \( \mathcal{C} \), such that the diagram

\[
\begin{array}{ccc}
\mathbf{F}(X) & \xrightarrow{\eta_X} & \mathbf{G}(X) \\
\mathbf{F}(f) & \downarrow & \mathbf{G}(f) \\
\mathbf{F}(Y) & \xrightarrow{\eta_Y} & \mathbf{G}(Y)
\end{array}
\]

commutes for every morphism \( f : X \to Y \) in \( \mathcal{C} \). In the special case where each \( \eta_X \) is an isomorphism in \( \mathcal{D} \), then \( \eta \) is said to be a natural isomorphism and we write \( \mathbf{F} \cong \mathbf{G} \).

Every time we write \( \mathbf{F} \cong \mathbf{G} \) we mean there exists a natural isomorphism \( \eta : \mathbf{F} \Rightarrow \mathbf{G} \).

Definition 2.8. A pair of categories \( \mathcal{C}, \mathcal{D} \) are said to be equivalent if there exist functors \( \mathbf{F} : \mathcal{C} \to \mathcal{D} \) and \( \mathbf{G} : \mathcal{D} \to \mathcal{C} \) such that \( \mathbf{F} \circ \mathbf{G} \cong \mathbf{1}_\mathcal{D} \) and \( \mathbf{G} \circ \mathbf{F} \cong \mathbf{1}_\mathcal{C} \). In the special case where \( \mathbf{F} \circ \mathbf{G} = \mathbf{1}_\mathcal{D} \) and \( \mathbf{G} \circ \mathbf{F} = \mathbf{1}_\mathcal{C} \), the categories \( \mathcal{C} \) and \( \mathcal{D} \) are said to be isomorphic.

Remark 2.9. A category \( \mathcal{P} \) is a poset if and only if \( \mathcal{P} \) is both thin and skeletal.

2.2 Coproducts

We define the notion of a coproduct of objects in a category \( \mathcal{C} \). This is the dual notion of a product. However we focus only on the definition of coproducts since this is the only notion we use.

Definition 2.10. Let \( X_1, \ldots, X_n \) be objects in \( \mathcal{C} \). An object is called the coproduct of \( X_1, X_2, \ldots, X_n \), written \( \coprod X_i \), if there exist morphisms \( \iota_j : X_j \to \coprod X_i, j = 1, 2, \ldots, n \) satisfying the following universal property: for any object \( Y \) and any pair of morphisms \( f_j : X_j \to Y, j = 1, 2, \ldots, n \), there exists a unique morphism \( f : \coprod X_i \to Y \) such that the diagrams

\[
\begin{array}{ccc}
X_j & \xrightarrow{\iota_j} & \coprod X_i \\
\downarrow & & \downarrow f \\
Y & & Y
\end{array}
\]

commute for \( j = 1, 2, \ldots, n \).

Note that by the universal property of coproducts, the morphisms \( \iota_j \) are uniquely defined up to a unique natural isomorphism. The morphisms \( \iota_j : X_j \to \coprod X_i \) are called coprojections.

Definition 2.11. An object \( X \) is said to be decomposable if it is isomorphic to a coproduct of \( n \) objects in \( \mathcal{C} \) where \( n \geq 2 \). Otherwise \( X \) is said to be indecomposable.

Example 2.12. Here we give some basic examples of categorical coproducts.

- If \( \mathcal{C} \) is the category of all sets \( \textbf{Set} \), then the coproduct is given by the disjoint union \( \coprod \) of sets.
- If \( \mathcal{C} \) is the category of groups \( \textbf{Groups} \) then the coproduct is given by the free product \( \ast \) of groups.
- If \( \mathcal{C} \) is the category of abelian groups \( \textbf{Ab} \) then the coproduct is the direct sum \( \oplus \) of abelian groups.
3 Rooted Reeb graphs

In this section we consider the setting of Reeb graphs as developed by V. de Silva, A. Patel and E. Munch in \[14\]. We define simple rooted Reeb graphs and examine how they relate to rooted directed acyclic graphs. Then we show that simple rooted Reeb graphs with a fixed edge-labelling form a thin category inside of which any simple rooted Reeb graph decomposes as a coproduct of merge trees.

3.1 Reeb graphs versus directed acyclic graphs

The main tool we use to visualize relationships among objects is a graph. A graph \( G = (V, E) \) consists of a collection \( V \) of objects called vertices, e.g. \( v_1, v_2, \ldots \) and a set \( E \) of connections \( e_1, e_2, \ldots \) between vertices called edges. A graph can have multiple edges connecting a pair of vertices and edges connecting a vertex with itself called self loops. A graph with no multiple edges and no self loops is said to be simple. To distinguish the simple graph from the general case of a graph, a graph that is allowed to have multiple edges or self loops is known in the literature as a multigraph. In the literature, however, when we talk about a graph we always assume it is simple. Hence any edge \( e \) of a graph always has exactly two incident vertices, say \( x, y \). So, we often denote an edge \( e \) by a set with two elements \( e = \{x, y\} \). If we allow the edges \( e \) of a graph to have a direction, i.e. \( x \xrightarrow{e} y \) then the resulting graph is said to be a digraph and the edges are denoted by an ordered tuple \((x, y)\). A directed path of length \( n \) on a digraph is a sequence of arrows \((x_0, x_1), (x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n)\) in \( G \). A directed cycle \((x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n), (x_n, x_0)\) that starts and ends at the same vertex is called a directed cycle. A digraph with no directed cycles is said to be a directed acyclic graph (DAG). A DAG is said to be rooted if there exists a vertex \( v_0 \), called the root, such that every vertex \( v \) is connected with \( v_0 \) by a directed path that starts at \( v_0 \) and ends at \( v \). The vertices that have a directed path with \( v_0 \) of maximum length are called leaves. Rooted DAGs find a plethora of applications in biology including phylogenetics. A phylogenetic network is a rooted DAG whose leaves are bijectively labelled by a set of integers \( 1, \ldots, n \). These networks often arise from biological entities (such as genes) that evolve through time. Thus, it is natural to assume that phylogenetic networks are equipped with a function that keeps in track of the time instances that each evolutionary relationship occurs. Hence, we are interested in considering phylogenetic networks equipped with a ‘height function’, where the word ‘height’ stands for time. We believe that a natural mathematical model to describe these ‘time-evolving’ phylogenetic networks is a Reeb graph.

Reeb graph

Reeb graphs have originally studied in the context of Morse theory and recently have found a lot of applications. More generally we can define a Reeb graph as a graph \( X \) together with a real valued map \( f : X \to \mathbb{R} \) which is strictly monotone when restricted to edges. However with this definition we are not making precise the exact way \( X \) can be construct in conjunction with the map \( f \) being monotone restricted to edges. Making this precise specifically for the rooted case is what we do in this paragraph.

