RICE UNIVERSITY

Scalable Methods for Phylogenetic Network Inference

by

Jiafan Zhu

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Dr. Luay K. Nakhleh (Chair),
J.S. Abercrombie Professor,
Computer Science, Rice University

Dr. Christopher Jermaine,
Professor,
Computer Science, Rice University

Dr. Todd Treangen,
Assistant Professor,
Computer Science, Rice University

Dr. Meng Li,
Noah Harding Assistant Professor,
Statistics, Rice University

Houston, Texas
May, 2019
Abstract

Scalable Methods for Phylogenetic Network Inference

by

Jiafan Zhu

Reticulate evolutionary histories, such as those arising in the presence of hybridization, are best modeled as phylogenetic networks, which take the shape of rooted, directed, acyclic graphs. Recently developed methods allow for statistical inference of phylogenetic networks while also accounting for other evolutionary processes, such as incomplete lineage sorting (ILS). These methods use two different types of input: unlinked bi-allelic markers (e.g., single nucleotide polymorphism data), and sequence alignments of multiple, unlinked loci. While these methods have good accuracy in terms of estimating the network and its parameters, likelihood computations and convergence remain major computational bottlenecks and limit the methods’ applicability and scalability.

The contributions of this thesis are threefold. First, I explore the challenge with viewing a phylogenetic network as an underlying phylogenetic tree with an additional set of “horizontal” edges. Furthermore, I demonstrate why likelihood computations of networks take orders of magnitude more time when compared to trees. Second, I develop an approach for inference of phylogenetic networks based on pseudo-likelihood using bi-allelic markers. I demonstrate the scalability and accuracy of phylogenetic network inference via pseudo-likelihood computations on simulated data, and I demonstrate aspects of robustness of the method to violations in the underlying assumptions of the employed statistical
model. Third, I introduce a novel divide-and-conquer method for scalable inference of phylogenetic networks from the sequence data of multiple, unlinked loci. The method infers networks on subproblems and then merges them into a network on the full set of taxa. To reduce the number of subproblems on which to infer subnetworks, a Hitting Set version of the problem of finding a small number of subsets is formulated, and a simple heuristic is implemented to solve it. I demonstrate the performance of the two-step algorithm, in terms of both running time and accuracy, on simulated as well as on biological data sets. The divide-and-conquer method accurately infers phylogenetic networks at a scale that is infeasible with existing methods.

I implemented and made available to the community all the algorithms in the publicly available software package PhyloNet. The contributions of my thesis provide a significant and promising step towards accurate, large-scale phylogenetic network inference.
Acknowledgments

First, I would like to thank my advisor Dr. Luay Nakhleh. He is a great advisor as well as a great collaborator. In these 4 years, he did not only teach me the domain knowledge of phylogenomics, but also improved my skills of communication, presentation, writing, and all other things a PhD candidate should learn. I came to the Rice CS Bioinformatics Group with little experience about research, and I had never published a paper before. He guided me to start my research from scratch and inspired me to explore at the boundary of human knowledge. The spirit and the methodology of research will benefit me a lot during my lifetime.

Then, I would like to thank my committee members: Dr. Christopher Jermaine, Dr. Todd Treangen and Dr. Meng Li. They spent their precious time to attend my talks, and they provided many valuable comments on my PhD proposal, which made my thesis work much better.

Also, I would like to thank Dr. Ron Goldman. He played a significant part for me to be admitted into the PhD program of Rice University. More importantly, I met my wife in his research group and started my romance.

Furthermore, I would like to thank two previous group members, Dr. Yun Yu and Dr. Dingqiao Wen, for their collaboration and great work that formed the basis for many of my contributions. As Isaac Newton had said, I can see further because I stand on the shoulders of giants. With their help, I was able to start my research immediately soon after I joined this group. I built my research and publications on top of their publications and projects as forerunners. I would like to thank all my collaborators, Dr. Huw Ogilvie, Ryan Leo
Elworth, and Xinhao Liu. My thesis would not have been possible without them. Also I would like to thank all other group members for their friendship and encouragement.

Last but not least, I would like to thank my wife Juan Du, all my family members and all my friends for their love and support.

Work in this thesis has been supported by NSF grants DBI-1355998, CCF1302179, CCF-1514177, CCF-1800723, and DMS-1547433. This thesis was supported in part by the Big-Data Private-Cloud Research Cyberinfrastructure MRI-award funded by NSF under grant CNS-1338099 and by Rice University.
Contents

Abstract ................................................................. ii
Acknowledgments .................................................... iv
List of Figures ........................................................ xx
List of Tables .......................................................... xxii

1 Introduction .......................................................... 1
   1.1 Contributions of the thesis ................................. 2

2 Background .......................................................... 5
   2.1 Phylogenetic networks ........................................ 5
   2.2 Inference under the MSNC model ......................... 6

3 In the Light of Deep Coalescence: Revisiting Trees Within Networks ..................... 8
   3.1 Methods ......................................................... 11
      3.1.1 Deep coalescence and the parental trees inside a network .......................... 12
      3.1.2 Inheritance probabilities and the multispecies network coalescent ................. 15
   3.2 Results and discussion ....................................... 16
      3.2.1 Phylogenetic networks and anomalies ...................... 16
3.2.2 On the backbone tree of a phylogenetic network .................................. 20
3.2.3 From gene trees to species networks via parental trees: A clustering approach ................................................................. 28
3.3 Discussion ........................................................................................................... 32

4 Inference of Species Phylogenies from Bi-allelic Markers Using Pseudo-likelihood 35
4.1 Methods ................................................................................................................ 35
4.1.1 When it comes to computational complexity, a network is not merely a tree with a few additional reticulations ........................................ 36
4.1.2 Pseudo-likelihood ............................................................................................. 41
4.1.2.1 Searching the phylogenetic network space ............................................... 43
4.2 Results .................................................................................................................. 43
4.2.1 Simulations ....................................................................................................... 43
4.2.2 Comparison to full likelihood computation ...................................................... 49
4.2.3 The effect of number of individuals sampled ................................................ 51
4.2.4 Robustness of inference to lack of independence ........................................... 54
4.2.5 The effect of limitation on number of reticulations ....................................... 56
4.2.6 Empirical data sets .......................................................................................... 58
4.3 Discussion ............................................................................................................. 62

5 A Divide-and-Conquer Method for Scalable Phylogenetic Network Inference from Multi-locus Data 65
5.1 Two related works ............................................................................................... 65
5.2 Method .................................................................................................................. 66
5.2.1 Reconciling and summarizing the subnetworks . . . . . . . . . . . 67

5.2.2 Generating a starting network and an order for leaf addition . . . 68

5.2.3 Iterative attachment of new taxa . . . . . . . . . . . . . . . . . . . 70

5.2.4 Asymptotic time complexity . . . . . . . . . . . . . . . . . . . . . 76

5.2.5 Reducing the number of subproblems . . . . . . . . . . . . . . . . . 76

5.2.6 An example of the merger algorithm . . . . . . . . . . . . . . . . . 82
  5.2.6.1 Reconciling and summarizing the subnetworks . . . . . . 85
  5.2.6.2 Generating a starting network and an order for leaf addition 88
  5.2.6.3 Adding B . . . . . . . . . . . . . . . . . . . . . . . . . 90
  5.2.6.4 Adding C . . . . . . . . . . . . . . . . . . . . . . . . . 92

5.3 Performance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
  5.3.1 Accuracy of the merger algorithm . . . . . . . . . . . . . . . . . . 96
  5.3.2 Accuracy on simulated multi-locus data sets . . . . . . . . . . . . 100
  5.3.3 Inference on an empirical data set . . . . . . . . . . . . . . . . . . 113

5.4 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 116

6 PhyloNet Implementation and Usage 117
  6.1 MLE_BiMarkers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
    6.1.1 Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
    6.1.2 Usage . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
      6.1.2.1 Search settings . . . . . . . . . . . . . . . . . . . . . . . . 118
      6.1.2.2 Inference settings . . . . . . . . . . . . . . . . . . . . . . . . 119
      6.1.2.3 Start State Settings . . . . . . . . . . . . . . . . . . . . . . . . 121
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.2.4</td>
<td>Data related settings</td>
<td>121</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Example</td>
<td>122</td>
</tr>
<tr>
<td>6.2</td>
<td>NetMerger</td>
<td>123</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Requirements</td>
<td>123</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Usage</td>
<td>124</td>
</tr>
<tr>
<td>6.2.2.1</td>
<td>The mode to reduce the number of trinets</td>
<td>124</td>
</tr>
<tr>
<td>6.2.2.2</td>
<td>The mode to divide the data set into subsets</td>
<td>125</td>
</tr>
<tr>
<td>6.2.2.3</td>
<td>The mode to merge subnetworks</td>
<td>126</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Example</td>
<td>127</td>
</tr>
<tr>
<td>6.2.3.1</td>
<td>Reduce the number of trinets</td>
<td>127</td>
</tr>
<tr>
<td>6.2.3.2</td>
<td>Divide data set into subsets</td>
<td>127</td>
</tr>
<tr>
<td>6.2.3.3</td>
<td>Merge subnetworks</td>
<td>128</td>
</tr>
</tbody>
</table>

7 Conclusions 130
List of Figures

3.1 (a) A phylogenetic network $\Psi$ with one reticulation node and (b-c) the two trees it displays. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

3.2 (a) The genealogy of a gene (blue lines) within the branches of a phylogenetic network. In this scenario, the two lineages from B and C failed to coalesce prior to the reticulation node (evolution is viewed backward in time, from the leaves toward the root). The resulting genealogy in this case is $((a,b),(c,d))$ and neither of the two trees in the set $U(\Psi)$ (shown in Fig. 3.1) capture this scenario. The length in coalescent units of the branch between the reticulation node and the MRCA of B and C is $y$. (b) An abstract representation of the network, assuming both reticulation edges have the same length $w$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.3 (a) The MUL-tree of the phylogenetic network $\Psi$ in Fig. 3.1(a). (b-e) The four trees that constitute set $W(\Psi)$, assuming one individual is sampled per species. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
3.4 The most likely gene tree given the phylogenetic network in Figure 3.2(b) (with \( w = 0 \)) with \( \gamma = 0.5 \) (left) and \( \gamma = 0.05 \) (right). The x-axis corresponds to branch length \( x \) and the y-axis corresponds to branch length \( y \). Yellow: gene tree \(( (a, d), (b, c) )\); Orange: gene trees \(( a, (b, (c, d)) )\) and \(( a, (c, (b, d)) )\); Brown: gene trees \(( (a, b), (c, d) )\) and \(( (a, c), (b, d) )\); Blue: gene tree \(( a, ((b, c), d) )\) in both panels, and gene tree \(( (a, (b, c)), d )\) additionally in the left panel. 20

3.5 Gene tree distribution for the phylogenetic network in Figure 3.2(b) \(( w = 0 )\) as \( x \to \infty \), for \( \gamma = 0.5 \) (left) and \( \gamma = 0.05 \) (right). The x-axis corresponds to branch length \( y \) and the y-axis corresponds to the probability of each gene tree topology (see Table 3.2). 24

3.6 The accuracy of three methods for inferring species trees in the presence of reticulation on data generated on the phylogenetic network of Fig. 3.2(b). Left column corresponds to setting \( \gamma = 0.5 \) and right column corresponds to setting \( \gamma = 0.05 \). Four settings for branch length \( y \) (on the \( x \)-axis) were used, and for each setting data sets with 25, 50, 100, and 200 loci were generated. See the text for definition of the accuracy measure. (Top) ASTRAL-II [MW15]; (Middle) The method of Steel \textit{et al.} [SLHS13] as implemented in Dendroscope [HS12]; (Bottom) Our own implementation of the method of Daskalakis and Roch [DR16]. 27
3.7 Performance of the clustering approach on the simulated data as a function of the number of gene trees. (a) The phylogenetic network used in the simulations. The lengths of the two reticulation edges were set to 0. The length of the edge going out of the reticulation node was set to 0.2. The inheritance probability of the left reticulation edge was set to 0.35. Ten networks were generated from this network by setting the length of each other internal branch to a random number uniformly sampled in the range [0.7, 1.3]. (b) The number of clusters identified (averaged over 300 data sets for each bar). (c) The number of correctly inferred parental trees (out of the maximum of four parental trees). (d) The error between the set of inferred trees from the identified clusters and the set of four parental trees of the network. The x-axis in panels (b)-(d) corresponds to the number of gene trees.

4.1 The ratio of the number of coalescent histories on a network to the number of coalescent histories on the underlying tree. (A) The results are based on 30 random 20-leaf phylogenetic networks and 100 gene trees for each network. Each of the 30 networks was obtained by adding a single reticulation to an underlying tree. (B) A phylogenetic network with a single reticulation. The diameter of the reticulation is the number of edges on the paths marked with the red cycle. The taxa under the reticulation correspond to the leaves marked with the blue line.
4.2 A phylogenetic network and its 3-taxon subnets. Phylogenetic network

Ψ induces four subnets Ψ₁, Ψ₂, Ψ₃, and Ψ₄. The inheritance probabilities

are the same among the network and subnets for corresponding reticulation

edges, and the times of all corresponding nodes are the same among the

network and its three subnets. .................................................. 41

4.3 Running times of computing the pseudo-likelihood of networks of varying numbers of taxa and leaves. The running times are reported in seconds. They were measured on a desktop computer with 16G RAM and

INTEL XEON E3-1245 @ 3.5GHz, and 8 threads were used. ............ 44

4.4 The true network with 16 taxa used for assessing the accuracy of inferences based on pseudo-likelihood. The branch lengths of the phylogenetic

networks are measured in units of expected number of mutations per site

(scale is shown). The inheritance probabilities are marked in blue. ....... 46

4.5 Accuracy of the inference on simulated data. The blue region corresponds to the number of times the true network was returned as the optimal

network (the one with the highest pseudo-likelihood) after the search. The

orange region corresponds to the number of times the true network is not

the optimal network found by the search, but but is among the top 5 species

networks under maximum pseudo-likelihood. All other scenarios are rep-

resented by the grey region. .................................................. 48
4.6 **The true network with 5 taxa used for assessing the accuracy of inferences based on pseudo-likelihood.** The branch lengths of the phylogenetic networks are measured in units of expected number of mutations per site (scale is shown). The inheritance probabilities are marked in blue.

4.7 **Accuracy of the inference on simulated data: pseudo-likelihood vs. full likelihood.** The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but is among the top 5 species networks. All other scenarios are represented by the grey region. P: pseudo-likelihood, F: full likelihood.

4.8 **Accuracy of the inference on simulated data: Effect of the number of sampled individuals.** The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but is among the top 5 species networks. All other scenarios are represented by the grey region. On the horizontal axis, 1, 2, 3 and 4 represent the number of individuals of sampled from the hybrid taxon.
4.9 Accuracy of the inference on simulated data: Robustness to violation in the independent-loci assumption. The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but is among the top 5 species networks. All other scenarios are represented by the grey region. On the horizontal axis, 1, 10 and 100 represent the number of markers generated from a single gene tree.

4.10 Accuracy of the inference on simulated data: Setting different limits on the number of reticulations during inference. The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but is among the top 5 species networks. All other scenarios are represented by the grey region. On the horizontal axis, 1, 2 and 3 represent the maximum number of reticulations during inference.

4.11 Improvement with more reticulations. On the horizontal axis, 1, 2 and 3 represent the maximum number of reticulations during the estimation. At each point, the corresponding network topology is shown. The blue edges represent the additional reticulations to the true network.
4.12 The phylogenetic network with maximum pseudo-likelihood for the subset with the hybrid *O. × cockayneana* (*Meudt 175a, MPN 29710*) and putative parents. The width of each tube is proportional to the population mutation rate of each branch, which is printed on each tube. The length of each tube is proportional to the length of the corresponding branch in units of expected number of mutations per site (scale shown). Blue arrows indicate the reticulation edges and their inheritance probabilities are printed in blue.

4.13 The phylogenetic network with maximum pseudo-likelihood for the subset with the hybrid *O. × prorepens* (*Meudt 203a, MPN 29774*) and putative parents. The width of each tube is proportional to the population mutation rate of each branch, which is printed on each tube. The length of each tube is proportional to the length of the corresponding branch in units of expected number of mutations per site (scale shown). Blue arrows indicate the reticulation edges and their inheritance probabilities are printed in blue.

4.14 Convergence of the proportions of marginalized site patterns in the data to their expectations. Every point is the empirical frequency of a marginalized 3-taxon site pattern minus (and normalized by) the (theoretical) expectation of that frequency on the corresponding subnet in the true network.
5.1 **The true network and its subnetworks.** (A) The true network $\Psi$, whose height of each node is indicated by the ticks. (B) The true subnetworks, whose height of each node is indicated by the ticks. 83

5.2 **The input to the merger algorithm.** Values near nodes of subnetworks are the height of each node. Colored dots indicate the corresponding nodes belong to the same disjoint set. 84

5.3 **The subnetworks after reconciling their heights.** Values near nodes of subnetworks are the height of each node. 85

5.4 **A starting network and a guide graph for leaf addition.** (A) The starting network, and its heights of nodes are represented by the nearby values. (B) The guide graph whose nodes are represented by circles. 89

5.5 **Attachments of B.** Colored nodes and edges belong to an attachment of B in a subnetwork. Blue nodes are in set $rt(B)$ of corresponding subnetwork, and red nodes are in set $it(B)$ of corresponding subnetwork. Attachments surrounded by dotted rectangles are selected in $H(B)$. 90

5.6 **Enumeration for B.** (A) The stored network when $(4.60, A)$ is resolved. (B) The draft network when examining $at_{\Psi_8}(B)$ with $(4.60, A)$ resolved. The arrows indicate the direction of branches in the attachment. The dotted branch is removed when storing this network. (C) The stored network when all HT pairs are resolved. (D) The new backbone network after adding taxon B and reconciling heights of nodes. 92
5.7 **Attachments of C.** Colored nodes and edges belong to an attachment of C in a subnetwork. Blue nodes are in set $rt(C)$ of corresponding subnetwork, and red nodes are in set $it(C)$ of corresponding subnetwork. Attachments surrounded by dotted rectangles are selected in $H(C)$. . . . . . . . . . . . . . 93

5.8 **Enumeration for C.** (A) The stored network when $(4.60, A)$ is resolved. (B) The stored network when $(4.60, A)$ and $(2.93, D)$ are resolved. (C) The attachment $at_{\Psi_{10}}(C)$ and the permutation of $rt_{\Psi_{10}}(C)$ with indices shown in parentheses. (D)(E) Draft networks when trying $at_{\Psi_{10}}(C)$. (F) stored network when trying $at_{\Psi_{10}}(C)$. (G) The final output. . . . . . . . . . . . . . 95

5.9 **Correctness of inferred networks from correct trinets, categorized by the number of reticulations in the true networks.** The numbers of data sets on which the inferred network is identical to or different from the true one are shown in blue and orange, respectively. . . . . . . . . . . . . . . . 97

5.10 **Correctness of inferred networks from correct trinets, categorized by the number of reticulations in the true networks.** (a) Results from 100 41-taxon networks. (b) Results from 100 81-taxon networks. Blue: the number of cases where the inferred network is identical to the true one when using either the full or reduced set of trinets. Orange: the number of cases where the inferred network is identical to the true one only when the full, but not reduced, set of trinets is used. Grey: the number of cases where the inferred network is different from the true one, regardless of whether the full or reduced set of trinets was used. . . . . . . . . . . . . . . . . . . 99
5.11 **Running times and accuracy for the inferred trinets.** (a) The total running time in CPU-hours to infer all trinets for each data set. (b) Accuracy of the inferred trinets. The number of data sets where the inferred trinet is correct (blue), the inferred trinet is inside the true network (orange), and all other cases (grey), are shown.

5.12 **Accuracy of the inferred networks, and comparison to maximum pseudo-likelihood.** (a) The false positives and false negatives for the data sets where the inferred network is not identical to the true network. Squares correspond to hard networks, crosses correspond to medium-difficulty networks and triangles correspond to easy networks. Blue, red and green correspond to results based on the full and reduced sets of trinets, and maximum pseudo-likelihood, respectively.

(b) The accuracy of our method on the full set of trinets (left set of bars) and on the reduced set of trinets (middle set of bars), and the accuracy of maximum pseudo-likelihood (right set of bars). Blue corresponds to the data sets where the inferred network is identical to the true network; orange corresponds to the data sets where the inferred network contains a backbone network that is present in the true network; grey corresponds to all other cases.

5.13 **The inferred network for the empirical data set.** The reticulation, with inheritance probabilities (blue), is shown by the dashed line.
5.14 The inferred networks for the empirical data set using maximum pseudo-likelihood. Top: the inferred network when no reticulation was allowed. Middle: the inferred network when 1 reticulation was allowed. Bottom: the inferred network when 2 reticulations were allowed. The reticulations, with inheritance probabilities (blue), are shown by the dashed lines.
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Probabilities of 15 rooted gene trees given the phylogenetic network $\Psi$ of Fig. 3.2 (b) ($w = 0$). The quantity $g_{ij}(t)$ is the probability that $i$ lineages coalesce into $j$ lineages within time $t$ [Ros02].</td>
</tr>
<tr>
<td>3.2</td>
<td>Probabilities of 15 rooted gene trees given the phylogenetic network $\Psi$ of Fig. 3.2 (b) ($w = 0$) as $x \to \infty$.</td>
</tr>
<tr>
<td>4.1</td>
<td>Site patterns on the phylogenetic network $\Psi$ of Fig. 4.2 and its marginalized patterns on the four subnets.</td>
</tr>
<tr>
<td>5.1</td>
<td>The computation of score of each subnetwork for starting network selection. The score of each network is the sum of two terms in the corresponding column.</td>
</tr>
</tbody>
</table>
5.2 **Results of merger algorithm for large networks.** Full and Reduced correspond to the full set of trinets and the reduced set of trinets, and \( n \) is the number of leaves in the network. Each batch consists of multiple trinet inferences that are all run in parallel. ‘Candidates enumerated’ is the number of new backbone networks that are proposed and examined by the algorithm during the full network construction. Accuracy is measured as the percentage of data sets in which the constructed network is identical to the true network. The average running time in seconds is the time it took to construct the full network from the set of trinets.

