Counting Galled Networks

Rathin J
Roll No. 15PT30

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

FIVE YEAR INTEGRATED
M.Sc THEORETICAL COMPUTER SCIENCE

OF ANNA UNIVERSITY

NOVEMBER 2018

DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES

PSG COLLEGE OF TECHNOLOGY
(Autonomous Institution)

COIMBATORE – 641 004

PSG COLLEGE OF TECHNOLOGY
(Autonomous Institution)

COIMBATORE – 641 004

Seventh Semester
Project work

Counting Galled Networks

Bona fide record of work done by

Rathin J
Roll No. 15PT30

Submitted in partial fulfillment of the requirements for the degree of

FIVE YEAR INTEGRATED
M.Sc THEORETICAL COMPUTER SCIENCE
of Anna University

NOVEMBER 2018

_____ _____

Faculty Guide **Head of the Department**

Submitted for the Viva-Voce Examination held on _____

_____ _____

Internal Examiner **External Examiner**

Faculty of Science
Department of Mathematics
Tel: (65)-65166579
Email: matzlx@nus.edu.sg

NUS National University of Singapore

24 Oct. 2018

To whom it may concern,

This is to certify that Rathin J, student of M.Sc in Theoretical Computer Science, PSG College of Technology, Coimbatore, India, interned as a visiting research student at the Department of Mathematics, National University of Singapore under my supervision from May 28, 2018 to November 24, 2018.

During this period, Rathin has done a survey of the knowledge and concepts on phylogenetic networks including clusters, trees, network classes and tree reconciliation. He has investigated the problem of counting galled networks. He has also explored problems in network reconstruction and metrics on phylogenetic tree space. The results he obtained are valuable. A paper is in preparation.

During his visit, he regularly participated in research meetings held weekly in my research group. In summary, his performance is excellent and the objectives of his visit are met. He is ready for conducting further research in the theoretical aspects of phylogenetics.

Best regards.

Yours Sincerely

Louxin Zhang, Ph.D.
Professor
http://www.math.nus.edu.sg/~matzlx/

Contents

Acknowledgements

Firstly, I am much obliged to Dr. R. Rudramoorthy, Principal, PSG College of Technology for providing me with the opportunity to undertake my research project at The National University of Singapore.

I am greatly thankful to Dr. R. Nadarajan, Professor and Head, Department of Applied Mathematics and Computational Sciences for his invaluable lessons, insightful quips and for his buzzing personality which infects everyone with positivity.

I am deeply indebted to Dr. Louxin Zhang, Professor, Department of Mathematics, National University of Singapore, for his patience and continuous guidance. He has been an exemplary source of inspiration and I am greatly honoured to have had the opportunity to work with him during this project.

I am also extremely thankful to Dr. Andreas Gunawan for his endless help and support and for his unwaning enthusiasm to discuss new ideas.

I have no words to express my profound gratitude to my guide and mentor, Dr. R. S. Lekshmi, who has always been a pillar of support and a consistent motivator. Thank you for introducing me to the world of research and thank you for helping me shape who I am today.

I am grateful to all faculty members and staff of the Department of Applied Mathematics and Computational Sciences, PSG College of Technology for their constant words of encouragement.

Finally, but not the least, I would like to share my immense love towards my family. To my parents, thank you for always being there to look up to and thank you for everything you have sacrificed for me. I hope to make you proud and give everything back twofold one day. To my brother, thank you for never ceasing to amaze me. And to all my near and dear friends, thank you for being with me throughout, and I know, will stay with me till the end.

Abstract

Galled networks are a significant yet simple class of phylogenetic networks. They are of biological interest, due to their topological property that reticulate nodes are delineated from each other. The complexity of reconstructing a phylogenetic network on a set of species from an input set of data, as a galled network can be characterized by enumerating the space of galled networks. This prompts the question - 'How many galled networks are there?'.

Motivated by the question, this report aims at answering the same. Based on the decomposition theorem introduced by Gunawan et al. in [1], the notion of a *Pyramid network* is introduced, which aids in the component-wise building of a galled network. Correspondence between a pyramid network and a *Multi-Labelled tree* is established, and built on this, the theorems for counting pyramid networks and hence galled networks are proved. A fixed-parameter tractable algorithm for computing galled networks, loosely based on the FPT algorithm presented in [2], is outlined.

Keywords Galled Networks . Decomposition of Networks . Multi-Labelled Trees . Network Reconstruction . Pyramid Networks

Chapter 1

Introduction

1.1 Theory and Background

Phylogenetics is the study of the evolutionary relationships among various biological entities. Usually, the evolutionary history of a set of species is described by a rooted phylogenetic tree. Despite its relative ease and simplicity in representation, evolutionary events are not always tree-like. *Reticulate* events, such as horizontal gene transfer (HGT), recombination, hybridization, duplication, loss are less suited to be represented and modelled as a tree [3].

One solution to represent reticulate events in the evolutionary history of a set of species is to represent them as phylogenetic networks rather than trees. Phylogenetic networks are directed acyclic graphs which include additional nodes, known as *reticulate* nodes to serve this purpose. Thus, phylogenetic networks, or just *networks* are of interest in order to understand and model

reticulate evolution through time.

But, representing, or more accurately, reconstructing the evolutionary history of a set of species as a network is not easy. The main goal is to reconstruct a phylogenetic network which explains the data provided. But the primary caveat is the handling of reticulate events. In a biological perspective, reticulate events such as HGT, hybridization etc. are rare events and hence considered costly. Explaining the data set, albeit with many such reticulate events might be futile. Considering this, a parsimonious solution is preferred, where the reconstruction of the network explaining the evolutionary history of the set of species based on the input data, minimizes the number of such reticulate events/nodes.

But, this is shown to be NP-Hard [4]. Thus, it is imperative to investigate the tractability of the problem of parsimonious reconstruction of networks. Galled networks, a class of phylogenetic networks are of interest. The topological constraints in limiting the class of galled networks are attractive, as each reticulation event, is in a way, delineated from another. This makes the study of galled networks an appealing prospect.

The space of galled networks, can give a hindsight on the underlying hardness in the process of reconstruction of a galled network from an input data set. This is the motivation behind the main results presented in this report - 'The Counting of Galled Networks'. The next section outlines the organization of the report.

1.2 Organization of the Report

This report is broadly organized into three further chapters. In chapter two, a broad literature review of the theory surrounding phylogenetic networks is provided. The concept of a phylogenetic network is introduced formally and related topics, including the decomposition theorem for phylogenetic networks and various network classes, are introduced. This chapter concludes by providing some properties and results which will be used in the later chapters.

