NOVEL COMPUTATIONAL FRAMEWORKS FOR ANALYZING COMPLEX NON-TREE-LIKE EVOLUTION IN GENOMIC SEQUENCE DATA

By

Qiqige Wuyun

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Doctor of Philosophy

2025

ABSTRACT

Phylogenetics studies the evolutionary relationships among species, often represented as phylogenetic trees. However, traditional tree-like models fall short when dealing with interspecific gene flows, such as hybridization/introgression in eukaryotes and horizontal gene transfer (HGT) in prokaryotes, which necessitate the use of phylogenetic networks to capture complex, reticulate evolutionary histories. Although numerous methodologies have been developed to analyze non-tree-like evolutions, the advent of high-throughput sequencing technologies has introduced two primary scalability challenges: the large number of taxa present in the data and the intricate and diverse gene flow among these taxa. These challenges have considerably constrained the scope and precision of non-tree-like phylogenetic inference and analysis.

This dissertation addresses these limitations by introducing novel techniques for analyzing non-tree-like evolutions in large-scale genomic studies. We developed PHiMM, an introgression detection tool that combines coalescent-based approximations with hidden Markov models (HMMs) to improve scalability and detection accuracy. Comparisons with state-of-the-art methods on both simulation and empirical datasets indicated that PHiMM significantly outperforms previous methods like PhyloNet-HMM in terms of runtime and memory usage, while maintaining comparable inference accuracy to PhyloNet-HMM.

To further enhance PHiMM's performance, we integrated it with the SERES resampling tool, significantly improving introgression inference accuracy under various model conditions. Simulation experiments demonstrated that combining the SERES resampling approach with PHiMM substantially improves introgression inference accuracy compared to standalone PHiMM, although it results in longer runtimes.

One major limitation of PHiMM is its requirement for a phylogenetic network as input to constitute the structure of the HMM. To address this, we extended PHiMM to DACS by integrating phylogenetic network inference with introgression detection. To further improve the scalability and accuracy of introgression mapping on ultra large datasets, we adopted divide-and-conquer and subsampling techniques, allowing us to efficiently handle the complexity of the data while

maintaining accuracy.

Moreover, we applied these approaches to metagenomic studies, where data often include thousands of species derived from complex microbial communities. After assembling the sequencing reads into metagenome-assembled genomes (MAGs), we employ DACS to identify reticulate evolutionary events, such as introgression or HGT, under challenging scenarios characterized by noise, incomplete data, and large numbers of taxa. By adapting our methods to accommodate the scale and complexity of metagenomic datasets, we provide a powerful framework for elucidating reticulate evolutionary histories in diverse microbial communities.

These advancements provide deeper insights into genetic and biological processes and offer robust tools for a wide range of biological and medical applications.

ACKNOWLEDGEMENTS

First, I would like to express my sincere thanks to my advisor, Professor Kevin Liu, for his continuous training and mentorship throughout my Ph.D. studies. With his patience and immense knowledge, Professor Liu has taught me to learn from every mistake and to care about details, guiding me to become a competent scientist.

I am deeply indebted to my committee members, Professor Emily Dolson, Professor Mohammad Ghassemi, Professor Elizabeth Heath-Heckman, and Professor Adina Howe, for their generous service, valuable comments, and thoughtful questions. It has been a great honor and privilege to have them on my doctoral committee.

I am grateful to BEACON, EEB, and HPCC for their generous support during my Ph.D. studies. I would also like to thank the professors and staff at the Department of Computer Science and Engineering, especially Professor Eric Torng and Professor Sandeep Kulkarni, for their invaluable assistance.

My deepest gratitude also goes to Professor Jishou Ruan, who supervised my master's thesis at the Department of Mathematics, Nankai University. Professor Ruan has provided invaluable patience and feedback. Without his precious support and inspiration, I would not have achieved the current results. Although he is no longer with us, he will always remain in my heart.

Lastly, I had the pleasure of working with my excellent colleagues — Hussein, Wei, Julia, Meijun, Ahmad, Md, Rei, Byungho, Preethi, and many others — who have impacted and inspired me greatly.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Phylogenetics is the study of relationships among different groups of species and their evolutionary developments, and attempts to trace the evolutionary history in the Tree of Life. A phylogeny represents the evolutionary history for a set of organisms and is often depicted in a diagram known as a phylogenetic tree. Phylogenies are used in a range of applications, including studying pathogens [1], representing the relationships between species in the tree of life, studying speciation and extinction [2], studying the spread of antibiotic resistance between species [3], identifying genes and non-coding RNAs in newly sequenced genomes [4], and reconstructing ancestral genomes [5]. These widespread applications have led to the development of many sophisticated methods for phylogeny reconstruction, reflecting the importance of accurate and efficient inference algorithms.

Most phylogenetic reconstruction methods assume that the underlying biological data follows a tree-like structure. However, it is known that this assumption does not always hold due to interspecific gene flows such as hybridization/introgression in eukaryotes and horizontal gene transfer (HGT) in prokaryotes. Introgression, or introgressive hybridization, denotes the process of gene flow from one species into the gene pool of another through the recurrent backcrossing of an interspecific hybrid with one of its parental species; HGT is the nonsexual movement of genetic material between distantly related organisms, which enables the rapid transmission of genes between species and plays an essential role in the adaptation of microbes to novel environments. Interspecific gene flow is thought to be especially important in the evolution of a wide range of different species, including humans [6], mice [7], butterflies [8], fungi [9], and bacteria [10]. When reticulate events such as introgression or HGT occur, the evolutionary history can no longer be accurately represented by a simple, bifurcating tree. Instead, a phylogenetic network is required to capture these complex genetic exchanges. A phylogenetic network extends the conventional tree framework by incorporating reticulation nodes and reticulation edges, which explicitly capture gene flow. This more general model accurately reflects both vertical (tree-like) and non-vertical (reticulate) evolutionary processes.

Over the past decade, researchers have developed various computational methods aimed at detecting and analyzing non-tree-like evolutionary events [11–13]. However, the field has recently encountered significant challenges stemming from high-throughput sequencing technologies. Two key scalability issues have emerged. First, there is often a large number of taxa (i.e., species or strains) present in the data, potentially spanning hundreds or thousands of organisms. Second, the complexity and diversity of gene flows among these taxa amplify the computational burden. These challenges can undermine the performance of traditional non-tree-like phylogenetic methods, limiting both the size of datasets that can be analyzed and the accuracy of the resulting inferences [11, 14, 15].

1.1 Contributions

To address these scalability and complexity challenges, this dissertation focuses on developing novel phylogenetic methods and models designed for large-scale genomics and metagenomics studies involving non-tree-like evolutionary processes.

We first concentrate on introgression, a prevalent form of gene flow in eukaryotes. We introduce **PHiMM**, an introgression detection tool tailored to identify introgression in genomic datasets containing dozens of DNA sequences. PHiMM incorporates a newly devised coalescent-based approximation technique with a Hidden Markov Model (HMM) under a joint model of genetic drift, substitutions, recombination, and gene flow. This integrated approach reduces model complexity and enhances the scalability of introgression detection compared to existing methods. PHiMM requires a phylogenetic network to define the structure of the underlying HMM. Comparisons with other state-of-the-art tools on both simulated and empirical datasets suggest that PHiMM offers significantly improved runtime and memory usage over PhyloNet-HMM while maintaining comparable inference accuracy.

Next, we seek to boost PHiMM's introgression inference performance and learning capabilities by leveraging **SERES**, a random-walk-based resampling tool. As a data perturbation technique, SERES is used to perform resampling on the input alignment to generate multiple replicates. PHiMM is then run on these resampled SERES replicates. The introgression probabilities inferred by PHiMM are aggregated across all SERES replicates to obtain final results. Simulation experiments demonstrate that combining the SERES resampling approach with PHiMM substantially improves introgression inference accuracy under various model conditions compared to standalone PHiMM, although this benefit does come at the cost of increased runtimes.

A primary limitation of PHiMM is its dependence on a known phylogenetic network as an input. Consequently, we introduce **DACS**, an extension of PHiMM that integrates phylogenetic network inference with introgression detection. By embedding network construction and introgression analysis into a unified framework, DACS bypasses the need for a predefined network. Additionally, DACS employs divide-and-conquer strategies and subsampling techniques to handle ultra-large datasets more efficiently. These approaches allow us to break down large datasets into smaller, manageable subproblems, each of which can be analyzed independently before combining the results, thereby efficiently handling the complexity of the data while maintaining accuracy.

The advent of high-throughput metagenomic technologies has brought us tons of biological data, which have experienced exponential growth in recent years. The data generated by metagenomic experiments are also inherently noisy and partial, sometimes containing as many as 10,000 species [16]. Extracting useful biological information from datasets of such magnitude poses substantial computational hurdles for researchers. To address these difficulties, we apply our proposed method, DACS, to metagenomic data. The first thing we need to do is to process the metagenomic data into metagenome-assembled genomes (MAGs), from which the aforementioned methods should be easily employed. Once MAGs are obtained, DACS can be applied to detect introgression or horizontal gene transfer (HGT) among diverse microbial communities. We further study and discuss the differences in terms of input requirements, accuracy, scalability, and application conditions of proposed methods between genomics and metagenomics.

1.2 Organization

The dissertation is organized into the following chapters:

- **Chapter 2** provides a background and foundation for the research presented in this dissertation. We introduce the fundamental concepts behind non-tree-like evolution, including phylogenetic network, multi-species network coalescent (MSNC) model, and introgression.

- **Chapter 3** details the development and design of **PHiMM**, our novel introgression mapping algorithm. We demonstrate how PHiMM significantly expands the scalability of introgression detection and achieves competitive or superior performance compared to existing tools.
- **Chapter 4** focuses on empirical validations of PHiMM. We apply PHiMM to a variety of real-world datasets, indicating its effectiveness in practical scenarios.
- **Chapter 5** explains the application of **SERES** random walk-based resampling approach, illustrating how integrating SERES with PHiMM can further boost introgression inference accuracy in challenging cases.
- **Chapter 6** introduces **DACS**, an improved introgression mapping framework that integrates phylogenetic network inference and adopts divide-and-conquer and subsampling techniques. We highlight its ability to handle ultra-large datasets without prior knowledge of the underlying network.
- **Chapter 7** explores the application of DACS-based introgression detection within the context of metagenomic studies. We elaborate on how our method is applied to microbial communities featuring vast species diversity and high levels of reticulation.
- **Chapter 8** concludes the dissertation with a summary of our research findings and a discussion of potential future directions.

This dissertation thus aims to bridge the gap between the theoretical frameworks of reticulate evolution and their practical implementation in large-scale genomics and metagenomics. Through the development of scalable algorithms and robust computational tools, we provide researchers with new ways to decode the complexities of gene flow across the Tree of Life.

CHAPTER 2

BACKGROUND

2.1 Phylogenies

2.1.1 Phylogenetic Tree

A phylogenetic tree represents a branching structure that depicts the evolutionary relationships between diverse biological species, genes, populations, or even individuals based on similarities and differences in their physical or genetic characteristics.