5.3 **24 true networks.** The 24 networks selected as the true networks in simulation study, followed by their subjective difficulties.
Chapter 1

Introduction

Phylogenetic networks model non-treelike evolutionary histories, such as those arising when hybridization occurs, and take the shape of a rooted, directed, acyclic graph. Phylogenetic network inference in the genomic era is most often carried out from data obtained from multiple unlinked loci across the genomes of species of interest. To account for the fact that processes such as incomplete lineage sorting (ILS) could co-occur with hybridization, the multispecies network coalescent (MSNC) model was introduced to turn phylogenetic networks into a generative model of gene genealogies, and, subsequently, a wide array of methods for statistical inference of phylogenetic networks under MSNC were introduced to evaluate the full likelihood (all of the aforementioned methods, except for the pseudo-likelihood method of [YN15]) suffer from several major performance bottlenecks. Methods that evaluate the full likelihood (all of the aforementioned methods, except for the pseudo-likelihood method of [YN15]) suffer from...
the prohibitive computational requirements of likelihood calculations [ZN18, EOZN18]. Currently, computing network likelihood is feasible only for fewer than 10 species and a very small number of reticulations. Second, all the aforementioned methods traverse the space of phylogenetic networks that is much larger than the space of phylogenetic trees, whose size is already exponential in the number of taxa. While the pseudo-likelihood method of [YN15] circumvents the likelihood calculations, albeit in an approximate manner, it does not overcome the problem of exploring the space of the phylogenetic networks. Third, for Bayesian methods, exploring the trans-dimensional space of phylogenetic networks (the number of reticulations changes during the exploration) leads to poor mixing.

1.1 Contributions of the thesis

In Chapter 3, I show that in the presence of coalescence effects, the set of displayed trees is not sufficient to capture the network. I formally define the set of parental trees of a network and make three contributions based on this definition. First, I extend the notion of anomaly zone to phylogenetic networks and report on anomaly results for different networks. Second, I demonstrate how coalescence events could negatively affect the ability to infer a species tree that could be augmented into the correct network. Third, I demonstrate how a phylogenetic network can be viewed as a mixture model that lends itself to a novel inference approach via gene tree clustering. Our results demonstrate the limitations of focusing on the set of trees displayed by a network when analyzing and inferring the network. Our findings can form the basis for achieving higher accuracy when inferring phylogenetic networks and open up new venues for research in this area, including new
problem formulations based on the notion of a network’s parental trees.

In Chapter 4, I first demonstrate why likelihood computations of networks take orders of magnitude more time when compared to trees. I then propose an approach for inference of phylogenetic networks based on pseudo-likelihood using bi-allelic markers. I demonstrate the scalability and accuracy of phylogenetic network inference via pseudo-likelihood computations on simulated data. Furthermore, I demonstrate aspects of robustness of the method to violations in the underlying assumptions of the employed statistical model. Finally, I demonstrate the application of the method to biological data. The proposed method allows for analyzing larger data sets in terms of the numbers of taxa and reticulation events. While pseudo-likelihood had been proposed before for data consisting of gene trees, the work here uses sequence data directly, offering several advantages as I discuss.

In Chapter 5, I propose a method for large-scale phylogenetic network inference that ameliorates all three challenges. The method divides the set of taxa into small, overlapping subsets, builds accurate subnetworks on the subsets, and finally agglomerates the subnetworks into a network on the full set of taxa. By focusing on three-taxon subsets in this thesis, the likelihood calculations become very fast, exploring the space of all phylogenetic networks on large numbers of taxa is completely sidestepped. Also, mixing is improved because more iterations of the RJMCMC sampler can be run on three-taxon networks, especially since different subsets can be analyzed independently in parallel. Furthermore, to avoid building all \( \binom{n}{3} \) trinets, a Hitting Set formulation of a problem for reducing the number of trinets based on gene trees is provided, and I demonstrate that the number of trinets can be reduced significantly without much effect on accuracy. I studied the accuracy and efficiency of the proposed algorithm. When making use of error-free trinets, I show
that the algorithm infers the correct network in all cases, whether making use of all trinets or a significantly reduced subset. When making use of inferred trinets, the algorithm has very good accuracy, where in many cases the correct network is inferred and in all others, a network with small error rate is inferred. This demonstrates the importance of inferring the trinets accurately. Equally important, the method allows for inferring large-scale networks whose inference is infeasible using existing statistical methods.

I implemented the two methods of Chapter 4 and 5 as two commands in the publicly available software package PhyloNet [WYZN18]. In Chapter 6 I demonstrate the usage of two commands and I provide examples to run them.
Chapter 2

Background

2.1 Phylogenetic networks

A phylogenetic network $\Psi$ on set $\mathcal{X}$ of taxa is a rooted, directed, acyclic graph (DAG) in which every internal node, except for the root, has in-degree 1 and out-degree 2 (tree node) or in-degree 2 and out-degree 1 (reticulation node). The root has in-degree 0 and out-degree 2, and each leaf has in-degree 1 and out-degree 0. Edges incident into reticulation nodes are the reticulation edges of the network, and all other edges are its tree edges. The leaves of the network are bijectively labeled by the elements of $\mathcal{X}$.

For a full probabilistic model, the edges of the network are also associated with continuous parameters as follows. For a given phylogenetic network $\Psi$, we denote by $V(\Psi)$, $E(\Psi)$, and $\mathcal{X}(\Psi)$ the network’s nodes, edges, and leaf labels, respectively. Each edge $b = (u, v)$ in $E(\Psi)$ has a length which is defined by the difference of heights of $u$ and $v$, which are denoted by $h(u)$ and $h(v)$. Each pair of reticulation edges $e$ and $e'$ incident into the same reticulation node have inheritance probabilities $\gamma_e$ and $\gamma_{e'}$ associated with them.
which are two non-negative numbers that satisfy $\gamma_e + \gamma_{e'} = 1$. Roughly speaking, $\gamma_e$ denotes the proportion of the genome (in the hybrid population denoted by the relevant reticulation node) that was inherited along edge $e$, and $\gamma_{e'}$ denotes the proportion of the genome that was inherited along edge $e'$. The network’s topology, branch lengths, and inheritance probabilities fully define the multispecies network coalescent (MSNC) and allows for deriving gene tree probability distributions under ILS and hybridization [YDN12, YDLN14].

For $x \in \mathcal{X}$, we denote by $A_{\Psi}(x)$ and $AR_{\Psi}(x)$ the sets of nodes and reticulation nodes, respectively, on all paths from the leaf labeled by $x$, or node $x$, to the root of $\Psi$ ($AR_{\Psi}(x) \subseteq A_{\Psi}(x)$). Additionally, we denote $R(\Psi)$ to be the set of reticulation nodes in $\Psi$, with $r(\Psi) = |R(\Psi)|$.

### 2.2 Inference under the MSNC model

The data in phylogenomic inferences involves $m$ independent loci (genomic regions) consisting of $\mathcal{X} = \{S_1, \ldots, S_m\}$, where $S_i$ is the sequence data for locus $i$. Most commonly, $S_i$ could be an alignment of sequences from each of the species under consideration, or $S_i$ is data from a single bi-allelic marker (a vector of 0’s and 1’s), such as a single nucleotide polymorphism (SNP).

The model consists of $\Psi$, the phylogenetic network (topology and its continuous parameters such as divergence times), and vector $\Gamma$ of the inheritance probabilities. The likelihood of the model is given by

$$p(\mathcal{S} | \Psi, \Gamma) = \prod_{i=1}^{m} \int_{G} p(S_i | g) p(g | \Psi, \Gamma) dg,$$

where the integration is taken over all possible gene trees, $p(S_i | g)$ is the probability of the
sequence alignment $S_i$ given a particular gene tree $g$ [Fel81], and $p(g|\Psi, \Gamma)$ is the density of the gene tree (topologies and branch lengths) given the model parameters [YDLN14].

The posterior $p(\Psi, \Gamma|\mathcal{S})$ of the model is proportional to

$$p(\Psi, \Gamma|\mathcal{S}) \propto p(\mathcal{S}|\Psi, \Gamma)p(\Psi)p(\Gamma) = p(\Psi)p(\Gamma) \prod_{i=1}^{m} \int_{G} p(S_i|g)p(g|\Psi, \Gamma)dg,$$  \hspace{1cm} (2.2)

where $p(\Psi)$ and $p(\Gamma)$ are the priors on the phylogenetic network (and its parameters) and the inheritance probabilities, respectively.

As discussed above, statistical inference methods under this model suffer from the computational complexity of computing the likelihood, and the challenges with exploring the astronomical and jagged space of phylogenetic networks. Next we describe our method that ameliorates the problem to infer a large network via a two-step approach in which subnetworks are first inferred on smaller data sets of taxa and then the subnetworks are combined to produce the full network.
Chapter 3

In the Light of Deep Coalescence:

Revisiting Trees Within Networks

Evolutionary, or explicit, phylogenetic networks are graphical models that model reticulate evolutionary histories \cite{HB06,Nakl0a,BvIJ13}. Such evolutionary histories arise when processes such as horizontal gene transfer or hybridization occur. Research into mathematical properties, complexity results, and algorithmic techniques has exploded recently, as evident by the publication of three recent books on the subject \cite{HRS10,Mor11,Gus14}. A main premise behind the use of phylogenetic networks is that when a single tree is not sufficient to model the evolutionary history of a set of sequences or characters, a phylogenetic network that encompasses several trees is used. For example, the phylogenetic network in Fig. 3.1(a) depicts an evolutionary history that involves hybridization between taxon D and the most recent common ancestor (MRCA) of taxa B and C.

Central to research on phylogenetic networks has been the notion of trees displayed by

\footnote{Results in this Chapter are based on \cite{ZYN16}.}
a phylogenetic network. We say that a phylogenetic network displays a tree if the tree can be obtained by removing a set of “reticulation edges” of the network. Fig. 3.1 shows the two trees displayed by the network given in the figure. Given a phylogenetic network $\Psi$, we denote by $U(\Psi)$ the set of all trees displayed by $\Psi$. When incongruence in the gene trees inferred on different genomic regions across a genome alignment is assumed to be caused only by reticulation (e.g., hybridization), then the observed gene trees are taken to be a subset of the set of trees displayed by the (unknown) phylogenetic network for the set of genomes. This is why the set $U(\Psi)$ has played a fundamental role in most results established for phylogenetic networks. Examples of the prominent use of $U(\Psi)$ include: (1) Parsimonious phylogenetic networks that fit the evolution of a set of sequences under the infinite sites model [WZZ01, NRW05, GBBS07, GEL03, SDG+06, SH03, SH04, SH05]; (2) extending the maximum parsimony and maximum likelihood criteria from trees to networks [Hei90, NJZMC05, INST06a, INST06b, INST07b, INST07a]; (3) inferring minimal networks from sets of gene trees [BSS06, HR08, VIKRH10, Wu13]; (4) establishing identifiability results related to networks [PS15]; (5) establishing complexity results related to
networks [KNX06, BS07, KNTX08, KNX08, VISS10, VIK11]; and (6) identifying special trees within the network [SLHS13, DR16, DVMW15, FS15].

One of the evolutionary phenomena that has been extensively documented in recent analyses and targeted for computational developments is deep coalescence, or incomplete lineage sorting [Ros02]. This phenomenon amounts to gene tree incongruence due to population effects (determined by factors such as the sizes of ancestral populations and/or the times between subsequent speciation events). When this phenomenon is present in a reticulate evolutionary history, a major challenge faces all the aforementioned works: The set of trees displayed by a network is no longer adequate to fully capture gene evolution within the network.

To resolve this issue, we define the set of parental trees of a phylogenetic network to supplant the set of displayed trees (this is the same as the set of weakly displayed trees defined by Huber et al. [HMSW16]). Based on this set, we make three contributions. First, we extend the concept of anomaly zone to phylogenetic networks and establish results based on this concept. It is important to note here that Solís-Lemus et al. [SLYA16] recently discussed the issue of anomaly in the presence of reticulation where they focused on the “species tree” inside the network. Here, we define the anomaly zone in terms of the set of all parental trees and do not designate a species tree inside the network. Second, we address the problem of inferring a backbone tree inside the network that could serve as a starting tree for network searches and/or provide information on the history of speciation events in the presence of gene flow. As in the first contribution, the work here differs from that of [SLYA16] in focusing on all trees displayed by a network, rather than just a designated species tree. Third, we propose a novel clustering-based approach to phylogenetic network
inference from gene trees by which the gene trees are first clustered, parental trees are inferred from the clusters, and then the parental trees are combined into a phylogenetic network. Gori et al. [GSA+16] recently studied the performance of various combinations of dissimilarity measures and clustering techniques in clustering gene trees. Our work differs from that of [GSA+16] in that our focus is on phylogenetic network inference via clustering.

We believe our work will open up new venues for research into computational methods and mathematical results for reticulate evolutionary histories.

3.1 Methods

As we discussed above and illustrated in Fig. 3.1, the notion of trees displayed by a network has played a central role in analyzing and inferring networks.

**Definition 1** Let $\Psi$ be a phylogenetic network. A tree $t$ is displayed by $\Psi$ if it can be obtained by removing for each reticulation node exactly one of the edges incident into it followed by repeatedly applying forced contractions until no nodes of in- and out-degree 1 remain. A forced contraction of a node $u$ of in-degree 1 and out-degree 1 consists of (1) adding an edge from $u$’s parent to $u$’s child, and (2) deleting node $u$ and the two edges that connect it to its parent and child. We denote by $\mathcal{U}(\Psi)$ the set of all trees displayed by $\Psi$.

Fig. 3.1 shows a phylogenetic network $\Psi$ along with $\mathcal{U}(\Psi)$. 
3.1.1 Deep coalescence and the parental trees inside a network

Let us consider tracing the evolution of a recombination-free genomic region of four individuals \(a, b, c, \) and \(d\), sampled from the four taxa A, B, C and D within the branches of the phylogenetic network \(\Psi\) of Fig. 3.1. If \(b\) and \(c\) coalesce at the most recent common ancestor (MRCA) of B and C, and no events such as deep coalescence or duplication/loss occur anywhere in the phylogenetic network, then the genealogy of the genomic region is one of the two trees in the set \(U(\Psi)\). This is precisely the reason why much attention has been given to the set \(U(\Psi)\), as discussed in the Background section.

However, let us now consider a scenario where \(b\) and \(c\) did not coalesce at the MRCA of B and C. One potential outcome in terms of the resulting genealogy for \(a, b, c, \) and \(d\) is illustrated in Fig. 3.2(a). The probability that \(b\) and \(c\) fail to coalesce at the MRCA of B and C has to do with the quantity \(y\) in the figure: The smaller it is, the more likely it is that \(b\) and \(c\) would fail to coalesce [DS05]. Interestingly, for the scenario illustrated in Fig. 3.2(a), neither of the two trees in the set \(U(\Psi)\) can capture the shown genealogy. This brings us to define the set of parental trees inside a phylogenetic network to appropriately represent the network as a mixture of trees that adequately model the evolution of genes in the presence of deep coalescence. Parental trees are what Huber et al. referred to as weakly displayed trees in [HMSW16].

Yu et al. [YDN12] gave an algorithm for the simple task of converting a phylogenetic network \(\Psi\) to a multi-labeled tree, or MUL-tree, \(T\). Proceeding from the leaves of the network toward the root, the algorithm creates two copies of each subtree rooted at a reticulation node, attaches them to the two parents of the reticulation node, and deletes the two
Figure 3.2: (a) The genealogy of a gene (blue lines) within the branches of a phylogenetic network. In this scenario, the two lineages from B and C failed to coalesce prior to the reticulation node (evolution is viewed backward in time, from the leaves toward the root). The resulting genealogy in this case is \(((a,b),(c,d))\) and neither of the two trees in the set \(\mathcal{U}(\Psi)\) (shown in Fig. 3.1) capture this scenario. The length in coalescent units of the branch between the reticulation node and the MRCA of B and C is \(y\). (b) An abstract representation of the network, assuming both reticulation edges have the same length \(w\).
Algorithm 1: NetworkToMULTree.

**Input**: Phylogenetic $\mathcal{X}$-network $\Psi$ and its branch lengths $\lambda$.

**Output**: MUL-tree $T$ and its branch lengths $\lambda'$.

$T \gets \Psi$;

$\lambda' \gets \lambda$;

while traversing the nodes of $T$ bottom-up do

if node $h$ has two parents, $u$ and $v$, and child $w$ then

Create a copy of $T_w$ whose root is new node $w'$;

Add to $T$ two new edges $e_1 = (u, w)$ and $e_2 = (v, w')$;

$\lambda'_{(u,w)} \gets \lambda_{(u,h)} + \lambda_{(h,w)}$; $\lambda'_{(v,w')} \gets \lambda_{(v,h)} + \lambda_{(h,w)}$;

Delete from $T$ node $h$ and edges $(u, h)$, $(v, h)$, and $(h, w)$;

Delete $\lambda'_{(u,h)}$, $\lambda'_{(v,h)}$, $\lambda'_{(h,w)}$;

return $T$;

from $T$ by retaining, for each taxon $x \in \mathcal{X}$, $n_x$ leaves labeled by $x$, where $1 \leq n_x \leq a_x$, and deleting the remaining $(a_x - n_x)$ leaves labeled by $x$, followed by repeatedly applying forced contractions until no nodes of in- and out-degree 1 remain.

**Definition 2** Let $\Psi$ be a phylogenetic network on set $\mathcal{X}$ of taxa and $T$ be its MUL-tree. A parental tree inside $\Psi$ is a tree $t$ such that $t = T|_{(\mathcal{X},a)}$. We denote by $\mathcal{W}(\Psi)$ the set of all parental trees inside $\Psi$.

Fig. 3.3 shows the set $\mathcal{W}(\Psi)$ for the phylogenetic network in Fig. 3.1. The gene genealogy shown in Fig. 3.2(a) can be captured by the parental tree in Fig. 3.3(d). Indeed, Yu et al. [YDN12, YDLN14] gave mass and density functions for gene trees on phylogenetic
networks in terms of the set of parental trees inside the network. While it is obvious that $U(\Psi) \subseteq W(\Psi)$, the two sets can differ significantly in terms of their properties. For example, if $\Psi$ has $k$ reticulation nodes, then $|U(\Psi)| \leq 2^k$. However, $|W(\Psi)|$ could be much larger than $2^k$, as it is a function of the numbers of leaves under the reticulation nodes as well as the numbers of individuals sampled per species.

One rather interesting result is that while the problem of testing whether a tree is displayed by a network is NP-hard [KNTX08], testing whether a tree is a parental tree of (equivalently, weakly displayed by) a network can be done in polynomial time [HMSW16].

### 3.1.2 Inheritance probabilities and the multispecies network coalescent

Given a species tree topology $\Psi$ and its branch lengths $\lambda$, the gene tree topology $G$ can be viewed as a discrete random variable whose mass function $P_{\Psi,\lambda}(G = g)$ was derived in [DS05]. In the case of phylogenetic networks, we also associate with every pair of edges $b_1 = (u_1, v)$ and $b_2 = (u_2, v)$ that are incident into the same reticulation node $v$ nonnegative real values $\gamma_{b_1}$ and $\gamma_{b_2}$ such that $\gamma_{b_1} + \gamma_{b_2} = 1$ [YDN12, YDLN14]. These quantities,
which we call inheritance probabilities, indicate the proportions of lineages in hybrid populations that tracks each of the two parents of that population. In this case, the phylogenetic network’s topology $\Psi$ and branch lengths $\lambda$, along with the vector of inheritance probabilities $\Gamma$, are sufficient to describe the mass function of gene trees $P_{\Psi,\lambda,\Gamma}(G = g)$ under the multispecies network coalescent [YDN12, YDLN14].

3.2 Results and discussion

In this section we describe the three main contributions of this work. First, we extend the concept of anomaly zones [DR06] to phylogenetic networks and establish conditions for their existence. Second, we address the question of whether it is possible, from an inference perspective, to obtain a tree that can be augmented into the correct network by adding reticulation edges between pairs of the tree’s edges. Third, we propose a clustering approach to network inference by clustering the gene trees, inferring parental trees, and then combining the parental trees into a network. These results have direct implications not only on understanding the relationships between trees and networks, but also the practical task of developing computational methods for network inference.

3.2.1 Phylogenetic networks and anomalies

In a seminal paper, Degnan and Rosenberg [DR06] showed that the branch lengths of a species tree could be set such that the most likely gene tree disagrees with the species tree. Such a gene tree is called an anomalous gene tree and the set of all branch length settings that result in an anomalous gene tree is the anomaly zone.
We now provide what, to the best of our knowledge, is the first definition of anomaly zones for phylogenetic networks. Note that in [SLY16], Solís-Lemus et al. discussed anomalous gene trees in the presence of ILS and gene flow. However, in their work, the anomaly was still defined with respect to a designated species tree (they viewed the phylogenetic network as a species tree with additional horizontal edges between pairs of its branches). Here, we do not designate any of the parental trees of the network as a species tree; instead, we define the anomaly zone directly in terms of the entire set of parental trees.

The guiding principle behind our definition is the question: Is the most likely gene tree to be generated by a phylogenetic network necessarily a parental tree inside the network?

**Definition 3** Let $\Psi$ be a phylogenetic network, $\lambda$ be its branch lengths, and $\Gamma$ be the inheritance probabilities associated with its reticulation edges. We say gene tree topology $g$ is anomalous for $(\Psi, \lambda, \Gamma)$ if

$$P_{\Psi,\lambda,\Gamma}(G = g) > P_{\Psi,\lambda,\Gamma}(G = t) \quad \forall t \in \mathcal{W}(\Psi).$$

(3.1)

A phylogenetic network $\Psi$ is said to produce anomalies if there exists branch lengths $\lambda$ and inheritance probabilities $\Gamma$ such that there exists an anomalous gene tree $g$ for $(\Psi, \lambda, \Gamma)$. The anomaly zone for a phylogenetic network $\Psi$ is a set of $(\Lambda, \Gamma)$ values for which $\Psi$ produces anomalies.