The main results on the counting of galled networks are enlisted in chapter three. Initially, some crucial definitions aiding in the counting process are elaborated. This is followed by the recursive expressions for the counting of galled networks. The chapter concludes by highlighting inferences derived from the main result.

In chapter four, the problem of parsimonious reconstruction of galled networks from input data is outlined. A new approach for this problem, based on the results and intuition from the previous chapter is reported. This approach is loosely based on an existing algorithm detailed in [2].

The report is concluded by summarizing the contents and by shedding light upon the inferences and future scope of the work presented here. Selected material for related and further reading are listed.

Chapter 2

Literature Review

This chapter gives a brief overview of the basic concepts and definitions with regard to phylogenetic networks. Initially, the formal definition of a phylogenetic network is provided along with certain properties associated with them such as node visibility. A decomposition theorem for phylogenetic networks described in [1] is discussed and certain phylogenetic network classes with structural and biological context are detailed. The concept of *Splits* and *Clusters* are introduced, which will aid in Chapter 4.

2.1 Phylogenetic Networks

X is defined to be a set of taxa or species, whose evolutionary history is to be reconstructed. Given a taxa set X, a *Phylogenetic Network* or more simply, a *network* N on X is a directed acyclic graph having (a) a unique node of in-degree zero, known as the *root*, (b) every node of out-degree zero, denoted as a *leaf*, are bijectively labelled to a taxa in X and (c) every internal

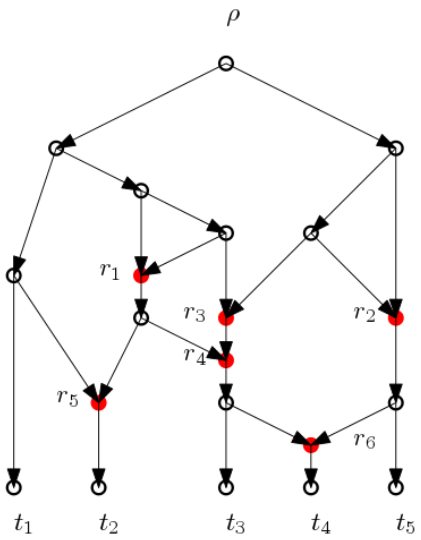

Figure 2.1: A Phylogenetic network N on the taxa set $X = \{t_1, t_2, t_3, t_4, t_5\}$. There are six reticulate nodes, labelled from r_1 to r_6.

node (non root and non leaf) either have in-degree one or out-degree one. An *internal* node is a *tree node*, if it has an in-degree of one or a *reticulate node*, if it has an out-degree of one. For simplicity, the *root* is considered to be a *tree* node. The leaves, tree nodes and reticulate nodes are denoted by $L(N)$, $T(N)$ and $R(N)$ respectively. $V(N)$ represents all the vertices of N, and $E(N)$ the edges.

A node u is the *parent* of a node v if there is a directed edge originating at u and terminating at v. Alternatively in the same context, it can be said that v is a *child* of u. If two nodes u and v share a common parent, they are denoted as *sibling* nodes. More generically, u is an *ancestor* of a node v if there is a directed path from u to v. Equivalently, v can be described as a *descendant* of u. An edge e is defined to be a *tree edge* if it originates from a tree node and a *reticulate edge*, if it originates from a reticulate node.

A network N is said to be a *Phylogenetic tree*, if it contains no reticulate nodes. Also, a network is *binary* if every tree node has out-degree two and every reticulate node has an in-degree equal to two. It can be observed that in every binary network N, $T(N) = L(N) + R(N) - 1$ and $E(N) = 2T(N) + R(N)$ or $E(N) = 2L(N) + 3R(N) - 2$.

Cherry Let N be a phylogenetic network on a taxa set X. A *Cherry* in N is defined as follows. Let $a, b \in X$ and $u, v \in V(N)$ be the nodes (leaves) that are mapped to them respectively. u, v is a *cherry* in N, if there is a node $c \in V(N)$, such that c is the parent of u and v.

Node Visibility A node u of N is said to be a visible ancestor of another node v, if every directed path from the root to v in N passes through u. Also, an internal node u is said to be *visible*, if it is the visible ancestor of some leaf l. In Figure (2.1), internal tree nodes having both children as reticulate nodes are not visible on any leaf t_i. The notion of node visibility plays a significant role in structurally classifying networks into different classes.

Since, each leaf l of a network N is mapped to some taxa $x \in X$, the terms leaf and taxa are used alternatively.

2.2 The Decomposition Theorem

The decomposition theorem for networks was introduced in [1]. Let N be a phylogenetic network with n leaves and r reticulations. It is easy to observe that the removal of every reticulate node from N yields a forest

where the nodes are either tree nodes or leaves. Such a component of the forest is denoted as a *Tree component*. Similarly, removing all the tree nodes and leaves from N produces a forest where each component consists of only reticulate nodes, referred to as a *Reticulate component*. Also, a tree (resp. reticulate) component is *trivial* if it contains only one node.

Decomposition Theorem Let N be a network with $V(N) = n$, $n \geq 1$. The tree components and reticulate components are ordered as $T_1, T_2, ..., T_m$ and $R_1, R_2, ..., R_k$ respectively. Then,

- $m - 1 \leq k \leq m - 1 + L(N)$
- $T(N)$ is the union of all the disjoint tree components of N.
- $R(N)$ is the union of all the disjoint reticulate components of N.

The decomposition theorem thus decomposes or splits the network into disjoint tree or reticulate components.

Network Compression Using the decomposition theorem, a network N can be compressed to produce a new network $\tilde{N}$. First, a reticulate component R_i is compressed as follows. Replace R_i with a single node r_i while redirecting all edges leading to and emanating from all nodes in R_i to r_i. The compression of tree components can be done similarly. It is easy to observe that the graph obtained by compressing every tree and reticulate component $\tilde{N}$ is a network, although it need not be binary.

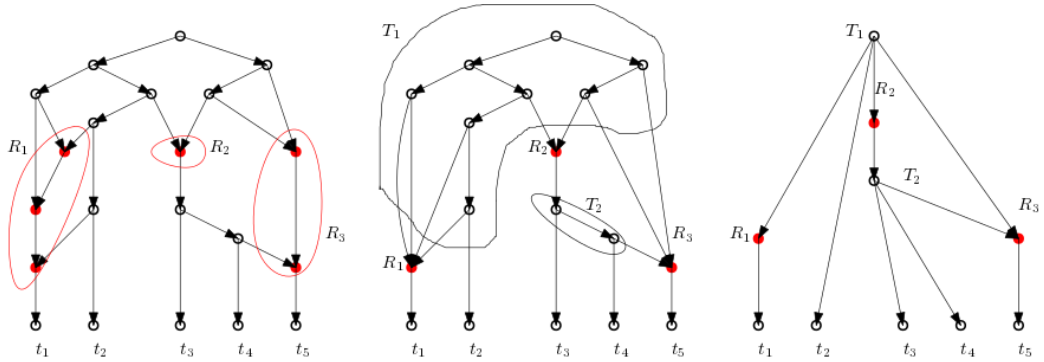

Figure 2.2: The network N on the left has three reticulate node components (R_1, R_2, R_3). These components are compressed to obtain the network $N^{'}$ in the middle. Now, the tree components (T_1, T_2) of $N^{'}$ are identified and compressed to obtain $N^{''}$ on the right. Note that $N^{''}$ has several parallel edges, which have been removed for the purpose of representation.

