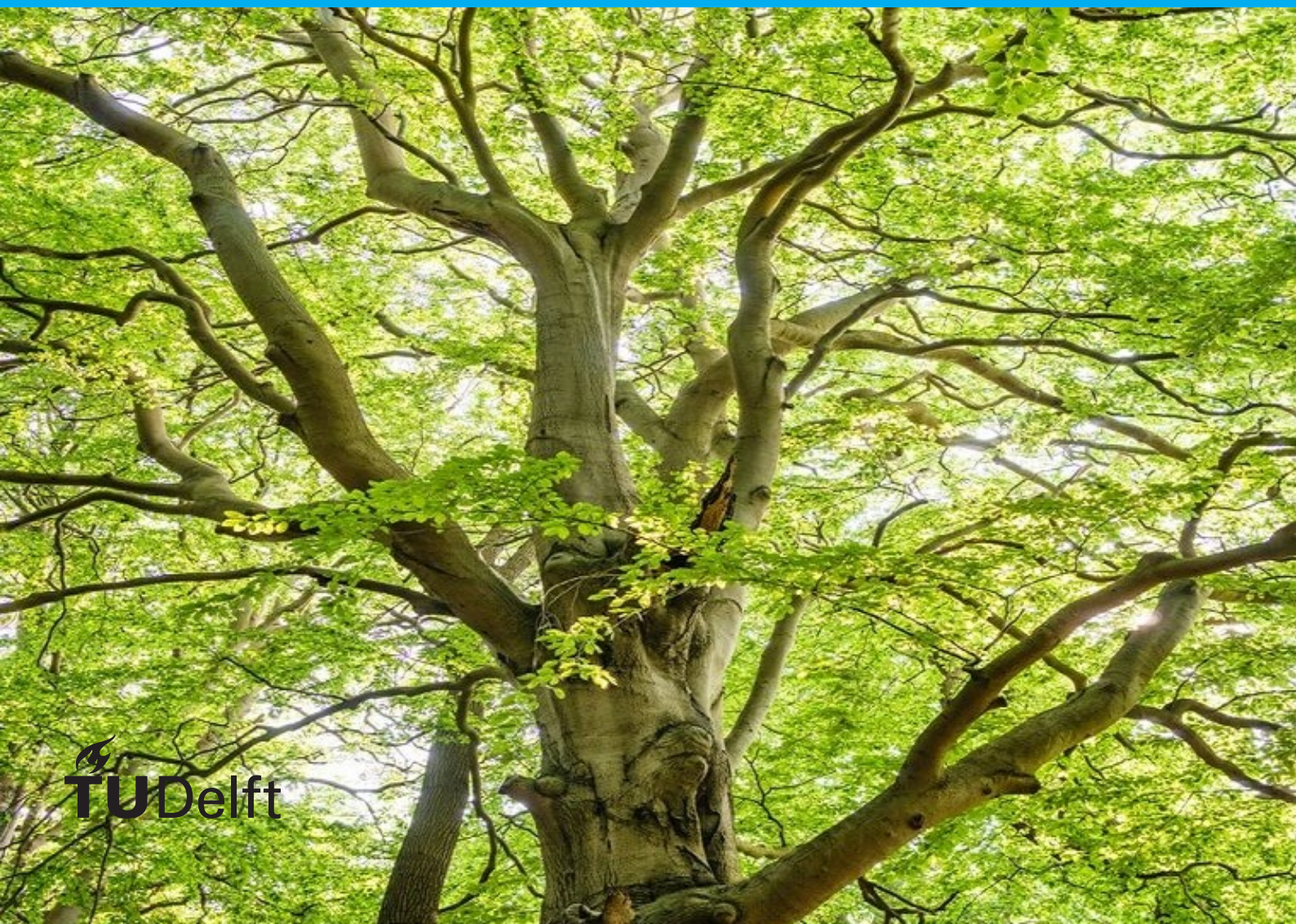
Identifiability of Phylogenetic Trinets

Guuske Anne Kouwenhoven

Bachelor Thesis
Applied Mathematics
Technische Universiteit Delft
June 2025



Identifiability of Phylogenetic Trinets

by

Guuske Anne Kouwenhoven

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Friday June 20, 2025 at 13:00.

Student number:	5877229	
Project duration:	April 22, 2025 – June 20, 2025	
Thesis committee:	Dr. ir. L. J. J. van Iersel,	TU Delft, supervisor
	Ir. N. A. L. Holtgrefe,	TU Delft, supervisor
	Dr. ir. G. F. Nane,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Lay summary

Phylogenetics is the study of how species are related through evolution. These relationships are traditionally represented using branching diagrams called phylogenetic trees. However, certain evolutionary processes, such as hybridization or horizontal gene transfer, cannot be represented by trees alone. Therefore, phylogenetic networks - which allow additional connections between branches - are used to capture more complex evolutionary histories

As phylogenetic networks become more complex, it becomes harder to determine whether a particular network can be uniquely reconstructed from observed DNA sequence data. This leads to the concept of identifiability. A network is said to be identifiable if, in theory, it can be uniquely determined from the data it generates. If two different networks produce the same data under a given evolutionary model, they are not identifiable from that data. Identifiability is essential for developing reconstruction methods that aim to infer evolutionary relationships from DNA.

This thesis investigates a specific class of phylogenetic networks, known as trinets. A trinet is a small subnetwork that describes the evolutionary relationship between just three species. Trinets are useful building blocks for understanding and reconstructing larger phylogenetic networks. The main question studied in this work is whether these trinets are identifiable. We investigate whether specific types of trinets can be distinguished from simpler tree structures, using a mathematical approach based on phylogenetic invariants. These are special algebraic expressions that relate to the probabilities of observing certain DNA patterns. By evaluating such invariants, we may be able to detect whether the data came from a network or from a tree.

This thesis provides new insights into the identifiability of trinets. It shows that a specific type of trinet can be distinguished from a simple tree using an invariant. This is an important step toward understanding how more complex evolutionary networks can be identified from biological data from genes.

Contents

1

Introduction

During the past few decades, phylogenetic analysis has become an important tool used in evolutionary biology to describe the relationships among genes or species [10]. Until recently, the links in history among a set of taxa, the species under consideration, are mostly represented in a strictly branching diagram, known as a rooted phylogenetic tree. However, phylogenetic trees cannot represent reticulate events, which are events where two evolutionary lineages converge again to create a new species, such as hybridization, horizontal gene transfer and recombination [9]. Such events cannot be represented by trees since they lead to cycles in the underlying undirected graphs, which are not allowed in trees. Capturing these special reticulate events is crucial to understand the evolutionary history of many classes of organisms [19], such as bacteria, viruses, plants, fish, birds and primates.

Therefore, phylogenetic networks are used to describe a more complicated evolutionary history than portrayed by a simple tree diagram [4]. Researchers can estimate these relationships by analyzing the DNA sequences of different taxa. By aligning and comparing the DNA sequences, one can attempt to reconstruct the evolutionary past of taxa [5]. As network structures become more complex, it becomes harder to find out which network could have produced the observed data [6].

An example of an evolution network, can be found in Figure 1.1. Here, the lines represent evolutionary relationships among various Xiphophorus species, a group of fishes that includes swordtails and platyfishes.

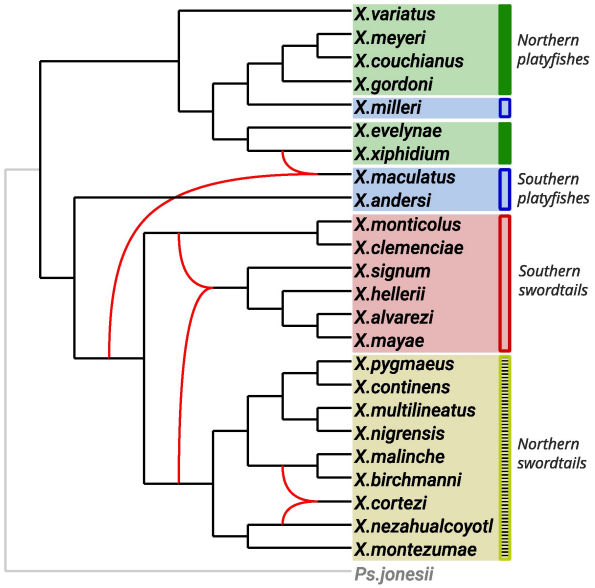


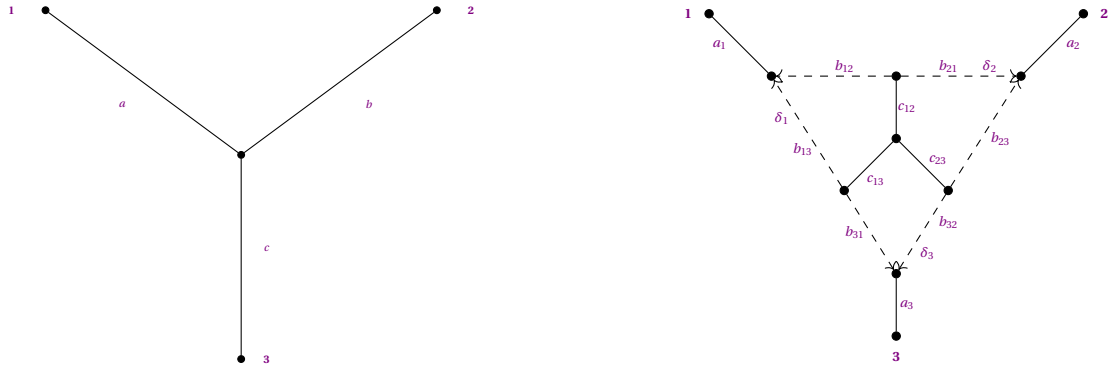
Figure 1.1: Phylogenetic network on the genus Xiphophorus. The four major lineages are indicated by the different shaded areas. The reticulation edges are curved, while the edges leading to the outgroup Pseudoxiphophorus jonesii are in grey. Holtgrefe et al. [11]

Recent studies focus on understanding which classes of networks are identifiable from the type of data

available. Roughly speaking, a network class is said to be identifiable if, in theory, sufficiently large data sets generated under a given model can distinguish between two different networks in this class. This is an important characteristic for the development of statistically consistent reconstruction algorithms.

In recent years, an important line of research has focused on reconstructing phylogenetic networks from smaller substructures. Since networks with many taxa can be complex, subnetworks offer a way to break down the reconstruction problem into smaller parts. In practice, subnetworks are often used as building blocks for reconstructing larger networks. Several algorithms and software tools adopt this approach by first identifying subnetworks from the data and then assembling them into a full network that is consistent with the inferred local structures [7]. Examples include the methods implemented in `Lev1athan` [12], `TriLoNet` [17], `Squirrel` [11] and `NANUQ` [2].

In this thesis, we will focus on the identifiability of semi-directed subnetworks. We will consider semi-directed networks, which are partially directed networks in which the root is suppressed, since the root location is not identifiable under many models [18]. The network is a connected semi-directed acyclic graph, with directed and undirected edges. The directed edges represent the reticulate events. An example of a semi-directed network with reticulate events can be found in Figure 2.6. This thesis has a special focus on trinets. Trinets are subnetworks that focus on the relationship between three different species, represented by the leaves in a network. In the context of this thesis, we call a trinet *level-k* if it has at most k reticulate events.



(a) Tree with three leaves 1, 2, and 3

(b) Symmetric level-3 trinet with three leaves

Figure 1.2: A simple three-leaf tree and a symmetric level-3 trinet on the same set of leaves.

From a biological perception, we want to know whether we can distinguish tree-like from network-like evolution. In this thesis, we will show that, in theory, for some cases we can. We investigate the identifiability question using phylogenetic invariants, which are, roughly speaking, polynomials that characterize the probability distributions of observed patterns of DNA nucleotides. We can determine these patterns from DNA data and evaluate the invariant which may tell us something about the evolutionary relationships. For phylogenetic trees, invariants have been used successfully to infer evolutionary relationships [1, 20]. More recently, phylogenetic invariants have also been applied to network models to distinguish specific classes of networks, such as cycle networks [9]. In a paper by Englander et al. [6] it is shown that invariants can in theory be used to distinguish between level-1 or level-2 networks and trees under the Jukes-Cantor model, a simple Markov model of evolution.

This thesis will extend one of the results given in [6], which shows how to distinguish between a three-leaf tree and a level-1 or level-2 trinet under the JC model. In this thesis, we will look at a specific symmetric strict level-3 trinet to determine whether we can use the same invariant to distinguish that trinet from a three-leaf tree, both trinets are given in Figure 1.2. We will show that the symmetric level-3 trinet is distinguishable from the three-leaf tree, which is a first result considering the distinguishability of level-3 networks. It can be a first step for extending results to level-k networks. These type of trinet results can also be used to prove identifiability of n-leaf networks, by using various trinets that are subnetworks of the bigger network (see [6]). Moreover, we will look at the more general K2P model and whether we can extend the result in [6] using the K2P model instead of the JC model. Unfortunately, we show that the approach we used for the JC model does not directly give a similar result for the K2P model.