Degnan and Rosenberg [DR06] showed that three-taxon and symmetric four-taxon species trees have no anomaly zones, but that non-symmetric four-taxon trees and all species trees with five or more taxa have anomaly zones. One practical implication of these results was that the simple approach of sampling a very large number of loci, building gene trees and
taking the most frequent gene tree as the species tree (an approach dubbed “the democratic
dvote” method) does not always work.

Since the multispecies coalescent is a special case of the multispecies network coales-
cent, it immediately follows that any phylogenetic network with \( n \geq 5 \) leaves produces anomalies. We now show that three-taxon phylogenetic networks do not produce anomalies, but that symmetric phylogenetic networks with \( n = 4 \) leaves could produce anomalies. Note that according to [SLYA16], 3-taxon networks could still generate anomalous gene trees. The seeming discrepancy between the two results is due to the fact that here we define the anomaly zone in terms of all the parental trees inside the network and not just a single designated species tree.

**Lemma 1** A phylogenetic network \( \Psi \) on 3 taxa does not produce anomalies.

**Proof 1** Let \( \Psi \) be a phylogenetic networks on 3 taxa, and consider the set \( W(\Psi) \) when restricted only to the distinct topologies. We have \( 1 \leq |W(\Psi)| \leq 3 \).

If \( |W(\Psi)| = 3 \), then the topology of every gene tree on the same set of 3 taxa is an
element of \( W(\Psi) \). Therefore, no gene tree can satisfy Eq. (3.1).

If \( |W(\Psi)| = 2 \), without loss of generality, let the two parental trees be \(((A, B), C)\) and
\((A, (B, C))\). If \( \Psi \) produces an anomaly, then it must be that the anomalous gene tree is
\(((a, c), b)\). To obtain this gene tree, \( a \) and \( c \) must coalesce above the root in both parental
trees. Since for the other two gene trees the coalescence events could occur under or
above the root, the probability of each of them is bounded from below by the probability of
\(((a, c), b)\). Therefore, \(((a, c), b)\) is not anomalous.

If \( |W(\Psi)| = 1 \), without loss of generality, let the parental tree topology be \(((A, B), C)\).
If $\Psi$ produces an anomalous gene tree, then it must be that the anomalous gene tree is either $((a,c),b)$ or $(a,(b,c))$. To obtain $((a,c),b)$, $a$ and $c$ must coalesce above the root in the parental tree. And to obtain $(a,(b,c))$, $b$ and $c$ must also coalesce above the root in the parental tree. Since for $((a,b),c)$ the coalescence events could occur under or above the root, its probability is bounded from below by the maximum of the probabilities of $((a,c),b)$ and $(a,(b,c))$. Therefore neither $((a,c),b)$ nor $(a,(b,c))$ is anomalous.

Consider now the symmetric phylogenetic network $\Psi$ in Fig. 3.2(b) and whose set of parental trees is given in Fig. 3.3. The four gene trees that are identical to the parental trees of the network are $((a,(b,c)),d)$, $(a,((b,c),d))$, $((a,b),(c,d))$ and $((a,c),(b,d))$. We plotted in Fig. 3.4 the anomaly zone for this network in terms of small values for $x$ and $y$ ($\leq 1.0$) and for two values of the inheritance probability $\gamma$. The yellow and orange regions correspond to the anomaly zone of this network. This figure clearly shows the existence of an anomaly zone of the network in Fig. 3.2(b) (where $w$ is set to 0), which means that symmetric phylogenetic networks with $n = 4$ leaves could produce anomalies.
Figure 3.4: The most likely gene tree given the phylogenetic network in Figure 3.2(b) (with $w = 0$) with $\gamma = 0.5$ (left) and $\gamma = 0.05$ (right). The x-axis corresponds to branch length $x$ and the y-axis corresponds to branch length $y$. Yellow: gene tree $((a, d), (b, c))$; Orange: gene trees $(a, (b, (c, d)))$ and $(a, (c, (b, d)))$; Brown: gene trees $((a, b), (c, d))$ and $((a, c), (b, d))$; Blue: gene tree $a, ((b, c), d))$ in both panels, and gene tree $((a, (b, c)), d)$ additionally in the left panel.

3.2.2 On the backbone tree of a phylogenetic network

A very important question in the area of phylogenetic network inference is whether there exists a tree that can be augmented into the network by adding reticulation edges between pairs of the tree’s edges. Here, we refer to such a tree as the network’s backbone tree. A biological significance of this tree lies in its potential designation as the species tree (e.g., see the species tree underlying the phylogenetic network of mosquitos in [FPS+15]).

Francis and Steel [FS15] recently introduced the notion of tree-based networks to cap-
ture those networks that can be obtained by augmenting a backbone tree (they called it the “base tree”). Zhang [Zha16] and Jetten and van Iersel [JvI18] provided necessary and sufficient conditions for tree-based networks.

The blue regions in the two panels of Fig. 3.4 correspond to the parameter zones where the most likely gene tree is one of the two backbone trees. However, the other regions correspond to parameter zones where the most likely gene tree is not a backbone of the network. We now provide more details on this issue.

Let us consider again the network of Fig. 3.2(b). This network is tree-based and each of the two trees in Fig. 3.1 could serve as its backbone (indeed, the same network is drawn in Fig. 3.1 in a way that clearly demonstrates that it is tree-based). The probabilities of all 15 gene trees under this phylogenetic network are given in Table 3.1.
Table 3.1: Probabilities of 15 rooted gene trees given the phylogenetic network $\Psi$ of Fig. 3.2(b) ($w = 0$). The quantity $g_{ij}(t)$ is the probability that $i$ lineages coalesce into $j$ lineages within time $t$ \cite{Ros02}.

| Gene Tree $T_i$ | $P(T_i | \Psi, x, y, \gamma)$ |
|-----------------|-------------------------------|
| $T_1 = (((b, c), a), d)$ | $g_{21}(y)\left[\gamma(g_{21}(x) + g_{22}(x)\frac{1}{3}) + (1 - \gamma)(g_{22}(x)\frac{1}{3})\right]$ + $g_{22}(y)\left[\gamma^2(g_{31}(x)\frac{1}{3} + g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6})\right]$ + $(1 - \gamma)^2(g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6})$ + $2\gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{1}{3})$ |
| $T_2 = (((b, c), d), a)$ | $g_{21}(y)\left[(1 - \gamma)(g_{21}(x) + g_{22}(x)\frac{1}{3}) + \gamma(g_{22}(x)\frac{1}{3})\right]$ + $g_{22}(y)\left[(1 - \gamma)^2(g_{31}(x)\frac{1}{3} + g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6})\right]$ + $\gamma^2(g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6})$ + $2\gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{1}{3})$ |
| $T_3 = ((a, b), (c, d))$ | $g_{22}(y)\left[(\gamma^2 + (1 - \gamma)^2)(g_{31}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6})\right]$ + $\gamma(1 - \gamma)(g_{21}(x)g_{21}(x) + g_{21}(x)g_{22}(x)\frac{1}{3} + g_{22}(x)g_{21}(x)\frac{1}{3} + g_{22}(x)g_{22}(x)\frac{2}{3})$ + $\gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3})$ |
| $T_4 = ((a, c), (b, d))$ | $g_{22}(y)\left[(\gamma^2 + (1 - \gamma)^2)(g_{31}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6})\right]$ + $\gamma(1 - \gamma)(g_{21}(x)g_{21}(x) + g_{21}(x)g_{22}(x)\frac{1}{3} + g_{22}(x)g_{21}(x)\frac{1}{3} + g_{22}(x)g_{22}(x)\frac{2}{3})$ + $\gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3})$ |
| $T_5 = (((a, b), c), d)$ | $g_{22}(y)\left[\gamma^2(g_{31}(x)\frac{1}{3} + g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6}) + (1 - \gamma)^2(g_{33}(x)\frac{1}{6})\right]$ + $\gamma(1 - \gamma)(g_{21}(x)g_{22}(x)\frac{1}{3} + g_{22}(x)g_{22}(x)\frac{1}{6})$ + $\gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{1}{6})$ |
| $T_6 = (((a, c), b), d)$ | $g_{22}(y)\left[\gamma^2(g_{31}(x)\frac{1}{3} + g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6}) + (1 - \gamma)^2(g_{33}(x)\frac{1}{6})\right]$ + $\gamma(1 - \gamma)(g_{21}(x)g_{22}(x)\frac{1}{3} + g_{22}(x)g_{22}(x)\frac{1}{6})$ + $\gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{1}{6})$ |
| $T_7 = (a, (b, (c, d)))$ | $g_{22}(y)\left[(1 - \gamma)^2(g_{31}(x)\frac{1}{3} + g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{6}) + \gamma^2(g_{33}(x)\frac{1}{6})\right]$ + $\gamma(1 - \gamma)(g_{21}(x)g_{22}(x)\frac{1}{3} + g_{22}(x)g_{22}(x)\frac{1}{6})$ + $\gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{1}{6})$ |
| $T_8 = (((b, d), c), a)$ | $g_{22}(y)[(1 - \gamma)^2(g_{31}(x)\frac{1}{3} + g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{6}{5} + \gamma^2(g_{33}(x)\frac{6}{5} + g_{22}(x)\frac{2}{3}) + \gamma(1 - \gamma)(g_{22}(x)\gamma_{22}(x)\frac{2}{3} + g_{22}(x)g_{22}(x)\frac{1}{3}) + \gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3} + g_{22}(x)g_{22}(x)\frac{1}{3})]$

| $T_9 = (((a, d), (b, c)) | g_{21}(y)[\gamma g_{22}(x)\frac{1}{3} + (1 - \gamma)g_{22}(x)\frac{1}{3}]$

| $T_{10} = (((a, b), d), c) | g_{22}(y)[\gamma^2(g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{3} + g_{33}(x)\frac{6}{5} + \gamma^2(g_{33}(x)\frac{6}{5} + g_{22}(x)\frac{2}{3}) + \gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3} + g_{22}(x)g_{22}(x)\frac{1}{3})]$

| $T_{11} = ((b, (a, c, d))) | g_{22}(y)[(1 - \gamma)^2(g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{1}{3} + g_{33}(x)\frac{6}{5} + \gamma^2(g_{33}(x)\frac{6}{5} + g_{22}(x)\frac{2}{3}) + \gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3} + g_{22}(x)g_{22}(x)\frac{1}{3})]$

| $T_{12} = (((a, d), b), c) | g_{22}(y)[\gamma^2(g_{33}(x)\frac{6}{5} + g_{33}(x)\frac{1}{3} + g_{33}(x)\frac{6}{5} + \gamma^2(g_{33}(x)\frac{6}{5} + g_{22}(x)\frac{2}{3}) + \gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3} + g_{22}(x)g_{22}(x)\frac{1}{3})]$

| $T_{13} = (((b, d), a), c) | g_{22}(y)[\gamma^2(g_{33}(x)\frac{6}{5} + g_{33}(x)\frac{1}{3} + g_{33}(x)\frac{6}{5} + \gamma^2(g_{33}(x)\frac{6}{5} + g_{22}(x)\frac{2}{3}) + \gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3} + g_{22}(x)g_{22}(x)\frac{1}{3})]$

| $T_{14} = (((a, c), d), b) | g_{22}(y)[(1 - \gamma)^2(g_{32}(x)\frac{1}{3} + g_{32}(x)\frac{1}{3} + g_{33}(x)\frac{6}{5} + \gamma^2(g_{33}(x)\frac{6}{5} + g_{22}(x)\frac{2}{3}) + \gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3} + g_{22}(x)g_{22}(x)\frac{1}{3})]$

| $T_{15} = (((a, d), c), b) | g_{22}(y)[\gamma^2(g_{33}(x)\frac{6}{5} + g_{33}(x)\frac{1}{3} + g_{33}(x)\frac{6}{5} + \gamma^2(g_{33}(x)\frac{6}{5} + g_{22}(x)\frac{2}{3}) + \gamma(1 - \gamma)(g_{22}(x)g_{22}(x)\frac{2}{3} + g_{22}(x)g_{22}(x)\frac{1}{3})]$

|
While there are 15 possible gene tree topologies on taxa $a$, $b$, $c$, and $d$, as branch length $x$ in the network tends to infinity, the probabilities of seven of the 15 gene tree topologies converge to 0 and only eight gene trees have non-zero mass: $(((a, (b, c)), d), (a, ((b, c), d)), ((a, c), (b, d)), (((a, b), c), d), ((a, c), b), (a, (b, c, d)))$, and $(a, (c, (b, d)))$. The probabilities in this case are given in Table 3.2 and visualized as a function of varying branch length $y$ for two different settings of $\gamma$ in Fig. 3.5.

Figure 3.5: Gene tree distribution for the phylogenetic network in Figure 3.2(b) ($w = 0$) as $x \to \infty$, for $\gamma = 0.5$ (left) and $\gamma = 0.05$ (right). The x-axis corresponds to branch length $y$ and the y-axis corresponds to the probability of each gene tree topology (see Table 3.2).

When $\gamma = 0.5$ and $\frac{1}{4}e^{-y} > \frac{1}{2} - \frac{5}{12}e^{-y}$, which is equivalent to $y < 0.288$, the most likely gene tree given $\Psi$ is not one of its backbone trees (that is, the network cannot be obtained by adding a single reticulation edge to the most likely gene tree). This also demonstrates that if we defined anomalies in terms of the set $\mathcal{U}(\Psi)$ instead of set $\mathcal{W}(\Psi)$, the phylogenetic network would still produce anomalous gene trees.
Table 3.2: Probabilities of 15 rooted gene trees given the phylogenetic network $\Psi$ of Fig. 3.2 (w = 0) as $x \to \infty$.

| Gene Tree $T_i$ | $P(T_i|\Psi, y, \gamma)$ |
|-----------------|--------------------------|
| $T_1 = (((b, c), a), d)$ | $\gamma - (\gamma - \frac{\gamma^2}{3})e^{-y}$ |
| $T_2 = (((b, c), d), a)$ | $(1 - \gamma) - (\frac{\gamma^2}{3} - \frac{\gamma}{3} + \frac{2}{3})e^{-y}$ |
| $T_3 = ((a, b), (c, d))$ | $\gamma(1 - \gamma)e^{-y}$ |
| $T_4 = ((a, c), (b, d))$ | $\gamma(1 - \gamma)e^{-y}$ |
| $T_5 = (((a, b), c), d)$ | $\frac{\gamma^2}{3}e^{-y}$ |
| $T_6 = (((a, c), b), d)$ | $\frac{\gamma^2}{3}e^{-y}$ |
| $T_7 = (a, (b, (c, d)))$ | $\frac{(1-\gamma)^2}{3}e^{-y}$ |
| $T_8 = (((b, d), c), a)$ | $\frac{(1-\gamma)^2}{3}e^{-y}$ |
| $T_9 = ((a, d), (b, c))$ | 0 |
| $T_{10} = (((a, b), d), c)$ | 0 |
| $T_{11} = (b, (a, (c, d)))$ | 0 |
| $T_{12} = (((a, d), b), c)$ | 0 |
| $T_{13} = (((b, d), a), c)$ | 0 |
| $T_{14} = (((a, c), d), b)$ | 0 |
| $T_{15} = (((a, d), c), b)$ | 0 |

Given that the most likely gene tree is not necessarily a backbone of the phylogenetic network, we now turn our attention to three recent methods whose goal is to infer a species tree despite horizontal gene transfer. It is very important to point out upfront that the assumptions of these methods do not necessarily match the scenarios we investigate here, but
our goal is to assess how well they do at recovering a backbone tree inside the network of Fig. 3.2(b). In [DVMW15], Davidson et al. showed that ASTRAL-II [MW15] performed best among species tree inference methods in terms of recovering the species tree in the presence of reticulation (under a specific model of horizontal gene transfer). They further proved that the method is statistically consistent in terms of recovering the species tree under the same model. In [SLHS13], Steel et al. showed that triplet-based approaches to species tree inference are consistent in terms of inferring a species tree in the presence of horizontal gene transfer (also under a specific model). This technique was implemented as the “primordial tree” in Dendroscope [HS12]. Both ASTRAL-II and the primordial tree method in Dendroscope take gene trees as input. The method of Daskalakis and Roch [DR16] takes as input gene trees with branch length and compute the distance between every two taxa \(u\) and \(v\) as the median of the gene-tree distances between \(u\) and \(v\) over all gene trees in the data set (given a gene tree with branch lengths, the gene-tree distance between two leaves is the sum of the branch lengths on the simple path between the two leaves).

We simulated gene tree data sets under the phylogenetic network of Fig. 3.2(b) using ms [Hud02] while varying branch length \(y\) to take on values from the set \(\{0.1, 0.2, 0.5, 1.0\}\) (\(w\) was set to 0 and \(x\) was set to 1000 so as to rule out deep coalescence involving the two branches incident with the root). Data sets with 25, 50, 100 and 200 gene trees were generated, and for each configuration of branch length \(y\) and number of gene trees, 100 data sets were simulated. The accuracy of each method for a setting of branch length \(y\) and number of gene trees is the fraction, out of the 100 data sets, of times that the method returned one of the two trees displayed by the network. The results for all three methods on the simulated data are shown in Fig. 3.6.
Figure 3.6: The accuracy of three methods for inferring species trees in the presence of reticulation on data generated on the phylogenetic network of Fig. 3.2(b). Left column corresponds to setting $\gamma = 0.5$ and right column corresponds to setting $\gamma = 0.05$. Four settings for branch length $y$ (on the $x$-axis) were used, and for each setting data sets with 25, 50, 100, and 200 loci were generated. See the text for definition of the accuracy measure. (Top) ASTRAL-II [MW15]; (Middle) The method of Steel et al. [SLHS13] as implemented in Dendroscope [HS12]; (Bottom) Our own implementation of the method of Daskalakis and Roch [DR16].

The results show that when $y$ is very small, the methods perform poorly in terms of returning one of the two trees displayed by the network, especially in the case of $\gamma = 0.5$. 
This is expected as an inheritance probability of 0.5 is a huge deviation from the assumptions of the three methods. When \( \gamma = 0.5 \) and \( y \) is long enough (e.g., 1), ASTRAL-II and the method of [DR16] do a perfect job, while the method of [SLHS13] does not perform as well. For smaller values of \( y \) and with \( \gamma = 0.5 \), the method of [DR16] consistently performs better than the other two methods. For \( \gamma = 0.05 \), which is closer to the assumptions of the methods, all three of them perform well, even when \( y = 0.5 \) (in this case, the most likely gene tree is also a backbone tree). For smaller values of \( y \) in this case, ASTRAL-II and the method of [DR16] do almost equally well, and slightly better than the method of [SLHS13].

Our results are in agreement with the findings in [SLYA16], where the authors showed, additionally, that methods for phylogenetic network inference (specifically, they evaluated the maximum likelihood method of [TRN08] in PhyloNet [YDLN14]) do a better job at recovering a species tree in the presence of gene flow than methods that infer (species) trees.

### 3.2.3 From gene trees to species networks via parental trees: A clustering approach

Given our discussion above of the set of parental trees, one can view a phylogenetic network \( \Psi \) as a mixture model with \( |\mathcal{W}(\Psi)| \) components and each component as a distribution on gene trees defined by the parental tree corresponding to that component. This view gives rise to a novel approach for reconstructing phylogenetic networks from a set \( \mathcal{G} \) of gene trees when both deep coalescence and reticulation could be both at play:
1. Cluster the gene trees into clusters $C_1, C_2, \ldots, C_k$;

2. Infer a parental tree $T_i$ for cluster $C_i$ under the multispecies coalescent;

3. Combine the trees $T_1, T_2, \ldots, T_k$ into a phylogenetic network $\Psi$.

The rationale behind this approach is that clustering would identify the components of the mixture model, where the gene trees belonging to a component differ only because of incomplete lineage sorting (ILS), but not because of hybridization. That is why in Step (2) a tree is inferred for each component under the multispecies coalescent, which only handles ILS. In the third step, disagreements among the $k$ inferred trees are assumed to be all due to the hybridization events, and are used to obtain the final network. A parsimony approach, for example, to Step (3) would be formulated as follows.

**Definition 4** The Parental Tree Network Problem is defined as:

**Input:** A set $\mathcal{P}$ of parental trees.

**Output:** A phylogenetic network $\Psi$ with the smallest number of reticulation nodes such that $\mathcal{P} \subseteq \mathcal{W}(\Psi)$.

Establishing the computational complexity of this newly defined problem and devising algorithms and heuristics for solving it are beyond the scope of this chapter.

In [GSA+16], Gori et al. studied the performance of various combinations of clustering methods and dissimilarity measures on gene tree topologies as well as gene trees with branch lengths. In our work here, the focus is on phylogenetic network inference and our simulation study in what follows is preliminary and aimed at demonstrating the viability of this approach in terms of identifying the true set of parental trees.
We used 10 phylogenetic networks (Fig. 3.7(a)), and within each, we generated 30 data sets of 50 gene trees each, 30 data sets of 250 gene trees each, 30 data sets of 500 gene trees each, and 30 data sets of 1000 gene trees each.

Figure 3.7: Performance of the clustering approach on the simulated data as a function of the number of gene trees. (a) The phylogenetic network used in the simulations. The lengths of the two reticulation edges were set to 0. The length of the edge going out of the reticulation node was set to 0.2. The inheritance probability of the left reticulation edge was set to 0.35. Ten networks were generated from this network by setting the length of each other internal branch to a random number uniformly sampled in the range \([0.7, 1.3]\). (b) The number of clusters identified (averaged over 300 data sets for each bar). (c) The number of correctly inferred parental trees (out of the maximum of four parental trees). (d) The error between the set of inferred trees from the identified clusters and the set of four parental trees of the network. The x-axis in panels (b)-(d) corresponds to the number of gene trees.

For each gene tree data set, pairwise Robinson-Foulds (RF) [RF81] distances were computed between the gene trees, and the pairwise distances were converted into 3-dimensional points in Euclidean space using multidimensional scaling (MDS) as implemented in the MDSJ package [AG] (we also conducted clustering directly on the RF distances, and found...
a significant improvement in the results after applying MDS). We implemented the $k$-means clustering algorithm \cite{M+67} and used it to cluster the gene trees based on the Euclidean distances from MDS using $k = 2, 3, \ldots, 10$. We implemented the silhouette method \cite{Rou87} and the number of clusters with the maximum average silhouette (based on the pairwise RF distances) was selected as the number of clusters identified and the corresponding clustering as the identified clusters.