A tree is a connected acyclic graph. Let $T = (V, E)$ be a phylogenetic tree, V be a set of vertices or nodes, and E be a set of edges or branches. Each edge $e \in E$ is defined by two connected vertices $e = (v_1, v_2)$, where $v_1, v_2 \in V$. The leaf nodes or leaves represent present-day taxa, such as species, populations, genes, or individuals, while the internal nodes typically symbolize extinct ancestors whose sequence data remain unavailable and inaccessible. An internal node is also termed as the most recent common ancestor (MRCA) of its descendants. The ancestor of all sequences is the root of the tree. The branches or edges represent the time estimates of the evolutionary relationships among taxa. A branch or edge that is incident with a leaf is called an external branch or external edge, whereas one connecting two internal nodes is called an internal branch or internal edge.

A tree in which the root is specified is known as a rooted tree, whereas a tree lacking a determined root is referred to as an unrooted tree (Figure 2.1). For n taxa, there are $(2n - 3)!!$ distinct rooted trees, and $(2n - 5)!!$ different unrooted trees.

For an unrooted tree, a commonly-used strategy to identify the root and produce a rooted tree is outgroup rooting. An outgroup is a species distantly related to all species in an unrooted tree. Through introducing an outgroup in the tree reconstruction, the root can be placed on the branch connecting the outgroup, resulting in a rooted subtree for the ingroups [17].

For readability and legibility in computer programs, rooted trees are commonly depicted using a Newick format. It utilizes a pair of parentheses to group sister taxa into one clade where taxa are split by a comma. A semicolon is used to mark the end of the tree. Branch lengths, if present, are prefixed by colons. Figure 2.1 shows an example of a rooted tree, which can be referred to as

"((((A, B), C), D), E);" without branch lengths visualization, or "((((A: 0.1, B: 0.2): 0.12, C: 0.3): 0.123, D: 0.4): 0.1234, E: 0.5);" showing branch lengths.

The Newick format can also be used to represent unrooted trees. An unrooted tree has several Newick representations due to the flexibility of root placement on the tree. For example, the unrooted tree shown in Figure 2.1 can also be represented as "((((A, B), C), D), E);".

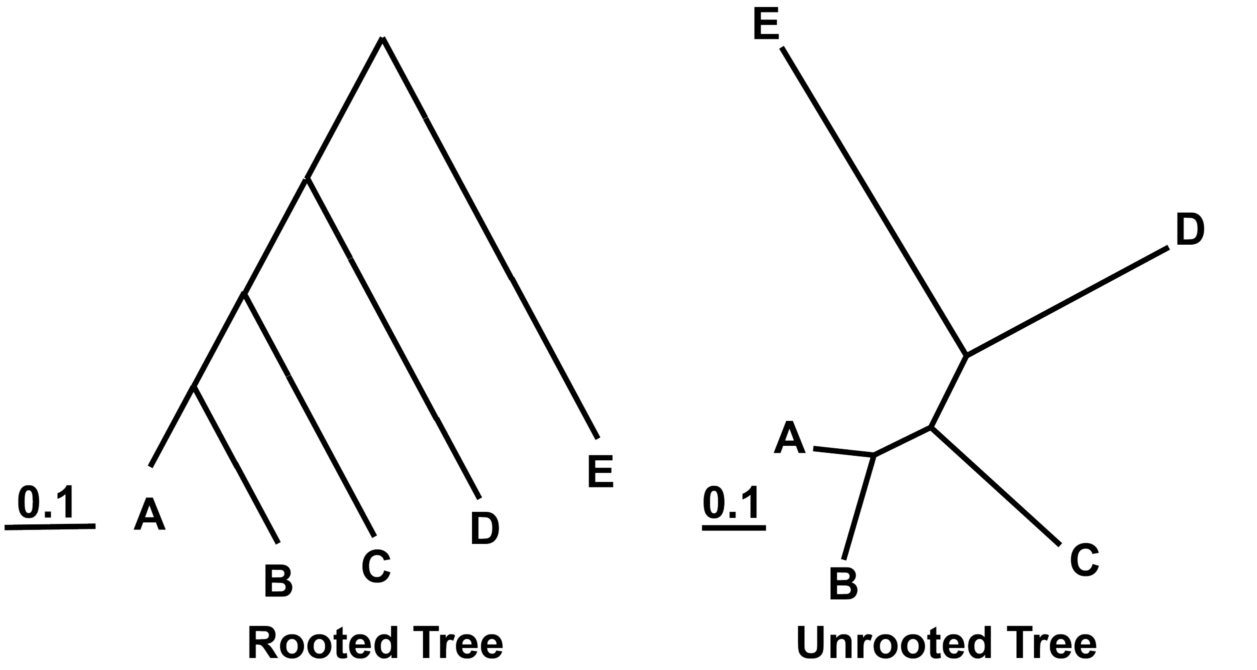


Figure 2.1 **The illustration of the rooted and unrooted phylogenetic trees.** In the rooted tree on the left, a specific ancestor is designated as the root (at the base of the diagram), implying a clear evolutionary direction as lineages diverge over time. In contrast, the unrooted tree on the right shows the same relationships among taxa (A, B, C, D, E) without specifying a common ancestor, thus omitting any time or evolutionary direction. Branch lengths can indicate the amount of sequence divergence (e.g., 0.1 substitutions per site). Taxa labeled A, B, C, D, and E represent different species or taxonomic units. This figure is reproduced from Yang [17].

2.1.2 Phylogenetic Network

Although phylogenetic trees are widely used to represent evolutionary relationships, the true evolutionary histories are not always tree-like. The phylogenetic networks are thus introduced to visualize evolutionary relationships when reticulation events, such as hybridization, introgression, and horizontal gene transfer (HGT), are involved. The phylogenetic networks explicitly model richly linked networks through the introduction of reticulation nodes to capture gene flow.

A phylogenetic network is a directed acyclic graph (DAG) with at least one reticulation event [18]. Let $N = (V, E)$ be a phylogenetic network with $V = r \cup V_L \cup V_T \cup V_N$, where

- r is the root of the network: $indegree(r) = 0$ (node has no parent);
- V_N is the set of reticulation nodes in N: $\forall v \in V_N, indegree(v) = 2$ (nodes have two parents) and $outdegree(v) = 1$ (nodes have one child);
- V_T is the set of tree nodes in N: $\forall v \in V_N$, $indegree(v) = 1$ (nodes have one parent) and $outdegree(v) \geq 2$ (nodes have at least two children);
- V_L is the set of leaves in N: $\forall v \in V_N$, $indegree(v) = 1$ (nodes have one parent) and $outdegree(v) = 0$ (nodes have no child).

$E \subseteq V \times V$ are the edges of network N. There are two distinct types of edges: reticulation edges, which direct to reticulation nodes, and tree edges, which point to tree nodes.

A phylogenetic network can be characterized based on the set of induced tree topologies. The set of trees can be obtained by the following steps. First, the phylogenetic network is converted to a multilabeled tree (MUL-tree) where leaves are not uniquely labeled by a taxon set [19, 20]. Then, a set of true trees is obtained by keeping only one valid way of taxa mapping in the MUL-tree. For simplicity, the set of induced trees is also called the set of MUL-trees. As shown in Figure 2.2, the set of MUL-trees, T_1 and T_2, are induced from the network N.

An extended Newick format can be employed to represent phylogenetic networks for easy use in computer programs [21]. It introduces the "#" symbol to annotate the reticulation nodes that are duplicated and numbered consecutively. For example, the network in Figure 2.2 can be represented as "((A:1, (B:0.5)X#H1:0.5):1, ((X#H1:0,C:0.5):0.5, D:1));" or "((A:1, X#H1:0.5):1, (((B:0.5)X#H1:0,C:0.5):0.5, D:1));".

2.1.3 Gene Tree, Species Tree, and Species Network

The phylogeny that illustrates the relationships among a group of species is referred to as the species tree or network. The tree corresponding to a set of gene sequences from the species is known as the gene tree.

For a species tree or network, the leaf nodes represent species, which include the entire popu-

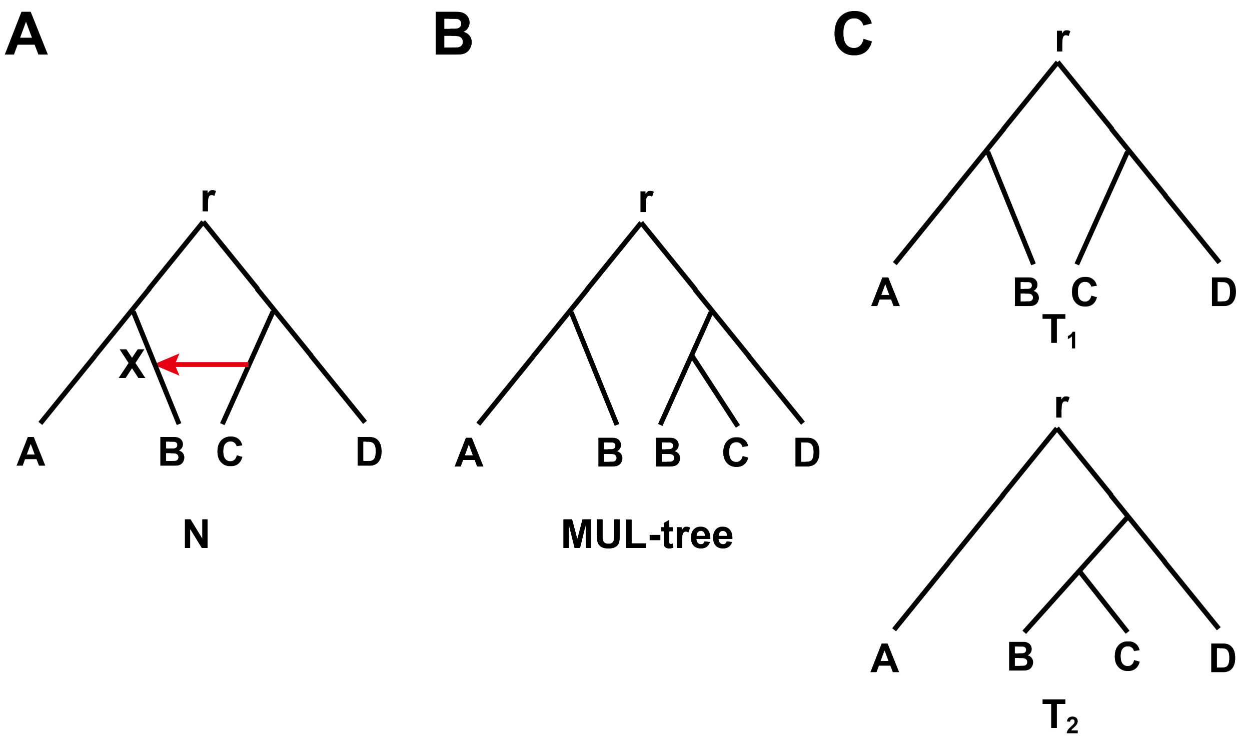


Figure 2.2 **The illustration of a phylogenetic network.** (A) A phylogenetic network *N*, where *r* is the root node, and *X* is a reticulation node. (B) The MUL-tree induced from the network *N*. (C) The induced trees T_1 and T_2 from network *N*.

lation of this species. Each internal node symbolizes a speciation event, which results in the split of one species' population into subsequent species.