The outline of this report is as follows. In the next Chapter, we start with a background on graph theory, phylogenetic trees and networks, Markov models on trees and networks and the Fourier transform used by, e.g., Ardiyansyah [3]. Then, Chapter 3 starts with two Lemmas showing that two functions are strictly positive

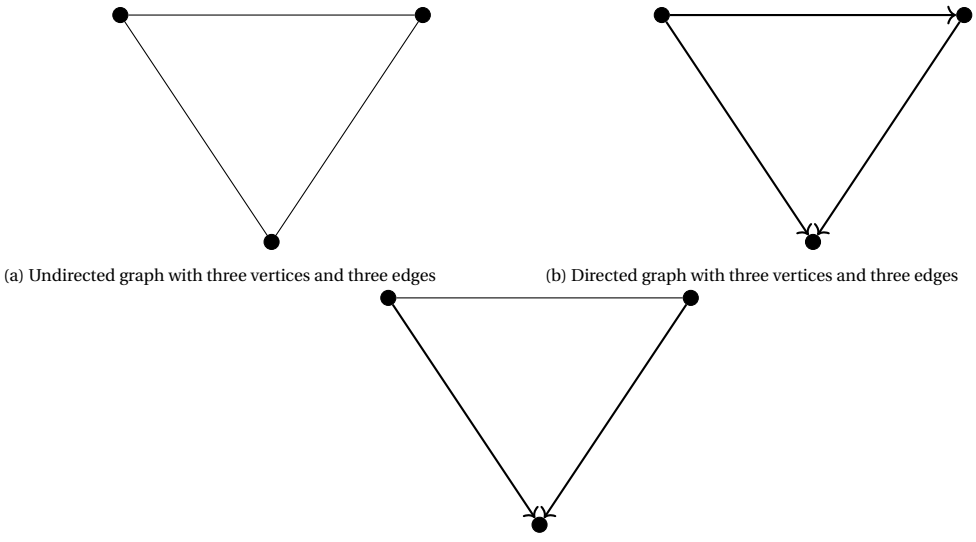
on the interval $(0,1)$. These auxiliary results will be used in Chapter 3 to show that the considered invariant is strictly positive for a symmetric level-3 trinet and therefore this level-3 trinet distinguishable from a three-leaf tree. In Chapter 4, we adapt the invariant given in [6] in an attempt to use it for the K2P model and show that this invariant cannot be used to distinguish between a three-leaf tree and a level-1 trinet. We end with a discussion.

2

Preliminaries and definitions

In this Chapter, we introduce some definitions required for the study of phylogenetic networks. These include definitions from graph theory and phylogenetic trees and networks, mostly following [3, 6, 8, 13].

We start with some basic terms used in graph theory. A *graph* is a mathematical way of representing relationships between objects. A graph is a pair $G = (V, E)$, where V is a set of elements called *vertices*, the objects, and E is a collection of unordered vertex pairs, $\{v_1, v_2\}$, where each pair represents an *edge*, which represents a relationship between two vertices. A *directed graph* is a graph in which all edges have directions. If the edges do not have any directions, we call it an *undirected graph*. Lastly, a *semi-directed graph* has both directed and undirected edges, see Figure 2.1. In an undirected graph, each edge $e \in E$ is a set $\{v, w\}$ of two vertices $v, w \in V$. The vertices v and w are called the *endpoints* of the edge, and we say that the edge *connects* v and w. Both vertices are said to be *incident* to the edge. Two edges are called *adjacent* if they share a common endpoint, and two vertices are adjacent if they are connected by an edge. The *degree* of a vertex v is defined as the number of edges that are incident to it, see Figure 2.2.



(a) Undirected graph with three vertices and three edges

(b) Directed graph with three vertices and three edges

(c) Semi-directed graph with two directed edges and one undirected edge

Figure 2.1: Examples of undirected, semi-directed and directed graphs

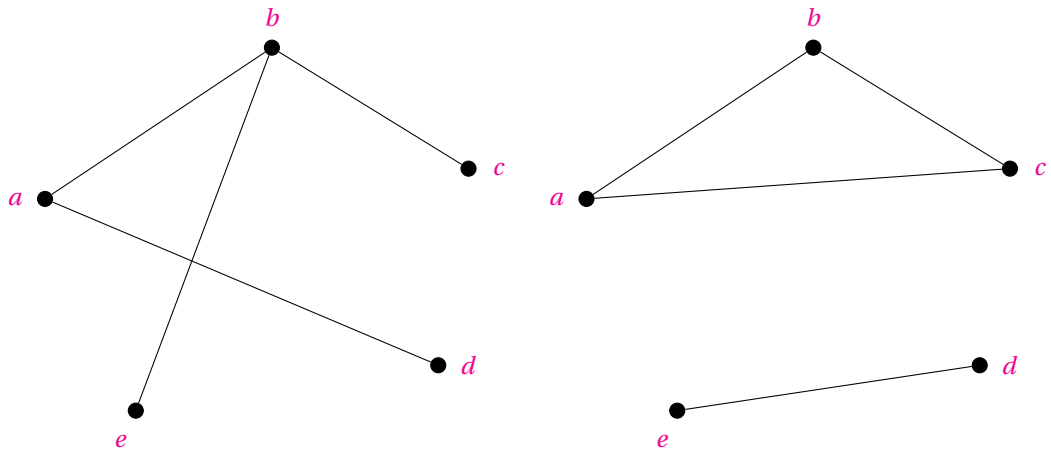
In an undirected graph, a pair of two vertices x and y is said to be *connected* if there exists a path between them. An undirected graph $G = (V, E)$ is called *connected* if every pair of distinct vertices $v, w \in V$ in the graph is connected. If there exists at least one pair of vertices that is disconnected, the graph is called *disconnected*. A *path* in a graph is a sequence of vertices and edges

$$P = (v_0, e_1, v_1, e_2, \ldots, e_k, v_k)$$

such that each edge e_i connects the vertices v_{i-1} and v_i, and no edge appears more than once in the path. If a path exists between two nodes v and w, we say the path *connects* v and w.

A *cycle* is a path in which the first and last node are the same, i.e., $v_0 = v_k$, and no other node occurs more than once. See figure 2.2 (b) for an example of a cycle. A directed graph is said to be *acyclic* if it does not contain any directed cycles.

An edge e in a network N is said to *separate* two disjoint subsets of vertices A and B if every path between any vertex $a \in A$ and $b \in B$ contains e. In this case, the edge e is called a *cut-edge*, and the network N is said to have an *A–B split*.



(a) Connected graph, where the *degree* of vertex b is three

(b) Disconnected graph with a cycle a, b, c

Figure 2.2: A connected graph and disconnected graph with cycle

Now that we have the basic graph theory, we can use these terms to introduce definitions that are used to study phylogenetic networks and trees.

We start with the definition of a tree and some related terms and then we will look at phylogenetic networks.

Definition 2.1 (Directed phylogenetic tree)**.** A *directed phylogenetic tree* is a rooted, directed acyclic graph that contains no underlying undirected cycles.

A directed phylogenetic tree has a *root*, which is the distinguished vertex of a graph with in-degree zero and out-degree two. The tree can be interpreted as a directed graph in which all edges are directed away from the root. This root represents the most recent common ancestor of all taxa (species) represented in the tree. The direction of the edges thus indicates the flow of evolutionary time.

The vertices of a tree with in-degree one and out-degree zero are called the *leaves* of the tree. In the context of phylogenetics, these leaves typically represent the extant species for which data is available in a phylogenetic analysis. Consequently, each leaf is often assigned a unique label from a set of labels corresponding to the species.

In theoretical settings, we often consider the set of leaf labels to be $[n] := \{1, 2, \ldots, n\}$, and refer to the resulting tree as an *n-leaf phylogenetic tree*. Two such trees are considered distinct if their leaf-labeling differ, even if the unlabeled graph structures are the same. More formally, two n-leaf phylogenetic trees are regarded the same if and only if there exists a graph isomorphism between them that also preserves the leaf labels according to Gross et al. [8].

We often focus on a particular class of trees called *binary trees*. A *binary tree* is a tree in which every vertex, except the root, has a degree of either one or three In such trees, internal vertices with degree three represent speciation events, indicating a point in time where a single species gave rise to two descendant species. The structure of the tree captures the timing and pattern of these divergence events.

Example 2.1 (Unrooting a tree)**.** We can unroot a tree by suppressing all degree two vertices. There is only one 3-leaf binary phylogenetic tree, whereas there are three different rooted 3-leaf binary phylogenetic trees. See Figure 2.3, where the leaves are labeled.

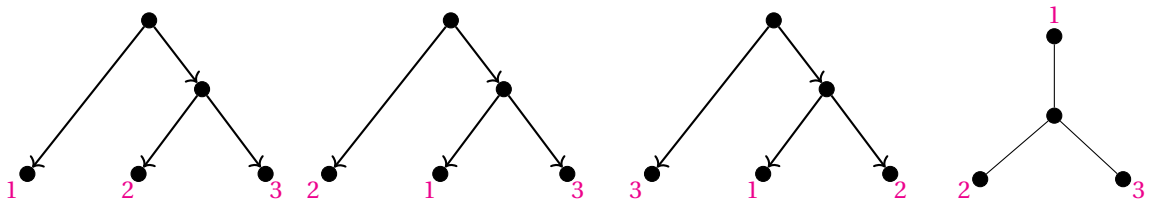


Figure 2.3: The three different rooted 3-leaved binary phylogenetic trees are shown on the left. The tree on the most right is the unrooted 3-leaved binary phylogenetic tree obtained by unrooting one of the rooted 3-leaved binary phylogenetic trees.

2.1. Phylogenetic networks

Before formally defining what a phylogenetic network is, we begin by introducing the notion of a *blob*. According to Englander et al. [6] a blob of a (partially) directed graph is a maximally connected subgraph that does not contain any *cut-edges*. That is, edges whose removal would disconnect the graph. A blob is referred to as an *m-blob* for some positive integer m, if it connects exactly m vertices outside the blob. If a blob consists of only a single vertex, it is said to be *trivial*.

Example 2.2. Let $G = (V, E)$ be an undirected graph with vertex set $V = \{a, b, c, d, e\}$ and edge set

$$E = \{\{a, b\}, \{b, c\}, \{c, a\}, \{c, d\}, \{d, e\}\}.$$

The subgraph induced by vertices $\{a, b, c\}$ forms a *blob*, as it is maximally connected and does not contain any cut-edges. Removing any edge within this subgraph does not disconnect it. The edge $\{c, d\}$ is a cut-edge, since its removal disconnects vertex d (and e) from the rest of the graph. Thus, the blob $\{a, b, c\}$ connects to the rest of the graph via a single edge and is therefore a *1-blob*.

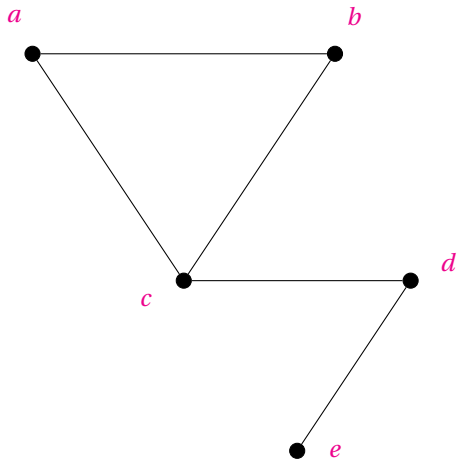


Figure 2.4: Example of a 1-blob $\{a, b, c\}$

Definition 2.2 (Directed rooted binary phylogenetic network)**.** A *rooted binary phylogenetic network* N on a set of leaves $[n]$ is a rooted, directed acyclic graph (DAG) without parallel edges, satisfying the following properties:

- The sum of in-degree and out-degree for any vertex is at most three.
- The network has a single root with in-degree zero and out-degree two.
- Each leaf is a vertex with out-degree zero and in-degree one, and the set of leaves is bijectively labeled by $[n]$.
- The network contains no 2-blobs and no 1-blobs, except for the leaves.

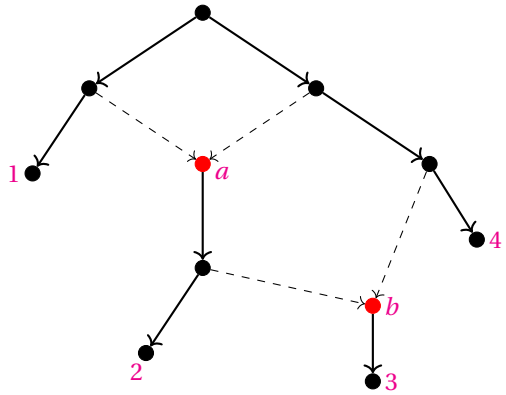


Figure 2.5: A rooted binary phylogenetic network with two reticulation vertices a, b in red and four leaves

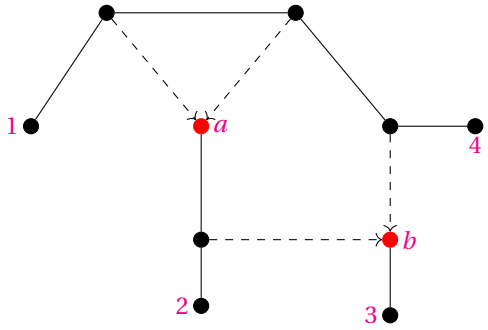


Figure 2.6: The semi-directed network obtained from Figure 2.5 with two reticulation vertices a, b in red and four leaves

We call a vertex in N with in-degree one and out-degree two a *tree vertex*, which represents a speciation event. The vertices with in-degree two and out-degree one are called *reticulation vertices* since they represent reticulation events. Additionally, an edge directed into a reticulation vertex is a *reticulation edge* and the other edges are called *tree edges*. All edges in N which are not directed into leaves are called *internal*. Note that the class of phylogenetic trees is a subset of a phylogenetic networks.