First we need to talk about the general setting of \( R \)-spaces. An \( R \)-space \((X, f)\) is a space \( X \) together with a real valued continuous map \( f : X \to \mathbb{R} \). A morphism of \( R \)-spaces \((X, f), (Y, g)\) also called a function preserving map— is an ordinary continuous map \( \varphi : X \to Y \) such that \( g \circ \varphi = f \). The collection of these objects forms the slice category \( \text{Top} \downarrow \mathbb{R} \). Now let us return to Reeb graphs. Specifically we focus on rooted Reeb graphs. Since rooted Reeb graphs can have roots with different heights and we don’t want to bound their height we ‘extend the root at infinity’ in the sense of the following definition.

Definition 3.1. An \( R \)-space \((X, f)\) is said to be a rooted Reeb graph if it is constructed by the following procedure, which we call a structure on \((X, f)\):

Let \( S = \{a_1 < \cdots < a_k\} \) be an ordered subset of \( \mathbb{R} \) called a critical set of \( X \).

- For each \( i = 1, \ldots, k \) we specify a set \( V_i \) of vertices which lie over \( a_i \),
- For each \( i = 1, \ldots, k - 1 \) we specify a set of edges \( E_i \) which lie over \([a_i, a_{i+1}]\)
- For \( i = k \) we specify a singleton set \( E_k \) that lie over \([a_k, \infty)\),
- For \( i = 1, \ldots, k \), we specify a down map \( D_i : E_i \to V_i \)
- For \( i = 1, \ldots, k - 1 \), we specify an upper map \( U_i : E_i \to V_{i+1} \).

The space \( X \) is the quotient \( \mathbb{U} / \sim \) of the disjoint union

\[
\mathbb{U} = \coprod_{i=1}^{k} (V_i \times \{a_i\}) \coprod_{i=1}^{k-1} (E_i \times [a_i, a_{i+1}]) \coprod (E_k \times [a_k, \infty])
\]
with respect to the identifications \((D_i(e), a_i) \sim (e, a_i)\) and \((U_i(e), a_{i+1}) \sim (e, a_{i+1})\), with the map \(f\) being the projection onto the second factor.

See Fig. 2 for an example of a Reeb graph.

**Definition 3.6.** A morphism of Reeb graphs \((X, f)\) and \((Y, g)\) is any function preserving map \(\varphi : (X, f) \rightarrow (Y, g)\) between these \(\mathbb{R}\)-spaces.

Thus, the collection of all rooted Reeb graphs forms a full subcategory \textbf{RootReeb} of \textbf{Top} \downarrow \mathbb{R}. Given a function preserving map \(\varphi : (X, f) \rightarrow (Y, g)\) since \(f\) and \(g\) are by definition the projections to the second coordinate of \(X\) and \(Y\) respectively, the map \(\varphi\) is given by

\[
\varphi : (X, f) \rightarrow (Y, g)
\]

\[
[(x, t)] \mapsto [\Phi(x), t],
\]

where \(\Phi = p_1 \circ \varphi\) and the bracket \([\ ]\) denotes an equivalence classes in the quotient space \(X = U/\sim\).

As shown in [14] Reeb graphs can be identified with nice \textbf{Set}-valued cosheaves on \(\mathbb{R}\). This equivalence of categories allows us to consider a useful combinatorial description of the morphisms of rooted Reeb graphs.

**Proposition 3.3** (Prop 3.12 in [14]). Let \((X, f), (Y, g)\) be a pair of Reeb graphs with a common critical set \(S = \{a_1, \ldots, a_k\}\). Any function preserving map \(\varphi : (X, f) \rightarrow (Y, g)\) of rooted Reeb graphs is completely determined by \(\Phi = p_1 \circ \varphi\) which, in turn, is determined by:

- Functions \(\varphi^X_i : V^X_i \rightarrow V^Y_i\)
- Functions \(\varphi^E_i : E^X_i \rightarrow E^Y_i\), satisfying the
- Consistency conditions: \(\varphi^X_i D^X_i = D^Y_i \varphi^Y_i\) and \(\varphi^X_i U^X_i = U^Y_i \varphi^Y_i\) for all \(1 \leq i \leq k - 1\).

Let \((X, f)\) be a rooted Reeb graph with critical set \(S = \{a_1, \ldots, a_k\}\).

**Definition 3.4.** A point \(x\) in \(X\) is said to be a node if its function value (‘height’) is a ‘critical value’, i.e. \(f(x) \in S\).

Given a pair of nodes \(x_1, x_2\) in \(X\) we can define joins and meets.

**Definition 3.5.** The join \(x_1 \vee x_2\) of \(x_1\) and \(x_2\), if it exists, is the unique node in \(X\) such that:

- \(f(x_1 \vee x_2) \geq f(x_1), f(x_2)\),
- \(x_1\) and \(x_2\) are connected with \(x_1 \vee x_2\) by two monotone increasing paths,
- the function value of \(x_1 \vee x_2\) is minimal.

**Definition 3.6.** The meet \(x_1 \wedge x_2\) of \(x_1\) and \(x_2\), if it exists, is the unique node in \(X\) such that:

- \(f(x_1 \wedge x_2) \leq f(x_1), f(x_2)\),
- \(x_1 \wedge x_2\) is connected with \(x_1\) and \(x_2\) by two monotone increasing paths,
- the function value of \(x_1 \wedge x_2\) is maximal.

Let \(x\) be a node of \(X\) with \(f(x) = a_i\). Then an edge \(e \in D^{-1}_i(x)\) (\(e \in U^{-1}_i(x)\)) is said to be an edge incident from above (below) of \(x\). Also the number of edges \(\text{indeg}(x) = |D^{-1}_i(x)|\) that are incident from above of \(x\) is called the indegree of \(x\). Similarly, the number of edges \(\text{outdeg}(x) = |U^{-1}_i(x)|\) is called the outdegree of \(x\). Since \(X\) is rooted, there are four cases for a node \(x\):

- If \(\text{indeg}(x) = 1\) and \(\text{outdeg}(x) = 1\) then \(x\) is said to be a regular-node.
- If \(\text{indeg}(x) = 1\) and \(\text{outdeg}(x) \geq 2\) is said to be a tree-node.
- If \(\text{indeg}(x) = 1\) and \(\text{outdeg}(x) = 0\) then \(x\) is said to be a leaf-node.
- If \(\text{indeg}(x) \geq 2\) then \(x\) is said to be a reticulation-node.

**Notation 3.7.** We denote by \(L(X)\) the set of all leaf-nodes of \(X\) and by \(R(X)\) the set of all reticulation-nodes of \(X\).