The gene tree shows the evolution path of one particular gene, which is a particular region on the genome of all involved species. The gene tree is not necessarily identical to the species tree or other gene trees. A variety of factors may render the gene tree different from the species tree, such as introgression, hybridization, horizontal gene transfer, recombination, incomplete lineage sorting, and gene duplication and loss.

Some gene tree discordances do not affect the tree structure of the species tree. However, some of the discordances caused by gene flow, where the genetic information transfers from one species to another species, cannot be properly represented in a tree structure on the species level. In these cases, a tree structure can depict the evolutionary traces of genes, but a network structure with reticulations may be a better form to visualize a species phylogeny.

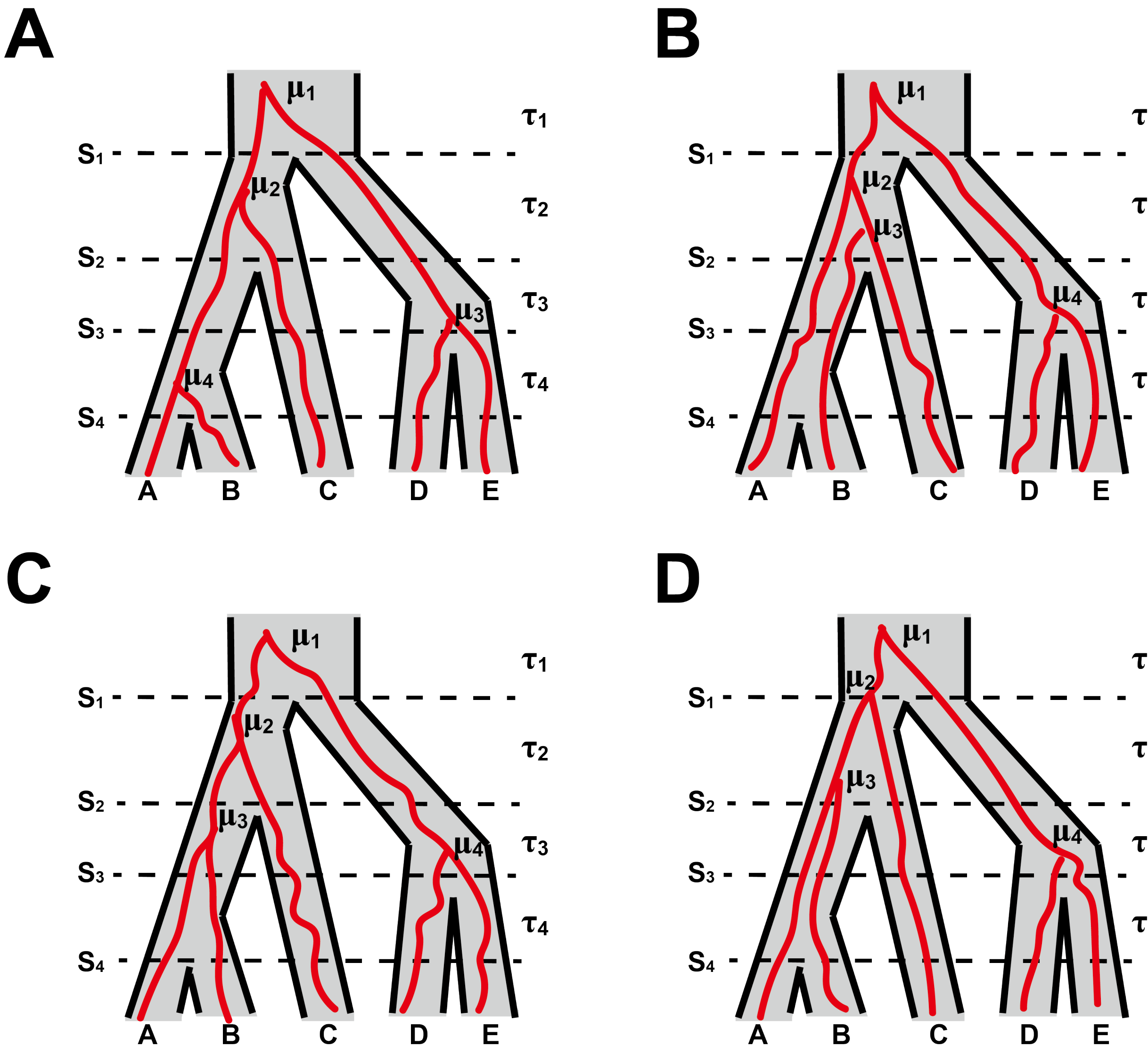


Figure 2.3 **Gene evolution within a species tree (((A,B),C),(D,E)) under the multi-species coalescent model.** (A) A scenario where the gene tree is the same as the species tree. (B) A scenario where the gene tree has a different tree topology from the species tree, resulting from ILS. (C) A scenario where the gene tree is also the same as the species tree. (D) A scenario where the gene tree also displays a deep coalescence, and yet the gene tree matches the species tree. This figure is reproduced from Stadler and Degnan [22].

2.1.4 Multi-Species Coalescent (MSC) Model

The multi-species coalescent (MSC) is introduced by Kingman [23], which offers a framework to analyze multi-locus genomic sequence data from different species in diverse inference problems, such as species tree inference.

In the MSC model, each species represents a population of individuals, where each gene involves a set of alleles. The species tree evolves forward in time, while the alleles trace their histories from leaves to the root backward in time. Each allele at a leaf can randomly pick its parent from the prior generation. Over time, two or more alleles may converge upon a common parent, giving rise to a coalescence event, wherein any two alleles have an equal probability of coalescing first. Eventually, all the alleles coalesce into a single allele. Thus, all of the alleles from different individuals constitute a gene tree, which matches inside the species tree.

For any two alleles, the initial chance of coalescence occurs on the edge located above their most recent common ancestor (MRCA). When two or more alleles do not coalesce on that first possible edge, but enter and coalesce on a subsequent edge (i.e., the one further back in time, and closer to the root), we call this event "deep coalescence". Since any two alleles are equally likely to coalesce first, deep coalescence has the potential to create gene trees that differ from the species tree and differ from each other, although the gene trees still fit inside the species tree. The deep coalescence is also called incomplete lineage sorting (ILS).

Figure 2.3 depicts four gene trees that evolved within the same species tree under the MSC. Of the four different gene trees, the gene trees in panels (A), (C) and (D) show the same tree topology as the species tree, while the incongruence between the gene tree and the species tree in panel (B) is due to deep coalescence or ILS. Interestingly, the gene tree in panel (D) also illustrates a deep coalescence event, where the alleles from A and B do not coalesce on the edge above the MRCA of A and B, despite the gene tree matching the species tree. Note that ILS can take place without altering the topology.

Mathematically, the MSC model specifies how lineages coalesce through time. The structure of the tree generated by the coalescent model is determined by coalescent times and how lineages are selected to merge at each coalescent event. The probability that two lineages coalesce in the generation immediately prior is equivalent to the chance they inherited their genetic material from the same parent sequence. Assuming a stable effective population size where each genetic locus has $2N_e$ possible parental sequences, there exist precisely $2N_e$ candidate parental copies in the previous generation. Under random mating conditions, the probability of two alleles descending from an identical parental sequence is therefore $1/(2N_e)$, and conversely, the probability they do not coalesce at this generation is $1 - 1/(2N_e)$. Considering generations sequentially, the coalescence probability follows a geometric distribution. Specifically, the probability that lineages coalesce exactly at generation t is given by the product of the non-coalescence probabilities across the preceding $(t - 1)$ generations and the coalescence probability in generation t itself:

$$P_c(t) = (1 - \frac{1}{2N_e})^{t-1}(\frac{1}{2N_e})$$

When the effective population size N_e is sufficiently large, this discrete probability distribution is closely approximated by a continuous exponential form:

$$P_c(t) = \frac{1}{2N_e}e^{-\frac{t-1}{2N_e}}$$

2.1.5 Multi-Species Network Coalescent (MSNC) Model

The MSC model has been introduced in the context of phylogenetic trees. It can be easily extended to the phylogenetic network [24], where the reticulation edges are used for tracing gene flow events, including hybridization, introgression, and horizontal gene transfer (HGT).

Figure 2.4 shows four scenarios where gene trees evolve within the same species tree under the multi-species network coalescent (MSNC) model. Of the four different gene trees, the gene tree in panel (A) has neither gene flow nor ILS, and matches the species network. Moreover, the gene tree in panel (C) has only gene flow, so it matches the species network with the reticulation edge. The gene tree in panel (B) displays only ILS, resulting in a mismatch to the species network. Interestingly, the gene tree in panel (D) also matches the species network, but includes both gene flow and ILS.

Formally, suppose we have m independent genomic loci, each represented by an alignment set denoted as $\mathscr{S} = S_1, S_2, \ldots, S_m$, where each S_i contains sequence information for locus i. Typically, each alignment S_i might consist of sequences from multiple species under study or represent data from a single bi-allelic genetic marker (e.g., a vector of binary indicators such as single nucleotide polymorphisms, SNPs). The MSNC model consists of two main components:

1. A phylogenetic network Ψ, characterized by its topological structure and associated continuous parameters, including divergence times;
2. A vector Γ that represents inheritance probabilities.

The likelihood under this model can be mathematically expressed as follows:

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

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 [25], and $p(g|\Psi,\Gamma)$ is the density of the gene tree (topologies and branch lengths) given the model parameters, defined as [26]:

$$p(g|\Psi,\Gamma) = \sum_{h\in H_\Psi(g)} \frac{w(h)}{d(h)} \prod_{b\in E(\Psi)} \frac{w_b(h)}{d_b(h)} \Gamma[b,j]^{u_b(h)} p_{u_b(h)v_b(h)}(\lambda_b)$$

where $H_\Psi(g)$ denotes the set of all coalescent histories that explain a gene tree g appearing inside a species tree with topology Ψ and a vector of branch length λ, $u_b(h)$ and $v_b(h)$ denote the number of lineages enter and exit edge b of Ψ under coalescent history h, $w_b(h)$ denotes the number of possible ways the coalescent events could have occurred consistently with the gene tree g, $d_b(h)$ denotes the number of sequences of coalescences that give the number of coalescent events specified by h, and $p_{u_b(h)v_b(h)}(\lambda_b)$ denotes the probability of u_b lineages entering the branch and v_b lineages exiting the branch over a branch of length λ_b. The posterior $p(\Psi,\Gamma|\mathscr{S})$ of the model is proportional to

$$p(\Psi,\Gamma|\mathscr{S}) \propto p(\mathscr{S}|\Psi,\Gamma)p(\Psi)p(\Gamma) = p(\Psi)p(\Gamma) \prod_{i=1}^{m} \int_g p(S_i|g)p(g|\Psi,\Gamma)dg$$

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

While such a full likelihood approach uses all the information in the data, it can be computationally intensive. One common strategy for speeding up inference is to preprocess each locus by estimating its local genealogies. The observed data are then summarized as a set of inferred gene trees $\mathscr{G} = \{G_1, G_2, ..., G_m\}$. In this case, the likelihood formulation simplifies to

$$p(\mathscr{G}|\Psi,\Gamma) = \prod_{i=1}^{m} p(G_i|\Psi,\Gamma)$$

where G_i is the gene tree that has been inferred for every locus, and $p(G_i|\Psi, \Gamma)$ is the density of the gene tree G_i (topologies and branch lengths) given the model parameters [26]. Because this approach relies on accurate gene tree inference, any errors in gene tree estimation can propagate to the final network inference.