Fig. 3.7(b) shows the results of identifying the number of clusters (the correct number is 4). As the figure shows, clustering in this case is performing very well, returning the correct number of clusters in almost all cases with 250 gene trees or more, and performing only slightly poorer in the case of 50 gene trees.

After the clusters were identified, we turned to the next natural question: Do the clusters correspond to the parental trees of the network? To investigate this question, we chose to apply the “minimizing deep coalescence” (MDC) method of \cite{TN09} as implemented in \cite{TRN08} (the heuristic version that uses only the clusters in the input gene trees) to infer a “species tree” on each cluster. We then quantified the number of true parental trees that were inferred by MDC on the clusters in each data set. The results are shown in Fig. 3.7(c). The results indicate a very good performance where all four true parental trees are almost always correctly inferred, particularly when 250 gene trees or more are used.

Finally, when this MDC-based analysis returns trees other than the true parental trees, how far are they from the true ones? To answer this question we compared the set of true parental trees and the set of trees inferred by MDC based on the identified clusters using the tree-based measure of \cite{NSW+03} (finding the min-weight edge cover of a bipartite graph whose two sets of nodes correspond to these two sets of trees and the weights of edges are
RF distance) as implemented in PhyloNet [TRN08]. The results are shown in Fig. 3.7(d). The results indicate a very good performance of about 2% error for data sets with 250 gene trees or more, and about 10% for data sets with 50 gene trees.

It is worth mentioning that if a network that displays all gene trees in the input was sought, the result would be a network that differs significantly from the true network, as each data set contained many distinct gene tree topologies. All the differences among gene trees (many of which are due to ILS) would be interpreted as signal for reticulations. This highlights the major difference between the current practice of seeking a network that displays all gene trees in the input and our proposed approach of seeking a network whose parental trees are obtained from the input gene trees.

3.3 Discussion

In this chapter, we showed that when deep coalescence occurs, inference and analysis of phylogenetic networks are more adequately done with respect to the set of parental trees of the network, rather than the common practice of using the set of trees displayed by the network. We described the simple procedure for enumerating the set of parental trees of a given network, and based on this set, we made three contributions. First, we defined the anomaly zone for a phylogenetic network topology as the region of branch lengths and inheritance probabilities under which the most likely gene tree is not one of the parental trees inside the network. We provided straightforward results on the anomaly zones for networks that mainly result from the fact that networks are an extension of trees. An important question is whether it is feasible that none of the trees displayed by a network has an anomaly
zone, yet the network itself has one.

In many cases, biologists are interested in identifying a species tree in the presence of gene flow. We demonstrated that in the presence of deep coalescence, the most likely gene tree is not necessarily one of the backbone trees inside the network. Furthermore, we studied the performance of three recently introduced methods in terms of their ability to recover a backbone tree inside the network. We found that none of these methods performs well when deep coalescence is extensive. It is important to point out, though, that none of these methods were designed specifically for cases of hybridization, where multiple genomic loci could be introgressed due to the same hybridization event. However, our findings here call for more research into the question of identifying a species tree inside the network, when one exists. However, biologically, reticulation could be extensive, such as reported recently in an analysis of a mosquito data set [FPS+15 WYHN16], in which case, designating a “species tree” might not be adequate [CM15]. Furthermore, as Solís-Lemus et al. [SLY16] showed, inferring a network does a better job at finding even the species tree when gene flow is at play. From a computational perspective, identifying such a tree aids significantly in searching for networks from data [YDL14 WYN16] as they can serve as the starting phylogeny to which reticulation edges could be added.

Finally, many existing approaches for network inference rely on the assumption that the input gene trees are a subset of the set of trees displayed by a network and, consequently, seek to infer a phylogenetic network that displays all the gene trees. In the presence of deep coalescence, this approach would result in very erroneous networks. We argued that in this situation, parental trees need to be inferred first from gene trees and then a network that contains the inferred parental trees could be estimated. To demonstrate the merit for this
approach, we introduced a method by which gene trees are first clustered and then parental trees are inferred for the clusters. The results were very promising for this clustering-based approach to be pursued further. In terms of network inference, this approach gives rise to a new computational problem in which a network is sought to contain a given set of parental trees. It is important to acknowledge here that our performance study of the clustering approach is very preliminary and is aimed at introducing the problem and demonstrating its merit in a relatively ideal setting. We identify as a direction for future research a thorough analysis that examines, among many other aspects, the effects of errors in gene tree estimates (as opposed to using true gene trees), larger variations in the network’s branch lengths, and the number of reticulations in the network, on the performance of the approach.
Chapter 4

Inference of Species Phylogenies from Bi-allelic Markers Using Pseudo-likelihood

4.1 Methods

[BBF+12] introduced an algorithm for analytically computing the integration in Eq. (2.1) for bi-allelic markers, thus avoiding the need to sample gene trees to estimate the integral. [ZWy+18] extended the method of Bryant et al. in novel ways so that the integration in Eq. (2.1) can be done analytically also when the species phylogeny is a network, and the likelihood function becomes Eq. (4.1).

1Results in this Chapter are based on [ZNI+18].
In this section, we first discuss factors that govern the computational complexity of full likelihood calculations on networks, and then propose a pseudo-likelihood function of phylogenetic networks and demonstrate its scalability, based on Eq. (4.1).

4.1.1 When it comes to computational complexity, a network is not merely a tree with a few additional reticulations

While the full likelihood computations of the algorithm of [BBF^12] allowed for inferring species trees with tens of taxa, the computational complexity of these computations exploded when the species phylogeny was a network. We now explain the explosion in the running time of computing the probability of a gene tree topology on a network as compared to that of computing the probability of a gene tree on the species tree “inside” the network. While the computations in this chapter do not involve gene tree probability computations, the limitations of the likelihood computations of [ZWY^18] follow the same rationale.

The probability mass function (pmf) \( p(g|\Psi) \), where \( g \) is a gene tree topology, is central to statistical inference of species trees and networks. In the case of species trees, \( \Psi \) is a tree, and \( \Gamma \) is irrelevant. As was shown in [DS05, YDLN14], \( p(g|\Psi) = \sum_{h \in H_{\Psi}(g)} p(h|\Psi) \), where \( H_{\Psi}(g) \) is the set of all coalescent histories of \( g \) inside \( \Psi \). Roughly speaking, a coalescent history of \( g \) is an embedding of \( g \) within the branches of \( \Psi \). The size of \( H_{\Psi}(g) \) for a gene tree topology \( g \) when \( \Psi \) is a tree is exponential in the number of leaves in the network.
gene tree [Ros07, TRIN07]. [Wu12] devised an efficient way of computing the pmf for species trees without explicitly enumerating all coalescent histories in the set $H_\Psi(g)$. Similarly, [YRN13] devised an efficient way of computing the pmf for species networks without enumerating all coalescent histories. However, with this improved method, likelihood calculations for species trees scale very well, whereas that is not the case for phylogenetic networks.

Here, we extended the algorithm of [TRIN07] to compute the size of $H_\Psi(g)$ for a gene tree topology $g$ and phylogenetic network $\Psi$. For simplicity, we assume one individual is sampled per species so that the leaves of the gene tree and phylogenetic network are labeled by the same set $\mathcal{X}$ of taxa (the algorithm can be trivially extended to the case where multiple individuals are sampled per species). Every edge $e$ in a gene tree defines a cluster, $c_e$, which is the set of leaves under the edge. We denote by $L(t)$ the set of all taxa labeling the leaves of tree $t$. Let $C_g$ be the set of all clusters of size $\geq 2$ of taxa in the gene tree $g$, and let $X_g = L(g)$. Assume $E(\Psi)$ includes a special edge $r$ that is incoming into the root. We write $(x, y)$ to denote a tree whose root has two children that are subtrees $x$ and $y$. We define function $\rho_c(e)$ to represent the number of coalescent scenarios of leaves in set $c \cap c_e$.

We define $\text{children}(e)$ for edge $e = (u, v) \in E(\Psi)$ as the set of all edges $(v, x) \in E(\Psi)$ for $x \in V(\Psi)$. For two edges $e_1, e_2 \in E(\Psi)$, $e_1 \neq e_2$, we say that $e_2 = (u_2, v_2)$ is on the path from $e_1 = (u_1, v_1)$ to the root, denoted by $e_1 \prec e_2$, if there is path from the root of $\Psi$ to $u_1$ that passes through both $u_2$ and $v_2$. If no such path exists, we write $e_1 \not\prec e_2$. Denoting
by $\zeta(e_1, e_2)$ the number of paths from $e_1$ to $e_2$, the quantity can be computed by:

$$\zeta(e_1, e_2) = \begin{cases} 
1, & \text{if } e_1 = e_2 \\
0, & \text{if } e_1 \neq e_2 \\
\sum_{e_k \in \text{children}(e_2)} \zeta(e_1, e_k), & \text{if } e_1 < e_2 
\end{cases} \tag{4.2}$$

Using the function $\zeta$, $\rho_c(e)$ is computed using Algorithm 2. The size of $H_{\Psi}(g)$ is given by

$$\sum_{e \in E(\Psi)} \rho_X(g)(e) \zeta(e, r).$$

Algorithm 2: Compute $\rho_c(e)$.

**Input:** Cluster $c \in C_g$ and edge $e \in E(\Psi)$.

**Output:** $\rho_c(e)$.

$\rho_c(e) \leftarrow 0$;

Let $T \leftarrow$ be the smallest subtree $T$ of $g$ such that $c \subseteq L(T)$;

if $T = (\ell_1, \ell_2)$ where $\ell_1, \ell_2 \in L(g)$ then

$e_1, e_2 \leftarrow$ the two external edges of $\Psi$ connected to $\ell_1, \ell_2$;

$\rho_c(e) \leftarrow \sum_{e, e_1 \neq e_2} \zeta(e_1, e) \zeta(e_2, e)$;

else if $T = (\ell_1, x)$ where $\ell_1 \in L(g)$ and $x \notin L(g)$ then

$e_1 \leftarrow$ the external edge of $\Psi$ connected to $\ell_1$;

$c_2 \leftarrow$ be the set of leaves in $x$;

$\rho_c(e) \leftarrow \sum_{e, e_2 \in E(\Psi)} \zeta(e_1, e) \rho_c(e_2) \zeta(e_2, e)$;

else

$c_1, c_2 \leftarrow$ the leaf-sets of the two subtrees of the root of $T$;

$\rho_c(e) \leftarrow \sum_{e, e_1, e_2 \in E(\Psi)} \zeta(e_1, e) \rho_{c_1}(e_1) \rho_{c_2}(e_2) \zeta(e_2, e)$;

return $\rho_c(e)$;
Fig. 4.1A demonstrates the effect on the size of set $H$ of the addition of a single reticulation to an underlying tree to form a network. For some networks, going from a tree to a network with a single reticulation increased the size of $H$ by ten million fold. The efficient algorithms of [Wu12], and [YRN13] ameliorate the computational complexity of computing the pmf by employing a bottom-up algorithm that stores values at nodes. In other words, they trade off memory for time. The reason that these algorithms scale in the case of trees but not networks is precisely the illustration in Fig. 4.1A.

An obvious factor that significantly affects the running time of computing the pmf is the number of taxa (leaves) in the network. However, in some cases, computing the likelihood of a network on 20 taxa could take less time than that of computing the likelihood of a network with, say, 10 taxa. The complexity of a phylogenetic network is governed by the diameters of the reticulation nodes and the number of leaves under the reticulation nodes (Fig. 4.1B). The larger either or both of these, the worse the explosion in the size of $H$ is and, consequently, the worse the likelihood calculations become in terms of time and memory requirements. These exorbitant computational costs are only exacerbated, in a potentially exponential manner, when the number of reticulations increases.
Figure 4.1: **The ratio of the number of coalescent histories on a network to the number of coalescent histories on the underlying tree.** (A) The results are based on 30 random 20-leaf phylogenetic networks and 100 gene trees for each network. Each of the 30 networks was obtained by adding a single reticulation to an underlying tree. (B) A phylogenetic network with a single reticulation. The diameter of the reticulation is the number of edges on the paths marked with the red cycle. The taxa under the reticulation correspond to the leaves marked with the blue line.
4.1.2 Pseudo-likelihood

Given a phylogenetic network $\Psi$ on set $\mathcal{X}$ of taxa, a subset $\Psi'$ of $\Psi$ on subset $\mathcal{X}' \subseteq \mathcal{X}$ of taxa is the phylogenetic network obtained by restricting $\Psi$ to the leaves in $\mathcal{X}'$ (if there are multiple paths between two leaves, we keep them all). We denote such a subnet by $\Psi|_{\mathcal{X}'}$. Fig. 4.2 shows a phylogenetic network and its four three-taxon subnets. It is important to note nodes and reticulation edges in the subnets have the same divergence times and inheritance probabilities as their corresponding nodes and reticulation edges, respectively, in the full phylogenetic network. We assume the same population size across all branches of the network.

While subnets with $0 \leq |\mathcal{X}'| \leq |\mathcal{X}|$ could be considered, here we consider only three-taxon subnets. For a phylogenetic network $\Psi$, we denote by $\mathcal{P}_3(\Psi)$ the set of all three-taxon subnets of $\Psi$, also called trinets.

Let $\Psi$ be a (parameterized) phylogenetic network on set $\mathcal{X}$ of taxa, and let $\mathcal{P}_3(\Psi)$ be the set of $\Psi$'s trinets. Consider a data set $\mathcal{J} = \{S_1, \ldots, S_m\}$ where $S_i$ consisting of the

Figure 4.2: A phylogenetic network and its 3-taxon subnets. Phylogenetic network $\Psi$ induces four subnets $\Psi_1$, $\Psi_2$, $\Psi_3$, and $\Psi_4$. The inheritance probabilities are the same among the network and subnets for corresponding reticulation edges, and the times of all corresponding nodes are the same among the network and its three subnets.
binary states of $m$ unlinked (independent) bi-allelic markers for a set $\mathcal{X}$ of taxa. We denote by $\mathcal{S}$ the set of all data sets obtained by restricting $\mathcal{I}$ to all three-taxon subsets. Table 4.1 shows an example of $\mathcal{I}$ and $\mathcal{S}$ for a four-taxon phylogenetic network (e.g., the one in Fig. 4.2). The pseudo-likelihood of a species phylogeny $\Psi$ (topology and parameters) is given by

$$
    \mathcal{P}.L(\Psi|\mathcal{I}) = \prod_{\Psi' \in \mathcal{S}^3(\Psi)} L(\Psi'|\mathcal{I}')
$$

where $\mathcal{I}'$ is the data set obtained by restricting $\mathcal{I}$ to the taxa in $\Psi'$, and $L(\Psi'|\mathcal{I}')$ is computed according to Eq. (4.1) using the algorithm of [ZWY+18]. It is important to note here that while the exposition is given in terms of phylogenetic networks and the algorithm of [ZWY+18], this same formulation applies to species trees and the algorithm of [BBF+12].

Table 4.1: Site patterns on the phylogenetic network $\Psi$ of Fig. 4.2 and its marginalized patterns on the four subnets.

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{I}$ ($\Psi$)</th>
<th>$\mathcal{S}_3$ ($\Psi_1$)</th>
<th>$\mathcal{S}_3$ ($\Psi_2$)</th>
<th>$\mathcal{S}_3$ ($\Psi_3$)</th>
<th>$\mathcal{S}_3$ ($\Psi_4$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1 s2 s3</td>
<td>1 0 0</td>
<td>1 0 0</td>
<td>1 0 0</td>
<td>1 0 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>A</td>
<td>1 0 0</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>B</td>
<td>0 1 0</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>C</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>D</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>
4.1.2.1 Searching the phylogenetic network space.

We use simulated annealing to search the phylogenetic network space with similar moves and setting to that of [YN15], with two main differences. First, when a delete-reticulation move is proposed, it is accepted with probability 0.01 no matter how the pseudo-likelihood changes. In our testing of the search strategy, we found that such a modification helps the search jump out of locally optimal states. Second, the optimal network with one fewer reticulation nodes than the the maximum number was kept during each iteration of search, and it was used as the starting state for every subsequent iteration except the first one. The search is conducted in a number of iterations, where each iteration walks the space of phylogenetic networks starting at some temperature that is reduced gradually during the iteration. So, for example, an iteration could inspect on the order of 50,000 points in the parameter space.

4.2 Results

4.2.1 Simulations

We first set out to assess the running time of computing the pseudo-likelihood of phylogenetic networks of varying sizes. To achieve this, we generated networks with 10, 20, 50, and 100 taxa and with 0, 1, 2, 3, and 4 reticulations. We used PhyloGen [Thi02] to first generate random species trees with 10, 20, 50, and 100 taxa. Then, for each species tree, we randomly added 0, 1, 2, 3, and 4 reticulations. To add a reticulation to a species network, we selected two edges uniformly at random and added an edge between their midpoints in
a direction that ensures no cycles are created. Inheritance probabilities were assigned 0.5 to reticulation edges. Then, we used PhyloNet [WYZN18] to simulate the evolution of 10000 bi-allelic markers on each species network. Finally, we computed the pseudo-likelihoods of each phylogenetic network using the data sets generated on it. The results are given in Fig. 4.3.

Figure 4.3: Running times of computing the pseudo-likelihood of networks of varying numbers of taxa and leaves. The running times are reported in seconds. They were measured on a desktop computer with 16G RAM and INTEL XEON E3-1245 @ 3.5GHz, and 8 threads were used.

As the results demonstrate, computing the pseudo-likelihood is very fast. In particular, it is worth pointing out that computing the full likelihood of networks of with more than 10 taxa and 3 or 4 reticulations could be impractical for many topologies. As discussed above, this is governed not only by the number of taxa and number of reticulation nodes,
but more importantly by the diameters of the reticulation nodes, the dependence among
the cycles in the underlying undirected graph of the network, and the numbers of taxa that
are descendants of reticulation nodes. The pseudo-likelihood calculations take fraction of
a second on phylogenetic networks with 20 or fewer taxa, regardless of the number of
reticulations. The calculations take about 10 seconds for 50-taxon networks, whereas the
running time jumps to over 100 seconds for phylogenetic networks with 100 taxa. This
increase in the running time with the number of taxa is expected, since the number of
trinets of a network on \( n \) taxa is \( \binom{n}{3} \). The very small increase in the running time, by
viewing it as a function of the increase in the number of reticulations, is because trinets
are small enough for full-likelihood calculations. Furthermore, even when a network has
4 reticulations, many trinets would have 0 or 1 reticulations, making their analysis even
faster.

While scalability in speed is impressive, the main question is: How accurate is infer-
ence of phylogenetic networks under pseudo-likelihood when using bi-allelic markers? To
answer this question, we generated 100 data sets, 20 replicates for each number of sites on
the network of Fig. 4.4. We simulated 100, 1000, 10000, 100000, and 1000000 bi-allelic
sites, with one haploid generated for every taxon. We set the mutation rates to \( u = 1 \) and
\( v = 1 \) (where \( u \) and \( v \) are the mutation rate from red allele to green allele and the muta-
tion rate from green allele to red allele, respectively, following the notation of [BBF +12]),
then we used \( \theta = 0.01 \) for every branch in the true network. We used following command
\((\text{numsites} \in \{ 100, 1000, 10000, 100000, 1000000 \})\) to simulate sites:

```
SimBiMarkersinNetwork -pi0 0.5 -sd seed -num numsites -tm <A
```
Figure 4.4: The true network with 16 taxa used for assessing the accuracy of inferences based on pseudo-likelihood. The branch lengths of the phylogenetic networks are measured in units of expected number of mutations per site (scale is shown). The inheritance probabilities are marked in blue.

For the value of `seed` in the “-sd” option, we used a different 8-digit integer for each of the 20 replicates. The value of `netstring` is the extended Newick string of the network in Fig. 4.4.

To test the ability of our algorithm to recover the topology of the true network, we ran the aforementioned simulated annealing procedure on the simulated data sets. The
maximum number of reticulations during the search was set to 3, since determining the true number of reticulation is beyond the capability of a maximum unpenalized (pseudo-)likelihood [WYZN18]. For each data set, the search was performed for 50 iterations. The 5 networks with highest pseudo-likelihood were saved during the search. We used the following command to generate the results:

```
MLE_BiMarkers -pseudo -mnr 50 -pi0 0.5 -mr 3 -pl 8 -ptheta 0.01 -thetawindow 0.01 -sd 12345678 -taxa (A_0,B_0,C_0,D_0,E_0,F_0,G_0,H_0,I_0,J_0,K_0,L_0,M_0,N_0,O_0,P_0)
-tm <A:A_0; B:B_0; C:C_0; D:D_0; E:E_0; F:F_0; G:G_0; H:H_0; I:I_0; J:J_0; K:K_0; L:L_0; M:M_0; N:N_0; O:O_0; P:P_0>
```

The results are shown in Fig. 4.5. As the results show, when 10,000 sites or more are used as input for inference, the method always infers the true phylogenetic network. When only 1,000 sites are used, the true network is one of the top five optimal networks inferred, but not the most optimal, in 20% of the cases, whereas the true network is not even among the top five optimal networks in the remaining 80% of the cases. When only 100 sites are used, the method cannot infer the true network.

These results can be put in the context of the performance of the method of [ZWY+18] in terms of how inference based on the full-likelihood calculations performs, and the method of [YN15] in terms of how inference based on pseudo-likelihood from gene tree estimates performs. As shown by [ZWY+18], accurate inference based on full-likelihood required at least 1,000 sites, even though the network considered in that work had only five taxa. In this regard, and given that the network considered here is much larger, it only
Figure 4.5: Accuracy of the inference on simulated data. The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but but is among the top 5 species networks under maximum pseudo-likelihood. All other scenarios are represented by the grey region.

makes sense that more than 1,000 sites are required. As for the method of [YN15], the authors showed that over 250 accurately estimated gene trees are required for accurate estimates of a 23-taxon phylogenetic network. An accurately estimated gene tree contains much more information than a single bi-allelic marker. In fact, a 23-taxon binary gene tree is equivalent to at least 21 different, yet compatible, bi-allelic markers, as each internal edge in the gene tree requires a bi-allelic marker to identify it.