More specifically, given a network N with tree components $\{\tau_i\}_{1\leq i\leq p}$ and reticulate components $\{\sigma_j\}_{1\leq j\leq q}$, a compressed network $\tilde{N}$ of N is a network with:

- $V(\tilde{N}) = L(N) \cup \{\tau_i^* : 1 \leq i \leq p\} \cup \{\sigma_j^* : 1 \leq j \leq q\}$, where τ_i^* and σ_j^* signify the single node used to relabel the corresponding tree and reticulate node component.

- $E(\tilde{N}) = \{(\kappa_1^*, \kappa_2^*)$, where $\kappa_1^*, \kappa_2^* \in V(\tilde{N})$ and κ_1^* is an ancestor of $\kappa_2^*\}$.

Depth of a Compressed Network Let N be a phylogenetic network and $\tilde{N}$ be its compressed network. The depth or the *Component Depth* of $\tilde{N}$ is defined to be the maximum number of tree node components in a directed path in $\tilde{N}$.

2.3 Network Classes

Based on certain structural properties, a network N can be categorized into one of many possible network classes. Structural characterization of a network class has meaning from a biological perspective also. Certain such network classes, which will be considered further on in the report are detailed here.

Reticulation-Visible Networks Recall the definition of node visibility. A *Reticulation-Visible Network* is a network N where every reticulate node r in N is visible. An observation from a reticulation-visible network is that, every reticulate node has either a leaf or a tree-node as its child. An immediate inference from this fact is that the reticulate node components of N through decomposition has a cardinality of one.

Tree-Child Networks N is a *Tree-Child Network* if every node of N is visible. A direct inference is that every node either has a leaf or a tree node as a child. Hence, there is a path consisting only of tree nodes from any internal node u to some leaf l of N. Also, it can be observed that every *Tree-Child Network* is *Reticulation-Visible*.

Galled Tree A network N is denoted as a *Galled Tree* if each reticulate node r is contained in exactly one unoriented cycle consisting of only tree nodes [2]. Equivalently, every bi-connected component of N contains exactly one reticulate node. An observation from galled trees is that every bi-connected component is node disjoint, and hence, every reticulate node is

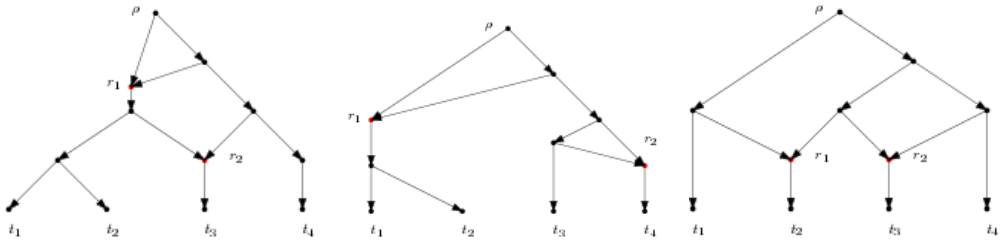

Figure 2.3: A Tree-Child Network (left), A Galled Tree (center) and a Galled Network (right) on the taxa set $X = \{t_1, t_2, t_3, t_4\}$.

delineated from each other. Also, each tree cycle along with the reticulate node in it is denoted as the *gall* of the galled tree.

Galled Networks A *Galled Network* is a network N where, for each reticulate node r there is a tree node u such that there are two node disjoint directed paths consisting of only tree nodes from u to r. Alternatively, every reticulate node r in a galled network N is contained in a tree cycle. Thus, it can be observed that every galled tree is a galled network.

2.4 Splits and Clusters

As outlined briefly in the introductory chapter, the reconstruction of phylogenetic networks from input data requires minimizing the number of reticulation events in history to provide a parsimonious solution. The input data for the reconstruction problem can be in many forms, the main of which are splits and clusters.

Splits Let X be the taxa set, whose evolutionary history is to be reconstructed. A *Split* on X is defined to be a bi-partition of X into two disjoint subsets A and B such that $A \cup B = X$. For the purpose of

representation, a split S on X is denoted as $\frac{A}{B}$. A pair of splits $s_1 = \frac{A}{B}$ and $s_2 = \frac{C}{D}$ are said to be compatible if any one of $A \cap C$, $A \cap D$, $B \cap C$, $B \cap D$ is empty.

Clusters Given the set of taxa X, a cluster C on X is defined to be a non-empty proper subset of X. A cluster is trivial if its cardinality is one. Two clusters c_1, c_2 from the taxa set X are said to be *compatible*, if $c_1 \cap c_2 = \emptyset$ or if $c_1 \in c_2$ or vice-versa.

Realization of Splits and Clusters Let N be a network on X. Let $T = \{T_1, T_2, ..., T_k\}$ be the set of spanning trees in N. A split $S = \frac{A}{B}$ on X, $A, B \subset X$, is realized in N if, for some spanning tree $T_i \in T$, the deletion of an edge e from T_i results in two sub-trees, whose corresponding label sets are A and B.

Similarly, let N be a network on X. A cluster c is realized in N if there is some node u in N such that the sub-graph rooted at u contains the taxa in c. This the *Hardwired* realization of a cluster c. If the cluster is realized by some spanning tree T_i of N, then it said to be realized in the *softwired* sense. It is to be noted that the clusters $Hardwired(N) \subseteq Softwired(N)$.

2.5 Properties and Takeaways

Properties of the network classes mentioned in the previous section with regard to the decomposition theorem and other miscellaneous results are described in this section.

Property 2.4.1 A network N is galled, if both the parents of each reticulate node r belong to the same tree-node component [1].

Property 2.4.2 The compression $\tilde{N}$ of a galled network N is a tree [5].

Property 2.4.3 The number of rooted binary leaf-labelled trees on a taxa set $|X|= n$ is $(2n-3)!!$ [3].

Property 2.4.4 Given an input data set Δ of splits or clusters, a unique phylogenetic tree T exists that realizes Δ if and only if every pair of splits or clusters in Δ are compatible [3].