Despite these optimizations, exact likelihood and posterior computations can still pose significant computational challenges. Hence, *pseudo-likelihood* methods have been developed. For example, Yu and Nakhleh [27] introduced the pseudo-likelihood of phylogenetic network Ψ and inheritance probabilities Γ given a set of gene trees G:

$$L(\Psi, \Gamma|G) = p(G|\Psi, \Gamma) = \prod_{\{X,Y,Z\}\subseteq\mathbb{X}} f(\rho(XY|Z, G), \rho(XZ|Y, G), \rho(YZ|X, G)|\Psi, \Gamma)$$

where $\mathbb{X}$ is the set of taxa, $XY|Z$, $XZ|Y$ and $YZ|X$ are binary triples with $\{X, Y, Z\} \subseteq \mathbb{X}$, and ρ is the number of times a binary triple is induced by G.

2.2 Phylogenetic Inference

2.2.1 Simulation Studies

Many methods have been developed to address the phylogeny problems. Each represents different tradeoffs of phylogeny accuracy with respect to the true phylogeny, as well as computational requirements.

Simulation studies are one category of widely-used tools to assess the performance of different methods by directly comparing the estimated phylogeny produced by different methods against a true phylogeny. By the generation of synthetic or simulated datasets, simulation studies provide predominant insights into the relative performance of different phylogeny inference methods. A simulation study proceeds according to the following steps:

(1) A model phylogeny (tree or network) is simulated randomly, where r8s [28] tool is often utilized to generate a random tree, while steps listed in Hejase et al. [11] can be used to add reticulations on the tree to generate a network.

(2) Local genealogies or gene trees are simulated for independent and identically distributed (i.i.d.) loci following a species phylogeny under an MSC or MSNC model. In this step,

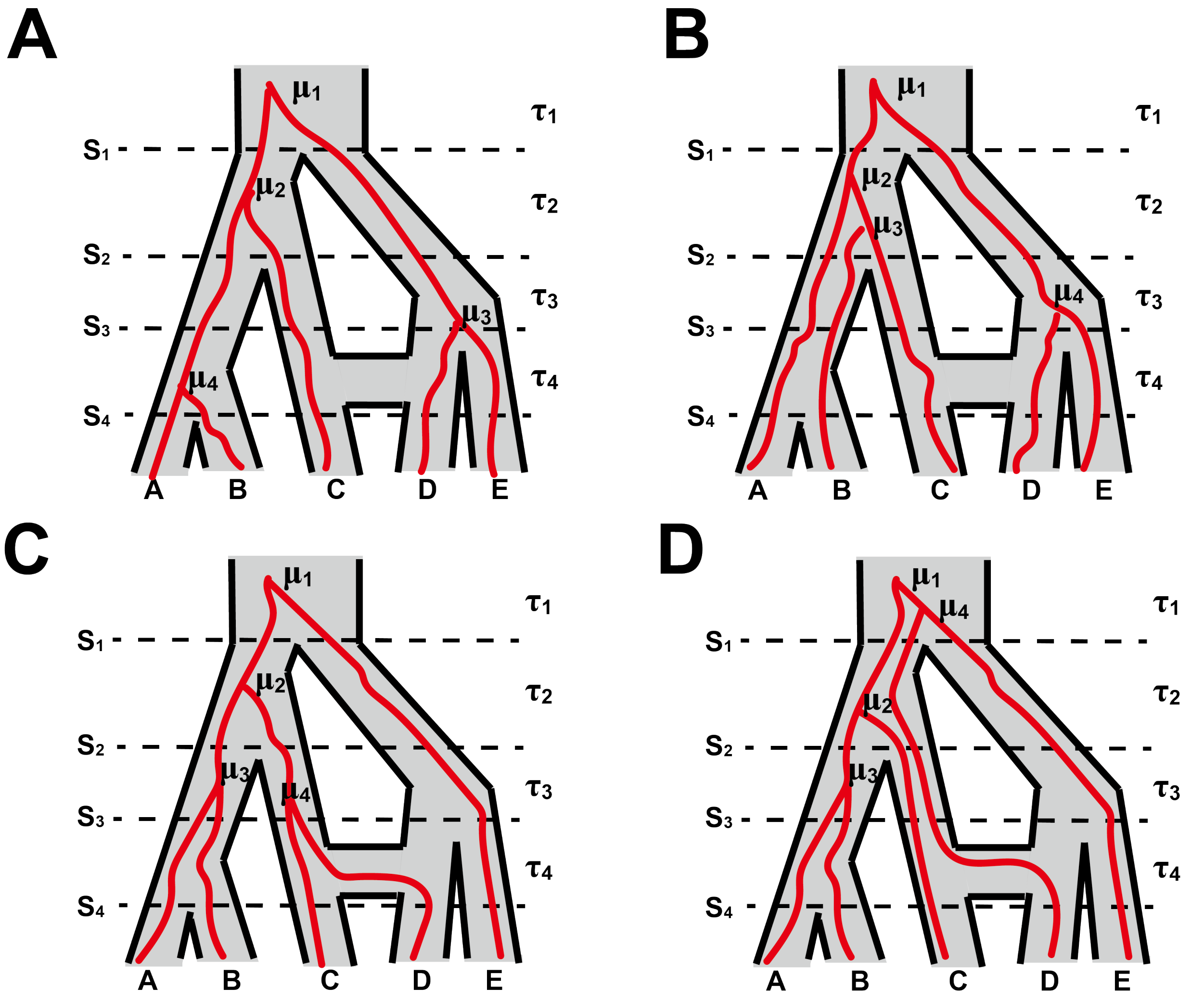


Figure 2.4 **Gene evolution within a species network under the multi-species network coalescent (MSNC) model.** (A) A scenario where the gene tree has neither gene flow nor ILS, and matches the species network. (B) A scenario where the gene tree displays only ILS, resulting in a mismatch to the species network. (C) A scenario where the gene tree has only gene flow, and matches the species network with the reticulation edge. (D) A scenario where the gene tree also matches the species network, but includes both gene flow and ILS. This figure is reproduced from Stadler and Degnan [22].

msmove [29] and ms [30] are often employed to simulate the local genealogies.

(3) DNA sequences are evolved following local gene trees under a specific substitution model [31], in which seq-gen [32] and INDELible [33] are two popularly used tools.

2.2.2 Phylogenetic Tree Inference

2.2.2.1 Gene Tree Inference

The inference of phylogenetic trees is a basic and fundamental problem in phylogenetic studies. A variety of methods have been developed to tackle this issue. These methods can be categorized into two main groups: distance-based methods and optimization-based methods.

The first category of phylogenetic tree inference methods is distance-based methods. In those methods, a distance matrix is first formed by calculating the distances of pairwise sequences, and then converted into a phylogenetic tree by clustering algorithms. The most popular methods within this category are UPGMA (Unweighted Pair Group Method using Arithmetic mean) [34] and neighbor-joining [35].

Other methods are optimization-based methods, which employ an optimality criterion or objective function to measure a tree's compatibility with the data. The tree that achieves the optimal score is considered the estimate of the true tree. Examples in this category include maximum parsimony [36], maximum likelihood (ML) [25], and Bayesian methods [37]. In the maximum parsimony method, the tree score is the minimum number of character changes required for the tree, and the maximum parsimony tree or most parsimonious tree is the tree with the smallest tree score. The ML method uses the log-likelihood value of the tree to measure the fit of the tree to the data, and the maximum likelihood tree is the tree with the highest log likelihood value. The ML method of gene tree estimation was first introduced by Felsenstein [25] and has been implemented in programs such as PAML [38], PhyML [39], RAxML [40], RAxML-NG [41], IQ-Tree [42], and FastTree [43]. In the Bayesian method, the posterior probability of a tree is the probability that the tree is true given the data. The tree with the maximum posterior probability is the estimate of the true tree. The Bayesian method was introduced into molecular phylogenetics in the 1990s [37] and has been implemented in programs such as MrBayes [44], RevBayes [45], BEAST [46, 47], and

PhyloBayes [48, 49].

Distance-based methods are often computationally faster than optimization-based methods, and can be easily applied to analyze different kinds of data as long as pairwise distances can be calculated [17]. Among optimization-based methods, parsimony is known to be more prone than likelihood methods to systematic errors, including long branch attraction (LBA) [50], which is the phenomenon where two branches that are in truth not sisters are inferred to be sister branches when using maximum parsimony inference (see Figure 2.5). This occurs because, unlike likelihood, parsimony does not take into account branch lengths when computing the parsimony score. The situations in which phylogenetic trees produce data susceptible to this failure are referred to as "Felsenstein Zone" [51].