These two contexts combined show the inference based on the pseudo-likelihood from bi-allelic markers is not only very accurate, but is also competitive with full-likelihood-
based inferences. It is worth repeating that the method of [YN15] requires very accurate
gene tree estimates. It is not uncommon to have error rates upwards of 60% in gene tree
estimates on closely related species.

4.2.2 Comparison to full likelihood computation

To compare inferences based on pseudo-likelihood and full likelihood as given by the
method of [ZWY+18], we simulated 100, 1000, 10000, 100000, and 1000000 bi-allelic
sites on the network in Fig. 4.6 with one haploid generated for every taxon. We used
\( \theta = 0.006 \) as the population mutation rate for external branches and the root, and \( \theta = 0.005 \)
for internal branches, both in the unit of population mutation rate per site. We set the muta-
tion rates to \( u = 1 \) and \( v = 1 \). We generated 100 data sets, 20 replicates for each number of
sites. We used following command (\( \text{numsites} \in \{100, 1000, 10000, 100000, 1000000\} \)) to
simulate sites:

```
SimBiMarkersinNetwork -pi0 0.5 -sd seed -num numsites -tm <A
\( \leftrightarrow A_0;C:C_0;L:L_0;Q:Q_0;R:R_0> \) -truenet netstring -out
\( \leftrightarrow \text{"markers.txt"}; \)
```

For the value of \textit{seed} in the “-sd” option, we used a different 8-digit integer for each of the
20 replicates. The value of \textit{netstring} is the extended Newick string of network in Fig. 4.6.

Then we ran maximum likelihood inference on those data sets with 10 iterations with
following commands, using both pseudo-likelihood and full likelihood computations (“-
pseudo” is removed for the latter):

```
MLE_BiMarkers -pseudo -mnr 10 -pi0 0.5 -mr 1 -pl 8 -ptheta
```
Figure 4.6: The true network with 5 taxa used for assessing the accuracy of inferences based on pseudo-likelihood. The branch lengths of the phylogenetic networks are measured in units of expected number of mutations per site (scale is shown). The inheritance probabilities are marked in blue.

Results of the comparison are shown in Fig. 4.7. As the results show, when 10,000 sites or more are used, inference based on pseudo-likelihood results in very accurate results that are identical to those obtained by full likelihood inference. When only 1000 sites were used, only in 6 out of 20 cases did pseudo-likelihood inference not result in the true network, but in all 6 cases that true network was one of the top 5 inferred. Again, in this case, the
Figure 4.7: **Accuracy of the inference on simulated data: pseudo-likelihood vs. full likelihood.** The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but is among the top 5 species networks. All other scenarios are represented by the grey region. P: pseudo-likelihood, F: full likelihood.

Results are comparable to those based on full likelihood. In data sets consisting of only 100 sites, inferences result in poor networks regardless of whether full- or pseudo-likelihood is used. To summarize these results, inference based on pseudo-likelihood is comparable in accuracy to that based on full likelihood.

### 4.2.3 The effect of number of individuals sampled

As described in [ZWY + 18], sampling more individuals from the hybrid species helps improve the accuracy of the inferred network based on full-likelihood computations. In par-
ticular, sampling more individuals allow the usage of fewer sites, a result that has important practical implications. We now set out to study this trend in the case of inferences based on pseudo-likelihood. We sampled one haploid individual for each of the four taxa L, A, R, and C, and 1, 2, 3, 4 haploid individuals for hybrid taxon Q in the network in Fig. 4.6. We generated 100, 500, 1000, 2000 sites for each of the 4 individual settings, with 10 replicates. We used following command (\(\text{numsites} \in \{100, 500, 1000, 2000\}\)) to simulate sites (in option “-tm”, individuals of Q: “Q_1,Q_2,Q_3” were removed according to 4 individual settings):

```
SimBiMarkersinNetwork -pi0 0.5 -sd seed -num numSites -tm <A
  ↠:A_0;C:C_0;L:L_0;Q:Q_0,Q_1,Q_2,Q_3;R:R_0> -truenet
  ↠netstring -out "markers.txt";
```

For the value of seed in the “-sd” option, we used a different 8-digit integer for each of the 20 replicates. The value of netstring is the extended Newick string of network in Fig. 4.6.

Then we ran maximum pseudo-likelihood estimation with 10 iterations on each of those 160 data sets using the same command as above. The accuracy of the inferred networks is reported in Fig. 4.8.

The results show a clear benefit to sampling multiple individuals, especially when the number of sites is small. Once again, when only 100 sites are used, the performance is not very good, since this number of sites is too small. However, when only 500 sites are used, sampled only individual results in obtaining the true network in only 6 out of 10 cases, whereas sampling 4 individuals results in accurate inferences in all 10 cases. Furthermore, as the number of sites increases, the positive effect of increasing the number of sampled
Figure 4.8: **Accuracy of the inference on simulated data: Effect of the number of sampled individuals.** The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but is among the top 5 species networks. All other scenarios are represented by the grey region. On the horizontal axis, 1, 2, 3 and 4 represent the number of individuals of sampled from the hybrid taxon.

Individuals starts diminishing.
4.2.4 Robustness of inference to lack of independence

As given by Eq. (2.1), the individual sites are assumed to be independent. Here, we set out to study the accuracy of inference when this assumption is violated. We simulated dependent sites as done by [ZWY+18]. We generated 1000, 5000, and 10000 sites in the network of Fig. 4.6 under different conditions: (a) single marker was generated from a gene tree (this is the case of independent loci), (b) 10 markers were generated from a gene tree, (c) 100 markers were generated from a gene tree. Note that gene trees were generated independently. We used following command

\[(\text{numgt, sitespergt}) \in \{(1000, 1), (5000, 1), (10000, 1), (100, 10), (500, 10), (1000, 10), (10, 100), (50, 100), (100, 100)\}\]


For the value of seed in the “-sd” option, we used a different 8-digit integer for each of the 20 replicates. The value of netstring is the extended Newick string of the network in Fig. 4.6

we ran maximum pseudo-likelihood estimation with 10 iterations on each of those data sets using the same command as above. Results are shown in Fig. 4.9 The results clearly show that the method is very robust to violation in the independent-loci assumption when 5000 sites or more are used. In the case of 1000 sites, the performance gets affected negatively only slightly. This result has great practical implications: In practice, when a large number of sites is sampled, one cannot ensure the sites are independent. The results show
Figure 4.9: **Accuracy of the inference on simulated data: Robustness to violation in the independent-loci assumption.** The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but is among the top 5 species networks. All other scenarios are represented by the grey region. On the horizontal axis, 1, 10 and 100 represent the number of markers generated from a single gene tree.

That even if independence is violated for a large number of sites, the method is robust in that case. This is consistent with what the authors observed with full likelihood computation in [ZWY+18].
4.2.5 The effect of limitation on number of reticulations

As we discussed in [WYZN18], inference of phylogenetic networks based on (unpenalized) likelihood cannot estimate the true number of reticulations simply because adding more reticulations only makes the model a better (or, at least as good a) fit for the data. This is why in the above results, we limited the number of reticulations that the method explores during inference to the true number (which is known since the data were simulated).

To understand how the method performs when the limit on the number of reticulations during inference is set higher than the true number, we compared different limitations on the number of reticulations using the network in Fig. 4.6. We simulated 100, 1000, 10000, 100000, and 1000000 bi-allelic sites with 20 replicates for each number with the same command used for comparing pseudo-likelihood and full likelihood computation, then set the maximum number of reticulations to 1, 2 and 3 for each settings. We then ran maximum pseudo-likelihood inference with 10 iterations on those data sets using the same command as above, and the “-mr” (maximum reticulations) option is changed to 1, 2 and 3 accordingly. The results are shown in Fig. 4.10. The results make complete sense as setting a higher limit on the number of reticulations allowed during the search for optimal networks guides the method toward networks with more reticulations since those would have better pseudo-likelihoods. To illustrate the behavior of the method, we took a replicate with 10,000 sites as an example and plotted the pseudo-likelihoods of the best inferred networks with 1, 2, and 3 reticulations, as well as the networks themselves in Fig. 4.11. The figure shows two important points. First, as more reticulations are allowed, the pseudo-likelihood of the best networks found improves, but the improvement starts slowing down with the
Figure 4.10: **Accuracy of the inference on simulated data: Setting different limits on the number of reticulations during inference.** The blue region corresponds to the number of times the true network was returned as the optimal network (the one with the highest pseudo-likelihood) after the search. The orange region corresponds to the number of times the true network is not the optimal network found by the search, but is among the top 5 species networks. All other scenarios are represented by the grey region. On the horizontal axis, 1, 2 and 3 represent the maximum number of reticulations during inference.

addition of more reticulations. Second, while more reticulations that the true number (one) are added, the true network is “inside” the optimal ones being identified with more reticulations. This is an important result because it illustrates how the true network is first identified and, then, when the method starts adding more reticulations, it does so to the underlying true network.
Figure 4.11: **Improvement with more reticulations.** On the horizontal axis, 1, 2 and 3 represent the maximum number of reticulations during the estimation. At each point, the corresponding network topology is shown. The blue edges represent the additional reticulations to the true network.

### 4.2.6 Empirical data sets

We also analyzed the two data sets in [ZWY+18] using our algorithm, including two different hybrid individuals *O. × cockayneana* and *O. × prorepens*. The hybrid origins of these two individuals are supported by both morphological [Meu06] and molecular (Meudt unpubl.) data. These two individuals were formally named along with putative parents. The first data subset comprises the following five individuals: *O. macrocarpa* (voucher: *Meudt 133a*, MPN 29546; herbarium codes follow [Thied] continuously updated), *O. macrophylla* subsp. *lactea* (Cameron 13392, AK 294893), hybrid *O. × cockayneana* (Meudt
175a, MPN 29710), O. caespitosa (Meudt 174a, MPN 29705), and O. calycina (Meudt 176a, MPN 29713). The number of loci in this data set is 802. The second data subset comprises O. sessilifolia subsp. splendida (Heenan s.n., MPN 32149), O. macrocarpa (Meudt 133a, MPN 29713), hybrid O. × prorepens (Meudt 203a, MPN 29774), O. sessilifolia subsp. sessilifolia (Meudt 199a, MPN 29771), and O. caespitosa (Meudt 196a, MPN 297695). The number of loci in this data set is 820.

Each data subset comprised five diploid individuals in total, which means ten haploid individuals were effectively analyzed due to the correction for dominant markers. Observe that while we could combine the two data sets into a larger one to study the scalability of the method on a larger data set, the number of sites available is too small for accurate inferences by pseudo-likelihood, as supported above by the simulation study.

The search was performed for 50 iterations for each data set. The maximum number of reticulations was set to 1. The results are given in Fig. 4.12 and Fig. 4.13.

Both results show that hybrids are correctly detected. The topology of the network in Fig. 4.13 with the hybrid O. × prorepens is consistent with Fig. 20 in [ZWY+18], while the topology of the network in Fig. 4.12 with the hybrid O. × cockayneana is slightly different from that in Fig. 19 in [ZWY+18]. Notably, the topology of Fig. 19 in [ZWY+18] is inferred as the second-best network by our algorithm, and its pseudo-likelihood value is smaller than that of the best network by only 0.2.
Figure 4.12: The phylogenetic network with maximum pseudo-likelihood for the subset with the hybrid *O. × cockayneana* (*Meudt 175a, MPN 29710*) and putative parents. The width of each tube is proportional to the population mutation rate of each branch, which is printed on each tube. The length of each tube is proportional to the length of the corresponding branch in units of expected number of mutations per site (scale shown). Blue arrows indicate the reticulation edges and their inheritance probabilities are printed in blue.
Figure 4.13: The phylogenetic network with maximum pseudo-likelihood for the subset with the hybrid *O. × prorepens* (*Meudt 203a, MPN 29774*) and putative parents.

The width of each tube is proportional to the population mutation rate of each branch, which is printed on each tube. The length of each tube is proportional to the length of the corresponding branch in units of expected number of mutations per site (scale shown). Blue arrows indicate the reticulation edges and their inheritance probabilities are printed in blue.
4.3 Discussion

The results above on simulated data and empirical data demonstrate very good accuracy of phylogenetic network inference using pseudo-likelihood based on bi-allelic markers. In terms of computing the pseudo-likelihood, the computation scales up to hundreds of taxa. Of course, searching the space of 100-taxon networks is a different challenge that requires novel techniques beyond scaling up likelihood computations. The simulation results, in particular, clearly demonstrate that as the number of bi-allelic markers increases, the accuracy improves significantly. We now turn to investigating the convergence of the inferred phylogenetic network onto the true network as the number of sites increases.

Let $\Psi$ be a phylogenetic network on set $\mathcal{X}$ of taxa and consider a subset $\mathcal{X}' \subseteq \mathcal{X}$ with $|\mathcal{X}'| = 3$. Both $\Psi$ and trinet $\Psi' = \Psi|_{\mathcal{X}}$ can be viewed as generative models for bi-allelic markers. It is important here to remind the reader that all parameters of $\Psi'$ are transferred from $\Psi$. Let $g$ be a gene tree on set $\mathcal{X}'$ of taxa. We have $p(g|\Psi) = p(g|\Psi')$, since the density $p(g|\Psi)$ does not involve any branches (or their parameters) that are in $\Psi$ but not in $\Psi'$ and the length of a branch in $\Psi'$ corresponds to the length of either a branch in $\Psi$ or a path that results from removing nodes when obtaining $\Psi'$. It follows from this that $P(s|g)p(g|\Psi) = P(s|g)p(g|\Psi')$ for a bi-allelic marker on the set $\mathcal{X}'$ of taxa. Therefore, we obtain from this that $P(s|\Psi) = P(s|\Psi')$. In other words, as the number of sites goes to infinity, for every trinet of the true network, the proportions of marginalized site patterns converge to their expectation in this trinet of the true network.

To study the convergence empirically, we generated one data set with each of 100, 1000, 10000, 100000, 1000000 sites of bi-allelic markers using the network of Fig. 4.4.
Then for each data set, for every trinet of that network, we compute the proportion of every corresponding marginalized site pattern, and compare it to the expected frequency of that pattern in the trinet. We normalized the differences by their corresponding expected frequencies. We plotted the normalized differences between these proportions in Fig. 4.14.

As the results show, the proportions of site patterns of trinets converge very fast to their theoretical expectation, with very negligible variance in the differences once 100,000 sites or more are used. These results are compatible with the accuracy of the method discussed above on the simulated data.

Statistical approaches for inferring phylogenetic networks offer a great promise in terms of utilizing the data to estimate not only the topology of the network, but also evolutionary parameters of interest to the biologist, such as population mutation rates and divergence times. Statistical approaches are based on evaluating the likelihood of phylogenetic network candidates during search of the parameter space. Except for small networks with fewer than ten taxa and two or three reticulations, these approaches are hard to apply in practice given the prohibitive computational requirements. The pseudo-likelihood methods of [YN15] and [SLA16] offered a way to ameliorate this issue. However, these methods make use of gene tree estimates as the input data. Under conditions of extensive incomplete lineage sorting, the individual loci could have very little signal to obtain accurate gene trees.

The major contribution of this chapter is the introduction of a pseudo-likelihood method based on the sequence data directly. We demonstrated that evaluating the pseudo-likelihood of a phylogenetic network is very fast and allows scaling to very large networks. We also demonstrated the accuracy of maximum pseudo-likelihood inference of phylogenetic net-
works from bi-allelic data.

Figure 4.14: **Convergence of the proportions of marginalized site patterns in the data to their expectations.** Every point is the empirical frequency of a marginalized 3-taxon site pattern minus (and normalized by) the (theoretical) expectation of that frequency on the corresponding subnet in the true network.
Chapter 5

A Divide-and-Conquer Method for Scalable Phylogenetic Network Inference from Multi-locus Data

5.1 Two related works

The closest works to our proposed method here are those of [HvIM+17, HVBL18]. In [HvIM+17], the authors devised an algorithm that is restricted to combining binet and trinet topologies (no divergence times) into level-1 networks (A phylogenetic network is level-1 if no two cycles in its underlying undirected graphs share a node). The work of [HVBL18] proposed another divide-and-conquer method to infer subnetworks and combine them. However their method makes use of the subnetwork topologies and requires specifying the number of reticulations a priori.

\[\text{Results in this Chapter are based on [ZLON19] (to appear)}.\]
The divide-and-conquer method we present here is not only designed to be scalable and make possible the inference of large phylogenetic networks, it also makes use of divergence times so that the estimated network has a time scale. It therefore represents substantial improvement over the previous likelihood-based methods limited in scalability and previous heuristic or summary methods limited in their utility.

5.2 Method

Our divide-and-conquer approach to large-scale phylogenetic network inference on set $\mathcal{X}$ of taxa takes the following steps:

1. determine a collection of overlapping subsets $\mathcal{X}_1, \ldots, \mathcal{X}_k$ of taxa;

2. for each set $\mathcal{X}_i$ of taxa, infer an accurate phylogenetic network $\Psi_i$ (topology, divergence times, and inheritance probabilities) from the sequence data of $\mathcal{X}_i$;

3. Combine the $k$ subnetworks $\Psi_1, \ldots, \Psi_k$ into a phylogenetic network on the full set $\mathcal{X}$ of taxa.

A key issue here is that the sets $\mathcal{X}_i$ are small enough so that accurate inference methods, such as [WN18], can efficiently and accurately estimate $\Psi_i$. In this work, we first show the performance when we consider all $\binom{|\mathcal{X}|}{3}$ 3-taxon subsets, and then propose a technique for reducing this number.

For $Y \subseteq \mathcal{X}$, we denote by $\Psi|_Y$ the phylogenetic network restricted to only the leaves labeled by elements of $Y$. We formulate Step 3 in our proposed approach as follows:

- **Input:** Subnetworks $\Psi_1, \ldots, \Psi_k$ on overlapping sets $\mathcal{X}_1, \ldots, \mathcal{X}_k$ of taxa.
• **Output:** Phylogenetic network $\Psi$ with the fewest nodes and edges such that $\Psi|_{x_i} = \Psi_i$ for $i = 1, \ldots, k$.

We now describe an iterative algorithm for this problem of combining subnetworks into a full network. The algorithm proceeds in three steps: (1) reconciling and summarizing the node heights across the subnetworks; (2) selecting a starting backbone network (a 3-taxon network in our case) and an order to add taxon-labeled leaves to it; and, (3) iteratively attaching new leaves ($n - 3$ of them) according to the computed order until a network on the full set of taxa is obtained.

### 5.2.1 Reconciling and summarizing the subnetworks

Although two nodes in different subnetworks can correspond to the same node in the true network, a degree of uncertainty is associated with the inferred parameters (mainly their heights) of the two nodes and so they will not exactly match. Those inexact heights will mislead a naïve algorithm that treats differences in heights as strictly pertaining to different nodes, therefore we need to reconcile the parameter estimates in each subnetwork first.

We construct a set $\mathcal{N}$ of disjoint sets of nodes (each node in each subnetwork has its height). Initially,

$$\mathcal{N} = \{\{v\}|v \in V(\Psi_j), 1 \leq j \leq k\};$$

that is, $\mathcal{N}$ is a set of singletons, one for each node in each of the subnetworks. For every pair $(\Psi_i, \Psi_j)$ of subnetworks, if $|\mathcal{X}(\Psi_i) \cap \mathcal{X}(\Psi_j)| > 1$, we obtain $\Psi_i'$ and $\Psi_j'$ by restricting $\Psi_i$ and $\Psi_j$ to $\mathcal{X}(\Psi_i) \cap \mathcal{X}(\Psi_j)$, respectively. By such a restriction, we have two injective mappings from the nodes of $\Psi_i'$ and $\Psi_j'$ to their corresponding nodes in $\Psi_i$ and $\Psi_j$, respec-
\[ m_i : V(\Psi'_i) \rightarrow V(\Psi_i) \text{ and } m_j : V(\Psi'_j) \rightarrow V(\Psi_j). \]

If \( \Psi'_i \) and \( \Psi'_j \) are identical in topology, let \( m' : V(\Psi'_i) \rightarrow V(\Psi'_j) \) be a bijection between their node-sets. Then for every node \( v_i' \in V(\Psi'_i) \), we find the two disjoint sets in \( \mathcal{N} \) containing \( m_i(v_i') \) and \( m_j(m'(v_i')) \), and replace these two sets with their union. If \( \Psi'_i \) and \( \Psi'_j \) are not identical, we ignore them.

In the end, for every node in every disjoint set in \( \mathcal{N} \), we assign the average height of nodes in the same set.

To summarize the height of each node in each subnetwork, here we introduce the “extended height matrix,” or EHM. An EHM \( \mathcal{M}_\Psi \) of a network \( \Psi \) with \( n \) leaves is an \( n \times n \) matrix, where element \( \mathcal{M}_\Psi(x, y) \), for taxa \( x, y \in \mathcal{X}(\Psi) \), is a sorted list of heights of tree nodes which are common ancestors of \( x \) and \( y \) in the binet obtained by restricting \( \Psi \) to \( \{x, y\} \). We combine \( \mathcal{M}_{\Psi_1}, \ldots, \mathcal{M}_{\Psi_k} \) into an EHM \( \mathcal{M} \) for the full network as follows. For \( x, y \in \mathcal{X} \), we set \( \mathcal{M}(x, y) \) to be the longest list among \( \mathcal{M}_{\Psi_1}(x, y), \ldots, \mathcal{M}_{\Psi_k}(x, y) \). If there are multiple longest lists, the list with smallest lexicographic rank is chosen. For example, if two longest lists \((0.1, 0.2, 0.4, 0.9)\) and \((0.1, 0.2, 0.3, 1.0)\) exist, the latter is chosen. We also define the “pairwise distance sum,” or PDS, for a subnetwork to be the sum of the height of the most recent common ancestor of every pair of taxa in the subnetwork.