**Definition 3.8.** A rooted Reeb graph \((X, f)\) with no reticulation nodes (no cycles) is said to be a merge tree.
Figure 1: On the left a non-simple rooted graph. On the right a simple rooted graph.

If we forget the height function \( f \) of a Reeb graph, then topologically \( X \) is a graph on its own. Rooted Reeb graphs might have multiple edges. However, here we restrict to rooted Reeb graphs whose underlying graph \( X \) is simple. See Fig. 1 for an example of a simple rooted graph. In additional to graph simplicity, we also want to ‘remove’ the regular-nodes: up to isomorphism we can consider that the down and upper maps restricted to regular-nodes are identities, i.e. for each regular-node \( x \) with \( f(x) = a_i \) we have that \( x = U_{i}^{-1}(x) = D_{i}^{-1}(x) \). The idea is: given a regular-node \( x \) with function value \( a_i \), located between two non-regular nodes, say \( u, v \), we can consider an isomorphic rooted Reeb graph by replacing \( x \) and the values of the singletons \( U_{i}^{-1}(x), D_{i}^{-1}(x) \) by a single ‘edge point’ \((u, v)\). By doing that for each regular-node, we obtain a new structure for our Reeb graph. Any structure on \( X \) of this form is said to be minimal.

**Definition 3.9.** A rooted Reeb graph \((X, f)\) is said to be simple if its structure is minimal and its underlying graph \( X \) is simple.

In the setting of simple rooted Reeb graphs an edge of \( X \) although is denoted by \((u, v)\) (where \( u, v \) are the source and target nodes respectively of the edge) is considered as a point in some level set of \( X \). So, for convenience we will often call it an edge-point. In a simple rooted Reeb graph all edges are denoted by an ordered tuple \((x, y)\) of non-regular nodes. Furthermore each regular-node is identified with an edge point. To make all of these definitions and facts clear we consider the following toy example.

**Example 3.10.** Consider the simple rooted graph \((X, f)\) located on the right side of Fig. 1. Assume that it is a Reeb graph with critical set \( \{a_1, a_2, a_3, a_4\} \) as in Fig. 2. Now we construct a minimal structure on \( X \). On the right we denote the node-sets \( \mathbb{V}_i \) in blue color and the edge-sets \( \mathbb{E}_i \) in green color. In Fig. 2 we describe how the down maps \( D_i \) and upper maps \( U_i \) work. Note that when restricted to regular-nodes we consider them as identities, that is: If \( x \) is a regular-node inside some edge-point \((y, z)\) then we identify \( x \) with that edge-point \((y, z)\). The node-sets are \( \mathbb{V}_1 = \{l_1\}, \mathbb{V}_2 = \{t_1, r_1\}, \mathbb{V}_3 = \{(t_2, r_1), t_1\} \) and \( \mathbb{V}_4 = \{t_2\} \). The edge-sets are \( \mathbb{E}_1 = \{(r_1, l_1)\}, \mathbb{E}_2 = \{(t_2, r_1), (t_1, r_1), (t_1, l_2)\}, \mathbb{E}_3 = \{(t_2, r_1), (t_2, t_1)\} \) and \( \mathbb{E}_4 = \{\bullet\} \), where the ‘infinite’ edge we identify with a bullet. Now construct the coproduct \( \mathbb{U} \) as in Defn. 3.1. Then, to obtain the simple rooted Reeb graph \( X \) on the left side of the figure we consider the following identifications on \( U \) as in Defn. 3.1 (and ignore the identities):

- \((D_1((r_1, l_1)), a_1) \sim (l_1, a_1)\)
- \((U_1((r_1, l_1)), a_2) \sim (r_1, a_2) \sim (D_2((t_2, r_1)), a_2) \sim (D_2((t_1, r_1)), a_2) \text{ and } (D_2((t_1, l_2)), a_2) \sim (l_2, a_2)\)
- \((U_2((t_1, l_1)), a_3) \sim (U_2((t_2, l_2)), a_3) \sim (t_1, a_3) \sim (D_3((t_2, t_1)), a_3)\)
- \((U_2((t_2, r_1)), a_4) \sim (U_2((t_2, l_1)), a_4) \sim (t_2, a_4) \sim (D_4(\bullet), a_4)\)

The map \( f \) is given by the projection of \( X \) onto the second coordinate.
Figure 2: An example of a simple rooted Reeb graph
By construction, the underlying graph $X$ of a simple rooted Reeb graph $(X, f)$ can be thought of as a rooted directed acyclic graph $(V, E)$, where $V$ and $E$ are the sets of nodes and edges of $X$. This is important specifically for computing the number of cycles $s$ on a simple rooted Reeb graph $X$. From the theory of directed graphs we have the degree sum formula

$$\sum_{x \in V} \text{ indeg}(x) = \sum_{x \in V} \text{ outdeg}(x) = |E|$$

Furthermore since $X$ is always connected the Euler characteristic provides the formula

$$s = |E| - |V|.$$ 

By combining the previous equation we obtain the formula for $s$:

$$s = \sum_{x \in V} (\text{ indeg}(x) - 1).$$

Let $r_1, \ldots, r_m$ be the reticulation nodes of $X$ with indegrees $d_1, \ldots, d_m \geq 2$ respectively. Since all the nodes except the reticulation nodes have indegree 1 these nodes vanish in the summation formula. Thus we have the following simpler formula for computing the number of cycles in $X$:

$$s = \sum_{i=1}^{m} (d_i - 1).$$

Although a simple rooted Reeb graph can be thought of as a rooted DAG, the other direction is not true: not every rooted DAG can be thought of as a simple rooted Reeb graph. Even if we restrict to combinatorial Reeb graphs—that is Reeb graphs that have consecutive critical values represented by integers—there are DAGs that cannot be realized in any way as Reeb graphs as illustrated by Ex. 3.11.

**Example 3.11.** Consider a rooted DAG, $G$, with root $x_1$, that has edges $e_1 = (x_1, x_2), e_2 = (x_2, x_3)$ connecting $x_1$ with $x_2$ and $x_2$ with $x_3$ and an edge $e_3 = (x_1, x_3)$ connecting directly $x_1$ with $x_3$ with a single edge. That means, we cannot construct a height function $f$ on $G$ that is strictly monotone when restricted to edges. Hence $G$ cannot be realized as some rooted Reeb graph. So $G$ is a DAG that is not consistent with ‘time’ in the sense that the edge $e_2 = (x_2, x_3)$ represents a change in the nodes from $x_2$ to $x_3$ that happen instantaneously. In other words the edge $e_2$ is a ‘horizontal’ edge.