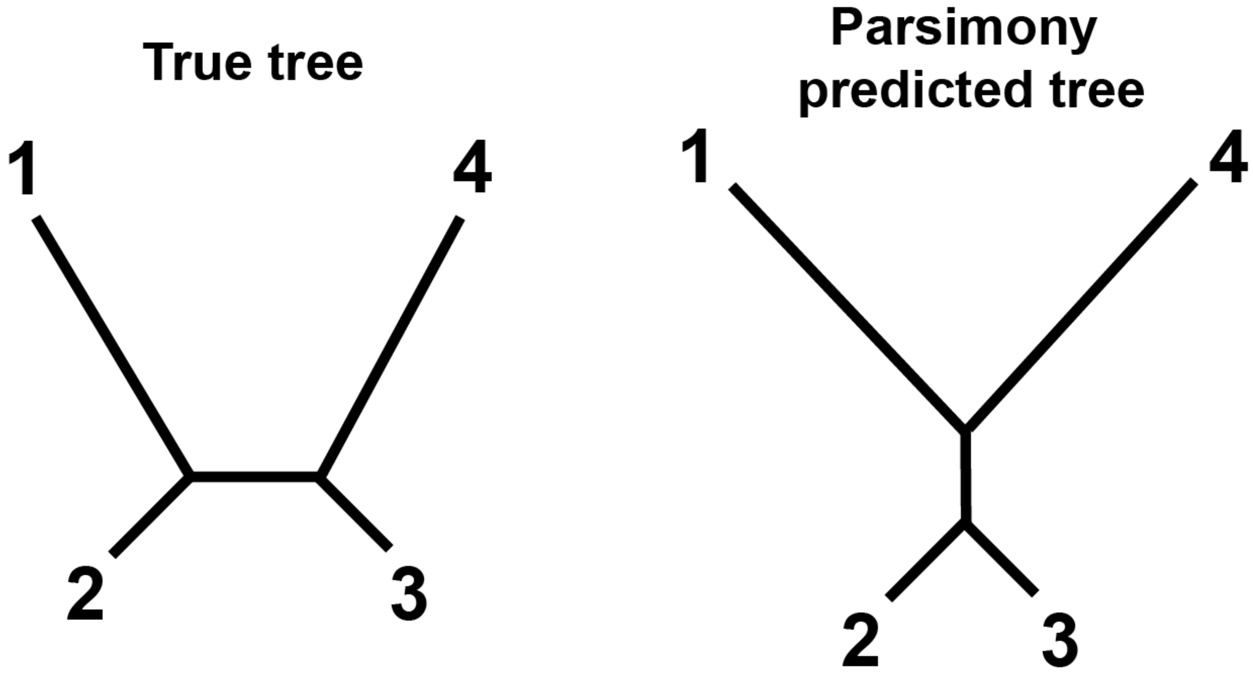


Figure 2.5 **An illustration of long branch attraction.** This figure contrasts the **true phylogenetic tree** (left) with a **parsimony-predicted tree** (right) to highlight the phenomenon of long-branch attraction where two branches that are in truth not sisters are inferred to be sister branches when using maximum parsimony inference. In the true tree, lineages 1 and 2 share a more recent common ancestor than either does with 3 or 4, yet in the parsimony-predicted tree, lineages 1 and 4 are incorrectly grouped as sister taxa. This occurs because, unlike likelihood, parsimony does not take into account branch lengths when computing the parsimony score. The situations in which phylogenetic trees produce data susceptible to this failure are referred to as "Felsenstein Zone" [51].

2.2.2.2 Species Tree Inference

Reconstruction of a species tree usually requires the analyses of a set of genes, due to the limited evolutionary signals provided by each gene and discordance between gene trees. To better infer the species tree, sufficient gene trees, associated with the corresponding gene tree distribution, are needed. Many methods have been developed to address the species tree inference. There are three main categories of these methods: concatenation methods, summary-based methods (or supertree methods), and co-estimation methods (see Figure 2.6).

In the concatenation methods, sequences from genes are first concatenated into one larger size of genome-level sequences, called a supermatrix. Then, a global estimate of the species tree is performed based on the supermatrix for taxa of interest. However, these methods ignore the discordance between gene trees, and may thus lead to statistically inconsistent trees and introduce poor accuracy [52]. Commonly-used programs implementing the multi-species coalescent (MSC) model include StarBEAST [53, 54] and BPP [55, 56].

For the summary-based methods, on the other hand, the gene trees are first estimated independently based on their sequences, and then summarized to construct the overall species tree. There are many methods in this category, such as "minimizing deep coalescence (MDC)"-based methods [57–59], and weighted quartet-based methods [60, 61]. Popular summary-based programs include ASTRAL [60, 61] and MP-EST [62]. These methods are computationally efficient and can analyze thousands of genes but may suffer from errors in reconstructed gene trees [50].

The co-estimation methods simultaneously infer both gene trees and the species tree based on the Bayesian model and the Markov Chain Monte Carlo (MCMC) algorithm, which have shown higher accuracy in species tree inference than solely concatenation methods [53, 63].

2.2.2.3 Metric

Sometimes we may wish to quantify the similarity between two trees. For instance, we might be interested in comparing the differences among trees derived from different methods or evaluating the dissimilarities between the true tree and the estimated tree in a simulation study in order to assess a tree inference method.

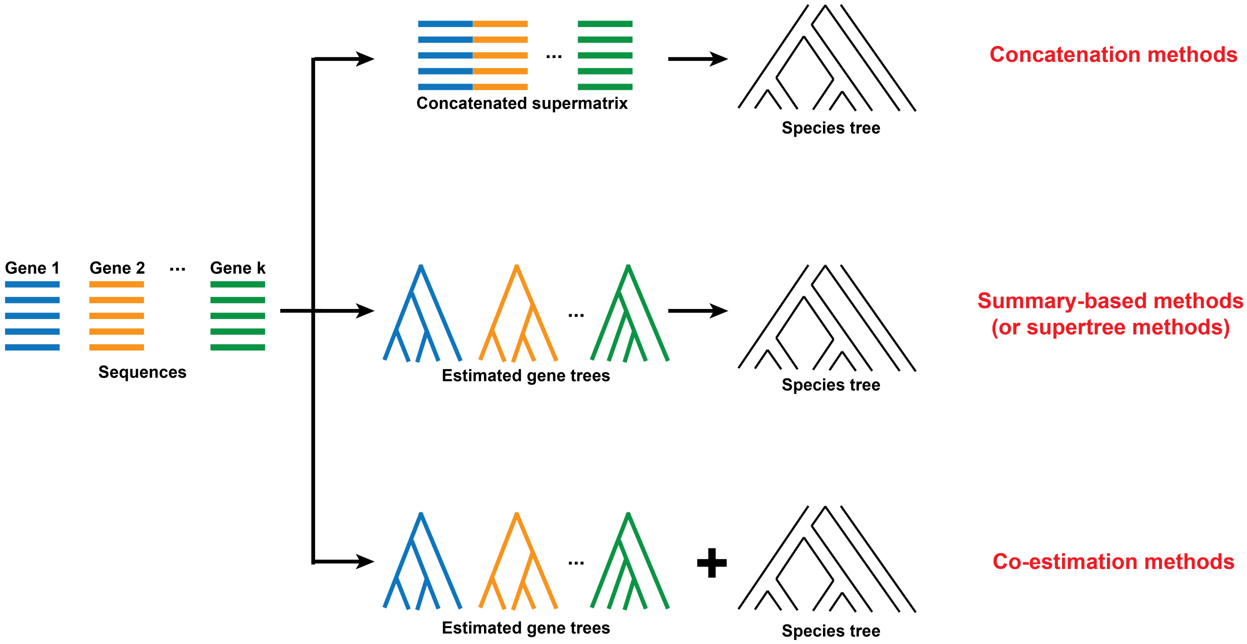


Figure 2.6 **Species tree inference methods, including concatenation methods, summary-based methods (or supertree methods), and co-estimation methods.** In the concatenation methods, sequences from genes are first concatenated into one larger size of genome-level sequences, called a supermatrix. Then, a global estimate of the species tree is performed based on the supermatrix for taxa of interest. However, these methods ignore the discordance between gene trees, and may thus lead to statistically inconsistent trees and introduce poor accuracy [52]. For the summary-based methods, on the other hand, the gene trees are first estimated independently based on their sequences, and then summarized to construct the overall species tree. These methods are computationally efficient and can analyze thousands of genes but may suffer from errors in reconstructed gene trees [50]. The co-estimation methods simultaneously infer both gene trees and the species tree based on the Bayesian model and the Markov Chain Monte Carlo (MCMC) algorithm, which have shown higher accuracy in species tree inference than solely concatenation methods [53, 63].

Phylogenetic trees can be represented by a collection of bipartitions. A bipartition refers to a distinct split of leaves obtained by removing one internal edge in an unrooted tree. Given an unrooted tree $T = (V, E)$, each edge defines a bipartition of taxa. We denote by $C(T)$ the set of non-trivial bipartitions of T, in which each non-trivial bipartition on the leaf set of T is generated by the removal of an internal edge.

The Robinson-Foulds (RF) distance [64] of two unrooted phylogenetic trees T_1 and T_2 is defined as

$$d(T_1, T_2) = |C(T_1) \backslash C(T_2)| + |C(T_2) \backslash C(T_1)|$$

That is, the RF distance is defined as the total number of false positive bipartitions and false negative bipartitions. For easy use, the RF distance can be normalized between 0 and 1 by dividing $d(T_1, T_2)$ by the maximal possible RF distance, whose value is $2n - 6$ for binary unrooted trees, where n is the number of leaves in a tree.

Although the RF distance here is defined for unrooted trees, the same definition can be easily applied to rooted trees by virtually attaching an outgroup to the root.

Furthermore, the missing branch rate is another commonly-used measure for evaluating the difference between the true tree and the estimated tree. The missing branch rate is computed as the proportion of bipartitions present in the true tree but absent in the estimated tree.

2.2.3 Phylogenetic Network Inference

2.2.3.1 Inference Methods

Numerous computational methodologies have been devised to infer phylogenetic networks from multi-locus data, including the maximum parsimony method, the maximum likelihood method, the maximum pseudo-likelihood method, and the Bayesian inference method.

The maximum parsimony (MP) methods employ the minimizing deep coalescence (MDC) criterion [57], which searches for the network that has the capability to minimize the occurrence of deep coalescences necessary to account for all input gene trees.

Maximum likelihood methods are proposed to calculate model likelihood under the multi-species network coalescent (MSNC) model, where only gene tree topologies (MLE) or gene tree topologies and branch lengths (MLE-length) are used [26]. One weakness of maximum likelihood methods is their high computational cost.

Maximum pseudo-likelihood (MPL) methods use pseudo-likelihood in the optimization criterion [27], which reduces the computational requirements but attains less accuracy than the MLE or MLE-length methods.

By leveraging the prior distribution in conjunction with a set of rooted gene tree topologies as input, the Bayesian inference method can achieve computational expediency, resulting in faster computations for inferring the posterior distribution of the network. The most widely-used method for the Bayesian inference of the phylogenetic network is the reversible-jump Markov chain Monte

Carlo (RJMCMC) [65].

2.2.3.2 Metric

The tripartition-based distance [18] counts the proportion of tripartitions that are not shared between the two networks.

For each node of the network, a tripartition consists of three sets of leaves: those that are strict descendants of it, those that are non-strict descendants of it, and those that are not descendants of it. Given a phylogenetic network $N = (V, E)$ with L as a set of leaves, an ancestor s of a node u in N is referred to as a strict ancestor if all the paths from the root of N to u contain s, otherwise a non-strict ancestor of u. The tripartition of edge $e = (u, v)$ is defined as $(A(e), B(e), C(e))$, where

- $A(e) = \{s \in L \mid u$ is a strict ancestor of $s\}$;
- $B(e) = \{s \in L \mid u$ is a non-strict ancestor of $s\}$;
- $C(e) = \{s \in L \mid u$ is not an ancestor of $s\}$.