Definition 2.3. The *reticulation number* of a phylogenetic network N is the total number of reticulation vertices of the network.

Definition 2.4. A *semi-directed network* N on a set of leaves $[n]$ is a partially directed graph that can be obtained from a rooted phylogenetic network on $[n]$ by suppressing all vertices of degree two, identifying parallel edges and removing edge directions of non-reticulation edges. The only directed edges are the edges representing the reticulation events.

Definition 2.5. Let N be a semi-directed network and k a positive integer. The network is *level-k* if there exists a maximum of k reticulation vertices in each blob of the network. Furthermore, we say that N is a *strict level-k* phylogenetic network if it is a level-k but not level-$(k-1)$ network.

Example 2.3 (Unrooting a phylogenetic network)**.** In Figure 2.5, you can find a rooted phylogenetic network with two reticulation vertices a, b in red. A semi-directed network can be obtained from the rooted network, the result is shown in Figure 2.6. The only edges with a direction in Figure 2.6 are the reticulation edges. Furthermore, the root has been suppressed because all vertices of degree two have to be removed. This network is an example of a level-2 network.

When analyzing phylogenetic networks, a useful notion is a *subnetwork* induced by a subset of the leaves. Such a subnetwork depicts only the evolutionary relationships between a subset of the species. As mentioned in the introduction, subnetworks are a useful tool for constructing larger networks.

We can restrict semi-directed networks to a smaller leaf set by taking a subnetwork. Let N be an n-leaf semi-directed network and S a subset of the leaves. We can obtain a semi-directed network on S by taking

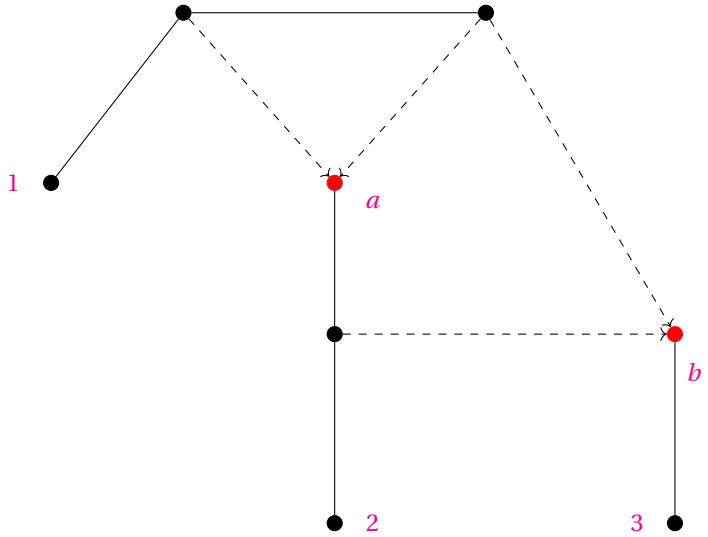


Figure 2.7: A level-2 trinet of the network in Figure 2.6 induces by a subset $S = \{1,2,3\}$

the union of all *up-down paths* connecting any two leaves in S. According to Englander et al. [6], an up-down path between two leaves x_i and x_j of a semi-directed network is a path of k edges where the first l edges are directed towards x_i and the last $k-l$ edges are directed towards x_j. Then, contract all degree two vertices to one of its neighbors and remove all parallel edges. We will look in this report at three-leaf subnetworks, which are called *trinets*.

Example 2.4 (Finding a trinet)**.** We can find a trinet from the network given in Figure 2.6. For example, take $S = \{1,2,3\}$ a subset of the leaves. To find the trinet, we only look at the edges making a path from one of the leaves in S. Therefore, vertex 4 and the connected edge will be removed. We then have a vertex with degree two, so we can suppress these two edges to one edge. The result can be found in Figure 2.7.

2.2. Markov model on trees

To describe the evolution of characters (such as DNA nucleotides) along the edges of a phylogenetic network, we use *Markov models*. We will discuss two different models, the Jukes-Cantor (JC) model, and the Kimura 2-parameter (K2P) model [6]. These models are both reversible, meaning that we can move interchangeably between rooted networks and semi-directed networks when discussing the parametrization of a phylogenetic network model. The models are statistical models, describing the probability distribution of the characters that can be observed at the leaves of a n-leaf network. In these models, characters evolve independently along the edges of the network according to the transition probabilities.

Formally, for each edge e in the network, a transition matrix M^e describes the probability of changing from one nucleotide to another. The resulting probabilities for observed patterns at the leaves can be used to infer or distinguish between different network topologies. To find these probabilities, we start with defining the Markov model. Given a rooted phylogenetic network $N = (V,E)$, each $v \in V$ is associated with a random variable X_v with state space $\Sigma = \{A,G,C,T\}$, which are the four DNA bases. We associate for every edge e a 4 x 4 transition matrix $M^e \in S$, where we let S_4 be the set of 4 x 4 stochastic matrices. Thus, the transition matrix is equipped for each edge $e = u \to w \in E$ such that $M^e_{ij} = P(X_w = j | X_u = i)$. Moreover, the root distribution is given as $\pi = (\pi_A, \pi_G, \pi_C, \pi_T) \subset [0,1]^4$.
Now, we can find the probability for a specific situation. Let T be a tree with vertex set $V(T)$ and edge set $E(T)$ and ϕ an assignment of $V(T)$ to states Σ. The probability that an assignment ϕ can be observed under our Markov model is

$$p(\phi) = \pi_{\phi(\rho)} \prod_{e \in E(T)} M^e_{\phi(w),\phi(u)}.$$

In this report, we are interested in the states at the leaves. So, we can marginalize the probabilities $p(\phi)$ to find the probability of a specific situation $w \in \Sigma^n$ at the leaves of T. We denote $\phi(X)$ as the restriction of

assignment ϕ to the leaves X. Then, the probability of observing w in T is given by

$$p_w(T) = \sum_{\phi:\phi(X)=w} p(\phi) = \sum_{\phi:\phi(X)=w} \pi_{\phi(\rho)} \prod_{e\in E(T)} M^e_{\phi(w),\phi(u)}$$

Until now, we have looked at trees that enforce a strict branching structure. However, more complex evolutionary processes, including reticulate events where lineages merge, are not represented in trees. These changes are taken into account in phylogenetic networks with so-called reticulation points.

2.3. Markov model on networks

In phylogenetic networks, reticulation events model evolutionary processes such as horizontal gene transfer, hybridization, or recombination. These events introduce cycles into the network structure and are represented by *reticulation vertices*. To incorporate such events in a probabilistic model of sequence evolution, we introduce *reticulation parameters*.

Let N be a rooted binary phylogenetic network with $r \geq 1$ reticulation vertices $v_1, \ldots, v_r$. Each v_i has two incoming edges, denoted by e_i^0 and e_i^1. We assign a probability parameter $\delta_i \in (0,1)$ to edge e_i^1, and $1-\delta_i$ to e_i^0. These parameters reflect the probability that a site follows one of the two alternative paths through the reticulation vertex.

To compute the site pattern probabilities, we consider all 2^r possible combinations of choices at reticulation vertices, represented by binary vectors $\sigma \in \{0,1\}^r$, where $\sigma_i = 0$ means edge e_i^0 was deleted (and e_i^1 kept), and vice versa for $\sigma_i = 1$. In total, there are 2^r possible combinations, since for all r vertices, there are two option (edge e_i^0 was deleted and e_i^1 kept, or vice versa). Each such configuration corresponds to a tree T_σ obtained from the network N by resolving the reticulation choices according to σ.

Example 2.5. In Figure 2.8 a 4-leaf semi-directed network is shown. The network has two reticulation vertices w_1 and w_2. There are four possible binary vectors of length two: $\alpha = (0,0), \beta = (0,1), \gamma = (1,0),$ *and* $\delta = (1,1)$. The four different trees from this network are shown in Figure 2.9 [3].

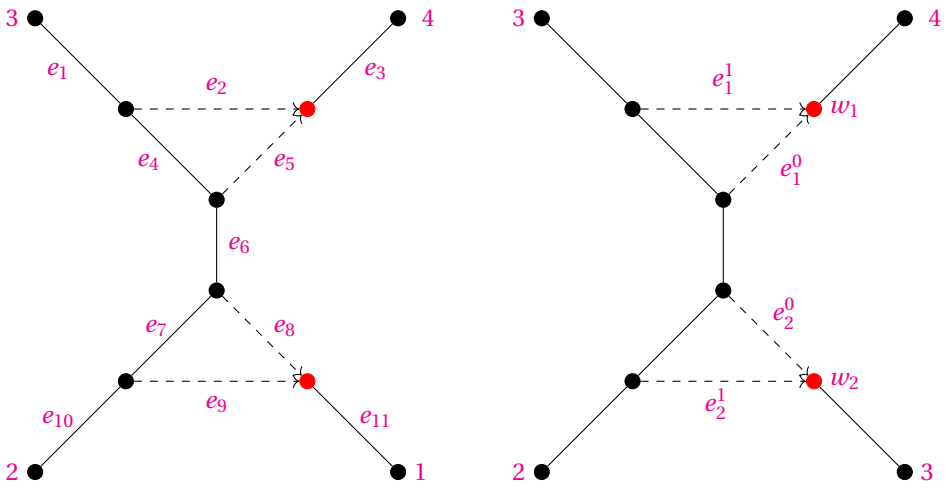


Figure 2.8: A 4-leaf semi-directed network with two reticulation vertices w_1 and w_2. The left figure displays the edge labeling, which will be used in example 2.3.3. The right figure shows the four reticulation edges [3].

Given a site pattern $\omega = (g_1, \ldots, g_n) \in \Sigma^n$, where $\Sigma = \{A, C, G, T\}$, the probability of observing ω at the leaves of N is

$$(p_N)_\omega = \sum_{\sigma\in\{0,1\}^r} \left(\prod_{i=1}^{r} \delta_i^{1-\sigma_i} (1-\delta_i)^{\sigma_i} \right) (p_{T_\sigma})_\omega,$$

where $(p_{T_\sigma})_\omega$ is the site pattern probability under the corresponding tree T_σ, computed using standard methods for phylogenetic trees as discussed previously.

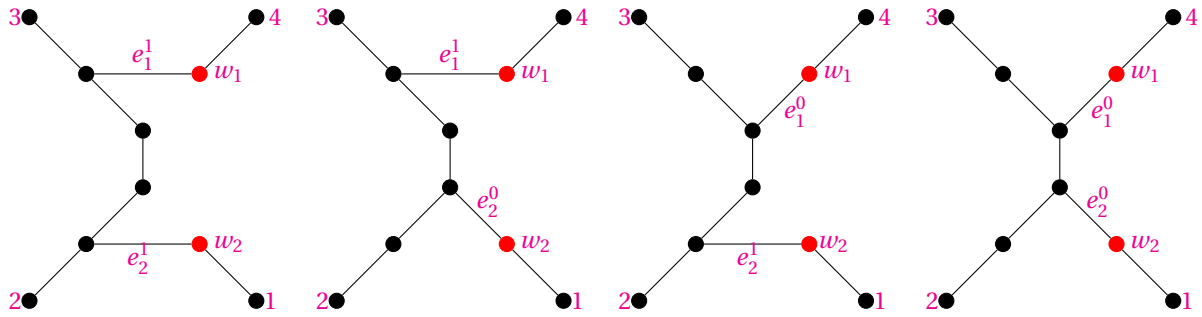


Figure 2.9: The trees T_α, T_β, T_δ and T_γ from left to right obtained from 2.8. For every tree, one of the four binary vectors is used [3].

Example 2.6. Consider a network N with one reticulation vertex v and incoming edges e^0, e^1. Let $\delta \in (0,1)$ be the probability of retaining edge e^1. Then the site pattern probability is:

$$(p_N)_\omega = \delta \cdot (p_{T_0})_\omega + (1-\delta) \cdot (p_{T_1})_\omega,$$

where T_0 and T_1 are the two trees resulting from choosing e^1 and e^0 respectively. This model captures the evolutionary ambiguity introduced by a single reticulation event.

2.3.1. JC model

We will consider two kind of submodels of the general Markov Model obtained by placing restrictions on the transition matrices. The simplest model is the Jukes-Cantor (JC) model. This model assumes that the transition probabilities M^e_{ij} is the same for all $i, j \in \Sigma, i \neq j, e \in E$, meaning that the probability of a transition or transversion is the same for all bases. This can be seen in the following matrix:

$$\begin{bmatrix} \beta & \alpha & \alpha & \alpha \\ \alpha & \beta & \alpha & \alpha \\ \alpha & \alpha & \beta & \alpha \\ \alpha & \alpha & \alpha & \beta \end{bmatrix}$$

Note that the probability of a mutation is α and $\beta + 3 \cdot \alpha = 1$.

2.3.2. K2P model

For the Kimura 2-parameter model (K2P), we add an extra parameter to differentiate between the probabilities of transitions (changes between either states A and G or C and T) and transversions. This can be seen in the transition matrix:

$$\begin{bmatrix} \beta & \alpha & \gamma & \alpha \\ \alpha & \beta & \alpha & \gamma \\ \gamma & \alpha & \beta & \alpha \\ \alpha & \gamma & \alpha & \beta \end{bmatrix}$$

Here we have $2\alpha + \beta + \gamma = 1$, with γ the transition probability. We can obtain the JC model by setting $\gamma = \alpha$.

2.3.3. Fourier transform

The Fourier transform is a linear transformation that converts the site pattern probabilities into a new coordinate system, called Fourier coordinates. That leads to a model which is monomial instead of a polynomial in its parameters, with the advantage of a simplified expression for the site pattern probabilities.