![Diagram](attachment:image.png)

This behaviour for phylogenetic networks is rather unrealistic to arise in real applications. This is one of the main reasons why we consider a simple rooted Reeb graph rather than a rooted DAG as a natural model for phylogenetic networks.

### 3.2 Tree-decomposition of simple rooted Reeb graphs

Fix an ordered subset $S = \{a_1 < \ldots < a_k\}$ of $\mathbb{R}$. Consider a Reeb graph as in Defn. 3.1 and consider

$$E = \{E_1 \times [a_1, a_2], \ldots, E_{k-1} \times [a_{k-1}, a_k], E_k \times [a_k, \infty)\}$$

the associated finite sequence of edge-sets that lie over each closed interval between consecutive elements of $S$ and $\infty$. We call $E$ an edge-sequence on a rooted Reeb graph.

Fix an edge-sequence $E$ and consider the subcategory $\text{SimpReeb}[E]$ of $\text{RootReeb}$ where:

- the objects are simple rooted Reeb graphs $(X, f)$ with its associated edge-sequence as in Defn. 3.1

$$EX = \{EX_1 \times [a_1, a_2], \ldots, EX_{k-1} \times [a_{k-1}, a_k], EX_k \times [a_k, \infty)\}$$

together with a family $\mu^X = \{\mu^X_i\}_{i=1}^{k}$ of bijections $\mu^X_i : EX_i \to E_i$, for all $i = 1, \ldots, k$, called an

$E$-edge labelling or simply an edge-labelling.

- the morphisms are function preserving maps $\varphi : (X, f) \to (Y, g)$ that preserve the edge-labelings, i.e. $\mu^Y(\varphi(e)) = \mu^X(e)$ for all $e \in EX$. 


It is straightforward to check that this is indeed a category on its own.

**Lemma 3.12.** The category \( \text{SimpReeb}[\mathcal{E}] \) is thin.

**Proof.** Let \( \varphi, \psi : (X, f) \to (Y, g) \) be two morphisms in \( \text{SimpReeb}[\mathcal{E}] \). By Prop. 3.3 we have to show that the maps agree at the edge-points and the nodes. Assume on the contrary that the edges \( \varphi(e) \neq \psi(e) \) for some \( e \in \mathcal{E}_n^X \) and \( i = 1, \ldots, k \). Then \( \mu^X(\varphi(e)) \neq \mu^X(\psi(e)) \) because \( \mu^X \) is bijective. Hence \( \mu^X(e) \neq \mu^X(e) \) a contradiction. Let \( v \) be a node in \( X \), say \( v \in \mathcal{V}_i^X \) for some \( i \). We claim that \( \varphi(v) = \psi(v) \). Since \( X, Y \) are rooted Reeb graphs \( v \) is either the down image of some edge \( e \in \mathcal{E}_n^X \) or the upper image of some edge \( e \in \mathcal{E}_n^{X-1} \). So, we have two cases:

- **Case 1:** \( v = D_i^X(e) \). By Prop. 3.3 we have that

\[
\varphi(v) = \varphi_i^Y(D_i^X(e)) = D_i^Y(\varphi_i^X(e)) = D_i^Y(\psi_i^X(e)) = \psi_i^Y(D_i^X(e)) = \psi(v).
\]

- **Case 2:** \( x = U_i^{X-1}(e) \). Proof works the same way. \( \Box \)

**Theorem 3.13.** Fix an edge-sequence \( \mathcal{E} \). Let \( n, s \geq 0 \). Inside \( \text{SimpReeb}[\mathcal{E}] \), every simple rooted Reeb graph with \( n \)-leaves and \( s \) cycles, decomposes into a coproduct of merge trees with \( (n + s) \)-leaves, i.e.

\[
(X, f, \mu) \cong \coprod_{T \in \mathcal{T}(X)} (T, f_T, \mu),
\]

for some set of merge trees \( \mathcal{T}(X) \).

**Proof.** Let \( X = (X, f, \mu_X) \) be a rooted Reeb graph with \( n \)-leaves and \( s \)-cycles and \( \mathcal{E} \)-edge labelling. Let \( L(X) = \{l_1, \ldots, l_t\} \), \( R(X) = \{r_1, \ldots, r_m\} \) be its sets of leaves, and reticulation nodes respectively. Let \( d_i \) be the indegree the reticulation node \( r_i \) for all \( i = 1, \ldots, m \). Also for any reticulation node \( r_i \), \( i = 1, \ldots, m \), let us denote by \( D^{-1}(r_i) = \{e^{(i)}_1, \ldots, e^{(i)}_{d_i}\} \) the set of all adjacent edges of \( r_i \) that are above it. We construct a collection of merge trees with \( (n + s) \)-leaves

\[
\mathcal{T}(X) = \{(T(w_1, \ldots, w_m), f(w_1, \ldots, w_m)) : 1 \leq w_i \leq d_i, \text{ for all } i = 1, \ldots, m\}
\]

associated to \( X \) by breaking up the reticulation nodes into pieces making a tree, by introducing new leaf-nodes in all possible ways. For simplicity of the proof, we denote an \( m \)-tuple \((w_1, \ldots, w_m)\) by \( w \). Let \( w \in \prod_{i=1}^{m}(1, \ldots, d_i) \). We construct a merge tree \( T = (T_w, f_w, \mu_w) \) with \( (n + s) \)-leaves and \( \mathcal{E} \)-edge labelling by changing the structure of \( X \) to make a tree, by taking the following steps:

1. Consider the vertex and edge sets \( V_j \) and \( E_j \) and the upper and down functions \( U_j^X \) and \( D_j^X \), \( j = 1, \ldots, k \) as in Defn. 3.1.

2. For any \( j = 1, \ldots, k \):

- Keep the edge sets \( E_j \) as they are; namely we keep the edge-sequence the same, i.e. \( \mathcal{E}^{T_w} = \mathcal{E}^X \).
- If \( V_j \) contains no reticulation nodes then set \( V_j^T = V_j \) and go to step \( j + 1 \),
- Else do the following:
  - replace the function \( D_j : E_j \to V_j \) with the bijection \( D_j^T : E_j \to V_j^T \), given by

\[
D_j^T(e) = \begin{cases} 
D_j(e), & \text{if } D_j(e) \text{ is not a reticulation node} \\
e, & \text{if } D_j(e) = r_i, \text{ for some } i = 1, \ldots, m
\end{cases}
\]

where \( V_j^T \) is the image of the bijection \( D_j^T \).
- replace the function \( U_j : E_j \to V_{j+1} \) with the function \( U_j^T : E_j \to V_{j+1}^T, e \mapsto U_j(e) \).
\[ U_j^w(e) = \begin{cases} U_j(e), & \text{if } U_j(e) \text{ is not a reticulation node} \\ \epsilon_{w, i}^{(i)}, & \text{if } U_j(e) = r_i, \text{ for some } i = 1, \ldots, m \end{cases} \]

3. \( \mathbb{T}_w \) is the quotient of the disjoint union

\[
\prod_{j=1}^{k} \left( \mathcal{V}_j \times \{ a_j \} \right) \prod_{j=1}^{k-1} \left( \mathcal{E}_j \times [a_j, a_{j+1}] \right) \prod_{j=1}^{k} \left( \mathcal{E}_j \times [a_k, +\infty) \right)
\]

with respect to the identifications \((U_j^w(e), a_{j+1}) \sim (e, a_{j+1}) \) and \((D_j^w(e), a_j) \sim (e, a_j)\).