For two phylogenetic networks N_1 and N_2, the tripartition-based distance can be calculated by

$$d(N_1, N_2) = \frac{1}{2} \times (\frac{|\{e_1 \in E(N_1)|\nexists e_2 \in E(N_2), A(e_1) = A(e_2), B(e_1) = B(e_2), C(e_1) = C(e_2)\}|}{|E(N_1)|} + \frac{|\{e_2 \in E(N_2)|\nexists e_1 \in E(N_1), A(e_1) = A(e_2), B(e_1) = B(e_2), C(e_1) = C(e_2)\}|}{|E(N_2)|})$$

where $E(N_1)$ and $E(N_2)$ define the set of edges of networks N_1 and N_2, respectively.

The reduction-based distance [66] is commonly used to measure the difference between phylogenetic networks. The reduction-based distance quantifies the dissimilarity between two phylogenetic networks by considering their reduced topologies.

We first define the reduction of a network. Given a phylogenetic network N, its reduced network N' can be obtained by creating symbolic leaf nodes to replace and represent each maximal subtree where reticulation nodes are not included.

For two reduced phylogenetic networks N_1 and N_2, the reduction-based distance can be calculated by

$$d(N_1, N_2) = \frac{1}{2} \times (\sum_{v \in U(N_1)} max\{0, \kappa(v) - \kappa(v')\} + \sum_{u \in U(N_2)} max\{0, \kappa(u) - \kappa(u')\})$$

where $U(N_1)$ and $U(N_2)$ define the set of unique nodes of networks N_1 and N_2, respectively. $v' \in U(N_2)$ is equivalent to $v \in U(N_1)$, while $u' \in U(N_1)$ is equivalent to $u \in U(N_2)$. Two nodes are considered equivalent if they are both leaf nodes and bear identical labels, or if they have the same number of children and their children are equivalent accordingly. $\kappa(v)$, $\kappa(v')$, $\kappa(u)$ and $\kappa(u')$ represent the number of nodes equivalent to v, v', u, and u' in networks N_1, N_2, N_2, and N_1, respectively.

Similar to the above-mentioned RF distance, the reduction-based distance roughly quantifies the number of rooted subnetworks that are present in one network but absent in the other [66].

2.3 Substitution Models

2.3.1 Definition

The substitution models are Markov models that characterize changes occurring over evolutionary time in macromolecules (e.g., DNA sequences) represented as sequences of symbols (A, C, G, and T in the case of DNA). Substitutions at any particular site are described by a Markov chain, wherein the symbols (A, C, G, and T in the case of DNA) serve as the states of the chain. The primary characteristic of a Markov chain is its lack of memory: "given the present, the future does not depend on the past". The sites in the sequence are commonly assumed to undergo evolution independently of each other.

2.3.2 Nucleotide Substitution Models

The most general and commonly-used model of nucleotide substitution is the general time-reversible (GTR) model [67]. It is a continuous-time reversible Markov model that is parameterized by a stationary base probability distribution over the nucleotides π and a 4×4 transition rate matrix Q,

$$Q = \{q_{ij}\} = \begin{array}{c} \\ T \\ C \\ A \\ G \end{array} \begin{array}{c} \begin{array}{cccc} T & C & A & G \end{array} \\ \begin{pmatrix} . & a\pi_C & b\pi_A & c\pi_G \\ a\pi_T & . & d\pi_A & e\pi_G \\ b\pi_T & d\pi_C & . & f\pi_G \\ c\pi_T & e\pi_C & f\pi_A & . \end{pmatrix} \end{array}$$

where each entry is the instantaneous substitution rate of one nucleotide to another (see Figure 2.7).

Jukes-Cantor (JC or JC69) model [31] represents the simplest submodel of the GTR model, as it assumes both equal transition rates and equal base frequencies, i.e., $a = b = c = d = e = f$ and $\pi_A = \pi_T = \pi_G = \pi_C = \frac{1}{4}$. The transition rate matrix Q is as follows (see also Figure 2.7),

$$Q = \{q_{ij}\} = \begin{array}{c|cccc} & T & C & A & G \\ T & . & \lambda & \lambda & \lambda \\ C & \lambda & . & \lambda & \lambda \\ A & \lambda & \lambda & . & \lambda \\ G & \lambda & \lambda & \lambda & . \end{array}$$

Kimura's two-parameter model, also known as K80 model [68], is another simple submodel of the GTR model that assumes that all of the bases are equally frequent ($\pi_A = \pi_T = \pi_G = \pi_C = \frac{1}{4}$). Let the substitution rates be α for transitions and β for transversions, where the transition means the substitution between two pyrimidines ($T \leftrightarrow C$) or between two purines ($A \leftrightarrow G$), while the transversion is the substitution between a pyrimidine and a purine ($T, C \leftrightarrow A, G$). The transition rate matrix Q is as follows (see also Figure 2.7),

$$Q = \{q_{ij}\} = \begin{array}{c|cccc} & T & C & A & G \\ T & . & \alpha & \beta & \beta \\ C & \alpha & . & \beta & \beta \\ A & \beta & \beta & . & \alpha \\ G & \beta & \beta & \alpha & . \end{array}$$

Let the transition/transversion rate ratio κ be α/β. The transition rate matrix Q can also be represented as,

$$Q = \{q_{ij}\} = \begin{array}{c|cccc} & T & C & A & G \\ T & . & \kappa & 1 & 1 \\ C & \kappa & . & 1 & 1 \\ A & 1 & 1 & . & \kappa \\ G & 1 & 1 & \kappa & . \end{array}$$

These nucleotide substitution models can be employed to simulate the evolutionary process of nucleotide sequences under a rooted tree T, where branch lengths represent the time estimating the evolutionary changes. First, the sequence at the root is randomly generated based on the stationary base probability distribution of nucleotides π. Then, the sequences on the descendant nodes are iteratively obtained according to the transition probability matrix $P(t) = e^{Qt}$ [69], where the time t represents the branch length until a set of simulated sequences at all nodes of T are created.

To incorporate rate heterogeneity across sites, the instantaneous substitution rates of any substitution model can be multiplied by a rate sampled from the Γ distribution for rate variation across sites [70]. For example, GTR+Γ is the GTR substitution model in combination with the Γ model to produce rate variation across sites.

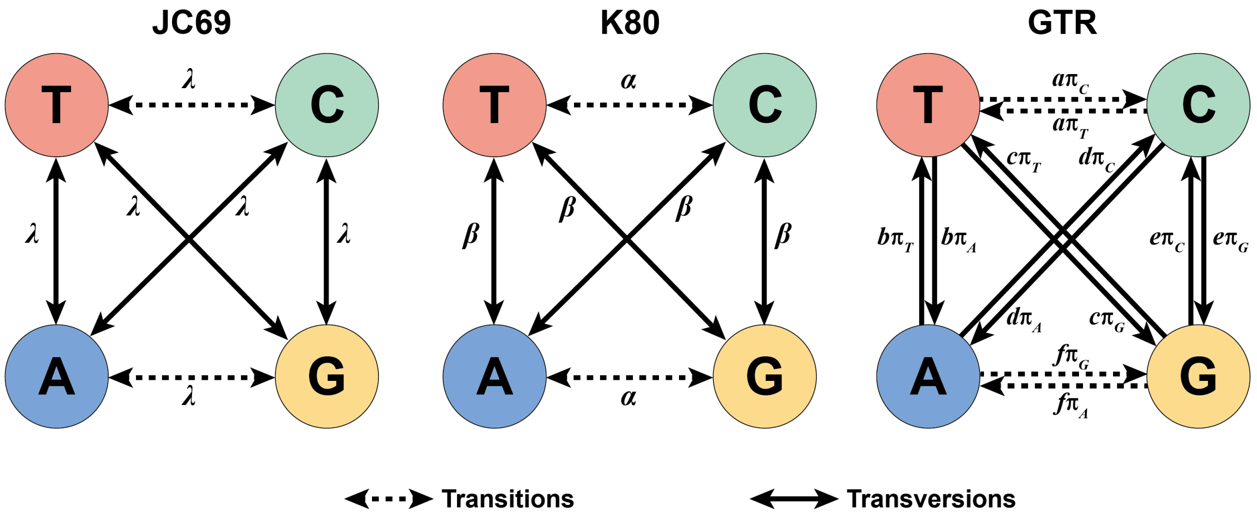


Figure 2.7 **An illustration of JC69, K80, and GTR substitution model.** In JC69 model (left), all substitutions — whether transition or transversion — are assumed to occur at the same rate λ. In K80 model (middle), transitions (dashed arrows) occur at rate α, whereas transversions (solid arrows) occur at rate β, thus distinguishing purine-purine or pyrimidine-pyrimidine substitutions from purine-pyrimidine substitutions. The GTR (General Time Reversible) model (right) is the most parameter-rich, allowing each of the six possible base substitutions to have its own rate parameter. In addition, GTR incorporates base frequency parameters $(\pi_A, \pi_T, \pi_G, \pi_C)$ to account for unequal nucleotide frequencies in the sequences being modeled. These models progressively increase in complexity to capture the diverse evolutionary patterns that arise in real genetic data.

2.4 Gene Flow

2.4.1 Definition

Gene flow is the transfer of genetic material from one population to another. The gene flow between species primarily involves horizontal gene transfer (HGT) in prokaryotes and introgression (also known as introgressive hybridization) in eukaryotes.

2.4.2 Introgression

2.4.2.1 Definition

Introgression, or introgressive hybridization, denotes the process of gene flow from one species into the gene pool of another through the recurrent backcrossing of an interspecific hybrid with one of its parental species. Introgression differs from hybridization, because simple hybridization results in a relatively even mixture of two parental species, while introgression results in a variable mixture involving a relatively small percentage from the donor species (Figure 2.8). Ancient introgression events have the potential to leave traces of extinct species in present-day genomes, a phenomenon referred to as ghost introgression [71].

Introgression can manifest as either neutral, where its effects on phenotypes go unnoticed, or adaptive, where it influences phenotypes in significant ways. The term “adaptive introgression” is used when the genetic transfer of minimal genomic regions from a donor species leads to positive fitness effects within the gene pool of the recipient species [72].

2.4.2.2 Inference Methods

The detection of introgression has become a major area of interest in evolutionary biology. Numerous computational methods have been proposed to distinguish true introgression events from incomplete lineage sorting (ILS), leveraging different features of genomic data, analytical formalisms, and statistical frameworks.