### 5.2.2 Generating a starting network and an order for leaf addition

Here we describe how (1) a starting backbone network is selected, and (2) an order for adding all taxa to it is generated. We assume that a designated taxon \( z \) has been identified \textit{a priori} to be a member of outgroup with at most 2 members. As this taxon, by definition, is farthest from all ingroup taxa, our task boils down to selecting one of the subnetworks
that have \( z \) as a taxon (when all \( \binom{n}{3} \) trinets are built, there are \( \binom{n}{2} \) trinets that have \( z \) as a leaf label). We now describe how to choose one of those as the backbone network.

Let \( \Psi_i \) be a subnetwork whose leaves are labeled by the outgroup taxon \( z \), and two other taxa \( x \) and \( y \). We define \( s(\Psi_i) \) to be 1 if either \( x \) or \( y \) is under a reticulation node in any of the \( k \) subnetworks; otherwise, \( s(\Psi_i) = 0 \). Furthermore, for two subnetworks \( \Psi_i \) and \( \Psi_j \), we define \( d(\Psi_i, \Psi_j) \) to be the topological difference \([Nak10b]\) of their corresponding restrictions to the set \( \mathcal{R}^- \) of leaves when \( |\mathcal{R}^- \cap \mathcal{R}^-| > 1 \), otherwise, \( d(\Psi_i, \Psi_j) = 0 \). We then take as the backbone network the subnetwork

\[
\arg\min_{\Psi_i} s(\Psi_i) + \sum_{1 \leq j \leq k, i \neq j} d(\Psi_i, \Psi_j),
\]

where \( \Psi_i \) iterates over all subnetworks that have \( z \) as a leaf label, and \( k \) is the number of subnetworks. If there are multiple subnetworks with the same criterion, the subnetwork with largest PDS is chosen.

Before we add new taxa into the starting backbone, we need to generate an order for attaching new taxa according to the topologies of subnetworks to maximize the correct placement of reticulation nodes. Given two taxa \( x, y \in \mathcal{R}^- \) and a collection \( \Psi_1, \ldots, \Psi_k \) of subnetworks, we say that \( x \) precedes \( y \), denoted by \( x \preceq y \), if \( AR_{\Psi_i}(x) \neq \emptyset \) and \( |AR_{\Psi_i}(x)| \leq |AR_{\Psi_i}(y)| \) for some \( \Psi_i \). We build a directed graph whose nodes are the taxa set \( \mathcal{R}^- \), and edge \( (x, y) \) is in the graph if and only if \( x \preceq y \). Then we perform a topological sorting on the directed graph to get an order of attaching missing taxa. Note that there may be cycles in the directed graph; in such a case, when the topological sorting cannot proceed due to a cycle, we break the cycle by removing node \( x \) (and its incident edges) that appears under a reticulation node in the largest number of subnetworks. The
final result is an order of the elements of $\mathcal{B}$ (minus the three taxa that label the leaves of the backbone network). We create a list of distinct nodes (leaves), each labeled by one taxon, sorted according to the order obtained. The taxa are added to the initial backbone network one at a time according to the computed order. We now describe how each single taxon is added.

5.2.3 Iterative attachment of new taxa

Given the backbone network and the remaining set of taxon-labeled leaves (with their order), we describe how to attach a new taxon to the iteratively growing backbone network.

We define the attachment of taxon $x$ that labels a leaf in subnetwork $\Psi_i$, denoted by $at_{\Psi_i}(x)$, as the set $it_{\Psi_i}(x) \cup rt_{\Psi_i}(x)$, where

$$it_{\Psi_i}(x) = (\mathcal{A}_{\Psi_i}(x) \setminus \bigcup_{y(\neq x) \in \mathcal{X}(\Psi_i)} \mathcal{A}_{\Psi_i}(y)) \cup \{x\},$$

and $rt_{\Psi_i}(x)$ are parent nodes not in $it_{\Psi_i}(x)$ of all nodes in $it_{\Psi_i}(x)$. The edges of the attachment, denoted by $E(at_{\Psi_i}(x))$, is the set of all edges of $\Psi_i$ that connect two nodes in the attachment.

We add (leaf labeled by) taxon $x$ to the current backbone $\Psi_B$ as follows. We first compute $at_{\Psi_i}(x)$ for all $k$ subnetworks $\Psi_i$. Assuming there are $\ell$ subnetworks that have $x$ as a leaf label, we cluster the $\ell$ attachments by their sizes (all attachments with the same number of nodes in $rt$ belong to one cluster), and then choose the single attachment per cluster in which the parent node of the leaf labeled by $x$ has the smallest height of all attachments in that cluster. In our implementation, we considered only attachments that have up to 5 nodes in $rt$. Let $H(x)$ be the set of all resulting attachments (in our implementation, $H(x)$
contains at most 6 attachments). For each attachment \( at(x) = (it(x) \cup rt(x)) \in H(x) \), we create a set of new backbone networks as follows:

1. For each leaf \( x' \in \mathcal{X}(\Psi_B) \), we generate height-taxon pairs, or HT pairs, according to the overall EHM \( \mathcal{M} \). The height of the pair is an element of \( \mathcal{M}(x, x') \), and the taxon of the pair is \( x' \). Each HT pair \( (h', x') \) indicates, there is a tree node at a certain height \( h' \) to be the common ancestor of both \( x \) and \( x' \). A HT pair \( (h', x') \) is resolved, when there is a tree node in \( \Psi' \) which is a common ancestor of both \( x \) and \( x' \), such that the height of that node is \( h' \) within a tolerance.

2. Resolve HT pairs by finding the set \( P \) of positions on the path from \( x' \) (taxon in the pairs) to the root of \( \Psi_B \) where the height of each element in \( P \) is the height in the pairs. Map the elements of \( rt(x) \) to the positions in \( P \) in multiple ways. Remove from all the resulting backbone networks any nodes of in-degree 0 except for the original root of the \( \Psi_B \). In our implementation, we copy current backbone network to a draft network \( \Psi' \). We modify the draft network by inserting nodes in \( rt(x) \) to \( \Psi' \), and leaf labeled by \( x \) is connected to the draft network and thereby resolves HT pairs. The height of inserted node is set to the height of HT pair. We resolve HT pairs from lowest height. We enumerate the ways to resolve HT pairs by permuting nodes in \( rt(x) \), then use depth first search algorithm to search where to connect a node in \( rt(x) \). \textbf{Enumerate} (Alg. 3) computes \( \mathcal{H} \) as well as calls the depth first search algorithm \textbf{EnumerateDFS} (Alg. 4).

3. Remove networks with same topology.

The outcome of this procedure, when applied to all attachments in \( H(x) \), is a set of can-
didate backbone networks $B(x)$. We then choose from set $B(x)$ the network $\Psi'$ whose score is minimum. The score of $\Psi'$ is defined as follows with respect to each subnetwork $\Psi_1, \ldots, \Psi_k$:

$$D(\Psi', \Psi_i) = \begin{cases} d(\Psi', \Psi_i) & \text{if } r(\Psi') \leq r(\Psi_i) \\ \min_{\Psi''} d(\Psi', \Psi'') & \text{otherwise} \end{cases},$$

where $d$ is the topological distance of [Nak10b] applied to two networks restricted to their shared leaf-set, and $\Psi''$ is taken over all subnetworks of $\Psi'|\mathcal{X}(\Psi_i) \cap \mathcal{X}(\Psi')$ that have $r(\Psi_i)$ reticulation nodes. We choose $\Psi^*_B$ from set $B(x)$ as the new backbone network on set $\mathcal{X}(\Psi_B) \cup \{x\}$ of leaves the network $\Psi'$ that minimizes

$$(r(\Psi'))^2 + \sum_{1 \leq i \leq k} D(\Psi', \Psi_i).$$

Finally, we reconcile the heights of nodes in $\Psi^*_B$ according to subnetworks, by generating a mapping from nodes in $\Psi^*_B$ to a set of nodes in the subnetworks, then assign the average of height in each set to the nodes. For inheritance probabilities, we do the same thing for edges in $\Psi^*_B$. 
Algorithm 3: Enumerate.

**Input:** Network $\Psi_B$, Full EHM $\mathcal{M}$. Taxon $x$. Tolerance $\epsilon$.

**Output:** Set of networks $\Omega$.

$\Omega \leftarrow \emptyset$;

$\mathcal{H} \leftarrow []$;

**foreach** Taxon $x' \in \mathcal{X}(\Psi_B)$ **do**

**foreach** Height $h \in \mathcal{M}(x, x')$ **do**

Append $(h, x')$ to $\mathcal{H}$;

Sort $\mathcal{H}$ according to heights of the entries in ascending order;

Draft network $\Psi' \leftarrow \Psi_B$;

**foreach** $at(x) = (it(x) \cup rt(x)) \in H(x)$ **do**

**foreach** Permutation $r = (r_1, r_2, \cdots)$ where $r_1, r_2, \cdots \in rt(x)$ **do**

Remove the height of $r_1, r_2, \cdots$;

$\Omega \leftarrow \Omega \cup \text{EnumerateDFS}(\Psi', at(x), r, \mathcal{H}, x, \epsilon)$;

return $\Omega$;
Algorithm 4: EnumerateDFS.

**Input:** Draft network $\Psi'$. Attachment $at$. List $r$ of $rt$ nodes in $at$. List of HT pairs $\mathcal{H}$.

Taxon $x$. Tolerance $\epsilon$.

**Output:** Set of networks $\Omega$.

EnumerateDFS($\Psi'$, $at$, $r$, $\mathcal{H}$, $x$, $\epsilon$)

1. $\Omega \leftarrow \emptyset$;
2. // Clone current draft network and check whether to store it.
3. $\Psi'' \leftarrow \Psi'$;
4. Remove nodes with in-degree 0 in $\Psi''$ until the only node with in-degree 0 is the root;
5. Remove binary nodes in $\Psi''$;
6. if $\Psi''$ has no cycle and $\Psi''$ has no negative branch length then $\Omega \leftarrow \Omega \cup \{\Psi''\}$;
7. // If no more nodes to connect, no more search.
8. if $r$ is empty then return $\Omega$;
9. $\mathcal{M}_{\Psi'} \leftarrow$ extended height matrix of $\Psi'$;
10. // Remove all resolved HT pairs.
11. while $\mathcal{H}$ is not empty do
12.   $(h', t') \leftarrow$ first HT pair in $\mathcal{H}$;
13.   if $h'$ exists in $\mathcal{M}_{\Psi'}(x, t')$ within tolerance $\epsilon$ then Remove $(h', t')$ from $\mathcal{H}$;
14. end
15. // If all HT pairs are resolved, no more search.
16. if $\mathcal{H}$ is empty then return $\Omega$;
17. // Otherwise resolve the unresolved HT pair with lowest height.
(h', t') ← first HT pair in \( \mathcal{H} \);

\( b \) ← first node in \( r \) and remove first node from \( r \);

// \( b \) must have one child according to definition of attachment.

\( b' \) ← child of \( b \);

\( h(b) \) ← \( h' \);

if \( b' \) is a tree node then \( b''_1, b''_2 \) ← children of \( b \);

// Enumerate the ways to resolve current HT pair.

for \( k \in \{1, 2, 3\} \) if \( b' \) is a tree node else \( k \in \{3\} \) do

    if \( k \leq 2 \) then Delete edge \((b', b''_k)\);

    foreach \( e = (u, v) \in \mathcal{E}(\Psi') \) and \( e \notin at \) do

        if \( h(u) > h' > h(v) \) then

            Break \((u, v)\) into \((u, b)\) and \((b, v)\);

            \( \Omega \leftarrow \Omega \cup \text{EnumerateDFS}(\Psi', at, r, \mathcal{H}, x, \epsilon) \);

            Restore \((u, v)\);

        if \( k \leq 2 \) then Restore edge \((b', b''_k)\);

    Restore \( b \) in \( r \);

return \( \Omega \);
5.2.4 Asymptotic time complexity

Here we provide a loose analysis of asymptotic time complexity of our merger algorithm if all input subnetworks are trinets. Let the total number of taxa be \( n \), and let the total number of reticulations in the true network be \( r \). Then it takes at most \( O((n + r)^2) \) to compute the topological difference \([\text{Nak10b}]\) for two networks which are subnetworks of the true network. Suppose the number of input trinets is \( k \). The major time consumption is from the enumeration and evaluation of candidates while attaching new taxa to the growing backbone network.

Suppose we have \(|rt(x)| \leq m\) for all attachment in \( H(x) \). For one attachment, there will be at most \( O(m! \times 3^m(n + r)^m) \) new backbone networks. In our implementation, we set \( m \) to 5, which makes the number of candidates \( O((n + r)^5) \). Note that there are far fewer candidates, as demonstrated by our simulation study. A loose upper bound on the time complexity for computing the score for a candidate is \( O(3^r k(n + r)^2) \).

The total (yet very loose in many cases) asymptotic time complexity of our merger algorithm is

\[
O((n + r)^5) \times O(3^r k(n + r)^2) \times O(k) = O(3^r k^2(n + r)^7).
\]

5.2.5 Reducing the number of subproblems

The first step of our method requires inferring a phylogenetic network for every combination of 3 taxa, and this causes the computational complexity of subnetwork inference to be \( O(n^3) \) given \( n \) total taxa. If there are 100 taxa, the number of subnetworks to infer will be \( \binom{100}{3} = 161,700 \), which is an overwhelmingly large number for researchers who do not
have access to the largest supercomputers. Therefore, it is important to reduce the number of subnetworks by precomputing which subnetworks are actually needed.

Let $g$ be a rooted, binary phylogenetic tree leaf-labeled by set $\mathcal{X}$ of taxa. For a node $u$ in $g$, we denote by $L(u)$ the set $X' \subseteq X$ that labels the leaves of $g$ that are under node $u$. Consider an internal edge $e = (u, v)$ in $g$ (that is, an edge that is not incident with a leaf). Let $v_1$ and $v_2$ be the two children of $v$, and let $u_1$ be the child of $u$ that is not $v$. We say that edge $e$ is defined by the set $\{L(v_1), L(v_2), L(u_1)\}$ (that is, it is a set of three sets of leaf labels). Finally, we say that a triplet of leaf labels $\{x_1, x_2, x_3\} \subseteq \mathcal{X}$ covers edge $e$ if

$$(x_1 \in L(v_1) \land x_2 \in L(v_2) \land x_3 \in L(u_1)).$$

The algorithm we propose for reducing the number of subproblems to solve on a data set of $m$ loci is as follows:

1. Let $\mathcal{G}$ be a set of $m$ estimated gene trees, and denote by $\mathcal{E}(\mathcal{G})$ the set of all internal edges in the gene trees in $\mathcal{G}$.

2. Compute a smallest set $\Delta = \{\{x_1, x_2, x_3\} : \{x_1, x_2, x_3\} \subseteq \mathcal{X}\}$ such that each edge $e \in \mathcal{E}(\mathcal{G})$ is covered by at least one element of $\Delta$.

3. Infer $|\Delta|$ trinets, one for each element of $\Delta$.

We show how computing set $\Delta$ can be posed as an instance of the Hitting Set Problem, which allows one to make use of many existing algorithmic developments for this problem. The Hitting Set Problem is defined as follows:

**Input:** A collection $C$ of subsets of $S$. 
**Output:** Smallest subset $S' \subseteq S$ that intersects every set in $C$.

To pose our problem of finding a smallest set of 3-taxon subproblems as an instance of the Hitting Set Problem, we define:

- $S$ is the set of all $\binom{|\mathcal{X}|}{3}$ three-taxon subsets of $\mathcal{X}$.
- Let edge $e \in \mathcal{E}(\mathcal{G})$ be defined by the the set $\{A, B, C\}$ of three sets of taxa, as described in the main text. We create set $C_e = \{\{a, b, c\} : a \in A, b \in B, c \in C\}$.

Then,

$$C = \bigcup_{e \in \mathcal{E}(\mathcal{G})} \{C_e\}.$$  

Finding a smallest subset $S' \subseteq S$ amounts to finding the smallest set of 3-taxon sets on which to infer trinets. We implemented the following greedy heuristic for solving the problem given an input set $\mathcal{G}$ of gene trees:

- $\Delta \leftarrow \emptyset$;

- **foreach** $e \in \mathcal{E}(\mathcal{G})$
  
  - if $e$ is not covered by an element in $\Delta$
    
    * Let $\{x_1, x_2, x_3\} \subseteq \mathcal{X}$ be an arbitrary element that covers $e$;
    
    * $\Delta \leftarrow \Delta \cup \{\{x_1, x_2, x_3\}\}$;

- **return** $\Delta$;

After $\Delta$ is generated by the greedy heuristic, we compute $\Delta'$ by the mapping $S$ from alleles in gene trees to leaves in the full network using another greedy heuristic:
• $\Delta' \leftarrow \emptyset$;

• foreach $\{x_1, x_2, x_3\} \in \Delta$

  - $x'_1, x'_2, x'_3 \leftarrow S(x_1), S(x_2), S(x_3)$

  - if $|\{x'_1, x'_2, x'_3\}| = 3$

    * $\Delta' \leftarrow \Delta' \cup \{\{x'_1, x'_2, x'_3\}\}$;

  - else if $|\{x'_1, x'_2, x'_3\}| = 2$

    * Replace one of repeated species with the outgroup;

    * $\Delta' \leftarrow \Delta' \cup \{\{x'_1, x'_2, x'_3\}\}$;

  - else if $|\{x'_1, x'_2, x'_3\}| = 1$

    * Discard;

• return $\Delta'$;

For certain networks (that are automatically identified by the algorithm), the smallest set $\Delta$ of trinets needs to be enriched with additional trinets that are identified in multiple rounds, a step that we discuss and describe as follows, along with the heuristic we implemented for solving the aforementioned problem.

To check whether the set of inferred subnetworks is sufficient to be used by the merger algorithm, we use the following algorithm **Enrich** to find additional subnetworks needed. **Enrich** is a naive version of merger algorithm: instead of building a network, it builds a binary tree according to $\Psi_1, \cdots, \Psi_{|\Delta'|}$. If it cannot build a binary tree, it provides a set of additional subnetworks needed.
• $\Delta'' \leftarrow \emptyset$;

• Reconcile heights of nodes in $\Psi_1, \cdots, \Psi_{|\Delta'|}$;

• Combine EHMs $\mathcal{M}_{\Psi_1}, \cdots, \mathcal{M}_{\Psi_{|\Delta'|}}$ into $\mathcal{M}$;

• Let $M$ be the height matrix with same dimension as $\mathcal{M}$, and let $M(x,y)$ to be the first item of $\mathcal{M}(x,y)$ if $\mathcal{M}(x,y)$ exists; otherwise $M(x,y) = \infty$;

• Select a starting network $\Psi_t$ from $\Psi_1, \cdots, \Psi_{|\Delta'|}$;

• Let $x_1, x_2, x_3$ be the taxon of $\Psi_t$ such that $M(x_1, x_2) < \min(M(x_1, x_3), M(x_2, x_3))$;

• Build a tree $T$ of $x_1, x_2, x_3$, whose internal node is the parent of $x_1$ and $x_2$ with height $M(x_1, x_2)$, and the height of its root is the same as $\Psi_t$;

• Let $Order$ be the order of leaf addition computed from $\Psi_1, \cdots, \Psi_{|\Delta'|}$;

• foreach $x$ in $Order$

  – $x' \leftarrow \arg \min_{x'} \{M(x, x')\}$ such that $x'$ is in $T$;

  – if $M(x, x') = \infty$

    * foreach node $p$ in $T$

      · $y, z \leftarrow$ two arbitrary taxa separated by $p$;

      · $\Delta'' \leftarrow \Delta'' \cup \{(x, y, z)\}$;

    * return $\Delta''$;

  – else if $\exists p$ such that $p$ is an ancestor of $x'$ and $|M(x, x') - h(p)| < \epsilon$

    * foreach pair of taxa $y$ and $z$ separated by $p$
\[ \Delta'' \leftarrow \Delta'' \cup \{x, y, z\} \]

* return \(\Delta''\);

- else

  * Add a leaf labeled by \(x\) to \(T\) by creating a common ancestor \(p\) of \(x\) and \(x'\) such that \(h(p) = M(x, x')\);

  * return \(\emptyset\);

Finally we can change our first step into ReducedSubnetworkInference. (Alg. 5). In this version, the first batch of subnetworks are inferred from \(\Delta'\). Then we call Enrich to enrich the set of subnetworks iteratively. After running this version of first step, the selected starting network and order of leaf addition from the last call of Enrich is kept, and the merger algorithm needs to use the same starting network and order of leaf addition.
Algorithm 5: ReducedSubnetworkInference.

Input: Multilocus sequence alignments $S_1, \cdots, S_m$.

Output: List of subnetworks.

g_1, \cdots, g_m \leftarrow \text{gene tree inference given } S_1, \cdots, S_m ;

Get $\Delta'$ using greedy heuristic from $g_1, \cdots, g_m$;

Infer the subnetwork $\Psi_1, \cdots, \Psi_{|\Delta'|}$ with subproblems in $\Delta'$;

$\Delta'' \leftarrow \text{Enrich}(\Psi_1, \cdots, \Psi_{|\Delta'|});$

while $\Delta'' \neq \emptyset$ do

Infer subnetworks $\Psi_{|\Delta'|+1}, \cdots, \Psi_{|\Delta'|+|\Delta''|}$ corresponding to additional triplets;

$\Delta' \leftarrow \Delta' \cup \Delta'';$

return $\Psi_1, \cdots, \Psi_{|\Delta'|};$

5.2.6 An example of the merger algorithm

Here we give a simple example to illustrate the merger algorithm. Fig. 5.1 shows the true network with 5 taxa as well as its all $\binom{5}{3} = 10$ subnetworks with 3 taxa. The subnetworks in Fig. 5.1(B) are obtained by restricting $\Psi$ to each combination of 3 taxa. Suppose the input sequence alignments are restricted to 10 combinations of 3 taxa, and inference algorithm is called. The inferred subnetworks are shown in Fig. 5.1. The inferred subnetworks are not necessarily identical to true subnetworks due to inference errors. Note that reticulation edges are missing in inferred $\Psi_1, \Psi_5$ and $\Psi_8$. 
Figure 5.1: The true network and its subnetworks. (A) The true network $\Psi$, whose height of each node is indicated by the ticks. (B) The true subnetworks, whose height of each node is indicated by the ticks.
Figure 5.2: **The input to the merger algorithm.** Values near nodes of subnetworks are the height of each node. Colored dots indicate the corresponding nodes belong to the same disjoint set.
5.2.6.1 Reconciling and summarizing the subnetworks

The first step is to reconcile the heights of nodes in each subnetwork. The disjoint sets of nodes are generated by mapping nodes in common binets in the subnetworks. Then the height of each node is assigned according to the average height of nodes in the same set. Fig. 5.3 shows the heights of nodes after reconciliation. Some disjoint sets are shown for illustration.