This method applies particularly well to models whose state space can be equipped with the structure of a finite abelian group. For example, in the JC and K2P models, the DNA bases are identified with the elements

of the Klein four-group $\mathbb{Z}_2 \times \mathbb{Z}_2$ via the correspondence

$$A = (0,0), C = (0,1), G = (1,0), T = (1,1).$$

We associate four Fourier parameters with each edge $e \in E$, denoted by a_A^e, a_G^e, a_C^e and a_T^e. These parameters describe the evolutionary behavior along the edge e in Fourier coordinates and satisfy model-specific constraints. Under the JC model, we have $a_C^e = a_G^e = a_T^e$, and under the K2P model we have that $a_G^e = a_T^e$, distinguishing transitions and transversions. Furthermore, $a_A^e = 1$ for biological parameters. All other parameters are in the interval $(0,1)$. The parameters cannot be one because that would correspond to a zero branch length, meaning that no evolutionary change or divergence has occurred along that branch.

Then, let $\omega = (g_1, g_2, \ldots, g_n)$ be a site pattern across the leaves of a tree T. Denote by $\Sigma(T)$ the set of splits induced by the edges of T. For each split $A|B \in \Sigma(T)$ we associate parameters $a_g^{A|B}$ for each group element $g \in \mathbb{Z}_2 \times \mathbb{Z}_2$, where $a_g^{A|B} = a_g^e$ for the edge e that induces the split. Then, the Fourier parametrization of $p_\omega(T)$ is given by

$$q_\omega(T) = \begin{cases} \prod_{\substack{e \in E(T) \\ e \text{ induces the split } A|B}} a^e_{\sum_{i \in A} g_i} & \text{if } \sum_{i=1}^n g_i = 0, \\ 0 & \text{otherwise} \end{cases}$$

where addition is in the group $\mathbb{Z}_2 \times \mathbb{Z}_2$. The condition $\sum_{i=1}^n g_i = 0$ ensures that the overall site pattern is compatible with a tree-based evolutionary history. Note that for a network with r reticulation vertices, each Fourier coordinate for the model will consist of 2^r distinct terms.

Example 2.7 (Fourier parametrization)**.** Suppose that we label the edges of N as displayed on the left in Figure 2.8. We want to compute the Fourier coordinate of observing nucleotides T, G, C, and A at the leaves 1, 2, 3 and 4, respectively. We will denote by a_g^i the parameter $a_g^{e_i}$ to simplify the notation. Then

$$\begin{aligned} q_{TGCA} = {} & \delta_1\delta_2(a_C^1 a_A^2 a_A^3 a_C^4 a_C^6 a_C^7 a_T^9 a_G^{10} a_T^{11}) + \delta_1\delta_2'(a_C^1 a_A^2 a_A^3 a_C^4 a_C^6 a_G^7 a_T^8 a_G^{10} a_T^{11}) \\ & + \delta_1'\delta_2(a_C^1 a_A^3 a_A^4 a_C^5 a_A^6 a_C^7 a_T^9 a_G^{10} a_T^{11}) + \delta_1'\delta_2'(a_C^1 a_A^3 a_A^4 a_C^5 a_A^6 a_C^7 a_G^8 a_T^{10} a_G^{10} a_T^{11}) \end{aligned}$$

In the above parameterization, the first, second, third and fourth terms correspond to the trees T_α, T_β, T_δ and T_γ in Figure 2.9, respectively [3].

3

Distinguishing symmetric level-3 trinets from three-leaf trees

In this Chapter, we will give a proof of the following Theorem, which is an extension of the Theorem given in Englander et al. [6]. We focus on a specific type of level-3 trinet depicted in Figure 3.2, which we refer to as a *symmetric level-3 trinet*, due to its rotational symmetry. We will show that we can distinguish the symmetric level-3 trinet from a three-leaf tree (depicted in Figure 3.1). Before we give the proof of Theorem 3.1 in Section 3.2, we will state and prove two Lemmas in Section 3.1. The Lemmas are used to show that an invariant applied to the symmetric level-3 trinet is strictly positive.

Theorem 3.1. *Let N_1 and N_2 be two binary semi-directed level-3 phylogenetic networks on the same three leaves. Let N_1 be the symmetric level-3 trinet with leaves $X = \{1,2,3\}$ from Figure 3.2 and N_2 the three-leaf tree with leaves $X = \{1,2,3\}$ from Figure 3.1. Then, under the JC model and for all parameter values in $(0,1)$, the polynomial invariant*

$$q_{011}q_{101}q_{110} - q_{111}^2$$

evaluates to zero for N_2 and is strictly positive for N_1. Hence, N_1 and N_2 are distinguishable under the JC model.

Note that we used a simplified notation for the Fourier coordinates under the JC model. For instance, we write q_{110} to denote any of the equivalent values $q_{CCA} = q_{GGA} = q_{TTA}$, since they are all the same under the Jukes-Cantor. In the same way, we define q_{101} and q_{011}. The coordinate q_{111} refers to any of the values q_{CGT}, q_{CTG}, q_{GCT}, q_{GTC}, q_{TGC}, q_{TCG}, which are again equal in this model. This compact notation makes it easier to visually distinguish between the different q-coordinates.

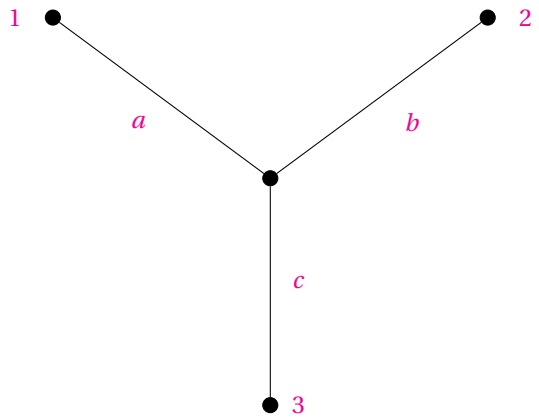


Figure 3.1: Tree with three leaves $1,2,3$

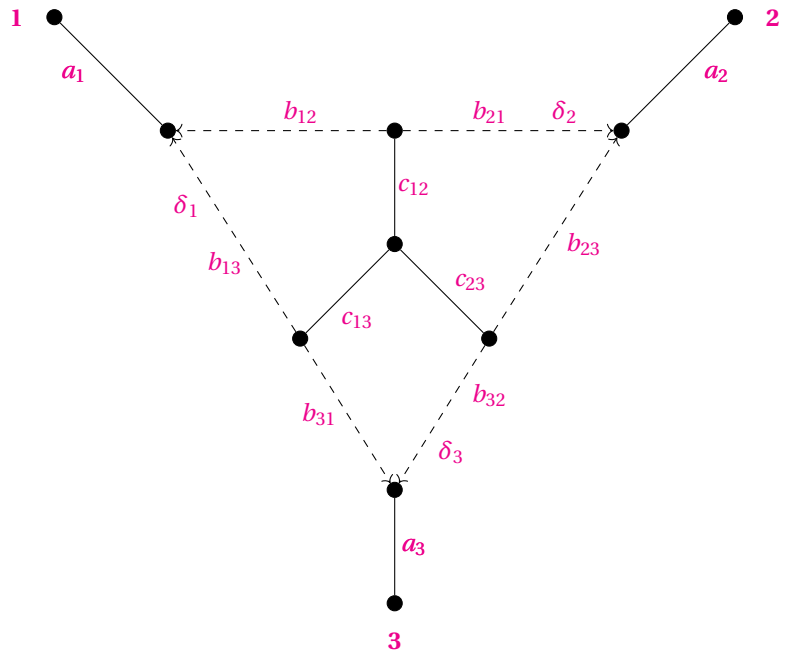


Figure 3.2: Symmetric level-3 trinet

3.1. Auxiliary Lemmas

In this section, we will look at two functions and show that these functions are strictly positive on the interval $(0,1)$. The first and second order partial derivatives will be calculated, to derive the Jacobian and Hessian matrix and the critical points. In the next section, we will use these Lemmas to show that the invariant of a symmetric level-3 trinet is strictly positive.

Lemma 3.1. *Let $x, y \in (0,1)$. Then, the following function is strictly positive in the interval $(0,1)^2$:*

$$f(x,y) = \frac{1}{2} + \frac{(2y^2-2y+1)x^2}{2} - xy. \tag{3.1}$$

Proof. To show strict positivity on the interval $(0,1)$, we have to evaluate the function. First, we have to find the critical points of the function f (given in 3.1) using the Jacobian matrix with the first-order partial derivatives. Then, we find the second-order partial derivatives to obtain the Hessian matrix, which we can use to determine what type of critical points the function has. In particular, we want to determine where local minima and saddle points occur and what the values of the critical points are. Lastly, we have to find the values of the boundary points, so that we can show that the function never gets below zero. This will then allow us to prove that this function is strictly positive in the open interval $(0,1)^2$.

We first start to compute the first-order partial derivatives:

$$\frac{\partial f}{\partial x} = (2y^2-2y+1)x - y \tag{3.2}$$

$$\frac{\partial f}{\partial y} = (2y-1)x^2 - x. \tag{3.3}$$

So the Jacobian matrix is

$$J_f(x,y) = \Big((2y^2-2y+1)x - y \quad (2y-1)x^2 - x\Big).$$

To find the critical points, we solve:

$$\frac{\partial f}{\partial x} = 0 \tag{3.4}$$

$$\frac{\partial f}{\partial y} = 0. \tag{3.5}$$

That is,

$$(2y^2-2y+1)x - y = 0 \tag{3.6}$$

$$(2y-1)x^2 - x = 0. \tag{3.7}$$

Solving equation 3.7 gives us:

$$x = 0 \quad \text{or} \quad (2y-1)x = 1.$$

If $x = 0$, equation 3.6 becomes $y = 0$. So the first critical point is $(0,0)$.
If $(2y-1)x = 1$, we get $x = \frac{1}{2y-1}$. Substitute this into equation 3.6:

$$\begin{aligned} (2y^2-2y+1)\cdot\frac{1}{2y-1} - y &= 0, \\ (2y^2-2y+1) - y(2y-1) &= 0, \\ -y+1 &= 0, \\ y &= 1. \end{aligned} \tag{3.8}$$

Substituting $y = 1$ into $x = \frac{1}{2y-1}$ gives $x = 1$. So the other critical point is $(1,1)$.
Now that we have the critical points, we have to find out what kind of points these are. Therefore, we compute the second-order partial derivatives:

$$\frac{\partial^2 f}{\partial x^2} = 2y^2 - 2y + 1, \tag{3.9}$$

$$\frac{\partial^2 f}{\partial y^2} = 2x^2, \tag{3.10}$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} = 2(2y-1)x - 1. \tag{3.11}$$

Thus, the Hessian matrix is:

$$H_f(x,y) = \begin{pmatrix} 2y^2 - 2y + 1 & 2(2y-1)x - 1 \\ 2(2y-1)x - 1 & 2x^2 \end{pmatrix}.$$

Now we can evaluate the critical points with the Hessian matrix.
At $(x,y) = (0,0)$:

$$H_f(0,0) = \begin{pmatrix} 1 & -1 \\ -1 & 0 \end{pmatrix}, \quad \det(H_f(0,0)) = 1 \cdot 0 - (-1)^2 = -1 < 0.$$

This Hessian is indefinite, so $(0,0)$ is a *saddle point.*

At $(x,y) = (1,1)$:

$$H_f(1,1) = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, \quad \det(H_f(1,1)) = 1 \cdot 2 - (1)^2 = 1 > 0, \quad \text{and } \frac{\partial^2 f}{\partial x^2} = 1 > 0.$$

The Hessian is positive definite, so $(1,1)$ is a *local minimum.*

We fill in the critical points to get the function values.

$$f(0,0) = \frac{1}{2}, \quad f(1,1) = 0.$$

Then, we have to find and evaluate the boundary points. Since we have $x, y \in (0,1)$, the boundary points are the following points:

1. $x = 0$, $y \in (0,1)$
2. $x = 1$, $y \in (0,1)$
3. $y = 0$, $x \in (0,1)$
4. $y = 1$, $x \in (0,1)$

The first boundary points give the value $f(0,y) = \frac{1}{2}$, which is positive.
The second boundary points give the value $f(1,y) = \frac{1}{2} + y^2 - y + \frac{1}{2} - y = y^2 - 2y + 1 = (y-1)^2$. This is strictly positive for $y \in (0,1)$, since $(y-1)^2$ is only zero for $y = 1$.
The third boundary points give the value $f(x,0) = \frac{1+x^2}{2}$, which is clearly positive.
The last boundary points give the value $f(x,1) = \frac{1}{2} + \frac{x^2}{2} - x = \frac{1}{2}(x-1)^2$. We can use the same argument as for the second boundary points that this is positive for $x \in (0,1)$.