Chapter 3

Counting Galled Networks

As described in the introduction, reconstruction of a phylogenetic network from input data, while minimizing the number of reticulate events is hard. But much work has not been done to analyze whether the parsimonious reconstruction problem is solvable for restricted classes of network, in this case, the *Galled Network*. An algorithm solving the reconstruction problem for galled networks has been introduced in [2], but it is polynomial time tractable on the size of the input data set Δ and the parameter k, which limits the number of reticulations. Identifying and bounding the space of galled networks can give an insight on the underlying complexity of solving the reconstruction problem for galled networks.

This chapter details on the theory and results surrounding the counting of galled networks. Some preliminary definitions, including that of Multi-Labelled trees and Pyramid networks are provided, which will play a key role in the enumeration of galled networks. The key lemmas and

theorems concerning the same are elaborated in the further sections. For a sense of generality, all networks and trees are considered to be *binary* and *rooted*, unless stated otherwise.

3.1 Multi-Labelled Trees (MUL-Trees)

As described in section 2.1, a *Phylogenetic Tree* T (or just *Tree*), is a phylogenetic network N on a taxa set X with no reticulate nodes. In such a tree T, the set of leaves L are bijectively labelled over X. Let T be a binary tree where $L(T)$, $I(T)$ and $E(T)$ denote its set of leaves, internal nodes and edges respectively. It is known that:

- $|I(T)| = |L(T)| - 1$
- $|E(T)| = |L(T)| + |I(T)| - 1$ or $|E(T)| = 2|L(T)| - 2$

A *Multi-Labelled Tree* or abbreviated as a *MUL-Tree* is a tree T on a taxa set X where it is allowed for more than one leaf to be labelled with the same taxa or label $x \in X$, but each leaf is labelled by a unique taxa. Specifically, a *k-MUL-Tree* is a MUL-Tree where each label appears in at least one and at most k leaves.

2-MUL-Tree A 2-MUL-Tree is a MUL-Tree where each label appears in at least one and at most two leaves. For the set of labels X, X_i is denoted to be the subset of labels which appear i times. Naturally $1 \leq i \leq k$ for a k-MUL tree and in the case of a 2-MUL tree, $i = \{1, 2\}$. The number of edges of a 2-MUL tree T on n labels and k labels appearing twice is $2n + 2k - 2$.

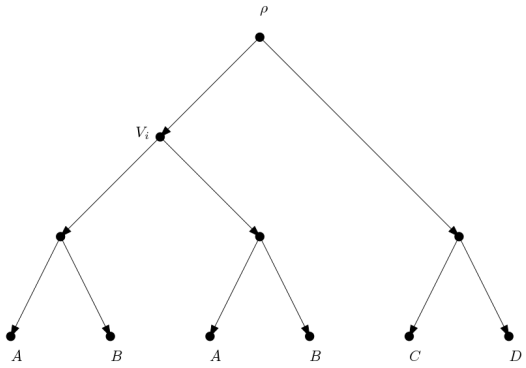

Figure 3.1: A 2-MUL tree on the taxa set $\{A, B, C, D\}$, with A and B labelled twice. Also, the sub-tree below the node V_i is a symmetric component of size two.

In a 2-MUL Tree, a *Twin Cherry* is defined to be a cherry, where both the labels are the same. Also, a 2-MUL Tree is *Twin Cherry Free (TCF)* if it has no twin cherries.

Symmetricity Let T be a 2-MUL tree on n taxa and k taxa labelled twice. T contains a symmetric component if there exists a node v such that its left and right sub-trees rooted at its children u and w respectively are isomorphic. This implies that the same taxa which are present as leaves in the sub-tree at u are the same as that of w and hence, these taxa must be labelled twice. Two edges are defined to be *Corresponding* or *Sibling* edges in a symmetric component if the deletion of either edge results in the same forest. The *Size* of a symmetric component is the number of taxa in it. In a symmetric component of size d, there are $2d - 1$ corresponding or sibling edge pairs. Refer Figure (3.1).

The following lemma outlines the ancestor-descendant relationship among the symmetric components in a 2-MUL-Tree.

Lemma 3.1.1. *Let T be a 2-MUL Tree and u and v be two vertices in T such that there are symmetric components of size d_1 and d_2 rooted at them. Then, u and v and hence the symmetric components, have no ancestor-descendant relationship.*

3.2 Pyramid Networks

A *Pyramid Network* is a galled network with exactly one tree node component. Alternatively, a Pyramid network is a galled network of depth one. Pyramid networks play a significant role in the building and counting of galled networks. It can be observed that each galled network of depth d can be visualized as a galled network of depth 1, each of whose leaves are galled networks of depth at most $d-1$. This observation underlines the role of pyramid networks in recursively constructing galled networks and will be used in the further sections for counting the space of galled networks.

Notation $P(n,k)$ denotes the number of pyramid network with n leaves and k reticulate nodes, where the n leaves are labelled over the taxa set X, $|X| = n$. Also, it can be observed that a reticulate node r appears in a pyramid network only as a parent of some leaf l.

Given a pyramid network P with n leaves and k reticulations, it can observed that, the number of edges $E(P) = 2n + 3k - 2$. A correlation between pyramid networks of size n having k reticulation nodes and 2-MUL trees on n labels and k labels occurring twice can be observed. This is detailed in the following section.

3.3 Correspondence between 2-MUL Trees and Pyramid Networks

A correspondence between pyramid networks and 2-MUL trees can be obtained as follows. A pyramid network P with n leaves and k reticulations can be constructed as a 2-MUL tree with n labels and k labels occurring twice, where each of the k labels are exactly the ones which are a child to the reticulate nodes in P. Also, the 2-MUL tree is *Twin-Cherry Free*. This is shown in the following lemma.

Lemma 3.3.1. *There is a one-to-one correspondence between pyramid networks over X having k reticulate nodes and TCF 2-MUL trees over the same taxa set X with $|X_i| = n - k$ and $|X_2| = k$.*

Proof. Let P be a pyramid network with n leaves and k reticulate nodes. This implies P has a single tree node component τ rooted at the root ρ of P. Also, τ has $n + k - 1$ tree nodes. Now, a 2-MUL tree T is constructed from P as follows:

- **Step 1** Set T to initially be the tree node component τ of P.
- **Step 2** For every node u in τ that has a leaf of label l as a child, a new leaf v is attached to u in T, labelled with the same label l.
- **Step 3** For every pair of nodes w_1, w_2 in τ which have the same reticulate node r_1 as a child, two leaves v_1, v_2 are attached to w_1, w_2 in T and the new leaves are labelled by the taxa l_r, where l_r is the label of the child of reticulate node r.