4. \( f_w \) is the projection of \( T_w \) to the second coordinate.

5. \( \mu_w = \mu_X = \mu \), since to obtain the tree \( T_w \) we have modified the structure on \( X \) only on the reticulation nodes and not on the edges.

By 1-5 we obtain the structure of each of the merge trees \((T_w, f_w, \mu_w)\) inside \( \text{SimpReeb}[\mathcal{E}] \).

Each merge tree \((T_w, f_w, \mu_w)\) is equipped with the obvious quotient map

\[ q_w : (T_w, f_w, \mu_w) \rightarrow (X, f, \mu_X) \]

\[ [(x, t)] \mapsto [(Q_w(x), t)] \]

where

- \( Q_w(x) = r_i \), if \( x = e_1^{(i)}, \ldots, e_{w-1}^{(i)}, e_w^{(i)}, \ldots, e_{q}^{(i)} \), for some \( i = 1, \ldots, m \), or
- \( Q_w(x) = x \), otherwise.

By definition the quotient map \( q_w \) is identity when restricted to edges and since \( \mu_w = \mu_X \) it follows that \( q_w \) is \( \mathcal{E} \)-edge labelling preserving, i.e. it forms a morphism in \( \text{SimpReeb}[\mathcal{E}] \).

Now we claim that the coproduct of all the merge trees \((T_w, f_w, \mu)\) in \( \text{SimpReeb}[\mathcal{E}] \) is isomorphic to the simple rooted Reeb graph \((X, f, \mu)\) with the coprojections morphisms being the quotient maps \( q_w \).

Let

\[ \varphi_w : (T_w, f_w, \mu) \rightarrow (Y, g, \mu_Y) \]

\[ [(x, t)] \mapsto [(\Phi_w(x), t)] \]

be a family of morphisms in \( \text{SimpReeb}[\mathcal{E}] \) (function preserving maps such that \( \mu_Y(\varphi_w(e)) = \mu(e) \) for all \( e \in \mathcal{E} \), where \( \Phi_w = p_1 \circ \varphi_w \)). For convenience we denote by \( \varphi_{w, i}^{Y} \) and \( \varphi_{w, i}^{E} \) the associated functions satisfying the consistency conditions as described in Prop. 3.3. We claim that there exists a morphism \( \varphi : (X, f, \mu) \rightarrow (Y, g, \mu_Y) \) in \( \text{SimpReeb}[\mathcal{E}] \) such that each of the diagrams

\[
\begin{array}{ccc}
(T_w, f_w, \mu) & \xrightarrow{\eta_w} & (X, f, \mu) \\
\downarrow \varphi_w & & \downarrow \varphi \\
(Y, g, \mu_Y) & & (Y, g, \mu_Y)
\end{array}
\]

commutes. Since the category \( \text{SimpReeb}[\mathcal{E}] \) is thin, this is equivalent to showing that for every \( w \) there exists a morphism \( \Tilde{\varphi}_w : (X, f, \mu) \rightarrow (Y, g, \mu_Y) \) such that the diagram

\[
\begin{array}{ccc}
(T_w, f_w, \mu) & \xrightarrow{\eta_w} & (X, f, \mu) \\
\downarrow \varphi_w & & \downarrow \varphi \\
(Y, g, \mu_Y) & & (Y, g, \mu_Y)
\end{array}
\]

commutes. Indeed if we show this, then since \( \text{SimpReeb}[\mathcal{E}] \) is thin we have

\[ \Tilde{\varphi}_w = \Tilde{\varphi}_w' \]
for all \( w \neq w' \). That is, we will have a single morphism that is independent of the index \( w \), which we can denote by \( \varphi \), which is exactly what we want.

Pick any \( w \). We claim that there exists a \( \hat{\varphi}_w \) such that the diagram

![Diagram](image)

commutes. To show the existence of such a map \( \hat{\varphi}_w \) by the universal property of quotient spaces it suffices to show that \( \varphi_w \) respects the identifications in \( T_w \) under the quotient map \( q_w \). Assume that \( (e^{(i)}_s, a_j) \sim (e^{(i)}_t, a_j) \) for some reticulation node \( r_i \) and some \( 1 \leq s \neq t \leq d_i \), with \( s, t \neq w_i \), where \( a_j = f(r_i) \) for some \( j = 1, \ldots, k \). Then by definition \( D_j(e^{(i)}_s) = e^{(i)}_s \) and \( D_j(e^{(i)}_t) = e^{(i)}_t \). We compute

\[
\varphi_w([(e^{(i)}_s, a_j)]) = [(\Phi_w(e^{(i)}_s), a_j)] \\
= [([\varphi_w^Y(e^{(i)}_s), a_j])] \\
= [([\varphi_w^Y(D_j^w(e^{(i)}_s)), a_j])] \\
= [([D_j^w(e^{(i)}_s), a_j])] \\
= [([e^{(i)}_s, a_j])] \\
= [([e^{(i)}_t, a_j])] \\
= [([D_j^w(e^{(i)}_t), a_j])] \\
= [([\varphi_w^Y(D_j^w(e^{(i)}_t)), a_j])] \\
= [([\varphi_w^Y(e^{(i)}_t), a_j])] \\
= [([\Phi_w(e^{(i)}_t), a_j])] \\
= \varphi_w([(e^{(i)}_t, a_j)]).
\]

Hence the map \( \hat{\varphi}_w \) exists. Again by the universal property of quotient maps, the map \( \hat{\varphi}_w \) is also function preserving. Finally, we show that the function preserving map \( \varphi_w \) is \( E \)-edge labelling preserving. Indeed. Let \( e \) be an edge in \( E^X \). We compute:

\[
\mu_v(\hat{\varphi}_w(e)) = \mu_v(\varphi_w(q_w(e))) \\
= \mu_v(\varphi_w(e)) \\
= \mu_w(e) \\
= \mu(e).
\]

**Remark 3.14.** By definition, each of the leaves of \( T_w \) which are additional to the \( n \)-leaves of \( X \) that we obtain from the Algorithm at Step. 2 is identified with an edge that is a down-image but not an upper-image, i.e. the ‘new’ leaves are exactly those edges \( e \) such that for some \( 1 \leq j \leq k \), \( D_j^w(e) = e \) but \( e \notin U_j^w(e') \) for any \( e' \in E_{j-1} \). That is, the set of these leaves is equal to the set of all edges that are incident from above of the reticulation nodes but are not equal to \( e^{(i)}_w \) for all \( i = 1, \ldots, r \). Therefore the cardinality of the set of the leaves that are additional to the \( n \)-leaves of \( X \) is exactly

\[
\sum_{i=1}^{m} (d_i - 1) = s,
\]

which is the number of the cycles in \( X \). Hence each merge tree \( T_w \) has \((n + s)\)-leaves.

\[\square\]

## 4 Phylogenetic networks

Using the tree-decomposition of simple rooted Reeb graphs in Thm. 3.13 we classify phylogenetic networks up to isomorphism. Then we examine this classification from the viewpoint of complexity theory.