So, the function 3.1 has a local minimum at $(1,1)$, where it attains the value $f(1,1) = 0$, and a saddle point at $(0,0)$ where it attains a value of $\frac{1}{2}$. Moreover, all other boundary points are positive. Since the function is continuous and the only saddle point has a positive value, the function is strictly positive on the open interval $(0,1)^2$. □

Lemma 3.2. *Let $x, y, z \in (0,1)$. Then, the following function is strictly positive in the interval $(0,1)^3$:*

$$f(x,y,z) = \left(\left(z^2 + \frac{1}{2}\right)y^2 - yz + \frac{1}{2}z^2\right)x^2 - yz(y+z)x + \frac{1}{2}y^2z^2 + \frac{1}{2}. \tag{3.12}$$

Proof. We use a similar strategy as in the previous Lemma. We first start to compute the first-order partial derivatives:

$$\frac{\partial f}{\partial x} = 2\left(\left(z^2+\frac{1}{2}\right)y^2 - yz + \frac{1}{2}z^2\right)x - yz(y+z), \tag{3.13}$$

$$\frac{\partial f}{\partial y} = x^2(2yz^2+y-z) - xz(2y+z) + yz^2, \tag{3.14}$$

$$\frac{\partial f}{\partial z} = x^2(2y^2z-y+z) - xy(y+2z) + y^2z. \tag{3.15}$$

So the Jacobian matrix is:

$$\nabla f(x,y,z) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix}.$$

To find the critical points, we solve:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = \frac{\partial f}{\partial z} = 0. \tag{3.16}$$

We get the following critical points inside $[0,1]^3$ using Wolfram Alpha [14],

$$(0,0,z), (1,0,0), (0,y,0) \text{ where } y \neq 0 \text{ and } (1,1,1).$$

We compute the Hessian matrix by taking the second-order partial derivatives of $f(x,y,z)$:

$$H_f(x,y,z) = \begin{bmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{yx} & f_{yy} & f_{yz} \\ f_{zx} & f_{zy} & f_{zz} \end{bmatrix},$$

where:

$$\begin{aligned}
\frac{\partial^2 f}{\partial x^2} &= 2\left(y^2\left(z^2+\frac{1}{2}\right) - yz + \frac{z^2}{2}\right) \\
\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} &= 2x\left(2y\left(z^2+\frac{1}{2}\right) - z\right) - yz - z(y+z) \\
\frac{\partial^2 f}{\partial x \partial z} = \frac{\partial^2 f}{\partial z \partial x} &= 2x\left(2y^2z - y + z\right) - yz - y(y+z) \\
\frac{\partial^2 f}{\partial y^2} &= 2x^2\left(z^2+\frac{1}{2}\right) - 2xz + z^2 \\
\frac{\partial^2 f}{\partial y \partial z} = \frac{\partial^2 f}{\partial z \partial y} &= x^2(4yz-1) - x(y+z) - x(y+z) + 2yz \\
\frac{\partial^2 f}{\partial z^2} &= x^2(2y^2+1) - 2xy + y^2.
\end{aligned}$$

Now, we can evaluate the critical points inside the interval $[0,1]^3$ to see where the saddle points and local minima and maxima are. First, we fill in the critical points to find the function values.

For $(1,1,1)$ we get $f(1,1,1) = 0$ and the Hessian is

$$H_f(x,y,z) = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

Table 3.1: For every critical point of the function f, the value of the function, the Hessian matrix, its eigenvalues, and the type of the critical point are listed.

Critical Point	$f(x,y,z)$	**Hessian Matrix**	**Eigenvalues**	**Type**
$(1,1,1)$	0	$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$	$\{4,1,1\}$	local minimum
$(0,0,z)$	$\frac{1}{2}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & z^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\{0,0,z^2\}$	inconclusive
$(0,y,0)$	$\frac{1}{2}$	$\begin{bmatrix} y^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & y^2 \end{bmatrix}$	$\{0,y^2,y^2\}$	inconclusive
$(1,0,0)$	$\frac{1}{2}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2} & -1 \\ 0 & -1 & 1 \end{bmatrix}$	$\{0,\frac{1}{4}(3\pm\sqrt{17})\}$	saddle point

This is a symmetric matrix with eigenvalues $\lambda = \{4,1,1\}$, all of which are positive. Therefore, the Hessian is positive definite at this point. The function has a local minimum at $(1,1,1)$.
We do the same for the other critical points. See table 3.1 for an overview of the critical points.

We have to look at the boundary points to check whether all boundary points are non-negative. The boundary of the open interval $(0,1)^3$ consist of the six faces of an open cube. The six faces are:

1. $x = 0,\ y,z \in (0,1)$
2. $x = 1,\ y,z \in (0,1)$
3. $y = 0,\ x,z \in (0,1)$
4. $y = 1,\ x,z \in (0,1)$
5. $z = 0,\ x,y \in (0,1)$
6. $z = 1,\ x,y \in (0,1)$

$$f(x,y,z) = \left(\left(z^2+\frac{1}{2}\right)y^2 - yz + \frac{1}{2}z^2\right)x^2 - yz(y+z)x + \frac{1}{2}y^2z^2 + \frac{1}{2}. \tag{3.17}$$

For the first face, we get the function value $f(0,y,z) = \frac{1}{2}y^2z^2 + \frac{1}{2}$. This is always positive for $y,z \in (0,1)$.
For the second face, we get $f(1,y,z) = \frac{3}{2}y^2z^2 + \frac{1}{2}y^2 - yz + \frac{1}{2}z^2 - y^2z - yz^2 + \frac{1}{2} = g(y,z)$. We can find with Wolfram Alpha that the global minimum of $g(y,z)$ is at $(y,z) = (1,1)$, where $g(1,1) = 0$. But, $(y,z) = (1,1)$ is not in our interval, so the boundary points in this face are strictly positive.
For the third boundary points, we get $f(x,0,z) = \frac{1}{2}x^2z^2 + \frac{1}{2}$. This is positive for the third face.
For the fourth face, we get $f(x,1,z) = \frac{3}{2}x^2z^2 - x^2z - xz^2 - xz + \frac{1}{2}x^2 + \frac{1}{2}z^2 + \frac{1}{2}$. This has the same structure as the second boundary points, so we conclude that the boundary points are strictly positive.
For the fifth face, we get $f(x,y,0) = \frac{1}{2}x^2y^2 + \frac{1}{2}$, which is positive as well.
For the sixth face, we get $f(x,y,1) = \frac{3}{2}x^2y^2 - x^2y - yz^2 - xy + \frac{1}{2}x^2 + \frac{1}{2}y^2 + \frac{1}{2}$. Again, this is the same structure as the second and fourth boundary points.

Therefore, all boundary points are strictly positive.

From the analysis of the critical points, we observe that the function attains in the domain $(0,1)^3$ its minimum value $f(x,y,z) = 0$ only at the point $(1,1,1)$, where the Hessian is positive definite, confirming a local minimum. All other critical points yield a strictly positive function value ($f = \frac{1}{2}$), with Hessians that are singular or indefinite. In particular, no other point inside the domain $(0,1)^3$ results in a function value of zero or a negative value, because the saddle point is at a positive value and all the boundary points are strictly negative. Therefore, we conclude that the function f cannot attain a negative value, so $f(x,y,z) > 0$ for all $(x,y,z) \in (0,1)^3$. □

3.2. Proof of main theorem

We will now give the proof of Theorem 3.1. We have to show that the invariant is strictly positive for the symmetric level-3 trinet, which we will do by splitting the invariant up in coefficients. Then, we will show that all coefficients are strictly positive, where we will use Lemma 3.1 and 3.2 for some of the coefficients.

Theorem (3.1)**.** *Let N_1 and N_2 be two binary semi-directed level-3 phylogenetic networks on the same three leaves. Let N_1 be the symmetric level-3 trinet with leaves $X = \{a,b,c\}$ from Figure 3.2 and N_2 the three-leaf tree with leaves $X = \{a,b,c\}$ from Figure 3.1. Then, under the JC model and for all parameter values in $(0,1)$, the polynomial invariant*

$$q_{011}q_{101}q_{110} - q_{111}^2 \tag{3.18}$$

evaluates to zero for N_2 and is strictly positive for N_1. Hence, N_1 and N_2 are distinguishable under the JC model.

Proof. We first recall the proof from [6], which shows that $q_{011}q_{101}q_{110} - q_{111}^2$ equals 0 for the three-leaf tree N_2. Let a, b and c be the nontrivial Fourier parameters associated to the three edges of N_2 (see also Figure 3.1). Then, under the JC model, the considered Fourier parametrizations of N_2 are $q_{111} = abc$, $q_{110} = ab$, $q_{101} = ac$ and $q_{011} = bc$. Clearly, we then get that $q_{011}q_{101}q_{110} - q_{111}^2 = 0$ for N_2.

Then, consider the symmetric level-3 trinet. See Figure 3.2. Let a_i, b_{ij} and c_{ij}, with $i,j \in \{1,2,3\}$, be the Fourier parameters associated to the edges of N (see Figure 3.2). Moreover, let δ_1, δ_2 and δ_3 be the reticulation parameters. The Fourier parametrization for N under the JC model is as follows:

$$\begin{aligned}
q_{011} &:= a_2a_3\big(b_{21}b_{31}c_{12}c_{13}\delta_2\bar{\delta}_3 + b_{21}b_{32}c_{12}c_{23}\delta_2\delta_3 + b_{23}b_{31}c_{13}c_{23}\bar{\delta}_2\bar{\delta}_3 + b_{23}b_{32}\delta_3\bar{\delta}_2\big)\\
q_{101} &:= a_1a_3\big(b_{12}b_{31}c_{12}c_{13}\bar{\delta}_1\bar{\delta}_3 + b_{12}b_{32}c_{12}c_{23}\delta_3\bar{\delta}_1 + b_{13}b_{32}c_{13}c_{23}\delta_1\delta_3 + b_{13}b_{31}\delta_1\bar{\delta}_3\big)\\
q_{110} &:= a_1a_2\big(b_{12}b_{23}c_{12}c_{23}\bar{\delta}_1\bar{\delta}_2 + b_{13}b_{21}c_{12}c_{13}\delta_1\delta_2 + b_{13}b_{23}c_{13}c_{23}\delta_1\bar{\delta}_2 + b_{12}b_{21}\delta_2\bar{\delta}_1\big)\\
q_{111} &:= a_1a_2a_3\big(\delta_1b_{13}\bar{\delta}_3b_{31}c_{13}(b_{21}c_{12}\delta_2 + b_{23}c_{23}\bar{\delta}_2) + \delta_3b_{32}\delta_1b_{13}c_{13}c_{23}(b_{21}c_{12}\delta_2 + b_{23}\bar{\delta}_2)\\
&\quad + \bar{\delta}_1b_{12}c_{12}b_{21}\delta_2(b_{31}c_{13}\bar{\delta}_3 + b_{32}c_{23}\delta_3) + \bar{\delta}_1b_{12}c_{12}c_{23}\bar{\delta}_2b_{23}(b_{31}c_{13}\bar{\delta}_3 + b_{32}\delta_3)\big).
\end{aligned}$$

Here we use the shorthand notation $\bar{\delta}_i = (1-\delta_i)$ for $i \in \{1,2,3\}$.
We compute $q_{011}q_{101}q_{110} - q_{111}^2$ and we assume that the parameter values are nontrivial (i.e. they are in $(0,1)$). To show that this invariant is strictly positive, we start by looking at the coefficients of the invariant for δ_1^2, $\delta_1\bar{\delta}_1$ and $\bar{\delta}_1^{\,2}$. The coefficients are found with the `collect` function in Maple [15].

To prove that the three coefficients of δ_1^2, $\delta_1\bar{\delta}_1$ and $\bar{\delta}_1^{\,2}$ are positive, we again split the three corresponding expressions up by considering their coefficients for δ_2^2, $\delta_2\bar{\delta}_2$ and $\bar{\delta}_2^{\,2}$. We repeat this a third time for the last reticulation parameter δ_3. Then, we have to consider 27 coefficients. This process of splitting up the coefficients is visualized by a tree diagram in Figure 3.3. In this Figure, the end points are coloured, where the same colour means that the coefficients are the same, up to labelling the parameters. We will now show that the coefficients corresponding to each of the endpoints are strictly positive. If all the coefficients are positive, we can conclude that the invariant $q_{011}q_{101}q_{110} - q_{111}^2$ is strictly positive.

We will explain how to find a specific coefficient, by giving an example. Specifically, we will consider the path in the middle of Figure 3.3 and then look at the coefficients corresponding to nodes 13, 14, and 15.

First, we split up the invariant over δ_1 and we have to find the coefficient of $\delta_1\bar{\delta}_1$.

The coefficient for $\delta_1\bar{\delta}_1$ is

$$\begin{aligned}
&a_2^2a_3^2\left(b_{21}b_{31}c_{12}c_{13}\delta_2\bar{\delta}_3 + b_{21}b_{32}c_{12}c_{23}\delta_2\delta_3 + b_{23}b_{31}c_{13}c_{23}\bar{\delta}_2\bar{\delta}_3 + b_{23}b_{32}\delta_3\bar{\delta}_2\right)a_1^2\left(b_{12}b_{31}c_{12}c_{13}\bar{\delta}_3 + b_{12}b_{32}c_{12}c_{23}\delta_3\right)\\
&\cdot\left(b_{13}b_{21}c_{12}c_{13}\delta_2 + b_{13}b_{23}c_{13}c_{23}\bar{\delta}_2\right) + a_2^2a_3^2\left(b_{21}b_{31}c_{12}c_{13}\delta_2\bar{\delta}_3 + b_{21}b_{32}c_{12}c_{23}\delta_2\delta_3 + b_{23}b_{31}c_{13}c_{23}\bar{\delta}_2\bar{\delta}_3 + b_{23}b_{32}\delta_3\bar{\delta}_2\right)\\
&\qquad\cdot a_1^2\left(b_{13}b_{32}c_{13}c_{23}\delta_3 + b_{13}b_{31}\bar{\delta}_3\right)\left(b_{12}b_{23}c_{12}c_{23}\bar{\delta}_2 + b_{12}b_{21}\delta_2\right)\\
&\quad - 2a_1^2a_2^2a_3^2\left(b_{12}c_{12}b_{21}\delta_2(b_{31}c_{13}\bar{\delta}_3 + b_{32}c_{23}\delta_3) + b_{12}c_{12}c_{23}\bar{\delta}_2b_{23}(b_{31}c_{13}\bar{\delta}_3 + b_{32}\delta_3)\right)\\
&\qquad\cdot\left(b_{13}\bar{\delta}_3b_{31}c_{13}(b_{21}c_{12}\delta_2 + b_{23}c_{23}\bar{\delta}_2) + \delta_3b_{32}b_{13}c_{13}c_{23}(b_{21}c_{12}\delta_2 + b_{23}\bar{\delta}_2)\right)
\end{aligned}$$

Then we split this up over δ_2 and look only at the coefficient of $\delta_2\bar{\delta}_2$.