It can be noted from step 1 that, every tree node with two tree node children in P has an out degree two in T. Moreover, if a tree node has k non-tree-node children, $k \in \{1, 2\}$, then from steps 2 and 3, k new children are attached to them, hence showing that T is binary. Also, a twin cherry is not created in any of the steps mentioned above and hence, the resulting tree is TCF (Twin-Cherry Free).

It is clear that every leaf l_r that is a child of a reticulate node in P is labelled exactly twice in T, from Step 3. Also, leaves of P that are children of a tree node are labelled exactly once in T from Step 2. This shows that T is a 2-MUL tree where the set of taxa labelled twice are those which appear as a child of a reticulate node in P, whose cardinality is equal to k.

This process can be easily reversed to show that a pyramid network can be constructed from a TCF 2-MUL tree, and hence the lemma follows.

□

Thus, it is imperative that the number of pyramid networks on n labels and k reticulations is the number of distinct 2-MUL trees with no twin cherries and having k taxa labelled twice.

3.4 Counting Pyramid Networks

As described in the previous section, pyramid networks can be counted by counting the the number of distinct TCF 2-MUL trees. Let $P(n, k)$ denote the number of distinct TCF 2-MUL trees T over taxa set $X = \{x_1, x_2, ..., x_n\}$,

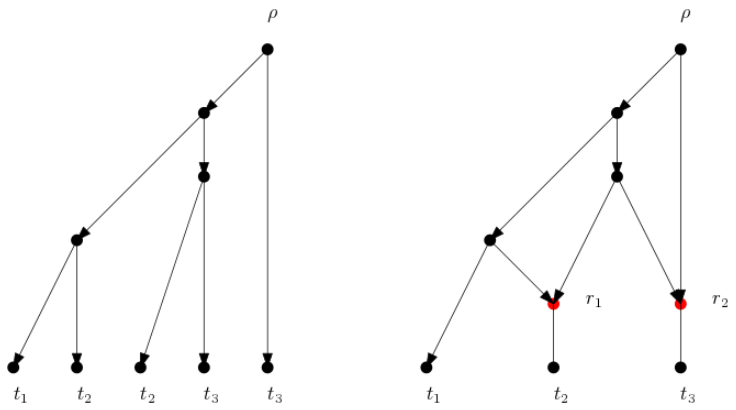

Figure 3.2: A 2-MUL tree on the taxa set $X = \{t_1, t_2, t_3\}$, where the taxa t_2 and t_3 are labelled twice. The corresponding Pyramid network P is provided in the right, on the same taxa set, having two reticulate nodes as parent for the leaves t_2 and t_3.

with $X_2(T) = \{x_1, x_2, ..., x_k\}$ or equivalently, pyramid networks with n leaves and k reticulate nodes. Then, the following lemma holds.

Lemma 3.4.1. *If $P(n,k)$ denotes the number of pyramid networks over X with k reticulate nodes, $|X| = n$, then:*

$$P(n,k) = \binom{n}{k} T(n,k)$$

Proof. There are $\binom{n}{k}$ ways to choose the reticulate leaves. And for each choice of reticulate nodes, $T(n,k)$ ways are there to construct the network. Thus, the lemma follows. □

It still remains to count $T(n,k)$, the number of distinct 2-MUL trees with n labels and k labelled twice. The following lemma counts the same.

Lemma 3.4.2.

$$T(n,k+1) = (n+k-1)T(n,k) + k(T(n,k-1)) - \frac{1}{2}\sum_{d=1}^{k}\binom{k}{d}(2d-1)!!\,[T(n-d+1,k-d) - T(n-d,k-d)] \quad (3.1)$$

where, $T(n,0) = (2n-3)!!$ *and* $k \in \{0,1,2,...,n\}$ *for each* $T(n,k)$.

Proof. If $k = 0$, that implies there are no taxa which are labelled twice. Thus, the number of ways to construct the tree is $(2n-3)!!$.
$T(n,k+1)$ can be obtained from $T(n,k)$ by adding another label for the $k+1^{th}$ taxa. To achieve this, first, a label of the $k+1^{th}$ taxa which is labelled twice in $T(n,k+1)$ is removed, and the operation of adding the $k+1^{th}$ taxa is explored. This can be detailed into five cases:

- **Case 0:** The operation of removing the $k+1^{th}$ taxa cannot be performed from $T(n,k+1)$ since the removal of either label results in a twin-cherry in $T(n,k)$. Thus there is no tree in $T(n,k)$ which generates the tree in $T(n,k+1)$. Such cases are denoted as C_0.

- **Case 1:** Only one label of the $k+1^{th}$ taxa can be removed as the removal of the other label results in a twin-cherry in $T(n,k)$. Another possibility is when both the labels of the $k+1^{th}$ taxa are present in the corresponding edges of the same symmetric component. This means in either possibility, there is only one tree in $T(n,k)$ which when added with the second label of the $k+1^{th}$ taxa generates the required tree. This case is denoted by C_1.

- **Case 2:** There are two sub-cases here.
  - **2a:** The first case is when either label of the $k+1^{th}$ taxa can be removed from $T(n,k+1)$ without any conflict. This implies that the removal of either of the $k+1^{th}$ label does not create a symmetric component in $T(n,k)$. Thus, there are two trees in $T(n,k)$, which when added with the second label of the $k+1^{th}$ taxa, generates the required tree in $T(n,k+1)$. This case is hence counted twice and represented by C_2.
  - **2b:** The second case is when the removal of one label of the $k+1^{th}$ taxa results in a twin-cherry and the removal of the other label results in a symmetric component with d labels. There are two ways to reconstruct $T(n,k+1)$ from $T(n,k)$ by adding the $k+1^{th}$ label to either of the corresponding symmetric edge pairs. This case is denoted as C_2^*.
- **Case 3:** This case occurs when removal of one of the $k+1^{th}$ label in $T(n,k+1)$ results in a symmetric component of size d and removal of the other does not result in a symmetric component in $T(n,k)$. Thus, addition of the $k+1^{th}$ label to three trees in $T(n,k)$ can generate the same tree in $T(n,k+1)$ and is hence counted thrice. This case is denoted as C_3.
- **Case 4:** The final case occurs when removal of either occurrence of the $k+1^{th}$ label in $T(n,k+1)$ results in two difference symmetric components with d_1 and d_2 labels respectively. Thus four trees in

$T(n,k)$ can generate the same tree in $T(n,k+1)$. This case is denoted by C_4.