Let us define phylogenetic networks.
**Definition 4.1.** A phylogenetic network with \( n \)-labelled leaves is a simple rooted Reeb graph \((X, f)\) together with a bijection \( \ell : L(X) \to \{1, \ldots, n\} \), called a leaf-labeling.

**Definition 4.2.** An isomorphism of phylogenetic networks with \( n \)-labelled leaves \((X, f, \ell)\), \((X', f', \ell')\) is an isomorphism \( \varphi : (X, f) \to (X', f') \) of their underlying Reeb graphs that preserves their leaf-labellings, i.e. \( \ell'(\varphi(v)) = \ell(v) \), for all \( v \in L(X) \).

### 4.1 Classification of phylogenetic networks

Let us denote by \( \text{Pow}(S) \) the power set of any set \( S \). Let \( n, s \geq 0 \) be positive integers. Consider the category \( \text{PhNet}(n, s) \) of isomorphisms of phylogenetic networks with \( n \)-labelled leaves and \( s \) cycles where only isomorphisms taken as the morphisms.

Let \( k \geq 0 \). We denote by \( \text{PhTree}(k) \) the collection of all isomorphism classes of phylogenetic trees with \( k \)-labelled leaves. That is, \( \text{PhTree}(k) = \text{PhNet}(k, 0) \). Our goal is to prove the following theorem.

**Theorem 4.3.** There exists an injection between isomorphism classes of phylogenetic networks with \( n \)-labelled leaves and \( s \) cycles, and sets of isomorphism classes of phylogenetic trees with \((n + s)\)-labelled leaves, i.e.

\[
\text{PhNet}(n, s) \hookrightarrow \text{Pow}(\text{PhTree}(n + s)) \\
N \mapsto T(N).
\]

**Proof.** Let \((X, f, \ell)\) be a phylogenetic network with \( n \)-labelled leaves and \( s \) cycles. Let \( E \) be its associated edge-sequence. Consider the identity labelling \( \mu : E \to E \). Then Thm. 3.13 applies so that

\[
(X, f, id) = \bigsqcup_w (T_w, f_w, id).
\]

By construction, the set of leaves of each merge tree \( T_w \) contains the set of leaves of \( X \) and also another \( s \) additional leaves. By Rmk. 3.14 the \( s \) additional leaves of \( T_w \) are identified with all the edges that are incident from above to the reticulations nodes of \( X \) except the edges \( e_w^{(i)} \), \( i = 1, \ldots, m \). We equip \( T \) with
a leaf-labeling $\ell_w$ on the leaves of $T_w$ given by:

\[
\begin{align*}
  l_1 & \mapsto 1 \\
  l_2 & \mapsto 2 \\
  \ldots \\
  l_n & \mapsto n \\
  e_1^{(1)} & \mapsto n + 1 \\
  \ldots \\
  e_{w_1-1}^{(1)} & \mapsto n + w_1 - 1 \\
  e_{w_1+1}^{(1)} & \mapsto n + w_1 \\
  e_{w_1+2}^{(1)} & \mapsto n + w_1 + 1 \\
  \ldots \\
  e_{d_1}^{(1)} & \mapsto n + (d_1 - 1) \\
  e_1^{(2)} & \mapsto n + (d_1 - 1) + 1 \\
  \ldots \\
  e_{w_2-1}^{(2)} & \mapsto n + (d_1 - 1) + w_2 - 1 \\
  e_{w_2+1}^{(2)} & \mapsto n + (d_1 - 1) + w_2 \\
  e_{w_2+2}^{(2)} & \mapsto n + (d_1 - 1) + w_2 + 1 \\
  \ldots \\
  e_{d_2}^{(2)} & \mapsto n + (d_1 - 1) + (d_2 - 1) \\
  \ldots \\
  e_1^{(m)} & \mapsto n + (d_1 - 1) + \ldots + (d_{m-1} - 1) + 1 \\
  \ldots \\
  e_{w_m-1}^{(m)} & \mapsto n + (d_1 - 1) + \ldots + (d_{m-1} - 1) + w_m - 1 \\
  e_{w_{m+1}}^{(m)} & \mapsto n + (d_1 - 1) + \ldots + (d_{m-1} - 1) + w_m \\
  e_{w_m+2}^{(m)} & \mapsto n + (d_1 - 1) + \ldots + (d_{m-1} - 1) + w_m + 1 \\
  \ldots \\
  e_{d_m}^{(m)} & \mapsto n + (d_1 - 1) + (d_2 - 1) + \ldots + (d_{m-1} - 1) + (d_m - 1) = n + s.
\end{align*}
\]

That way we obtain a phylogenetic tree $(T_w, f_w, \ell_w)$ with $(n + s)$-labelled leaves. So naturally, we define the map $\mathcal{T}$ as follows:

\[
\mathcal{T} : \text{PhNet}(n, s) \rightarrow \text{P(PhTree}(n + s)) \\
[(X, f, \ell)] \mapsto \{([T_w, f_w, \ell_w]) : 1 \leq w_i \leq d_i, \text{ for } i = 1, \ldots, m\}.
\]


First we claim the map $\mathcal{T}$ is well defined:

$\left[([X, f, \ell]) = ([X', f', \ell'])\right]$

$\Rightarrow ([X, f, \ell]) \xrightarrow{s_n} ([X', f', \ell'])$

$\Rightarrow ([X, f]) \xrightarrow{s_n} ([X', f', \ell'])$ such that $\ell'(\varphi(v)) = \ell(v)$, for all $v \in L(X)$.