![Diagram of subnetworks after reconciliation](image)

Figure 5.3: The subnetworks after reconciling their heights. Values near nodes of subnetworks are the height of each node.

Then, EHM is computed for every subnetwork.

\[
\mathcal{M}_{\psi} = \begin{pmatrix}
A & B & C \\
- & [5.03] & [5.03, 8.17] \\
[5.03] & - & [1.94, 8.17] \\
[5.03, 8.17] & [1.94, 8.17] & -
\end{pmatrix}
\]
\[
\begin{align*}
\mathcal{M}_2 &= \begin{pmatrix}
A & B & D \\
- & [4.60, 8.16] & [8.16] \\
[4.60, 8.16] & - & [5.86, 8.16] \\
[8.16] & [5.86, 8.16] & - \\
\end{pmatrix} \\
\mathcal{M}_3 &= \begin{pmatrix}
A & B & E \\
- & [4.60, 8.16] & [9.64] \\
[4.60, 8.16] & - & [9.64] \\
[9.64] & [9.64] & - \\
\end{pmatrix} \\
\mathcal{M}_4 &= \begin{pmatrix}
A & C & D \\
- & [5.31, 8.16] & [8.16] \\
[5.31, 8.16] & - & [2.93, 5.86, 8.16] \\
[8.16] & [2.93, 5.86, 8.16] & - \\
\end{pmatrix} \\
\mathcal{M}_5 &= \begin{pmatrix}
A & C & E \\
- & [5.03, 8.17] & [9.64] \\
[5.03, 8.17] & - & [9.64] \\
[9.64] & [9.64] & - \\
\end{pmatrix}
\end{align*}
\]

\[ \mathcal{M}_{\Psi_7} = \begin{pmatrix} B & C & D \\ - & [2.02, 5.86, 8.16] & [5.86, 8.16] \\ [2.02, 5.86, 8.16] & - & [2.93, 5.86, 8.16] \\ [5.86, 8.16] & [2.93, 5.86, 8.16] & - \end{pmatrix} \]


Finally, an overall EHM $M$ is computed. Take the entry $(B, C)$ for an example. The candidates for $(B, C)$ are $[5.03, 8.17]$, $[5.99, 8.15]$ and $[2.02, 5.86, 8.16]$. $[2.02, 5.86, 8.16]$ has the most elements, so it is selected as the entry in the overall EHM.

$$M_{\Psi,0} = \begin{pmatrix}
C & D & E \\
- & [2.93, 5.86, 8.16] & [9.64] \\
[2.93, 5.86, 8.16] & - & [9.64] \\
[9.64] & [9.64] & -
\end{pmatrix}$$

### 5.2.6.2 Generating a starting network and an order for leaf addition

The “outgroup” is set to E. The score of a subnetwork $\Psi_i$ is computed by

$$s(\Psi_i) + \sum_{1 \leq j \leq k} d(\Psi_i, \Psi_j).$$

Table 5.1 shows how the score of each subnetwork is computed. The subnetwork $\Psi_6$ with taxa A, D, and E, shown in Fig. 5.4(A) is selected as the starting network because it has the lowest backbone score. Then we need to generate an order of attaching new
Table 5.1: The computation of score of each subnetwork for starting network selection.

The score of each network is the sum of two terms in the corresponding column.

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(Ψ_i)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>\sum_{1 \leq j \leq k} d(Ψ_i, Ψ_j)</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

leaves. The guide graph whose nodes are the taxa set, and edges are computed according to reticulations in the subnetworks. Fig. 5.4(B) shows the guide graph. Topological sorting yields \([B, C]\) after removing A, D and E.

Figure 5.4: A starting network and a guide graph for leaf addition. (A) The starting network, and its heights of nodes are represented by the nearby values. (B) The guide graph whose nodes are represented by circles.
5.2.6.3 Adding B

Attachments of B is extracted from subnetworks, as shown in Fig. 5.5. Attachments are clustered by the number of blue nodes in them. Therefore there are two clusters: \{at_{\Psi_1}(B), at_{\Psi_7}(B)\} and \{at_{\Psi_4}(B), at_{\Psi_3}(B), at_{\Psi_8}(B), at_{\Psi_9}(B)\}. For the first cluster, at_{\Psi_1}(B) is chosen (the two items are equivalent, so either one can be chosen). For the second cluster, at_{\Psi_8}(B) is chosen, because the parent of B has the lowest height in it. Therefore, \(H(B) = \{at_{\Psi_1}(B), at_{\Psi_8}(B)\}\).

![Diagram](image)

Figure 5.5: Attachments of B. Colored nodes and edges belong to an attachment of B in a subnetwork. Blue nodes are in set \(rt(B)\) of corresponding subnetwork, and red nodes are in set \(it(B)\) of corresponding subnetwork. Attachments surrounded by dotted rectangles are selected in \(H(B)\).

Alg. 5 is called and sorted HT pairs in \(\mathcal{H}\) are

- (4.60, A)
Enumerate tries to connect attachments in $\mathcal{H}$ with the draft network.

1. $at_{\Psi_1}(B)$ is tried. (4.60, A) is resolved, and this yields the network in Fig. 5.6(A).

2. $at_{\Psi_8}(B)$ is tried. In the either permutation, after (4.60, A) is resolved, the draft network is in Fig. 5.6(B). The node with in-degree 0 is removed and this yields the network in Fig. 5.6(A). Then (5.86, D) and naturally all remaining HT pairs are resolved. This yields network in Fig. 5.6(C). The score of the network in Fig. 5.6(A) is 11, and the score of the network in Fig. 5.6(C) is 1, therefore the network in Fig. 5.6(C) is selected, and after reconciling its heights of nodes, we get a new backbone network in Fig. 5.6(D).
Figure 5.6: **Enumeration for B.** (A) The stored network when \((4.60, A)\) is resolved. (B) The draft network when examining \(at_{\Psi_8}(B)\) with \((4.60, A)\) resolved. The arrows indicate the direction of branches in the attachment. The dotted branch is removed when storing this network. (C) The stored network when all HT pairs are resolved. (D) The new backbone network after adding taxon B and reconciling heights of nodes.

### 5.2.6.4 Adding C

Attachments of C is extracted from subnetworks, as shown in Fig. [5.7](#). Attachments are clustered by the number of blue nodes in them. There are three clusters: \(\{at_{\Psi_8}(C)\}\), \(\{at_{\Psi_1}(C), at_{\Psi_5}(C), at_{\Psi_7}(C)\}\) and \(\{at_{\Psi_4}(C), at_{\Psi_10}(C)\}\). For the first cluster, \(at_{\Psi_8}(B)\) is chosen. For the second cluster, \(at_{\Psi_5}(C)\) is chosen (the three items are equivalent, so either one can be chosen). For the third cluster, \(at_{\Psi_10}(C)\) is chosen (the two items are equivalent, so either one can be chosen). Therefore, \(H(C) = \{at_{\Psi_4}(C), at_{\Psi_5}(C), at_{\Psi_10}(C)\}\).

Alg. 3 is called and sorted HT pairs in \(\mathcal{H}\) are

- (2.02, B)
- (2.93, D)
- (5.03, A)
Figure 5.7: Attachments of C. Colored nodes and edges belong to an attachment of C in a subnetwork. Blue nodes are in set $rt(C)$ of corresponding subnetwork, and red nodes are in set $it(C)$ of corresponding subnetwork. Attachments surrounded by dotted rectangles are selected in $H(C)$.

- (5.86, B)
- (5.86, D)
- (8.16, B)
- (8.16, D)
- (8.17, A)
- (9.64, E)

Enumerate tries to connect attachments in $\mathcal{H}$ with the draft network.

1. $at_{Ψ_8}(C)$ is tried. (2.02, B) is resolved, and this yields the network in Fig. 5.8(A).
2. \( at_{Ψ₃}(C) \) is tried. In the either permutation, \((2.02, B)\) is resolved, then \((2.93, D)\).

This yields both networks in Fig. \ref{fig:networks}. (A)(B).

3. \( at_{Ψ₁₀}(C) \) is tried. Take one permutation in Fig. \ref{fig:networks} (C) for an example. \((2.02, B)\) is resolved, and yields Fig. \ref{fig:networks} (A). Then \((2.93, D)\) is resolved and draft network is in Fig. \ref{fig:networks} (D), note that the node with height 3.97 is removed when storing the draft network, so it yields the network in Fig. \ref{fig:networks} (F). When resolving \((5.03, A)\), the draft network is in Fig. \ref{fig:networks} (E), and the node with height 3.97 creates negative branch length thereby the draft network is discarded.

Note that the topology of the networks in Fig. \ref{fig:networks} (B)(F) are identical, the only difference is the height of one reticulation node. The score of the network in Fig. \ref{fig:networks} (A) is 19, while the score of the network in Fig. \ref{fig:networks} (B)(F) is 4, therefore the network in Fig. \ref{fig:networks} (B)(F) is selected, and after reconciling the heights of nodes, we get a new backbone network in Fig. \ref{fig:networks} (G).

Since all taxon are in the network in Fig. \ref{fig:networks} (G), it is the final output of merger algorithm. Its topology of is identical to the true network, and the heights of its nodes are close to the true ones.
Figure 5.8: **Enumeration for C.** (A) The stored network when \((4.60, A)\) is resolved. (B) The stored network when \((4.60, A)\) and \((2.93, D)\) are resolved. (C) The attachment \(at_{\Psi_{10}}(C)\) and the permutation of \(rt_{\Psi_{10}}(C)\) with indices shown in parentheses. (D)(E) Draft networks when trying \(at_{\Psi_{10}}(C)\). (F) stored network when trying \(at_{\Psi_{10}}(C)\). (G) The final output.

### 5.3 Performance

The way we ran our method is as follows: For each subproblem, MCMC_SEQ [WN18] was run and a sample of subnetworks was collected from the posterior. We then selected one subnetwork randomly from the samples of each subset, and applied our merger algorithm. This step was repeated 100 times, and resulted in 100 candidate networks on the full set of taxa. We selected the final network as follows. if a network topology appeared in two thirds or more of the 100 networks, it was selected as the final result; otherwise, we identify the most common topology for each of the subnetwork distributions from MCMC_SEQ.
Then, we select the network which maximizes the number of subnetworks, contained in that network, which match those topologies. The parameters of the final network are averaged from the networks with same topology.

Since our algorithm for combining subnetworks into a network on the full set of taxa is a heuristic with no established theoretical guarantees, we first set out to study its accuracy on a large number of networks. We then studied the performance of our full approach on simulated multi-locus data sets, and finally analyzed a biological data set.

### 5.3.1 Accuracy of the merger algorithm

We generated 10,000 16-taxon networks using a birth-hybridization model, and for each network, an outgroup was added to create a 17-taxon network. We restricted each of the 10,000 17-taxa networks to every combination of 3 taxa to produce \( \binom{17}{3} = 680 \) trinets that were used as input to our merger algorithm that combines the trinets into a network on the full set of taxa. We then inspected the accuracy of the resulting networks. Fig. 5.9 shows the number of data sets on which the merger algorithm inferred the correct network with 10,000 17-taxon networks. As Fig. 5.9 shows, in total, 9,838 out of 10,000 inferred networks are identical to their corresponding true networks. When the true network had 0 or 1 reticulations, the algorithm always returned the correct network. Furthermore, the few cases where an incorrect network was returned mostly correspond to large numbers of reticulations (even in those cases, the computed network was very similar to the true one).

To examine the performance of the merger algorithm with and without reduced number of subproblems for large networks, we generated 100 41-taxon networks and 81-taxon
networks using a birth-hybridization model (each network had a designated outgroup that did not involve hybridization with any other taxa). We simulated 1,000 gene trees within the branches of each network, using the program ms [Hud02], and generated the full set of all true trinets as well as subset obtained by our algorithm for reducing the number of trinets. We used each set of trinets as input to our merger algorithm. We inspected the accuracy in terms of whether the inferred network is identical to the true network. The results, as well as other characteristics of the data, are shown in Table 5.2. When the full
set of trinets was used as input, all trinets were inferred in parallel in a single batch. When the reduced set of trinets was used as input, the first batch always consists of the set of reduced trinets being inferred in parallel. However, as we discussed above, in some cases, multiple rounds of enrichment of the reduced set of trinets are performed. Each such round corresponds to an addition batch where all new trinets in that round are inferred in parallel.

The table shows several important points. The algorithm achieves almost perfect accuracy on the 41-taxon networks, and perfect accuracy on the 81-taxon networks, when the full set of trinets is used. Our heuristic for reducing the number of trinets achieves two orders of magnitude reduction in the number of trinets, resulting in one or two orders of magnitude reduction in the running time. The accuracy decreases when the reduced set of trinets is used, since some information on the full network is lost by this reduction. We identify the problem of obtaining a better reduced set of trinets as a direction for future research.

One reason the algorithm performs better on the larger networks (81-taxon networks) is that for a fixed number of reticulations, those reticulations would be sparser on a network with 81 taxa than on a network with 41 taxa, making the inference of the former less challenging. Fig. 5.10 breaks the accuracy results of our algorithm on the 41- and 81-taxon networks by the number of reticulations in these networks.
Figure 5.10: **Correctness of inferred networks from correct trinets, categorized by the number of reticulations in the true networks.** (a) Results from 100 41-taxon networks. (b) Results from 100 81-taxon networks. Blue: the number of cases where the inferred network is identical to the true one when using either the full or reduced set of trinets. Orange: the number of cases where the inferred network is identical to the true one only when the full, but not reduced, set of trinets is used. Grey: the number of cases where the inferred network is different from the true one, regardless of whether the full or reduced set of trinets was used.
Table 5.2: **Results of merger algorithm for large networks.** Full and Reduced correspond to the full set of trinets and the reduced set of trinets, and \( n \) is the number of leaves in the network. Each batch consists of multiple trinet inferences that are all run in parallel. ‘Candidates enumerated’ is the number of new backbone networks that are proposed and examined by the algorithm during the full network construction. Accuracy is measured as the percentage of data sets in which the constructed network is identical to the true network. The average running time in seconds is the time it took to construct the full network from the set of trinets.

<table>
<thead>
<tr>
<th>( n )</th>
<th>Quantity</th>
<th>Full</th>
<th>Reduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>Number of trinets</td>
<td>10660</td>
<td>151 ( \sim ) 386</td>
</tr>
<tr>
<td></td>
<td>Number of batches</td>
<td>1</td>
<td>1 ( \sim ) 6</td>
</tr>
<tr>
<td></td>
<td>Candidates enumerated</td>
<td>39 ( \sim ) 225</td>
<td>39 ( \sim ) 228</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>98%</td>
<td>83%</td>
</tr>
<tr>
<td></td>
<td>Average running time (s)</td>
<td>50.93</td>
<td>3.57</td>
</tr>
<tr>
<td>81</td>
<td>Number of trinets</td>
<td>85320</td>
<td>347 ( \sim ) 772</td>
</tr>
<tr>
<td></td>
<td>Number of batches</td>
<td>1</td>
<td>2 ( \sim ) 9</td>
</tr>
<tr>
<td></td>
<td>Candidates enumerated</td>
<td>80 ( \sim ) 155</td>
<td>80 ( \sim ) 150</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>100%</td>
<td>88%</td>
</tr>
<tr>
<td></td>
<td>Average running time (s)</td>
<td>1077.16</td>
<td>10.90</td>
</tr>
</tbody>
</table>

### 5.3.2 Accuracy on simulated multi-locus data sets

We now set out to study the performance of our approach on simulated multi-locus sequence data, where the method is applied to the sequence data directly. Given that compu-
tational complexity of Bayesian inference of trinets [WN18], we focus our attention here on a subset of 24 phylogenetic networks that we sampled to reflect varying complexity levels. As discussed in [ZYN16, EOZN18], the complexity of phylogenetic networks arises not only from the number of leaves or number of reticulation nodes, but also in how the reticulation nodes are structured in the network. To allow for a careful assessment of the accuracy of our approach, we define a simple complexity measure of networks as follows.

We define the complexity of $\Psi$ as

$$\sum_{r \in R(\Psi)} |L(r)| + |L(p_1(r))| + |L(p_2(r))| + |B| \cdot |AR_\Psi(r)|,$$

where $L(u)$ is the set of leaves under node $u$, and $p_1(u)$ and $p_2(u)$ are the two parents of reticulation node $u$.

We selected the 24 networks from the 10,000 as follows. All simulated networks with 0 to 5 reticulation nodes were sorted by their complexities. For each of the six numbers of reticulation nodes, we selected four networks: the one with the minimum complexity, the one with the maximum complexity, and the two networks at tertiles. The 24 networks were divided into three groups of 8 “easy” networks (E), 8 “medium-difficulty” networks (M), and 8 “hard” networks (H), and are shown in Table 5.3. We used these 24 networks as the ground truth and simulated multi-locus sequence from these 24 networks.

For each of the 24 networks, we generated the full set of all true trinets as well as subset obtained by our algorithm for reducing the number of trinets. Then, for each set of trinets (full or reduced), we perturbed the heights of the nodes in each trinet randomly by 0.1% and repeated this 100 times to obtain 100 “ideal” MCMC-like samples of trinets. We then used the trinet sets as inputs to our merger algorithm and inspected the resulting networks.
The algorithm obtained the correct networks in all 24 cases regardless of whether the full or reduced set of trinet “samples” were used. While this result is perfect, Bayesian MCMC in practice is not guaranteed to yield as accurate a sample as the one we used here. Therefore, we next set out to study the performance of the method when we use sequence data of the multiple loci.

For each of the 24 networks, we simulated 100 gene trees, with two individuals per species, for 100 loci using the program **ms** [Hud02], and generated sequence alignments of length 1,000 for each locus using **Seq-gen** [RG97] under GTR model. In other words, each locus consists of 34 aligned sequences. For each data set, we inferred subnetworks using **MCMC-SEQ** [WN18] as implemented in **PhyloNet** [WYZN18] with $2 \times 10^6$ iterations, $1 \times 10^6$ burn-in iterations, and one sample collected per $5 \times 10^3$ iterations. To obtain the first state for the method, we inferred gene trees for the individual loci using **IQ-TREE** [NSvHM14], optimized their branch lengths using local search, and the resulting gene trees were used as the starting gene trees in the MCMC chain.

For each data set, the running time to infer all trinets is shown in Fig. 5.11(a). This analysis was performed on **NOTS** (Night Owls Time-Sharing Service), which is a batch scheduled High-Throughput Computing (HTC) cluster. The average cost to infer all trinets for a data set was 1636.82 CPU-hours, which means it takes about an hour to infer a trinet with a dual-core machine. Since the inferences of trinet are independent of each other, this task is embarrassingly parallel. Fig. 5.11(b) shows the accuracy of the inferred trinets. The figure shows that the more complex the true network, the harder it is to infer their subnetworks.

We then used the inferred trinets as input to our merger algorithm. The merger algo-
Figure 5.11: **Running times and accuracy for the inferred trinets.** (a) The total running time in CPU-hours to infer all trinets for each data set. (b) Accuracy of the inferred trinets. The number of data sets where the inferred trinet is correct (blue), the inferred trinet is inside the true network (orange), and all other cases (grey), are shown.

The algorithm ran on a Macbook Pro with 2.9 GHz Intel Core i5. We used both the full and reduced sets of inferred trinets. The reduced sets contain between 61 and 132 trinets, which is a major reduction (especially when considering the running time, as shown in Fig. 5.11(a)) over the full set, which contains 680 subnetworks. Most data sets only need one batch of inference, 3 data sets need 2 batches, and 1 data set needs 3 batches. The time that our algorithm took to merge the trinets into a full network (repeated 100 times) ranged between 148 and 1538 seconds when the full set of trinets was used, and between 44 and 141 seconds when the reduced set of trinets was used. This shows the additional efficiency gained by reducing the number of trinets.

Finally, we fed the full and reduced sets of trinets to our merger algorithm and compared the inferred networks to the true ones. In measuring the difference between a true network $\Psi_t$ and an inferred network $\Psi_i$, we quantified false positive and false negative rates...
as follows. We find the backbone $\Psi_i'$ of $\Psi_i$ and backbone $\Psi_t'$ of $\Psi_t$ whose topological differences [Nak10b] are smallest and have the largest number of reticulation nodes among all such pairs of backbones. If the topological difference is 0, the inferred network has a backbone inside the true network. We compute the true positives as the number of nodes remaining in $\Psi_t'$, minus the topological difference of $\Psi_i'$ and $\Psi_t'$. We compute the false positives as the number of nodes deleted from $\Psi_i$ to $\Psi_i'$, plus the topological difference of $\Psi_i'$ and $\Psi_t'$. The false negative rate is computed by normalizing the true positives by the number of nodes in $\Psi_t$ and subtracting it from 1, and the false positive rate is computed by normalizing the false positives by the number of nodes in $\Psi_i$.

The inferred network was identical to the true network in 12 out of 24 data sets when full set of trinets were used. When the reduced set of trinets was used, 9 inferred networks were identical to their corresponding true networks. We plot the false positives and false negatives for the data sets where the inferred network is not identical to the true one in Fig. 5.12(a). As the results show, not much accuracy is lost when using the reduced set of trinets. In particular, for four data sets, the false negative rate when using the full set of trinets is higher than its counterpart when using the reduced set. On the other hand, more networks inferred from the reduced set have slightly higher false positive rates. It is important to note here that these results combined with the fact that all 24 inferred networks are completely accurate when using error-free trinets shows that the error in the final networks is mainly due to inaccuracy of the trinets, rather than the merger algorithm.