Now, equation (3.1) can be rephrased as follows:

$$T(n,k\ +\ 1)\ \ =\ \ (n\ +\ k\ -\ 1)T(n,k)\ +\ C_0\ +\ \frac{1}{2}(C_1\ -\ C_3\ -\ 2C_4) \quad (3.2)$$

There are $2n+2k-2$ ways to attach the second copy of the $k+1^{th}$ taxa in $T(n,k)$, including an incoming edge to the root and excluding the pendant edge leading to the existing label of the $k+1^{th}$ taxa. Each case C_i mentioned earlier is counted i times in this expression. Since, case 2 accounts for the majority of the count of pyramid networks, it is not counted and obtained by subtracting the remaining cases from the total and dividing by two to counter the fact that it is generated twice. But sub-case C_2^* is essential in counting some remaining cases. The rephrased expression (3.2) thus follows. It still remains to count the number of cases for each C_i, $i \in \{0,1,3,4\}$.

- **Counting C_0:**

$$\binom{k}{2} T(n-1,k-2)$$

Both labels of the $k+1^{th}$ taxa are attached to twin-cherries. There is only one unique way of attaching the same. Out of the k taxa labelled twice (removing both labels of the $k+1^{th}$ taxa), two taxa are choosed and assumed to be leaves labelled once. The resulting tree has $n-1$ taxa and $k-2$ taxa labelled twice and the expression hence follows.

- **Counting C_2^*:**

$$k\sum_{d=2}^{k-1}\binom{k-1}{d}(2d-1)!!\,T(n-d,k-d-1)$$

Out of the k remaining taxa labelled twice, one is chosen as a twin-cherry. A symmetric component of size d must be constructed from the rest of the $k-1$ taxa. There are $(2n-3)!!$ ways to construct a symmetric component (here a tree with d leaves) and $(2k-1)$ ways to attach the $k+1^{th}$ taxa to it. The remaining tree contains $n-d$ taxa and $k-d-1$ taxa labelled twice. This is summed over all possible values of d.

- **Counting C_1:**

$$\sum_{d=2}^{k+1}\binom{k}{d-1}(2d-3)!!\,T(n-d+1,k-d+1)+k.T(n,k-1)-2C_0-C_2^*$$

The number of cases where both the labels of the $k+1^{th}$ taxa are present in the corresponding edges of a symmetric component are enumerated in the first term. $k(T(n,k-1))$ counts the number of trees where one of the labels of the $k+1^{th}$ taxa is in a twin-cherry. But this count includes the counts of C_2^* and twice of C_0 (Since the either twin-cherry in C_0 can be chosen first). The expression thus follows.

- **Counting C_4:**

$$\frac{1}{2}\sum_{d_1=2}^{k-2}\sum_{d_2=2}^{k-d_1}\binom{k}{d_1 d_2}(2d_1-1)!!\,(2d_2-1)!!\,T(n-d_1-d_2+2,k-d_1-d_2)$$

Both the labels of the $k+1^{th}$ taxa are attached to symmetric components in $T(n,k)$, of size d_1 and d_2. $\binom{k}{d_1 d_2}$ counts the number of ways to choose the taxa for the two symmetric components and each has $(2d_i-3)!\,!\,(2d_i-1)$ ways of constructing and attaching the label of the $k+1^{th}$ taxa ($i \in \{1,2\}$). Considering the symmetric components as leaves, $T(n-d_1-d_2+2, k-d_1-d_2)$ shows the number of parent trees to which the symmetric components can be attached. The case of double count when set of taxa in d_1 and d_2 are interchanged is eliminated by halving. This is summed over all possible values of d_1 and d_2.

- **Counting C_3:**

$$\sum_{d=2}^{k} \binom{k}{d} (2d-1)!\,!\,T(n-d+1, k-d) - 2C_4 - C_2^*$$

The approach for counting is as follows. First count the number of trees where one label of the $k+1^{th}$ taxa is in a symmetric component of size d. This count also includes the counts of C_2^* once and C_4 twice (either component can be chosen first). $\sum_{d=2}^{k} \binom{k}{d} (2d-1)!\,!\,T(n-d+1, k-d)$ gives the number trees with one label of $k+1^{th}$ taxa in a symmetric component, summed over all sizes of d. The expression then follows.

	k								
n	0	1	2	3	4	5	6	7	8
1	1								
2	1	1	3						
3	3	6	20	87					
4	15	45	189	993	6249				
5	105	420	2160	13407	97182	804585			
6	945	4725	28875	207135	1701855	15738765	161685045		
7	10395	62370	442260	3603915	33121890	338588685	3808469970	46726507485	
8	135135	945945	7640325	69757065	709428825	7946584695	97162333695	1287228175065	18363976595055

Table 3.1: $T(n,k)$ values for $1 \leq n \leq 8$ and $0 \leq k \leq 8$.

Substituting the expressions of each C_i, $i \in \{0,1,3,4\}$ in (3.2) and resultant simplification gives (3.1), hence proving the lemma.

□

Table 3.1 lists the values of $T(n,k)$. For each cell in Table 3.1, the number of pyramid networks is equal to $\binom{n}{k}T(n,k)$ for the corresponding values of n and k.

3.5 Counting Galled Networks

In the previous section, the recursive expression for counting pyramid networks given number of leaves and reticulations has been derived. Since, pyramid networks are galled networks with a single tree node component, or alternatively, with depth one, they can be counted as follows:

Lemma 3.5.1. *The number of galled networks of depth one G_n^1 is given by:*

$$G_n^1 = \sum_{k=0}^{n} P(n,k)$$

where $P(n,k)$ is the number of pyramid networks with n leaves and k reticulations.

Proof. $P(n,k)$ counts the number of pyramid networks on n leaves and k reticulations. The total number of galled networks with depth one is obtained by summing $P(n,k)$ values for all possible values of k, ranging from $0 \leq k \leq n$. □

An observation on galled networks is that, each galled network G_n^d, on n leaves and having depth at most d, can be visualized as a galled network of depth one, whose leaves are galled networks of depth at most $d-1$.

For this purpose, denote $h_n^d(m)$ to be a forest of n leaves and m components, such that each component is a non-trivial galled network of depth at most d. By non-trivial, the size of each component is at least two. The following lemma for counting all such $h_n^d(m)$ holds.

Lemma 3.5.2.

$$h_n^d(m) = \sum_{\substack{y_1,y_2,...,y_m \text{ s. t.} \\ y_1+y_2+...+y_m=n \\ y_1 \geq y_2 \geq .... \geq y_m \geq 2}} \left[\frac{\binom{n}{y_1,y_2,...,y_m}}{\prod_{j=1}^{n} \mu_j!} \prod_{i=1}^{m} G_{y_i}^d \right]$$

where $\mu_j = |\{i : y_i = j\}|$.