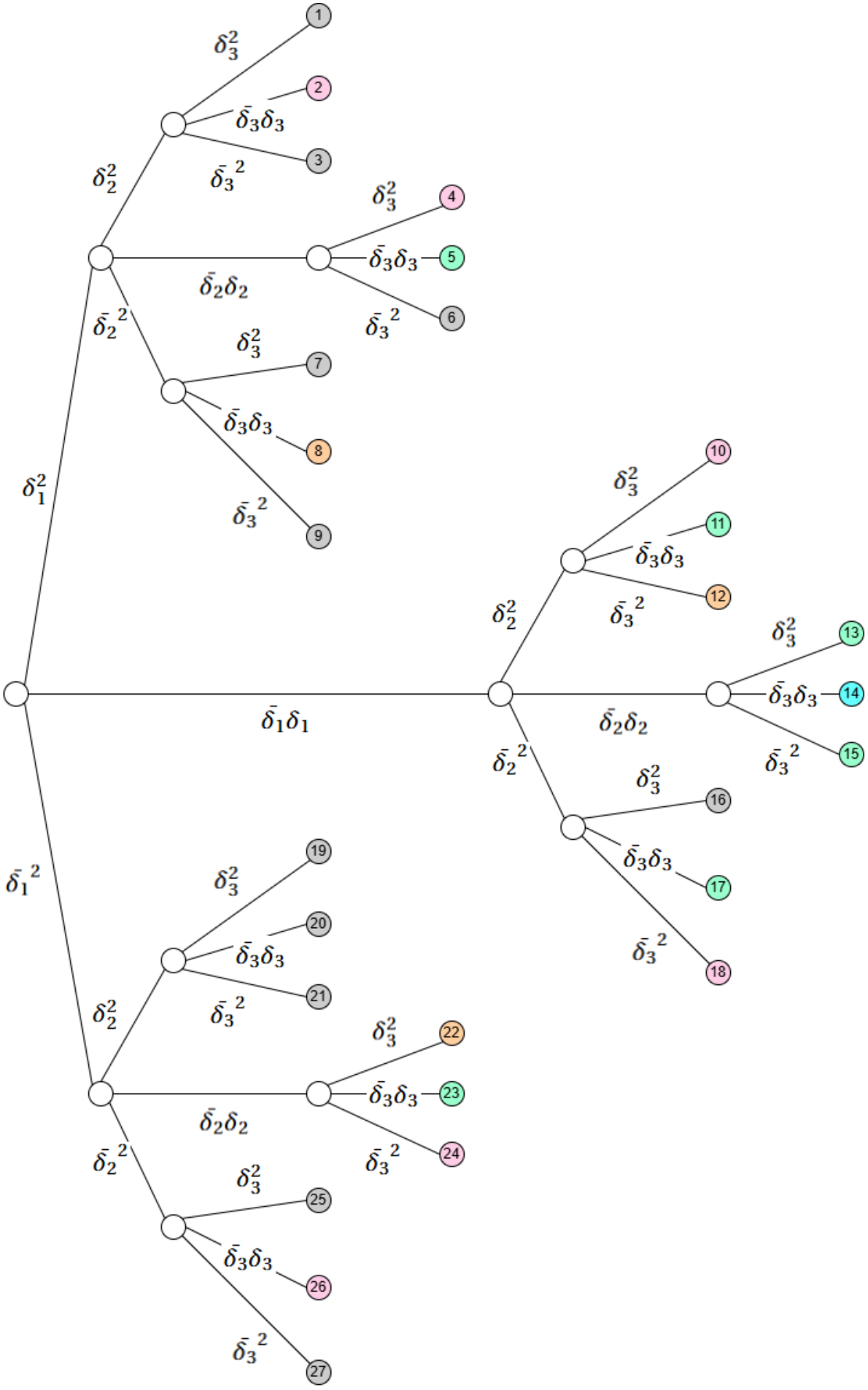


Figure 3.3: Tree of all coefficients of the invariant for the symmetric level-3 trinet

The coefficient of $\delta_2\bar{\delta}_2$ is

$$\begin{aligned}
&(b_{23}b_{31}c_{13}c_{23}\bar{\delta}_3 + b_{23}b_{32}\delta_3)(b_{12}b_{31}c_{12}c_{13}\bar{\delta}_3 + b_{12}b_{32}c_{12}c_{23}\delta_3)b_{13}b_{21}c_{12}c_{13}\\
&+ (b_{21}b_{31}c_{12}c_{13}\bar{\delta}_3 + b_{21}b_{32}c_{12}c_{23}\delta_3)(b_{12}b_{31}c_{12}c_{13}\bar{\delta}_3 + b_{12}b_{32}c_{12}c_{23}\delta_3)b_{13}b_{23}c_{13}c_{23}\\
&+ (b_{23}b_{31}c_{13}c_{23}\bar{\delta}_3 + b_{23}b_{32}\delta_3)(b_{13}b_{32}c_{13}c_{23}\delta_3 + b_{13}b_{31}\bar{\delta}_3)b_{12}b_{21}\\
&+ (b_{21}b_{31}c_{12}c_{13}\bar{\delta}_3 + b_{21}b_{32}c_{12}c_{23}\delta_3)(b_{13}b_{32}c_{13}c_{23}\delta_3 + b_{13}b_{31}\bar{\delta}_3)b_{12}b_{23}c_{12}c_{23}\\
&- 2\Big(b_{12}c_{12}c_{23}b_{23}(b_{31}c_{13}\bar{\delta}_3 + b_{32}\delta_3)(b_{13}b_{21}b_{32}c_{12}c_{13}c_{23}\delta_3 + b_{13}b_{21}b_{31}c_{12}c_{13}\bar{\delta}_3)\\
&\qquad + b_{12}c_{12}b_{21}(b_{31}c_{13}\bar{\delta}_3 + b_{32}c_{23}\delta_3)(b_{13}b_{23}b_{31}c_{13}c_{23}\bar{\delta}_3 + b_{13}b_{23}b_{32}c_{13}c_{23}\delta_3)\Big)
\end{aligned}$$

Splitting up with respect to the last reticulation parameter δ_3, gives us:

The coefficient of δ_3^2 is (node 13)

$$\begin{aligned}
&2b_{12}b_{13}b_{21}b_{23}b_{32}^2c_{12}^2c_{13}c_{23}^3 - 2b_{12}b_{13}b_{21}b_{23}b_{32}^2c_{12}^2c_{13}c_{23}^2 + b_{12}b_{13}b_{21}b_{23}b_{32}^2c_{12}^2c_{13}c_{23}\\
&- 2b_{12}b_{13}b_{21}b_{23}b_{32}^2c_{12}c_{13}c_{23}^2 + b_{12}b_{13}b_{21}b_{23}b_{32}^2c_{13}c_{23}\\
&= 2b_{32}^2c_{13}\left(\frac{1}{2} + (c_{23}^2 - c_{23} + \frac{1}{2})c_{12}^2 - c_{12}c_{23}\right)b_{13}b_{21}c_{23}b_{12}b_{23} \qquad (3.19)
\end{aligned}$$

The coefficient of $\delta_3\bar{\delta}_3$ is (node 14)

$$\begin{aligned}
&(b_{12}b_{23}b_{31}b_{32}c_{12}c_{13}c_{23}^2 + b_{12}b_{23}b_{31}b_{32}c_{12}c_{13})b_{13}b_{21}c_{12}c_{13} + (b_{13}b_{23}b_{31}b_{32}c_{13}^2c_{23}^2 + b_{13}b_{23}b_{31}b_{32})b_{12}b_{21}\\
&+ (b_{13}b_{21}b_{31}b_{32}c_{12}c_{13}^2c_{23} + b_{13}b_{21}b_{31}b_{32}c_{12}c_{23})b_{12}b_{23}c_{12}c_{23}\\
&- 2b_{21}b_{31}c_{12}^2c_{13}b_{12}b_{32}c_{23}b_{13}b_{23} - 2b_{12}c_{12}b_{21}b_{31}c_{13}^2b_{13}b_{23}b_{32}c_{23} - 2b_{12}c_{12}b_{21}b_{32}c_{23}^2b_{13}b_{23}b_{31}c_{13}\\
&= 2b_{31}b_{32}\left(\left(\left(c_{23}^2 + \frac{1}{2}\right)c_{13}^2 - c_{13}c_{23} + \frac{1}{2}c_{23}^2\right)c_{12}^2 - c_{13}c_{23}(c_{13} + c_{23})c_{12} + \frac{1}{2}c_{13}^2c_{23}^2 + \frac{1}{2}\right)b_{13}b_{21}b_{12}b_{23} \qquad (3.20)
\end{aligned}$$

The coefficient of $\bar{\delta}_3^2$ is (node 15)

$$\begin{aligned}
&2b_{12}b_{13}b_{21}b_{23}b_{31}^2c_{12}^2c_{13}^3c_{23} - 2b_{12}b_{13}b_{21}b_{23}b_{31}^2c_{12}^2c_{13}^2c_{23} + b_{12}b_{13}b_{21}b_{23}b_{31}^2c_{12}^2c_{13}c_{23}\\
&- 2b_{12}b_{13}b_{21}b_{23}b_{31}^2c_{12}c_{13}^2c_{23} + b_{12}b_{13}b_{21}b_{23}b_{31}^2c_{13}c_{23}\\
&= 2b_{31}^2c_{13}\left(\frac{1}{2} + \left(c_{13}^2 - c_{13} + \frac{1}{2}\right)c_{12}^2 - c_{12}c_{13}\right)b_{13}b_{21}c_{23}b_{12}b_{23} \qquad (3.21)
\end{aligned}$$

Using our auxiliary results from Section 3.1, we can prove that the coefficient of δ_3^2 (3.19) and $\bar{\delta}_3^2$ (3.21) is positive. In particular, we can use Lemma 3.1 with $y = c_{23}$, $x = c_{12}$ for the part between the brackets in coefficient 3.19, similarly we have $y = c_{13}$, $x = c_{12}$ for coefficient 3.21. We can conclude that the coefficients 3.19 and 3.21 are strictly positive, as the part in the brackets is strictly positive and the other parameters are all strictly positive. We can use Lemma 3.2 for the part in the brackets in coefficient 3.20 with $x = c_{12}$, $y = c_{13}$, $z = c_{23}$, and again the other parameters in the coefficient are in $(0,1)$. So, coefficient 3.20 is strictly positive as well. We can conclude that these three coefficients are strictly positive in the interval $(0,1)$ for the symmetric level-3 network.

We can find the other coefficients, which correspond to the coloured leaves shown in Figure 3.1, the same way as written above. As shown in Appendix A, all these coefficients are strictly positive or do not exist (grey nodes).

So, all coefficients of the symmetric level-3 trinet are strictly positive. Thus, we can write the invariant as a sum of positive parts and therefore we conclude that the symmetric level-3 network shown in Figure 3.2 and the three-leaf tree are distinguishable under the JC model.

□

4

Negative result for the K2P model

According to Englander et al. [6] we can distinguish a three-leaf tree N_1 from a strict level-1 trinet under the JC model by looking at the polynomial invariant

$$q_{011}q_{101}q_{110} - q_{111}^2. \tag{4.1}$$

In this Chapter, we will consider less restrictive assumptions and look at the K2P model. Surprisingly, we will show that extending invariant 4.1 to a seemingly equivalent polynomial that can be used for a K2P model will not distinguish between a three-leaf tree and a level-1 trinet.

4.1. Polynomial

Under the JC model, we could use a shorthand notation for the Fourier coordinates since we have $a_C^e = a_G^e = a_T^e$ for every edge e. However, under the K2P model we cannot use the shorthand notation as we did for the JC model, because we only have $a_T^e = a_G^e$ (and for example $a_C^e \neq a_G^e$). So, we must work with the full base-specific q-coordinates q_{ijk}, where $i, j, k \in \{A, C, G, T\}$.

If we want to use the original invariant 4.1, we have to replace the notation for the q-coordinates. We can, for example, replace q_{011} with q_{ACC}, and similarly for q_{101} and q_{110}. We can change invariant 4.1 to, for example, $q_{ACC}q_{CAC}q_{CCA} - q_{CGT}^2$. However, we cannot use this invariant to distinguish between a three-leaf tree and level-1 trinet. The invariant does not evaluate to zero for the three-leaf tree, because we do not have $a_G^e = a_T^e$. The problem here, is that we do not consider all the bases in the first part of the invariant. For example, we only have the C base in the first part of the invariant and did not take the other bases (G or T) into account. This issue does not arise under the JC model, since the parameters corresponding to C, G and T are identical due to the model's symmetry assumptions. Thus, we have to extend the invariant 4.1 to a new form that evaluates to zero for the three-leaf tree under the K2P model.