Finally, we compare the accuracy of the method to the only other statistical inference method that can scale to these data sets, namely maximum pseudo-likelihood [YN15]. As the method of [YN15] requires gene trees as input, we ran it on the gene trees inferred by
Figure 5.12: **Accuracy of the inferred networks, and comparison to maximum pseudo-likelihood.** (a) The false positives and false negatives for the data sets where the inferred network is not identical to the true network. Squares correspond to hard networks, crosses correspond to medium-difficulty networks and triangles correspond to easy networks. Blue, red and green correspond to results based on the full and reduced sets of trinets, and maximum pseudo-likelihood, respectively. (b) The accuracy of our method on the full set of trinets (left set of bars) and on the reduced set of trinets (middle set of bars), and the accuracy of maximum pseudo-likelihood (right set of bars). Blue corresponds to the data sets where the inferred network is identical to the true network; orange corresponds to the data sets where the inferred network contains a backbone network that is present in the true network; grey corresponds to all other cases.

IQ-TREE, with the maximum number of reticulations set to 5 and the number of runs set to 20. Fig. 5.12(b) shows the results of this comparison. These results clearly show that our approach here outperforms maximum pseudo-likelihood, and there could be several explanations for this. First, maximum pseudo-likelihood is not good at estimating the correct number of reticulations, so it could be that the networks obtained by the method have un-
necessary reticulation nodes. Second, maximum pseudo-likelihood searches the network space and could get stuck in local maxima, whereas our proposed approach here avoids such a search. It is important to also comment on the decreased accuracy of our approach when using a reduced set of trinets. As the set of trinets is much smaller than the full set, the method becomes more sensitive to inaccuracy in the inferred trinets, since when using the full set of trinets, signal from multiple trinets could mask the estimation error. All these results combined show that our proposed approach can produce very accurate results, especially when the individual trinets are accurately estimated.
Table 5.3: **24 true networks.** The 24 networks selected as the true networks in simulation study, followed by their subjective difficulties.
Table 5.3 Continued

<table>
<thead>
<tr>
<th>Reticulation</th>
<th>Difficulty</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Easy</td>
<td>The reticulation is hard to be identified since it only depends on D.</td>
</tr>
<tr>
<td></td>
<td>Middle</td>
<td>The reticulation is too deep to be correctly identified.</td>
</tr>
<tr>
<td>1</td>
<td>Easy</td>
<td>The reticulation is easy to identify.</td>
</tr>
<tr>
<td>Difficulty</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Easy</td>
<td>2 reticulations. Difficulty: Easy.</td>
<td></td>
</tr>
<tr>
<td>Middle</td>
<td>2 reticulations. Difficulty: Middle. The dependency of 2 reticulations makes inference not easy.</td>
<td></td>
</tr>
<tr>
<td>Middle</td>
<td>2 reticulations. Difficulty: Middle. The dependency of 2 reticulations makes inference not easy.</td>
<td></td>
</tr>
<tr>
<td>Hard</td>
<td>2 reticulations. Difficulty: Hard. The reticulations are deep and dependent.</td>
<td></td>
</tr>
</tbody>
</table>
3 reticulations. Difficulty: Easy.

3 reticulations. Difficulty: Middle. The dependency of up 2 reticulations makes inference not easy.

3 reticulations. Difficulty: Middle. The dependency of 2 reticulations above K and P makes inference not easy.

3 reticulations. Difficulty: Hard. The reticulations are deep and dependent.
Table 5.3 Continued

4 reticulations. Difficulty: Middle. The number of reticulations makes inference not easy.

4 reticulations. Difficulty: Hard. The number of reticulations and dependencies make inference hard.

4 reticulations. Difficulty: Hard. The number of reticulations and dependencies make inference hard.
| 5 reticulations. Difficulty: Middle. The number of reticulations makes inference not easy. |
| 5 reticulations. Difficulty: Hard. The number of reticulations and dependencies make inference hard. |
| 5 reticulations. Difficulty: Hard. The number of reticulations and dependencies make inference hard. |
| 5 reticulations. Difficulty: Hard. The number of reticulations and dependencies make inference hard. |
5.3.3 Inference on an empirical data set

We analyzed a data set of multi-locus sequence alignments of multiple Australian rainbow skinks \([BPS+18]\), where 11 taxa with 22 individuals were selected from the full data set. At first we computed the maximum pairwise distance of each locus using IQ-TREE \([NSvHM14]\), and we excluded the loci with maximum pairwise distance larger than 0.2, as that would imply impossible deep coalescence times. We then randomly selected 100 loci and used their sequence alignments as the input.

![Inferred network](image)

**Figure 5.13:** The inferred network for the empirical data set. The reticulation, with inheritance probabilities (blue), is shown by the dashed line.

The first step of our method is inferring subnetworks. So we restricted the data set with 11 taxa to every combination of 3 taxa, then we added *Lampropholis guichenoti* into every subproblem to root the subnetworks. Therefore for every subproblem, 4-taxon networks were inferred and the number of subproblems remains \(\binom{11}{3} = 120\). We ran MCMC-SEQ \([WN18]\) for 6,000,000 iterations with 3,000,000 burn-in steps, collecting a sample for ev-
We inferred gene trees using IQ-TREE [NSvHM14], and their branch lengths were optimized individually using local search. The resulting gene trees were used as the starting point of MCMC chain, and all gene tree topologies were fixed during Bayesian sampling. This analysis was performed on NOTS (Night Owls Time-Sharing Service). We used 2 CPU cores running at 2.6GHz, and 8G RAM for each subproblem. It took 3,670 CPU-hours to infer all subnetworks. Then we used the inferred subnetworks as the input to our merger algorithm to merge them on a Macbook Pro with 2.9 GHz Intel Core i5. It took 53.1 seconds to merge the subnetworks and generate the final result. The inferred network is shown in Fig.5.13. The ingroup result agrees with the known analysis of this data set. The topological relationships of the *Carlia* clade and the *Lygisaurus* clade are identical to Fig. 2 in [BPS+18].

For comparison, we also ran the maximum pseudo-likelihood method of [YN15] on this data set, using the inferred gene trees as the input. The number of runs was set to 10. The number of reticulations allowed was set to 0, 1 and 2. The inferred networks are shown in Fig. 5.14. The inferred species tree was identical to the backbone tree in the inferred network using our merger algorithm. However, that is no longer the case when reticulations are added by the method.
Figure 5.14: The inferred networks for the empirical data set using maximum pseudo-likelihood. Top: the inferred network when no reticulation was allowed. Middle: the inferred network when 1 reticulation was allowed. Bottom: the inferred network when 2 reticulations were allowed. The reticulations, with inheritance probabilities (blue), are shown by the dashed lines.
5.4 Discussion

In this chapter, we proposed a divide-and-conquer approach for large-scale phylogenetic network inference. The approach makes use of inferred subnetworks—topologies and divergence times—on overlapping subsets of the taxa to obtain a phylogenetic network on the full data set. We demonstrated the accuracy and efficiency of our approach on simulated and biological data sets.

While we illustrated the performance of the algorithm on subproblems of size 3 (three taxa), the merger algorithm we introduced works on subnetworks with any number of taxa. There is a tradeoff between the size of the subproblems, the running time, and the accuracy. If the number of taxa in the full data set is \( n \), then the full set of subnetworks on \( k \) leaves consists of \( \binom{n}{k} = O(n^k) \). For example, for \( n = 100 \) and \( k = 5 \), the algorithm would have to infer on the order of \( 10^{10} \) 5-subnetworks. Not only is this number large by itself, but the inference of each 5-subnetwork is much more demanding computationally than that of trinets.
Chapter 6

PhyloNet Implementation and Usage

The methods are implemented in the publicly available software package PhyloNet\cite{WYZN18}.

The method to perform maximum pseudo-likelihood estimation of a phylogenetic network given bi-allelic markers can be used by calling the command \texttt{MLE\_BiMarkers} in PhyloNet.

The method to merge subnetworks into a full network can be used by calling the command \texttt{NetMerger}. Their usages are described in the following sections.

6.1 \texttt{MLE\_BiMarkers}

6.1.1 Requirements

- Java 1.8 or higher
- Jeigen

6.1.2 Usage

### 6.1.2.1 Search settings

- **-mnr numRuns**
  The number of iterations of simulated annealing. The temperature of simulated annealing is reset in the beginning of each iteration, then the temperature reduces gradually as more states are examined. By doing this, the search can jump out of local optimum in the beginning of one iteration easily, then random walk in the space of phylogenetic networks is performed during each iteration.

- **-mec maxExaminationsCount**
  The maximum allowed times of examining a state during one iteration. During one iteration of simulated annealing, each state is obtained by random walk in the space of phylogenetic networks. A state is proposed by randomly altering the topology or parameters in the previous state, then the new state is examined, and can be accepted or rejected. If the number of states examined exceeds this limit, the current iteration
terminates, and a new iteration starts.

- **-mno numOptimums**

  The number of optimal networks to output. The optimal networks are outputted after every iteration. The optimal networks outputted are the optimal networks in any state examined in any iteration.

- **-mf maxFailures**

  The maximum allowed times of failures to accept a new state during one iteration. If the number of times when new purposed states are continuously rejected exceeds this limit, the current iteration terminates, and a new iteration starts.

- **-pl parallelThreads**

  The number of threads running in parallel. The computation of pseudo-likelihood is parallelized since the likelihood of trinets can be computed independently. This number of threads indicates how many threads are used for computation of pseudo-likelihood. However, more threads don’t necessarily mean faster computations usually. In practice, the user needs to figure out the best number of threads by experimenting on a smaller data set and see whether the inference is faster by increasing the number of threads.

### 6.1.2.2 Inference settings

- **-mr maxReticulation**

  The maximum number of reticulation nodes in the sampled phylogenetic networks. This number is a bound on the number of reticulations that the method explores dur-
ing the search. However, this does not mean that the inferred network has to have this number of reticulations. In theory, this number can be set to a very large value so as not to impose any real bound. However, in practice, the number of reticulations can affect the running time. Furthermore, in the absence of a real criterion for model selection, setting this parameter to a large value might result in overly complex networks. We recommend that the user sets the parameter at a value that is “reasonable” to them, based on knowledge of the data set.

- **-tm taxonMap**
  Gene tree / species tree taxa association. For example, if the gene tree is “(((a1,a2),(b1,b2)),c);” and the species tree is “((a,b),c);”, the command should be `-tm ⟨a:a1,a2; b:b1,b2;c:c⟩`. If the set of taxa appeared in this mapping is a subset of input data, the subset of input data will be used for the inference.

- **-fixtheta theta**
  Fix the population mutation rates associated with all branches of the phylogenetic network to this given value (theta). By default, the method estimates a constant population size across all branches.

- **-esptheta**
  If specified, the program estimates the mean value of prior of population mutation rates.
6.1.2.3 Start State Settings

- **-snet**
  Specify the starting network. The input network should be ultrametric with divergence times in units of expected number of mutations per site, inheritance probabilities and population sizes in units of population mutation rate (optional).

- **-ptheta startingThetaPrior**
  Specify the mean value of prior of population mutation rate (startingThetaPrior). If -esptheta is used, startingThetaPrior will be treated as the starting value, otherwise startingThetaPrior will be treated as the fixed mean value of prior of population mutation rates.

6.1.2.4 Data related settings

- **-diploid**
  Specify whether sequence sampled from diploids. If the sequence is from diploid and there are not dominant markers, the characters in the sequence should be ‘0’, ‘1’ or ‘2’. ‘0’ and ‘2’ are the homozygotes and ‘1’ is the heterozygote state.

- **-dominant dominantMarker**
  Specify which marker is dominant if the data is dominant. The dominant marker can either be ‘0’ or ‘1’. Only use when “-diploid” is specified. If this option is specified, the characters in the sequence should be ‘0’ or ‘1’.

- **-op**
  Specify whether or not to ignore all monomorphic sites. If this option is used, the
data will be treated as containing only polymorphic sites, and all monomorphic sites
are ignored. Then the frequencies of the monomorphic sites will be computed by the
likelihood function.

- **-pi0 value**

Specify the stationary distribution of marker ‘0’. Value should be between 0 and 1.
If not specified, the stationary distribution will be calculated from input data.

### 6.1.3 Example

```nexus
#NEXUS
Begin data;
Dimensions ntax=5 nchar=100;
Format datatype=dna symbols="012" missing=? gap=-;
Matrix

A_0 10010110101010110010000101010101
C_0 10011110111010110010010101010101
L_0 100101101010011100100001010101
Q_0 10010110101010110010010101010101
R_0 10010110101010110011010100010101
;End;
BEGIN PHYLONET;
```
MLE_BiMarkers -pseudo -mnr 10 -mec 50000 -mno 20 -mf 100 -
\[ \rightarrow \text{pi0 0.5 -dd -mr 1 -pl 8 -pttheta 0.006 -thetawindow} \]
\[ \rightarrow 0.006 -sd 12345678 -taxa (A_0,C_0,L_0,R_0,Q_0) -tm <A: \]
\[ \leftrightarrow \text{A}_0; \text{C:C}_0;\text{L:L}_0;\text{Q:Q}_0;\text{R:R}_0> ; \]
END;

This command will run maximum pseudo-likelihood estimation of 10 iterations with 20 optimal networks printed. And after 100 times of failure to accept a new state, or after 50000 examinations of new states, it will start a new iteration. We will estimate population mutation rates for all branches, and they are the same across all branches. The number of reticulation nodes is limited to 1. The starting value of population mutation rate is given by 0.006. We use the random seed of 12345678. In the end, we indicate the mapping from taxa to species.

### 6.2 NetMerger

#### 6.2.1 Requirements

- Java 1.8 or higher
6.2.2 Usage

6.2.2.1 The mode to reduce the number of trinets

This mode will generate a set of triplets using the heuristic to reduce the number of sub-problems, by covering the internal branches of gene trees.

```bash
```

- **-outgroup outgroup**
  
  The taxon in species network designated to be the “outgroup”, which is defined in Chapter 5 of this thesis.

- **-tm taxonMap**
  
  Gene tree / species tree taxa association of the full data set. The syntax is the same as “MLE_BiMarkers”.

- **-gts gtsFilePath**
  
  The path to the text file, in which each line is a gene tree.

- **-triplets tripletsFilePath**
  
  The path to the file to be written with triplets. Each line of the file has three species, separated by space.
6.2.2.2 The mode to divide the data set into subsets

In this mode, user can provide a “template” nexus file with regular “MCMC_SEQ” or “MCMC_BiMarkers” command. A “template” nexus file can be used as the input of PhyloNet with a large data set. PhyloNet should be able to start running with the “template” nexus file, but it does not necessarily generate results. Then this mode will generate a bunch of new nexus files, under the same folder as the template, and replace the taxon mapping in the template file by a subset of that mapping. Therefore if PhyloNet uses a new nexus file as an input, a trinet will be inferred. The user can specify whether the full set or reduced set of trinets needs to be inferred.

```
NetMerger -mode "Nex" [-nex nexFilePath] [-tm taxonMap] [-triplets tripletsFilePath]
```

- **-nex nexFilePath**
  The path to the “template” nexus file.

- **-tm taxonMap**
  Gene tree / species tree taxa association of the full data set. The syntax is the same as “MLE_BiMarkers”.

- **-triplets tripletsFilePath**
  The path to the text file with triplets. Each line of the file has three species, separated by space. This is optional: if not specified, the full set of subproblems will be generated.
6.2.2.3 The mode to merge subnetworks

Users can run PhyloNet with generated new nexus files. After trinets are inferred, those results generated by MCMC_SEQ or MCMC_BiMarkers can be merged by NetMerger. If the set of trinets needs to be enriched, a list of triplets will be provided, otherwise a final network is produced.

NetMerger -mode "Result" [-cl chainLength] [-bl burnInLength] [-sf sampleFrequency] [-outgroup outgroup] [-inputFolder inputFolder]

- -cl chainLength
  The length of the MCMC chain used in trinet inference.

- -bl burnInLength
  The number of iterations in burn-in period of MCMC inference used in trinet inference.

- -sf sampleFrequency
  The sample frequency in MCMC inference used in trinet inference.

- -outgroup outgroup
  The taxon in species network designated to be the “outgroup”, which is defined in Chapter 5 of this thesis.

- -inputFolder inputFolder
  The folder where results of MCMC_SEQ or MCMC_BiMarkers locate. Each file ends in “.out” is a single inference of a subnetwork.
6.2.3 Example

6.2.3.1 Reduce the number of trinets

```
#NEXUS

BEGIN PHYLONET;

NetMerger -mode "Triplets" -gts "~/Documents/BioinfoData/SuperNetwork/NetMerger/gts.txt"
-triplets "~/Documents/BioinfoData/SuperNetwork/NetMerger/triplets.txt"
-outgroup "Z"
-tm <A:A_0,A_1;B:B_0,B_1;C:C_0,C_1;D:D_0,D_1;E:E_0,E_1;F:F_0,F_1;G:G_0,G_1;H:H_0,H_1;I:I_0,I_1;J:J_0,J_1;K:K_0,
-K_1;L:L_0,L_1;M:M_0,M_1;N:N_0,N_1;O:O_0,O_1;P:P_0,P_1;
-Z:Z_0,Z_1> ;

END;
```

This command writes triplets to file “~/Documents/BioinfoData/SuperNetwork/NetMerger/triplets.txt” according to the gene trees in “~/Documents/BioinfoData/SuperNetwork/NetMerger/gts.txt”.

6.2.3.2 Divide data set into subsets

```
#NEXUS
```
BEGIN PHYLONET;

NetMerger -mode "Nex" -nex "/Users/zhujiafan/Documents/BioinfoData/SuperNetwork/NetMerger/newtest.nex"
-triplets "/Users/zhujiafan/Documents/BioinfoData/SuperNetwork/NetMerger/triplets.txt"
-tm <A:A_0,A_1;B:B_0,B_1;C:C_0,C_1;D:D_0,D_1;E:E_0,E_1;F:
  F_0,F_1;G:G_0,G_1;H:H_0,H_1;I:I_0,I_1;J:J_0,J_1;K:K_0,
  K_1;L:L_0,L_1;M:M_0,M_1;N:N_0,N_1;O:O_0,O_1;P:P_0,P_1;
  Z:Z_0,Z_1> ;

END;

This command produces “newtest_0.nex”, “newtest_1.nex”, etc, under the folder “/Users/zhujiafan/Documents/BioinfoData/SuperNetwork/NetMerger/”. The number of new nexus files equals to the number of triplets provided in “/Users/zhujiafan/Documents/BioinfoData/SuperNetwork/NetMerger/triplets.txt”. Each new nexus file can be used as the input of PhyloNet.

### 6.2.3.3 Merge subnetworks

#NEXUS

BEGIN PHYLONET;
This command parses all files end with “.out” under the folder “/Users/zhujiafan/Documents/BioinfoData/results/”, then merges those trinets generated by Bayesian inference. The Bayesian inference is performed by 6,000,000 iterations of MCMC with 3,000,000 burn-in iterations, and one sample is obtained for every 5,000 iterations. The command terminates and provides additional triplets if the set of trinets needs to be enriched, otherwise it produces the final network.
Chapter 7

Conclusions

In this thesis, I discussed the complexity of phylogenetic networks in terms of the parental trees. Then to improve the scalability to infer phylogenetic networks, I proposed two novel methods: the pseudo-likelihood of a phylogenetic network given bi-allelic markers, and the inference of phylogenetic networks based on divide-and-conquer technique. I implemented both methods in the publicly available software package PhyloNet.

However, my two methods also raise open questions. For the first method, even though the pseudo-likelihood formulation provides accurate inferences, it still does not circumvent the challenging problem of searching the space of large networks. Developing more efficient moves for walking the space of phylogenetic networks is necessary. Also, when the number of individuals of one species is larger, and the trinet is more complex, it is even very hard to compute the likelihood for a single trinet. Therefore methods to compute likelihood approximately or even to circumvent full likelihood computation should be developed.

For the second method, two bottlenecks of the method are the number of subproblems to analyze, and the time it takes to infer a subnetwork on each subproblem using compute-
heavy approaches such as Bayesian inference. To address the former, we introduced a formulation for reducing the number of subproblems to solve and demonstrated its effect on the efficiency and accuracy of the obtained results. However, our solution is a heuristic, and via our reduction of the problem to the Hitting Set Problem, one future direction is to explore the efficiency and accuracy of Hitting Set algorithms. For the latter bottleneck, and while subnetworks can be inferred in parallel on the subproblems, it is important to develop new techniques for accurate estimation of small networks—topologies and divergence times, as these are both used in our approach. Last but not least, while the efficiency of the merger algorithm could be improved, our analyses above show that the two aforementioned bottlenecks are the more important targets for further improvement.

Finally, it is worth mentioning that our merger algorithm makes no assumption on what evolutionary processes were accounted for in the subnetwork inference. In this sense, our merger algorithm can be applied to merge subnetworks inferred under a variety of models (e.g., ILS, gene duplication and loss, and hybridization), as long as the subnetworks’ topologies and divergence times are accurately estimated.

My thesis contributed novel methods for achieving scalability of phylogenetic network inference. Prior to the work introduced in this thesis, statistical inference of phylogenetic networks was limited to a handful of taxa and a very small number of reticulations. The methods I introduced allow for inferring phylogenetic networks on much larger data sets.

There are several directions for future research. First, it is worth exploring how pseudo-likelihood performs when it is incorporated into a fully Bayesian MCMC framework. In this case, the prior accounts for model complexity and the number of reticulations does not need to be specified a priori. However, to the best of our knowledge, using pseudo-
likelihood, instead of full likelihood, in phylogenetic MCMC has not been done before; therefore, there could be issues that arise and need to be handled carefully for this to work. Second, as we pointed out above, different techniques for reducing the number of trinets in the divide-and-conquer methods need to be explored in order to find ones that significantly reduce the number of trinets, yet maintain a good level of accuracy. Third, in our divide-and-conquer method, we assumed a single population mutation rate across all branches of the phylogenetic network. In practice, this parameter varies across the branches of the phylogeny. Accounting for this issue necessitates modifications to the trinet inference and merger algorithms.
Bibliography