$\Rightarrow ([X, f, \mu]) \xrightarrow{s_n} ([X', f', \mu'])$, where $\mu = \varphi_1 : E_X \rightarrow E_{X'}$ and $\mu' : E_{X'} \xrightarrow{id} E_{X'}$

$\Rightarrow ([X, f, \mu]) \xrightarrow{s_n} ([X', f', \mu'])$, an isomorphism in $\text{SimpReeb}[E_{X'}]$

$\Rightarrow \prod_w \left(\mathcal{T}_w, f_w, f'_w, \mu'_w\right) \xrightarrow{s_n} \prod_w \left(\mathcal{T}'_{\mu(w)}, f'_w, \mu'_w, \ell'_w\right)$ for all $j = 1, \ldots, d_i$, for all $i = 1, \ldots, m$ (*)

$\Rightarrow \mathcal{T}([([X, f, \ell])]) = \mathcal{T}([([X', f', \ell'])])$

Now we show that $\mathcal{T}$ is also injective:

$\mathcal{T}([([X, f, \ell])]) = \mathcal{T}([([X', f', \ell'])])\Rightarrow ([\mathcal{T}_w, f_w, \mu_w]) = ([\mathcal{T}'_{\mu(w)}, f'_w, \mu'_w, \ell'_w])$, for some permutation $\sigma$ and $\mathcal{T}_w, \mathcal{T}'_{\mu(w)}$ where $1 \leq w_i \leq d_i$

$\Rightarrow ([\mathcal{T}_w, f_w, \mu_w]) = ([\mathcal{T}'_{\mu(w)}, f'_w, \mu'_w, \ell'_w])$, for some isomorphisms $\varphi_w$, where $1 \leq w_i \leq d_i$

$\Rightarrow ([\mathcal{T}'_{\mu(w)}, f'_w, \mu'_w]) \xrightarrow{s_n} ([\mathcal{T}_w, f_w, \mu_w])$, such that $\ell_w(\varphi_w(v)) = \ell'_w(v)$, for all $v \in L(\mathcal{T}_w) \cap L(X')$ (*)

$\Rightarrow ([\mathcal{T}_w, f_w, \mu_w]) \xrightarrow{s_n} ([\mathcal{T}'_{\mu(w)}, f'_w, \mu'_w, \ell'_w])$, where $\mu_w : E_{\mathcal{T}_w} \xrightarrow{id} E_{\mathcal{T}'_{\mu(w)}}$ and $\mu'_w : E_{\mathcal{T}'_{\mu(w)}} \xrightarrow{id} E_{\mathcal{T'}_w}$

$\Rightarrow ([\mathcal{T}_w, f_w, \mu_w]) = ([\mathcal{T}'_{\mu(w)}, f'_w, \mu'_w, \ell'_w])$ in $\text{SimpReeb}[E_{\mathcal{T}_w}]$, where $E_{\mathcal{T}_w} = E_X, \mu_w = \mu_X, \mu'_w = \mu'_X$

$\Rightarrow ([\mathcal{T}_w, f_w, \mu_w]) \xrightarrow{s_n} ([\mathcal{T}'_{\mu(w)}, f'_w, \mu'_w])$, where $\varphi$ is the unique morphism: $\varphi \circ q_w = \varphi_w$, for all $w$

$\Rightarrow ([X, f', \mu']) \xrightarrow{s_n} ([X, f, \mu])$, where $\varphi$ is such that $\varphi \circ q_w = \varphi_w$

$\Rightarrow ([X, f', \mu']) \xrightarrow{s_n} ([X, f, \mu])$ and by (*) $\forall v \in L(X') : \ell(\varphi(v)) = \ell_w(\varphi(q_w(v))) = \ell_w(\varphi_w(v)) = \ell'_w(v) = \ell'(v)$

$\Rightarrow ([X, f, \ell]) \cong ([X', f', \ell'])$

$\Rightarrow ([X, f, \ell]) = ([X', f', \ell'])$.

Finally observe that because the rooted Reeb graphs we work on are always simple that means that the isomorphism classes of phylogenetic trees in $\mathcal{T}(X)$ are distinct, i.e. $\mathcal{T}(X)$ forms a set rather than a multiset of phylogenetic trees with $(n + s)$-labelled leaves. Hence, $\mathcal{T}(X)$ lies indeed in the power set of $\text{PhTree}[n + s]$.

**Remark 4.4.** We believe Thm. 4.3 is of particular importance to mathematical biology since it is the first structural theorem for phylogenetic networks. It is also important for defining and studying metrics on phylogenetic networks. As studied in 25, 4 the cophenetic map injects each isomorphism class of phylogenetic trees as a point in $\mathbb{R}^{n(n+1)/2}$. That way we can define $\ell'$-metrics on the isomorphism classes of phylogenetic trees. Combining this fact with the classification theorem, every phylogenetic network with $n$-labelled leaves and $s$ cycles injects as a finite set of points in $\mathbb{R}^{(n+s)(n+s+1)/2}$, where we can pull back the Hausdorff metric to define a metric for comparison on these networks.

**Example 4.5.** Consider a phylogenetic network $X = (X, f, \ell)$ with two labeled leaves and one reticulation node as shown in Fig. 3. We denote the edges incident from above of the reticulation node by $e_1$ and $e_2$ as in Fig. 3. By applying Thm. 4.3 we get two phylogenetic trees with 3-leaves in the tree-decomposition.
Figure 3: An example showing how, up to isomorphism, phylogenetic networks with \( n \)-labelled leaves and \( s \) cycles are in one to one correspondence to certain finite sets of phylogenetic trees with \((n + s)\)-leaves. In this case \( n = 2 \) and \( s = 1 \).

of \( X \). For each of the trees, the one additional leaf to the two leaves of \( X \) corresponds to one of the two edges \( e_1 \) and \( e_2 \). Thus there are exactly two such phylogenetic trees with 3-leaves.

### 4.2 Isomorphism complexity

We speculate that Thm. 4.3 can help for improving our understanding of the complexity of phylogenetic networks or study the tractability of the isomorphism under some fixed parameter. We have the following results.