For the following, note that since $q_{011}q_{101}q_{110} - q_{111}^2$ was an invariant for JC, $q_{011}^3 q_{101}^3 q_{110}^3 - q_{111}^6$ is also an invariant for JC. For the JC model, we can write $q_{011}^3 = q_{ACC} \cdot q_{ATT} \cdot q_{AGG}$, and similarly for the other q-coordinates. If we fill this in for the second invariant, we consider all the bases for every leaf (and not only C). Therefore, we can extend invariant 4.1 to the following polynomial:

$$q_{CCA} \cdot q_{ACC} \cdot q_{CAC} \cdot q_{TTA}^2 \cdot q_{ATT}^2 \cdot q_{TAT}^2 - q_{CTT}^2 \cdot q_{TTC}^2 \cdot q_{TCT}^2 \tag{4.2}$$

Note that q_{TTA}, q_{ATT} and q_{TAT} each occur squared, because they are the same as q_{GGA}, q_{AGG} and q_{GAG}, respectively. So, we have replaced q_{110} in the original invariant by $q_{CCA} \cdot q_{TTA}^2$ and we can do the same for q_{101} and q_{011}. Thus, polynomial 4.2 is an extended version of invariant 4.1 under the K2P model.

4.2. Fourier paramatrization

To define the Fourier parametrization, we use the notation introduced in Section 2.3.3. The tree with three leaves can be found in Figure 3.1 and the only level-1 trinet is shown in Figure 4.1. For the tree we have the general Fourier transformation given in formula 4.3, where g_i is one of the four DNA bases (A, C, G, T).

$$q_{g_1 g_2 g_3} = a_{g_1}^a a_{g_2}^b a_{g_3}^c \tag{4.3}$$

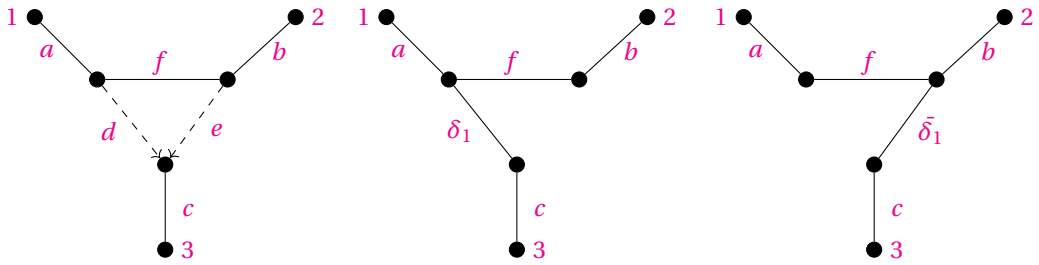


Figure 4.1: The only level-1 trinet on the left and its displayed trees with reticulation parameters δ_1 and $\bar{\delta_1}$ on the right

We can find the q-coordinates for the tree. We have the following Fourier parameterizations. We will use $a^e_A = 1$:

$$
\begin{aligned}
q_{CCA} &= a^a_C a^b_C \\
q_{CAC} &= a^a_C a^c_C \\
q_{ACC} &= a^b_C a^c_C \\
q_{TTA} &= a^a_T a^b_T \\
q_{TAT} &= a^a_T a^c_T \\
q_{ATT} &= a^b_T a^c_T \\
q_{CGT} &= a^a_C a^b_G a^c_T = a^a_C a^b_T a^c_T \\
q_{GTC} &= a^a_G a^b_T a^c_C = a^a_T a^b_C a^c_T \\
q_{TCG} &= a^a_T a^b_C a^c_G = a^a_T a^b_C a^c_T
\end{aligned}
$$

Now, we can look at the level-1 trinet. For the level-1 network we have the following general Fourier transform:

$$q_{g_1g_2g_3} = \delta_1(a^a_{g_1} a^b_{g_2} a^c_{g_3} a^d_{g_3} a^f_{g_2}) + \bar{\delta}_1(a^a_{g_1} a^b_{g_2} a^c_{g_3} a^e_{g_3} a^f_{g_1}) \tag{4.4}$$

We can find all the q-coordinates using the Fourier transformation based on the trinet given in Figure 4.1. That gives us the following:

$$
\begin{aligned}
q_{CCA} &= \delta_1(a_C^a a_C^b a_A^c a_A^d a_C^f) + \bar{\delta}_1(a_C^a a_C^b a_A^c a_A^e a_C^f) \\
&= \delta_1(a_C^a a_C^b a_C^f) + \bar{\delta}_1(a_C^a a_C^b a_C^f) = a_C^a a_C^b a_C^f \\
q_{CAC} &= \delta_1(a_C^a a_C^c a_C^d) + \bar{\delta}_1(a_C^a a_C^c a_C^e a_C^f) \\
&= a_C^a a_C^c(\delta_1(a_C^d) + \bar{\delta}_1(a_C^e a_C^f)) \\
q_{ACC} &= \delta_1(a_C^b a_C^c a_C^d a_C^f) + \bar{\delta}_1(a_C^b a_C^c a_C^e) \\
&= a_C^b a_C^c(\delta_1(a_C^d a_C^f) + \bar{\delta}_1(a_C^e)) \\
q_{TTA} &= \delta_1(a_T^a a_T^b a_A^c a_A^d a_T^f) + \bar{\delta}_1(a_T^a a_T^b a_A^c a_A^e a_T^f) \\
&= \delta_1(a_T^a a_T^b a_T^f) + \bar{\delta}_1(a_T^a a_T^b a_T^f) = a_T^a a_T^b a_T^f \\
q_{TAT} &= \delta_1(a_T^a a_T^c a_T^d) + \bar{\delta}_1(a_T^a a_T^c a_T^e a_T^f) \\
&= a_T^a a_T^c(\delta_1(a_T^d) + \bar{\delta}_1(a_T^e a_T^f)) \\
q_{ATT} &= \delta_1(a_T^b a_T^c a_T^d a_T^f) + \bar{\delta}_1(a_T^b a_T^c a_T^e) \\
&= a_T^b a_T^c(\delta_1(a_T^d a_T^f) + \bar{\delta}_1(a_T^e)) \\
q_{CGT} &= \delta_1(a_C^a a_G^b a_T^c a_T^d a_G^f) + \bar{\delta}_1(a_C^a a_G^b a_T^c a_T^e a_C^f) \\
&= a_C^a a_G^b a_T^c(\delta_1(a_T^d a_G^f) + \bar{\delta}_1(a_T^e a_C^f)) \\
q_{GTC} &= \delta_1(a_G^a a_T^b a_C^c a_C^d a_T^f) + \bar{\delta}_1(a_G^a a_T^b a_C^c a_C^e a_G^f) \\
&= a_T^a a_T^b a_C^c a_T^f(\delta_1(a_C^d) + \bar{\delta}_1(a_C^e)) \\
q_{TCG} &= \delta_1(a_T^a a_C^b a_T^c a_T^d a_C^f) + \bar{\delta}_1(a_T^a a_C^b a_T^c a_T^e a_T^f) \\
&= a_T^a a_C^b a_T^c(\delta_1(a_T^d a_C^f) + \bar{\delta}_1(a_T^e a_T^f))
\end{aligned}
$$

We will use these Fourier transformations to calculate the polynomial.

4.3. Negative result

Now that we have the Fourier transformations, we can have a look at the polynomial. Unfortunately, as stated in Theorem 4.1 the polynomial 4.2 cannot be used to distinguish between a level-1 trinet and level-0 trinet.

Theorem 4.1. *Under the K2P model, a three leaf tree N_1 and a level-1 trinet N_2 are **not** distinguishable with the polynomial*

$$q_{CCA} \cdot q_{ACC} \cdot q_{CAC} \cdot q_{TTA}^2 \cdot q_{ATT}^2 \cdot q_{TAT}^2 - q_{CGT}^2 \cdot q_{GTC}^2 \cdot q_{TCG}^2. \tag{4.5}$$

The polynomial can be both negative, positive or zero for N_2, and evaluates to zero for N_1.

Proof. We will show that the polynomial is zero for the level-0 trinet, a tree. We get the following if we fill in the q-coordinates, obtained in Section 4.2, in the polynomial 4.5:

$$
\begin{aligned}
&a_C^a a_C^b a_C^a a_C^c a_C^b a_C^c (a_T^a a_T^b a_T^a a_T^c a_T^b a_T^c)^2 - (a_C^a a_T^b a_T^c a_T^a a_C^b a_T^c a_T^a a_C^b a_T^c)^2 \\
&\quad = (a_C^a a_C^b a_C^c)^2 (a_T^a a_T^b a_T^c)^4 - (a_C^a a_C^b a_T^c (a_T^a a_T^b a_T^c)^2)^2 = 0
\end{aligned}
$$

We see that the polynomial becomes zero, as we expected.
Now, we have a look at the polynomial for the level-1 trinet.

$$
\begin{aligned}
&(a_C^b)^2 (a_C^c)^2 (a_T^b)^4 (a_C^a)^2 (a_T^c)^4 a_C^f (a_T^a)^4 (a_T^f)^2 \left(\delta_1 a_C^d a_C^f + \bar{\delta}_1 a_C^e\right) \left(\delta_1 a_T^d a_T^f + \bar{\delta}_1 a_T^e\right)^2 \left(\bar{\delta}_1 a_C^e a_C^f + \delta_1 a_C^d\right) \left(\bar{\delta}_1 a_T^e a_T^f + \delta_1 a_T^d\right)^2 \\
&- (a_C^a)^2 (a_T^b)^4 (a_T^c)^4 (a_T^a)^4 (a_C^c)^2 (a_T^f)^2 (a_C^b)^2 \left(\delta_1 a_T^d a_T^f + \bar{\delta}_1 a_T^e a_C^f\right)^2 \left(\delta_1 a_C^d + \bar{\delta}_1 a_C^e\right)^2 \left(\delta_1 a_T^d a_C^f + \bar{\delta}_1 a_T^e a_T^f\right)^2
\end{aligned} \tag{4.6}
$$

We can find examples in which the polynomial can be positive, negative or zero, for all parameters in $(0,1)$. For example, taking all parameters $\frac{1}{2}$ gives us an polynomial with value $3.947207004 \cdot 10^{-10}$, which is positive.

And changing $a_T^f = 0.9$ and keeping all the other variables $\frac{1}{2}$, gives a value of $-5.3204863 \cdot 10^{-10}$, which is negative. This means that the polynomial can become zero, because the polynomial is a continuous function (Intermediate Value Theorem). In Appendix B we further explore specific parameter values that evaluate at zero for this polynomial.

We conclude that the polynomial 4.5 cannot be used to distinguish between a level-1 and level-0 zero trinet under the K2P model, as the polynomial can attain the value zero for both trinets. □

5

Discussion

We have shown in Chapter 3 that a symmetric level-3 trinet is distinguishable from a three-leaf tree under the Jukes-Cantor model of evolution. This is a first distinguishability result on phylogenetic networks considering level-3 trinets. Since we have considered only one specific level-3 trinet, it remains open to prove a more general statement. We conjecture that all level-3 trinets are distinguishable from level-0 trinets. The particular case demonstrated in Chapter 3 is, intuitively, the hardest to distinguish as it seems most similar to the 3-leaf tree. In particular, the other level-3 cases tend to be less symmetric and therefore appear easier to distinguish. Since the invariant works for this case, we believe it is likely to work for all level-3 trinets. We expect that we can prove this by considering all level-3 trinet cases and showing that every case can be distinguished from a three-leaf tree, where we use the polynomial invariant (4.1). The same approach has been done in [6] for level-2 trinets. However, the number of level-3 trinets is still unclear and we do not know how they look like. A first step would be to find all level-3 trinets. A way to do this is to look at their generators. All semi-directed binary level-3 generators are presented by Nipius [16].
Furthermore, we can look at the analog of this result for arbitrary level-k trinets, where $k \in \mathbb{N}$. We conjecture that our result also hold for level-k trinets and it would be interesting to identify this.

Conjecture 5.1. *Let N_1 be a strict, semi-directed level-k trinet with leaves $X = \{a,b,c\}$ and N_2 the three-leaf tree with leaves $X = \{a,b,c\}$. Then, under the JC model and for all parameter values in $(0,1)$, the polynomial invariant*

$$q_{011}q_{101}q_{110} - q_{111}^2$$

evaluates to zero for N_2 and is strictly positive for N_1. Hence, N_1 and N_2 are distinguishable under the JC model.

To prove Conjecture 5.1, analyzing each network on a case by case basis, as it done for level-1 and level-2 trinets, is impossible. Even for small k, the case by case method is not convenient as the number of generators for level-k networks grows very rapidly, making a similar case analysis impossible in general. Therefore, it would be interesting to consider an inductive proof. We already know that we can distinguish a level-1 or level-2 trinet from a level-0 trinet as stated in Englander et al. [6], and our result for a symmetric level-3 trinet suggests that it is possible that this can be extended to level-3 trinets. In an inductive proof, we would use the result for level-1 trinets as the base case. Then, we assume that we can distinguish all level-k trinets from a three-leaf tree, and want to show that this also holds for a level-$(k+1)$ trinet. We can try to split the level-$(k+1)$ trinet up into level-k' trinets, where $k' < k$.