Proof. First, partition n into m components, each of size $y_1, y_2, ..., y_m$ such that the total amounts to n. Also, $y_1 \geq y_2 \geq ... \geq y_m \geq 2$. It is easy to see that the number of ways to do the above is $\binom{n}{y_1,y_2,...,y_m} / \prod_{j=1}^{n} \mu_j!$. This is equivalent to placing n unlabelled objects in m bins, each of size

$y_i, i \in \{1, 2, ..., m\}$. $\mu_j = |\{i : y_i = j\}|$ denotes the number of bins of size j. The number of ways to construct each component is given by $G^d_{y_i}$ and hence the lemma follows. □

The following theorem provides the recursive function which generates and counts all possible galled networks of size n.

Theorem 3.5.3.

$$G^d_n = G^1_n + \sum_{\substack{2 \le n' \le n, \\ 1 \le m \le \lfloor \frac{n'}{2} \rfloor \\ m \le r \le n-n'+m}} \binom{n}{n'} h^{d-1}_{n'}(m) \frac{\binom{n-n'}{r-m}}{\binom{n-n'+m}{r}} P(n-n'+m, r)$$

Proof. Let τ_0 denote the topmost tree-node component of G^d_n. If the depth $d = 1$, then, there are G^1_n ways to construct the galled network. Otherwise, construction of a galled network can be done in two steps.

First, a forest of galled networks, having a maximum depth of $d-1$ has to be constructed. This forest will consist of all tree-node components of G^d_n excepting τ. Let this forest contain m components and $n^{'}$ out of the n leaves. The number of ways to construct this equals $\binom{n}{n'} h^{d-1}_{n'}(m)$.

Next, each component of this forest is considered as a leaf for the topmost component. Thus, τ_0 has m leaves in addition to the $n - n^{'}$ leftover leaves. But, there is a constraint that all the m new leaves must be a child of a reticulate node.

The number of galled networks of depth one (pyramid networks) with $n - n^{'} + m$ leaves and r reticulate nodes is given by $P(n - n^{'} + m, r)$ (Lemma 3.4.1). It is necessary that $m \le r \le n - n^{'} + m$. Out of all these pyramid

	d					
n	1	2	3	4	5	6
1	1					
2	6	6				
3	168	240	240			
4	11550	19350	20502	20502		
5	1448370	2581230	2845950	2868990	2868990	
6	286250580	516934800	579614760	588577320	589130280	589130280

Table 3.2: Number of galled networks with n leaves and depth d, $n \leq 6$ and $1 \leq d \leq 6$.

networks, the fraction $\binom{n-n'}{r-m}\binom{m}{m}/\binom{n-n'+m}{r}$ of networks satisfy the condition that the corresponding m new leaves have a reticulation node as a parent. The theorem follows by taking a sum over all possible values of n', m and r. □

Table 3.2 lists the counts of galled networks, for values of $n \leq 6$. d refers to the maximum depth of the galled networks, with depth d signifying all galled networks with depth less than or equal to d.

Based on the first few values provided in the tables 3.1 and 3.2, it is imperative that a search of the space of galled networks, for identifying the most parsimonious or optimum network explaining the input data, is futile even for small values of n. Hence, as presented in the following chapter, emphasis is given to the identification of optimal galled networks explaining the input data.

Chapter 4

Reconstruction of Galled Networks

For an input data set on a collection of taxa or species, constructing an evolutionary model, or particularly, a phylogenetic network explaining the data is a fundamental problem in phylogenetic analysis. By the term 'explain', it is required of the network to highlight the possible speciation, mutation and reticulation events such as hybridization that could have possibly occurred in history among the considered species [2]. In a biological perspective, a reticulate event is considered costly in an evolutionary history, considering the rarity of such events through time. Hence naturally, it is required that any model or network explaining and input data set over a set of taxa must include as less reticulate events as possible, and this solution is known to be the parsimonious solution to the problem of phylogenetic network reconstruction [3].

But, the reconstruction of phylogenetic networks from input data while minimizing the number of reticulate events in history is NP-Hard [4]. Hence, as mentioned in earlier sections, emphasis has been given on the problem of reconstructing particular network classes from input data. An algorithm, to reconstruct galled networks from an input data set Δ has been introduced in [2]. The algorithm is polynomial time tractable on the size of the input data set Δ and the parameter k limiting the number of reticulate nodes.

The algorithm proceeds by identifying and constructing each bi-connected component of the galled network, with the parameter k limiting the number of reticulate nodes per bi-connected component. The network N is constructed by identifying a backbone tree on a subset of the taxa and modifying it by adding each reconstructed bi-connected component. This algorithm uses a decomposition theorem[1] which states that every split/cluster realized from the edges and nodes of a bi-connected component respectively are incompatible with each other (refer [2]).

In this chapter, a new approach for reconstructing galled networks from an input data set is presented. The approach is loosely based on the algorithm mentioned earlier, but uses the fundamental idea of counting galled networks introduced in the previous chapter. The run-time is still polynomial time tractable on the size of the input data set Δ and the parameter k limiting the number of reticulate nodes.

[1]Different from the one mentioned in section 2.2

4.1 Computing Pyramid Networks

Pyramid networks have a single tree-node component. Given an input set of clusters C, it is possible to identify the subset of clusters which can be realized in a pyramid network.

Ideal Let C be an input set of clusters on the taxa set X. A cluster $c \in C$ is said to be an *ideal*, if every other cluster $c^{'}$ in C is either $c^{'} \subseteq c$ or $c \subseteq c^{'}$. Naturally, the root cluster ρ and the trivial clusters (clusters of cardinality one) are ideals.

Observation: Let c_i be a non-trivial ideal in C. The set of clusters $C_i^{'} = \{c_s : c_s \in C \text{ and } c_s \subseteq c_i\}$ are realized in a pyramid network if none of c_s is a non-trivial ideal. Also, the root of the pyramid network realizes c_i as the cluster and taxa set $X_{C_i^{'}}$ is restricted on the labels in $C_i^{'}$. This observation is exploited in the next section in reconstructing galled networks.

First, the following algorithm outlines the process of construction of a pyramid network from the set of clusters $C_i^{'}$ on the restricted taxa set X_i. This is loosely derived from [2].