Early approaches for detecting introgression distinguish introgressive hybridization and ILS by using sequence data to construct gene tree topologies evolved down a phylogenetic network. Sang and Zhong [73] and Holder et al. [74] built and tested their model based on the idea that incongruent gene trees show a nearly equal distribution in the case of introgression but a significantly distinct

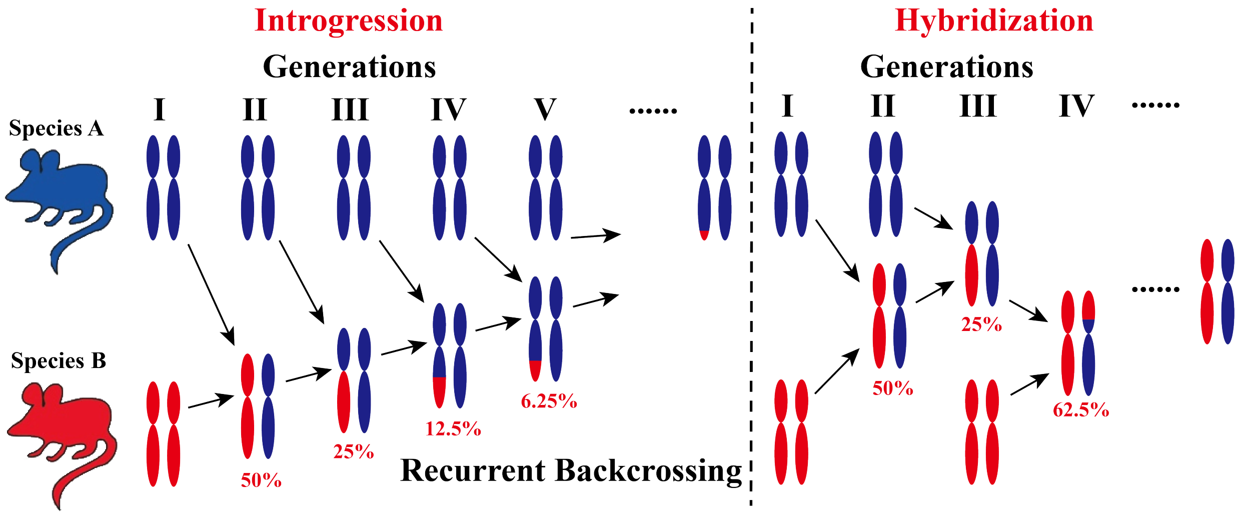


Figure 2.8 **An illustration of introgression and hybridization.** Introgression differs from hybridization, because simple hybridization results in a relatively even mixture of two parental species, while introgression results in a variable mixture involving a relatively small percentage from the donor species. For introgression on the left, a single hybridization event is followed by repeated backcrossing of hybrids to one parental species (Species A), causing the proportion of red chromosomes (from Species B) to diminish in each generation (e.g., 50%, 25%, 12.5%, 6.25%) while still persisting in the final population. For hybridization on the right, multiple or ongoing hybridization events occur between the two parental species across successive generations, leading to more complex admixture patterns. Here, the fraction of genetic material from each species can shift in different ways (e.g., returning to 62.5% red in one generation), reflecting repeated interbreeding rather than a single hybridization event followed strictly by backcrossing.

distribution for ILS. Patterson et al. [75] detected the introgressive hybridization by computing the genetic divergence of aligned sequences.

A notable class of recent methods employ hidden Markov models (HMMs) to analyze genomic data in the context of incomplete lineage sorting (ILS) [12, 76–79]. CoalHMM [76, 79] was originally proposed to tackle different population genetic inference issues but has since been expanded to address the challenge of statistical introgression mapping. Subsequently, Liu et al. [12] devised the PhyloNet-HMM method, which combined an HMM-based model with the phylogenetic networks as input to detect introgression in genomes. This method is accurate but time- and memory-consuming because it takes all possible gene trees evolved down the input phylogenetic network into account.

Other methods bypass the gene tree inference and calculate imbalances in minimum genetic distances to infer introgression. The method from Joly et al. [80] was based upon the fact that the calculated minimum pairwise sequence distance between two species is smaller for introgression events than for ILS. Joly [81] developed a software called JML based on the above scenarios.

Drawing inspiration from the reconciliation framework used in detecting horizontal gene transfer, Chauve et al. [82] introduced a method to reconcile gene trees within a species network, identifying gene flow events that yield consistent explanations for observed incongruences. This approach has been shown to detect extensive introgression even in large and complex datasets.

Taking advantage of the rapid progress in artificial intelligence technologies, Schrider et al. [83] introduced FILET, a machine learning-based pipeline designed to detect genomic introgression by integrating a comprehensive set of population genetic summary statistics. By capturing patterns of variation across multiple populations, FILET effectively identifies regions of introgressed ancestry. Expanding on this paradigm, Flagel et al. [84] proposed a convolutional neural network (CNN) framework that interprets population genomic alignments as images, thereby eliminating the need for predefined summary statistics. Through the automatic extraction of discriminative features across genomic windows, this CNN-based approach demonstrates high accuracy in detecting introgression events. More recently, Ray et al. [85] introduced IntroUNET, a deep learning framework that similarly treats alignments as images but employs semantic segmentation to pinpoint introgressed alleles with heightened resolution.

D-statistic is based on the idea that a statistically significant imbalance between two SNP patterns is indicative of an introgression event. D-statistic is first applied to hominoids using genome-scale SNP data and a sliding window analysis to detect introgression on four-taxon data despite the existence of ILS [6, 86, 87]. An extension of D-statistic (called D_{FOIL}) applying for the symmetric five-taxon phylogeny is developed by Pease and Hahn [88]. Inspired by D_{FOIL}, Elworth et al. [89] devised a general statistic, D_{GEN}, to automatically generate a statistic for detecting introgression in any number of genomes and any set of hybridization events. In recent years, many analogs to the D-statistic have been brought up to estimate the introgression regions [90–97]. The disadvantage of this type of statistical method is that it assumes an infinite-sites model and independence across loci.

2.4.3 Horizontal Gene Transfer

2.4.3.1 Definition

Horizontal gene transfer (HGT) or lateral gene transfer (LGT) is the nonsexual movement of genetic material between distantly related organisms. Other than the “vertical” transfer from parents to offspring, HGT enables the rapid transmission of genes between species and plays an essential role in the adaptation of microbes to novel environments [98]. In particular, there is a growing interest in HGT concerning the dissemination of antibiotic resistance genes among bacteria in diverse communities [99–102]. Most studies in genetics and biology have focused upon the “vertical” transfer, such as hybridization and introgression, but the importance of HGT among single-cell and even some multi-cell organisms is beginning to be acknowledged [103].

While HGT remains the primary mode of horizontal evolution scrutinized in genomic data, it is crucial to acknowledge several other mechanisms of horizontal evolutionary processes as well, including gene duplication and loss, and genetic recombination (Figure 2.9). Gene duplication is defined as the generation of new copies of a gene at distinct loci within the genome [104]. Gene loss is referred to as the absence or extinction of a gene that is identified when comparing different species, but can also encompass any allelic variant carrying a loss-of-function mutation found within a population [105]. Genetic recombination is the exchange of genetic material between different chromosomes, which renders different histories for neighboring segments within a gene [106].

Due to these processes, gene trees may be expected to be discordant with the species tree. As discussed before, the multi-species coalescent (MSC) model has served as a baseline to investigate gene tree discordance caused by incomplete lineage sorting (ILS), while the multi-species network coalescent (MSNC) model is proposed for analyzing the discordance introduced from gene flow, such as introgression, hybridization, and HGT. Furthermore, these models can also be extended to capture genetic recombination as well as gene duplication and loss. An example is shown in Figure 2.9.

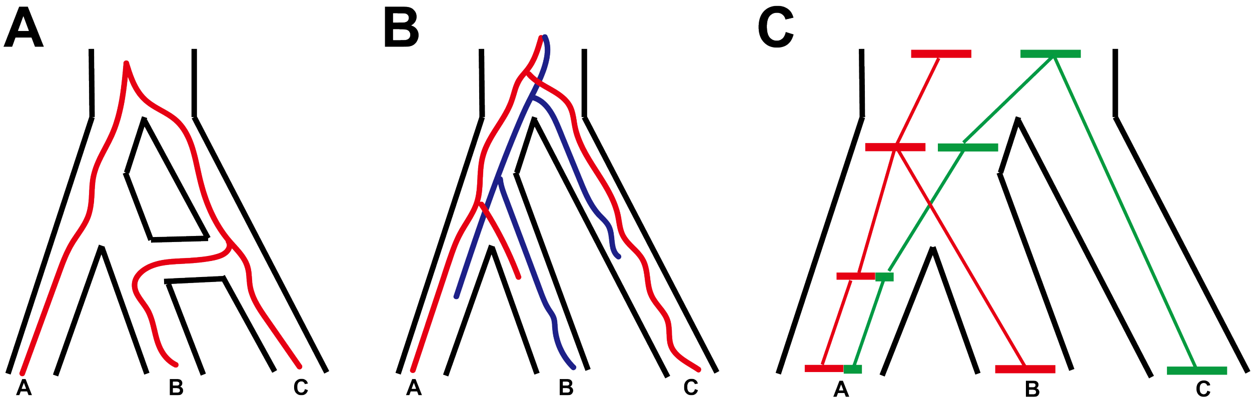


Figure 2.9 **Alternative sources of the discordance between the species tree and gene trees from horizontal evolutions.** (A) A scenario where an HGT event makes the gene tree (A,(B,C)) incongruent to the species tree ((A,B),C), but matches the species network ((A,(B)X#H1),(X#H1,C)). (B) A scenario where the incongruence is caused by the gene duplication and loss. (C) A scenario where the discordance comes from recombination. For the DNA segment depicted in red, the gene tree is ((A,B),C), but for the segment in green, the gene tree is ((A,C),B). This figure is reproduced from Degnan and Rosenberg [104].

2.4.3.2 Inference Methods

A number of approaches have been developed to detect HGT in genomics and metagenomics data. These methods can be broadly classified into two groups: sequence composition-based methods and phylogenetic methods (Figure 2.10).

Sequence composition-based methods search for fragments of a genome significantly different from the genomic average by computing sequence signatures, such as GC content or codon usage [107]. Genomic regions that exhibit distinct compositional characteristics compared to the background are referred to as "genomic islands". There are two commonly-used tools for the identification of genomic islands: GIST [108] and IslandViewer [109].

Phylogenetic methods analyze the evolutionary histories of genes involved and identify conflicting phylogenies. These methods can be further divided into explicit methods, which reconstruct and directly compare phylogenetic trees, and implicit methods, which employ surrogate measures in place of the phylogenetic trees [110]. For example, HGTector [111] and DarkHorse [112] are two popular implicit phylogenetic methods, which identify genes in genomes exhibiting taxonomically discordant similarity to genes present within a reference database; AnGST [113] and RANGER-DTL [114] are widely-used explicit phylogenetic methods that reconcile phylogenetic incongruencies between gene and species trees to detect HGT.

These methods address the detection of HGT from distinct perspectives, which may lead to remarkably different HGT inference results. However, these tools can be combined to produce more robust detections [115]. Based on this idea, MetaCHIP [116] has recently been proposed to identify HGT events in a natural community by combining an all-against-all BLASTN with the RANGER-DTL software.