We have not been able to prove distinguishablity of trinets under the K2P model. It is surprising that an extended invariant for the K2P model compared to the invariant used in the proof for JC model does not work. The invariant evaluates to zero for the three-leaf tree, but for the level-1 trinet it attains negative and positive values, and can become zero. It remains an open problem to find another polynomial invariant that distinguishes trees and networks, or another approach. It is also possible that the two trinets are not distinguishable at all. This is significantly more difficult to prove. This remains an interesting problem, as K2P is a more realistic model for phylogenetic analysis.

Bibliography

[1] Elizabeth S. Allman, Sonia Petrović, John A. Rhodes, and Seth Sullivant. Identifiability of two-tree mixtures for group-based models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8 (3):710–722, 2011. doi: 10.1109/TCBB.2010.79.

[2] Elizabeth S. Allman, Hector Baños, and John A. Rhodes. Nanuq: a method for inferring species networks from gene trees under the coalescent model. *Algorithms for Molecular Biology*, 14(1):24, 2019. URL https://doi.org/10.1186/s13015-019-0159-2.

[3] Muhammad Ardiyansyah. Distinguishing level-2 phylogenetic networks using phylogenetic invariants, 2021. URL https://arxiv.org/abs/2104.12479.

[4] Eric Bapteste, Leo van Iersel, Axel Janke, Stefanie M. Kelchner, Steven Kelk, et al. Networks: expanding evolutionary thinking. *Trends in Genetics*, 29(8):439–441, 2013. URL https://doi.org/10.1016/j.tig.2013.05.007.

[5] BioInteractive. Creating phylogenetic trees from dna sequences, n.d. URL https://www.biointeractive.org/classroom-resources/creating-phylogenetic-trees-dna-sequences. Accessed: 2025-06-12.

[6] Aviva K. Englander, Martin Frohn, Elizabeth Gross, Niels Holtgrefe, Leo van Iersel, Mark Jones, and Seth Sullivant. Identifiability of phylogenetic level-2 networks under the jukes-cantor model. *bioRxiv*, 2025. doi: 10.1101/2025.04.18.649493. URL https://www.biorxiv.org/content/early/2025/04/24/2025.04.18.649493.

[7] Martin Frohn, Niels Holtgrefe, Leo van Iersel, Mark Jones, and Steven Kelk. Reconstructing semi-directed level-1 networks using few quarnets. *Journal of Computer and System Sciences*, 152:103655, 2025. URL https://www.sciencedirect.com/science/article/pii/S0022000025000376.

[8] Elizabeth Gross, Colby Long, and Joseph Rusinko. Phylogenetic networks, 2019. URL https://arxiv.org/abs/1906.01586.

[9] Elizabeth Gross, Leo van Iersel, Remie Janssen, Mark Jones, Colby Long, and Yukihiro Murakami. Distinguishing level-1 phylogenetic networks on the basis of data generated by markov processes. *Journal of Mathematical Biology*, 83(3), September 2021. URL https://doi.org/10.1007/s00285-021-01653-8.

[10] David M Hillis. Phylogenetic analysis. *Current Biology*, 7(3):R129–R131, 1997. URL https://www.sciencedirect.com/science/article/pii/S0960982297700708.

[11] N. Holtgrefe, K. T. Huber, L. van Iersel, M. Jones, S. Martin, and V. Moulton. Squirrel: Reconstructing semi-directed phylogenetic level-1 networks from four-leaved networks or sequence alignments. *Molecular Biology and Evolution*, 42(4):msaf067, 2025. doi: 10.1093/molbev/msaf067.

[12] Katharina T. Huber, Leo van Iersel, Steven Kelk, and Radoslaw Suchecki. A practical algorithm for reconstructing level-1 phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(3):635–649, 2011. doi: 10.1109/TCBB.2010.17.

[13] Daniel H. Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge University Press, Cambridge, 2010. ISBN 978-0-521-75596-2. URL https://www.cambridge.org/9780521755962.

[14] Wolfram Research, Inc. Mathematica, Version 14.2. URL https://www.wolfram.com/wolfram-alpha-notebook-edition. Champaign, IL, 2024.

[15] Maplesoft, a division of Waterloo Maple Inc.. Maple. URL https://hadoop.apache.org.

[16] Leonie Nipius. Encoding undirected and semi-directed binary phylogenetic networks by quarnets. Msc thesis, Delft University of Technology, Delft, Netherlands, July 2022. URL https://resolver.tudelft.nl/eaeb4293-6349-4d6e-ab53-94b781fc65e1.

[17] James Oldman, Taoyang Wu, Leo van Iersel, and Vincent Moulton. Trilonet: Piecing together small networks to reconstruct reticulate evolutionary histories. *Molecular Biology and Evolution*, 33(8):2151–2162, 04 2016. URL https://doi.org/10.1093/molbev/msw068.

[18] Claudia Solís-Lemus and Cécile Ané. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS Genetics*, 12(3):e1005896, 2016. URL https://doi.org/10.1371/journal.pgen.1005896.

[19] Shannon M. Soucy, Jinling Huang, and Johann Peter Gogarten. Horizontal gene transfer: building the web of life. *Nature Reviews Genetics*, 16(8):472–482, 2015. ISSN 1471-0064. URL https://doi.org/10.1038/nrg3962.

[20] Bernd Sturmfels and Seth Sullivant. Toric ideals of phylogenetic invariants. *Journal of Computational Biology*, 12(2):204–228, 2005. doi: 10.1089/cmb.2005.12.204.

A

Coefficients for symmetric level-3 trinet

In Theorem 3.1 we showed that the invariant $q_{011}q_{101}q_{110} - q_{111}^2$ evaluates to zero under the JC model for the three-leaf tree and is positive for the symmetric level-3 trinet. In this appendix we provide all the coefficients of the invariant and show that they are positive, which is part of the proof of Theorem 3.1. Recall that we numbered all the coefficients in Figure 3.1.

We can find all coefficients displayed by a node in Figure 3.1, in the same way as we did for nodes 13, 14, 15 in the proof of Theorem 3.1. The coefficients are given in table A.1, where the coefficients identical to eachother, up to relabelling the parameters, are given the same colour in the third column.

The coefficients corresponding to a pink node $(2, 4, 10, 18, 24, 26)$ are strictly positive, since all parameters are strictly positive and the square $(c_{ij} - 1)^2$ is always positive. The same holds for the orange nodes $(8, 12, 22)$. The green nodes $(5, 11, 13, 15, 17, 23)$ are also strictly positive, because we can use Lemma 3.1 for the part in the brackets. Lastly, the blue node (14) is strictly positive as the part inside the brackets is strictly positive using Lemma 3.2 and the other parameters are strictly positive. The gray coefficients $(1, 3, 6, 7, 9, 16, 19, 20, 21, 25, 27)$ are zero and thus non-existent.

So, all coefficients are not existing or strictly positive.

Table A.1: Coefficients for each coulored node in the tree diagram (Figure 3.1) representing the coefficients of the invariant (equation 3.18). Nodes with the same color share structurally equivalent coefficients.

Node	Coefficient	Color
1	-	
2	$b_{13}^2 b_{21}^2 b_{31} b_{32} c_{12}^2 c_{13} c_{23} (c_{13}-1)^2$	
3	-	
4	$b_{23} b_{32}^2 b_{13}^2 c_{13}^2 c_{23} b_{21} c_{12} (c_{23}-1)^2$	
5	$2 c_{12} b_{13}^2 b_{23} b_{31} (\frac{1}{2} + (c_{13}^2 - c_{13} + \frac{1}{2}) c_{23}^2 - c_{13} c_{23}) c_{13} b_{21} b_{32}$	
6	-	
7	-	
8	$b_{13}^2 b_{23}^2 b_{31} b_{32} c_{13} c_{23} (c_{13} c_{23} - 1)^2$	
9	-	
10	$b_{21}^2 b_{32}^2 c_{12} c_{23}^2 b_{13} c_{13} b_{12} (c_{12}-1)^2$	
11	$2 b_{32} c_{12} b_{13} b_{21}^2 (\frac{1}{2} + (c_{12}^2 - c_{12} + \frac{1}{2}) c_{13}^2 - c_{12} c_{13}) c_{23} b_{12} b_{31}$	
12	$b_{12} b_{13} b_{21}^2 b_{31}^2 c_{12} c_{13} (c_{12} c_{13} - 1)^2$	
13	$2 b_{32}^2 c_{13} (\frac{1}{2} + (c_{23}^2 - c_{23} + \frac{1}{2}) c_{12}^2 - c_{12} c_{23}) b_{13} b_{21} c_{23} b_{12} b_{23}$	
14	$2 b_{31} b_{32} b_{13} b_{21} b_{12} b_{23} (((c_{23}^2 + \frac{1}{2}) c_{13}^2 - c_{13} c_{23} + \frac{1}{2} c_{23}^2) c_{12}^2 - c_{13} c_{23} (c_{13} + c_{23}) c_{12} + \frac{1}{2} c_{13}^2 c_{23}^2 + \frac{1}{2})$	
15	$2 b_{31}^2 c_{13} (\frac{1}{2} + (c_{13}^2 - c_{13} + \frac{1}{2}) c_{12}^2 - c_{12} c_{13}) b_{13} b_{21} c_{23} b_{12} b_{23}$	
16	-	
17	$2 b_{32} (\frac{1}{2} + (c_{23}^2 - c_{23} + \frac{1}{2}) c_{13}^2 - c_{13} c_{23}) c_{12} b_{13} b_{23}^2 c_{23} b_{12} b_{31}$	
18	$b_{12} b_{13} b_{23}^2 b_{31}^2 c_{12} c_{13} c_{23}^2 (c_{13}-1)^2$	
19	-	
20	-	
21	-	
22	$b_{23} b_{32}^2 b_{12}^2 c_{12} c_{23} b_{21} (c_{12} c_{23} - 1)^2$	
23	$b_{32} c_{13} c_{12} (\frac{1}{2} + (c_{12}^2 - c_{12} + \frac{1}{2}) c_{23}^2 - c_{12} c_{23}) b_{21} b_{23} b_{12}^2 b_{31}$	
24	$b_{12}^2 b_{21} b_{23} b_{31}^2 c_{12} c_{13}^2 c_{23} (c_{12}-1)^2$	
25	-	
26	$b_{12}^2 b_{23}^2 b_{31} b_{32} c_{12}^2 c_{13} c_{23} (c_{23}-1)^2$	
27	-	

B

Maple code

In Section 4.3 we show in Theorem 4.1 that we cannot use invariant 4.5 to distinguish between a three-leaf trinet and a level-1 trinet. The invariant evaluates to zero for both trinets. Here we will give an example for which parameters the invariant for the level-1 trinet is zero.

Let all parameters be $\frac{1}{2}$, except for a_T^f. We will find the value of a_T^f for a zero polynomial with the maple code below. We use the *solve* function in Maple [15], see Listing B.1. The only reasonable value, which is in our interval $(0,1)$, is $a_T^f = 0.8398428397$. This leads to an polynomial that is zero for the level-1 network. Therefore, we cannot distinguish the network from the tree.

We can use Maple to find general algebraic solutions. One of the solutions is a polynomial of a_C^d, expressed in the parameters $\delta_1, a_T^d, a_T^e, a_C^f, a_T^f, a_C^e$. The other parameters are free to choose. This expression is too long to write down here, but the maple code can be found in the Listing B.2.

```
# Define variables
CCA := aC*bC*fC;
CAC := aC*cC*(ddelta*eC*fC + d1*dC);
ACC := bC*cC*(d1*dC*fC + ddelta*eC);

TTA := aT*bT*fT;
TAT := aT*cT*(ddelta*eT*fT + d1*dT);
ATT := bT*cT*(d1*dT*fT + ddelta*eT);

CGT := aC*bT*cT*(d1*dT*fT + ddelta*eT*fC);
GTC := aT*bT*cC*(d1*dC*fT + ddelta*eC*fT);
TCG := aT*bC*cT*(d1*dT*fC + ddelta*eT*fT);

fC := 0.5;
eC := 0.5;
eT := 0.5;
dT := 0.5;
dC := 0.5;
bC := 0.5;
bT := 0.5;
cC := 0.5;
cT := 0.5;
aC := 0.5;
aT := 0.5;
ddelta := 0.5;
d1 := 0.5;

invariant := ACC*ATT^2*CAC*CCA*TAT^2*TTA^2 - CGT^2*GTC^2*TCG^2;

expanded_invariant := expand(invariant);
```

```
solve(invariant = 0);
```

Listing B.1: Maple code to find the values of fT for an invariant equal to zero under the K2P model for the level-1 trinet

```
# Define variables
CCA := aC*bC*fC;
CAC := aC*cC*(ddelta*eC*fC + d1*dC);
ACC := bC*cC*(d1*dC*fC + ddelta*eC);

TTA := aT*bT*fT;
TAT := aT*cT*(ddelta*eT*fT + d1*dT);
ATT := bT*cT*(d1*dT*fT + ddelta*eT);

CGT := aC*bT*cT*(d1*dT*fT + ddelta*eT*fC);
GTC := aT*bT*cC*(d1*dC*fT + ddelta*eC*fT);
TCG := aT*bC*cT*(d1*dT*fC + ddelta*eT*fT);

invariant := ACC*ATT^2*CAC*CCA*TAT^2*TTA^2 - CGT^2*GTC^2*TCG^2;

solve(invariant = 0);
```

Listing B.2: Maple code to find the zero points for the invariant under the K2P model for the level-1 trinet jajajjaa