Algorithm 1 Computing Pyramid Networks from Clusters

Data: Input set of clusters $C_i^{'}$ on the taxa set X_i

Result: Pyramid Network explaining $C_i^{'}$ minimizing the number of reticulate nodes, if exists

```
for k = 0, 1, ..., K do
    for each possible choice of a subset R ⊂ X_i of size k do
        if C'_{X_i/R} is compatible then
            Build a rooted backbone tree that realizes C'_{X_i/R}
            for each r ∈ R do
                Let B represent the set of all edges bridged by r in C'_{X_i/R}
                if B is contained in a path then
                    attach r to the two end edges of the shortest path
                else
                    Try the next choice of R
                end
            end
        else
    end
    if all r ∈ R can be attached and the resulting network realizes C' then
        return N
    else
        Try the next choice of k.
    end
end
return fail
```

The parameter K limits the number of reticulate nodes per pyramid network. Initially, the maximum subset of taxa from X_i which is compatible

is selected. $C'_{X_i/R}$ refers to the clusters in C' restricted to the taxa set X_i/R. Hence, it is possible to construct a tree with the constrained taxa set. This tree is denoted as the rooted backbone tree T. It just remains to add the remaining K taxa, in R to the backbone tree. An edge $e = (u, v) \in T$ is said to be *bridged* by the addition of the new taxa r if $c_u \cup \{r\}$ and $c_v \cup \{r\}$ is in $C'_{X_i/(R-\{r\})}$. c_u and c_v denote the clusters realized at the nodes $u, v \in T$. Each taxa r is attached to encompass all the edges in T that are bridged [2]. The algorithm fails if the pyramid network cannot be constructed. Else, the first pyramid network, which is generated is returned.

Lemma 4.1.1. *Algorithm 1 computes a minimal pyramid network explaining the input set of clusters C' using at most K, $0 \leq K \leq |X_i|$ reticulate nodes if exists, or returns fail, in polynomial time.*

Proof. The number of subsets of size at most K, $\sum_{k=1}^{K} \binom{|X|}{k}$ is polynomial in the size of $|X|$. The backbone tree can be constructed in polynomial time in the size of $C'_{X_i/(R)}$ and for each reticulate node r to be added to the backbone tree, the operation is polynomial in $|X|$ and $|C'|$. The first pyramid network explaining the input data is returned. □

4.2 Computing Galled Networks

The elementary observation in counting galled networks is that a galled network can be recursively constructed in a bottom-up manner using pyramid networks. Given an input set of clusters C, first, identify all the ideals c_i in C. The next task is to identify the ideals and the subset of clusters, which

form a pyramid network at the lowest level. Once these pyramid networks are reconstructed, each of them can be considered as a leaf, and hence, enabling the construction the next level of pyramid networks based on the remaining ideals in C. This process is continued until the root cluster ρ is reached, hence reconstructing the galled network in the process.

The following algorithm outlines the construction of galled networks from clusters based on the intuition provided above:

Algorithm 2 Computing Galled Networks from Clusters

Data: Input set of clusters C on the taxa set X

Result: Minimal Galled Network explaining C if exists

for *every non-trivial ideal* c_i *of increasing cardinality* **do**
    Construct the pyramid network P realizing c_i and all clusters $c_s \in C$ such that $c_s \subseteq c_i$.
    if *Success* **then**
        Remove all c_s from C. Consider c_i to be a new leaf.
    else
        return **fail**.
    end
end

if *all ideals* c_i *can be explained until the root cluster* ρ **then**
    return the constructed galled network N.
else
    return **fail**.
end

All the ideals are handled in their order of increasing cardinality. The ideals and the subset of clusters which form pyramid networks are reconstructed first. Thus, when the other ideals are encountered, they will form a pyramid network, since the pyramid sub-components beneath them can be considered as a set of new leaves. The *root cluster* is the final ideal to be encountered in the process of reconstruction.

Lemma 4.2.1. *Algorithm 2 returns a galled network G explaining the input data set C on X, if exists, in polynomial time.*

Proof. The ideal clusters in C can be identified in polynomial time corresponding to $|C|$. The number of ideal clusters i ranges from $1 \leq i \leq |C|$. For each ideal and associated clusters, a pyramid network can be constructed in time polynomial in $|C|$, $|X|$ and the number of reticulate nodes K in the pyramid network (refer Algorithm 1). Thus, the run-time of the algorithm is polynomial in the size of the input set of clusters C and the the cardinality of the taxa set X and hence the lemma follows. □

From a practical standoff, this approach gives a bottom-up method to reconstruct galled networks from input data. This does not improve the efficiency of the algorithm detailed in [2] but gives a more intuitive approach for handling the same.

Chapter 5

Conclusion and Future Work

The main results of the report are presented in chapter 3, which is preceded by an introduction to phylogenetic networks and related definitions and properties, including the decomposition theorem for networks and the definitions of splits and clusters. Counting of galled networks, is aided by the concept of a *Pyramid network*, a galled network with a single tree-node component. The notion of a multi-labelled (MUL) tree is introduced and the correspondence between a 2-MUL Tree and a pyramid network is established. This is used to derive the expression to count pyramid networks, with n leaves and k reticulations (Lemma (3.4.2)). The observation that galled networks can be recursively built up using pyramid networks is exploited to derive the expression to count galled networks, provided in theorem (3.5.3).

Chapter 4 deals with the problem of reconstruction of galled networks from input data. The algorithm presented is loosely based on the algorithm presented in [2] but it uses the intuition behind pyramid networks and

counting galled networks to provide a bottom-up approach in reconstruction.

Counting galled networks gives an insight to the complexity of searching for the optimum network in the space of galled networks. It still is unclear on the hardness of network reconstruction, when the network is constrained classes such as galled networks. Investigation of the above, including network classes such as *reticulation-visible* and *tree-child* networks is a good future scope. This can lead to the designing of better algorithms aiding in the parsimonious reconstruction of phylogenetic networks.

List of Figures

List of Algorithms

List of Tables

Bibliography

[1] A. D. M. Gunawan, B. DasGupta, and L. Zhang, "A decomposition theorem and two algorithms for reticulation-visible networks," *Information and Computation*, vol. 252, pp. 161–175, 2017.

[2] D. Huson and T. Klöpper, "Beyond galled trees-decomposition and computation of galled networks," in *Research in Computational Molecular Biology*, pp. 211–225, Springer, 2007.

[3] D. H. Huson, R. Rupp, and C. Scornavacca, *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge University Press, 2011.

[4] L. Wang, K. Zhang, and L. Zhang, "Perfect phylogenetic networks with recombination," vol. 8, no. 1, pp. 69–78, 2001.

[5] A. D. Gunawan, H. Yan, and L. Zhang, "The compressions of reticulation-visible networks are tree-child," *arXiv preprint arXiv:1806.07625*, 2018.

[6] É. Czabarka, P. Erdős, V. Johnson, and V. Moulton, "Generating functions for multi-labeled trees," *Discrete Applied Mathematics*, vol. 161, no. 1-2, pp. 107–117, 2013.

[7] D. H. Huson, T. Klöpper, P. J. Lockhart, and M. A. Steel, "Reconstruction of reticulate networks from gene trees," in *Research in Computational Molecular Biology* (S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P. A. Pevzner, and M. Waterman, eds.), (Berlin, Heidelberg), pp. 233–249, Springer Berlin Heidelberg, 2005.