**Upper bound on the complexity** By the classification theorem, two phylogenetic networks with \( n \)-leaves and \( s \) cycles are isomorphic if and only if their tree-decompositions are equal. Let \( C(n, s) \) be the number of steps it takes to decide if two phylogenetic networks with \( n \)-labelled leaves and \( s \) cycles are isomorphic. To check for isomorphism it takes at worst, as many number of steps as the cardinality of the set \( T(X) \) times the complexity of checking if two phylogenetic trees are isomorphic which takes at most \( n^2 \) steps. By construction, the cardinality of \( T(X) \) of a phylogenetic network \( X \) is equal to the product of indegrees of the reticulation nodes \( r_i \) constructed from Algorithm 2, for \( i = 1, \ldots, m \), i.e.

\[
|T(X)| = \prod_{i=1}^{m} d_i
\]

Therefore deciding if two phylogenetic networks with \( n \)-labelled leaves and \( s \) cycles and reticulation nodes with indegrees \( d_i, i = 1, \ldots, m \), and \( d'_j, j = 1, \ldots, m' \) respectively, are isomorphic takes time:

\[
\text{time} \leq n^2 \min\{\prod_{i=1}^{m} d_i, \prod_{j=1}^{m'} d'_j\}.
\]

**Fixed Betti tractability** However we want a closed form to bound the complexity that does not depend on the indegrees but purely only depends on \( n \) and \( s \) alone. The number of cycles \( s \) in a graph is known as the first Betti number or simply the **Betti number of the graph**. By the above upper bound we consider, it is easy to see that phylogenetic networks isomorphism is fixed Betti tractable: Since each of the indegrees of the reticulation nodes is \( d_i > 1 \) we have the bound \( d_i \leq 2^{d_i-1} \). Taking the product
over all these equations and since
\[ \sum_{i=1}^{m} (d_i - 1) = s, \]
is the number of cycles of \( G \), we have
\[ C(n, s) \leq n^22^s. \]
That means that by bounding the Betti numbers of two networks then deciding if they are isomorphic it is polynomial in \( n \).

5 Concluding remarks

We proposed an alternative approach to the extended Newick format that represents the isomorphism classes of phylogenetic networks with \( n \)-labelled leaves and \( s \) cycles by certain sets of phylogenetic trees with \((n+s)\)-leaves via an injection map. Furthermore, we have showed that with this injection we can pull back the Hausdorff metric to define a metric for comparison of these networks. We speculate that these two results would be useful for developing or enhancing the existing methods on phylogenetics as well as providing new insights on doing statistics or studying probability distributions on phylogenetic networks.

Future work In the near future we would like to expand the work in this paper with a follow up paper which will focus on three metrics on phylogenetic networks and study how they are related. The first two that would be studied are the Hausdorff distance arising from the finite set of points as mentioned in Remk. [14] and the network distance in the setting of general networks [11]. For the third one, we speculate that phylogenetic networks—just like Reeb graphs—are equipped with an interleaving distance [15], an extended pseudo-metric that arose from the field of topological data analysis (TDA) [8, 2].

Although the isomorphism problem of phylogenetic networks is GI-complete there are topological signatures from the field of TDA that are both stable to noise and efficient to compute. They have already been applied on phylogenetics [23]. Such signatures are: zigzag persistence homology [8] and path homology [10, 12] which we are going to exploit in the follow up work as well.

Acknowledgements

A. Stefanou was partially supported both by the National Science Foundation through grant NSF-CCF-1740761 TRIPODS TGDA@OSU and by the Mathematical Biosciences Institute at the Ohio State University. The author gratefully acknowledge F. Memoli, E. Munch, S. Kurtek, W. KhudaBukhsh and A. Foroughipour for many helpful discussions during this work.

References

[1] Louis J Billera, Susan P Holmes, and Karen Vogtmann. Geometry of the space of phylogenetic trees. Advances in Applied Mathematics, 27(4):733–767, 2001.
[2] Peter Bubenik, Vin De Silva, and Jonathan Scott. Metrics for generalized persistence modules. Foundations of Computational Mathematics, 15(6):1501–1531, 2015.
[3] Gabriel Cardona, Mercè Llabrés, Francesc Rosselló, and Gabriel Valiente. The comparison of treesibling time consistent phylogenetic networks is graph isomorphism-complete. The Scientific World Journal, 2014, 2014.
[4] Gabriel Cardona, Arnau Mir, Francesc Rosselló, Lucia Rotger, and David Sánchez. Cophenetic metrics for phylogenetic trees, after sokal and rohlf. BMC bioinformatics, 14(1):3, 2013.
[5] Gabriel Cardona, Francesc Rosselló, and Gabriel Valiente. Extended newick: it is time for a standard representation of phylogenetic networks. BMC bioinformatics, 9(1):532, 2008.
[6] Gunnar Carlsson and Vin De Silva. Zigzag persistence. Foundations of computational mathematics, 10(4):367–405, 2010.
[7] Gunnar Carlsson and Facundo Mémoli. Multiparameter hierarchical clustering methods. In Classification as a Tool for Research, pages 63–70. Springer, 2010.
Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. Springer, 2016.

Samir Chowdhury and Facundo Mémoli. Metric structures on networks and applications. In *Communication, Control, and Computing (Allerton)*, 2015 53rd Annual Allerton Conference on, pages 1470–1472. IEEE, 2015.

Samir Chowdhury and Facundo Mémoli. Persistent homology of directed networks. In *Signals, Systems and Computers, 2016 50th Asilomar Conference on*, pages 77–81. IEEE, 2016.

Samir Chowdhury and Facundo Mémoli. Distances and isomorphism between networks and the stability of network invariants. *arXiv preprint arXiv:1708.04727*, 2017.

Samir Chowdhury and Facundo Mémoli. Persistent path homology of directed networks. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1152–1169. SIAM, 2018.

Charles Darwin. *On the origin of species*, 1859. Routledge, 2004.

Vin De Silva, Elizabeth Munch, and Amit Patel. Categorified reeb graphs. *Discrete & Computational Geometry*, 55(4):854–906, 2016.

Vin de Silva, Elizabeth Munch, and Anastasios Stefanou. Theory of interleavings on categories with a flow. *arXiv preprint arXiv:1706.04095*, 2017.

Graham Dunn and Brian S Everitt. *An introduction to mathematical taxonomy*. Courier Corporation, 2004.

Daniel H Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: concepts, algorithms and applications*. Cambridge University Press, 2010.

Bo Lin, Anthea Monod, and Ruriko Yoshida. Tropical foundations for probability & statistics on phylogenetic tree space. *arXiv preprint arXiv:1805.12400*, 2018.

Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013.

Dmitriy Morozov and Gunther Weber. Distributed merge trees. In *Acm sigplan notices*, volume 48, pages 93–102. ACM, 2013.

Elizabeth Munch and Anastasios Stefanou. The $\ell^\infty$-cophenetic metric for phylogenetic trees as an interleaving distance. *arXiv preprint arXiv:1803.07609*, 2018.

Charles Semple, Mike A Steel, Richard A Caplan, Mike Steel, et al. *Phylogenetics*, volume 24. Oxford University Press on Demand, 2003.

Gurjeet Singh, Facundo Mémoli, and Gunnar E Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *SPBG*, pages 91–100, 2007.

Robert R Sokal. The principles and practice of numerical taxonomy. *Taxon*, pages 190–199, 1963.

Robert R Sokal and F James Rohlf. The comparison of dendrograms by objective methods. *Taxon*, pages 33–40, 1962.