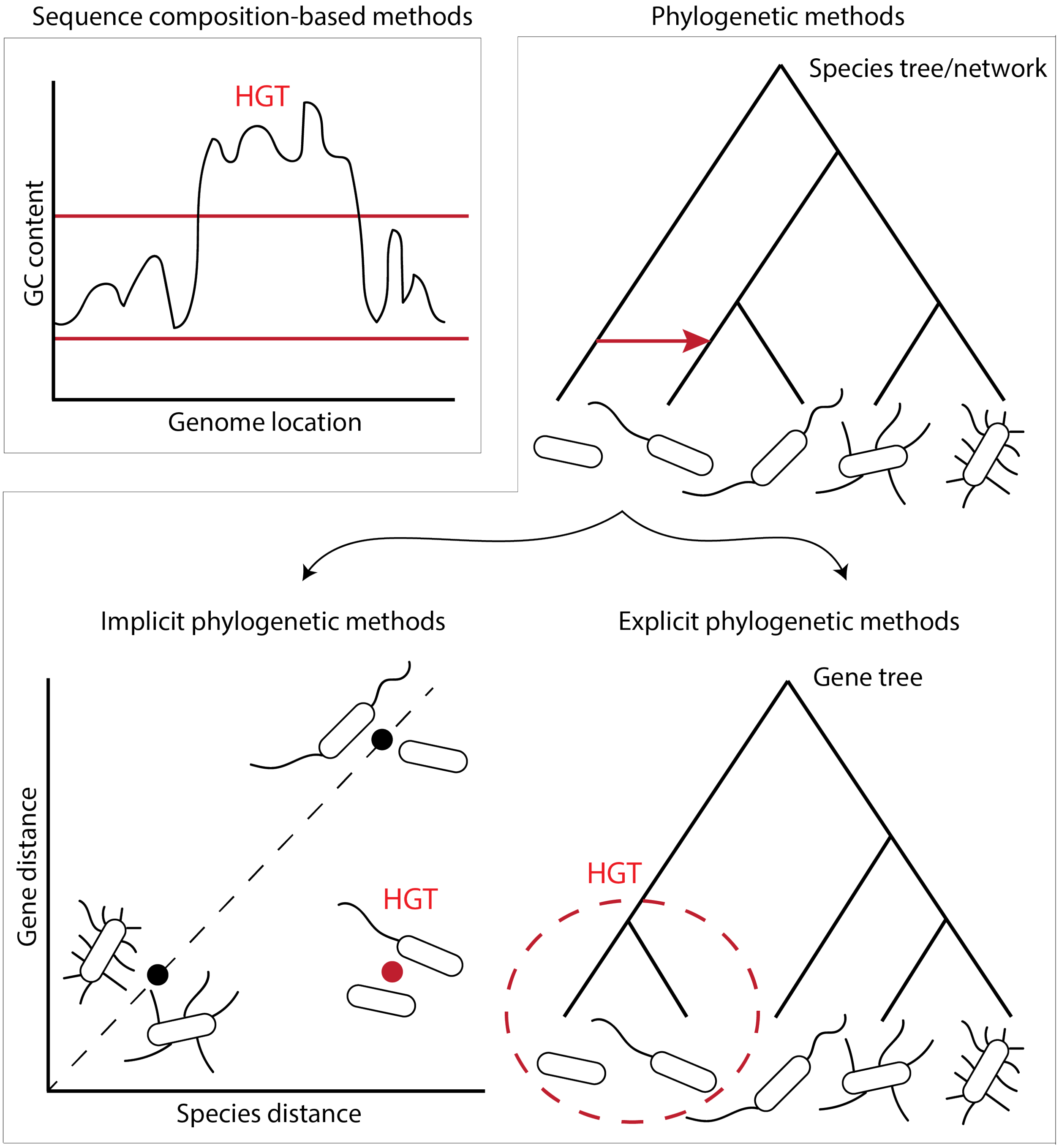


Figure 2.10 **An overview of HGT inference methods.** The HGT iference methods can be broadly classified into two groups: (1) sequence composition-based methods and (2) phylogenetic methods. (1) Sequence composition-based methods search for fragments of a genome significantly different from the genomic average by computing sequence signatures, such as GC content or codon usage [107]. Genomic regions that exhibit distinct compositional characteristics compared to the background are referred to as “genomic islands”. (2) Phylogenetic methods analyze the evolutionary histories of genes involved and identify conflicting phylogenies. These methods can be further divided into explicit methods, which reconstruct and directly compare phylogenetic trees, and implicit methods, which employ surrogate measures in place of the phylogenetic trees [110]. This figure is reproduced from Ravenhall et al. [117].

CHAPTER 3

PHIMM: SCALABLE STATISTICAL INTROGRESSION DETECTION USING THE APPROXIMATE COALESCENT-BASED INFERENCE

3.1 Introduction

Hybridization, defined as the sexual reproduction of two organisms from distinct species, is a key evolutionary process that can alter genetic and phenotypic diversity. Introgression (or introgressive hybridization) occurs when gene flow from one species enters the gene pool of another through recurrent backcrossing of hybrids [118]. It has been estimated that up to 25% of plant species and 10% of animal species are capable of interspecific hybridization and introgression [119], highlighting the evolutionary significance of this phenomenon.

Introgression regions in eukaryotic genomes are important, since the introgression can be neutral without showing in phenotypes but can also be adaptive and impact phenotypes. A widely-studied example of adaptive introgression is the house mice, i.e., *Mus musculus dometicus* (*M. m. dometicus*) and *Mus spretus* (*M. spretus*). The anticoagulant rodenticide resistance of *M. m. dometicus* is acquired through the introgression of the *Vkorc1* (vitamin K 2,3-epoxide reductase subcomponent 1) gene from *M. spretus* to *M. m. dometicus* [120]. Clearly, introgression plays a vital role in genetic evolution and adaptive divergence. Therefore, developing state-of-the-art techniques to detect introgression regions based upon the increasing explosion of genomic data is urgently needed.

Detecting the introgression region can be achieved by scanning the genomes of the species, and constructing the gene tree at each locus. The incongruence between gene trees evolved from different species trees offers an opportunity to detect introgression events. However, introgression detection remains a challenging problem, since distinguishing between hybrid introgression and incomplete lineage sorting (ILS) can be difficult. ILS is a type of deep coalescence that occurs in the speciation event, which can also result in the discordance between gene trees along the genomes. Although other processes, such as gene duplication and loss [121], may also cause the incongruence between gene trees and species trees, we focus on introgression and ILS here. Thus, a powerful approach for the detection of introgressed regions in the presence of ILS is highly needed but challenging.

Recent methods have been designed to disentangle introgression mapping in the presence of ILS, such as D-statistic, and an array of hidden Markov model (HMM)-based techniques (refer to Chapter 2.4.2.2). Of great relevance to our work, Liu et al. [12] introduced an introgression mapping method, PhyloNet-HMM, which combines the multi-species network coalescent (MSNC) model [20] with an HMM to detect introgression.

Although these methods have advanced the field, none are optimized for both speed and accuracy when analyzing datasets comprising dozens of DNA sequences. To bridge this gap, we developed PHiMM (fast PhyloNet + Hidden Markov Model) [122], a refined introgression detection framework that integrates a coalescent-based approximation with an HMM under a model in a combination of substitutions, recombination, and gene flow. The simulation study demonstrates that the PHiMM tool is able to decrease the model complexity, which can significantly expand the scalability of introgression detection while maintaining the inference accuracy. By offering an efficient and accurate tool for introgression detection in increasingly large genomic datasets, PHiMM contributes to a deeper understanding of the evolutionary processes underpinning species divergence and adaptation.

3.2 Related Work

3.2.1 PhyloNet-HMM Algorithm

The PhyloNet-HMM algorithm infers introgression regions based on a hidden Markov model. The algorithm was first proposed by Liu et al. [12], and then modified and implemented as a part of the PhyloNet version 3.6 package [13, 123].

The inputs of PhyloNet-HMM algorithm are the aligned DNA sequences A and a phylogenetic network N. The input alignment A can be defined as $\{A, C, T, G\}^{K \times L}$, where K is the number of taxa, and L is the length of genomic sequence alignment.

The HMM states in the PhyloNet-HMM algorithm correspond to all distinct pairs of a MUL-tree and a gene tree. The MUL-tree is encoded from the input species network N using some existing methods described by Huber et al. [19] and Yu et al. [20]. The set of MUL-trees can be used to represent the species network N. The gene tree can be any possible rooted binary tree on K leaves, so the total number of possible gene trees is $(2K-3)!!$, where K is the number of taxa. Gene flow directionality is reflected in reticulation edge directionality in an explicit phylogenetic network. Let m and n be the number of MUL-trees and gene trees, respectively. Thus, the total number of hidden states should be $m \times n$. As shown in Figure 3.1, the HMM hidden states can be represented by $s_{ij} = (T_i, G_j)$, where T_i is the i-th MUL-tree ($1 \leq i \leq m$) and G_j is the j-th gene tree ($1 \leq j \leq n$).

The transition of HMM from the start state to a state $s_{ij} = (T_i, G_j)$ can be calculated as follows:

$$t_{(T_i,G_j)} = \frac{z(s_{ij})}{\sum\limits_{k,l} z(s_{kl})}$$

where $z(s_{ij})$ is the probability of local gene tree G_j under MUL-tree T_i, which can be calculated using the approach in Yu et al. [20].

The transition from a state $s_{ij} = (T_i, G_j)$ to a state $s_{kl} = (T_k, G_l)$ ($1 \leq i, k \leq m$ and $1 \leq j, l \leq n$) occurs with the following probability:

$$a_{(T_i,G_j)\to(T_k,G_l)} = \epsilon(T_i, T_k)\delta(G_j, G_l)\frac{z(s_{kl})}{\sum\limits_{i,j} z(s_{ij})}$$

where the $\epsilon(T_i, T_k)$ and $\delta(G_j, G_l)$ can be calculated based on the following formulas:

$$\epsilon(T_i, T_k) = \begin{cases} 1 - \Delta_T & if \;\; i = k \\ \frac{\Delta_T}{m-1} & if \;\; i \neq k \end{cases}$$

$$\delta(G_j, G_l) = \begin{cases} 1 - \Delta_G & if \;\; j = l \\ \frac{\Delta_G}{n-1} & if \;\; j \neq l \end{cases}$$

where Δ_G represents the probability of switching between gene trees with different topologies (i.e., switching between columns in Figure 3.1), while Δ_T is the probability of switching between MUL-trees with different topologies (i.e., switching between rows in Figure 3.1).

Given a hidden state $s_{ij} = (T_i, G_j)$, the emission probability can be calculated based on the input observation sequence A, which is defined as $\{A, C, T, G\}^{K \times L}$, where K is the number of taxa and L is the length of genomic sequence alignment. We define each site of the observation sequence A as a_i $(1 \leq i \leq L)$. The emission probability at each site a_i $(1 \leq i \leq L)$ is calculated by:

$$e_{s,\phi}(a_i) = P[a_i|s, \phi] = P[a_i|\ell_T, \ell_G, \phi]$$

where ℓ_T are the branch lengths of the MUL-tree and ℓ_G are the branch lengths of the gene tree associated with state $s = (T, G)$. ϕ represents a specific substitution model under which emissions occur. In this study, the generalized time-reversible (GTR) model [67] is used.

Given the observation sequences A and the HMM structure, the model parameters θ are learned to maximize $P(A|\theta)$, i.e., $\text{argmax}_\theta P(A|\theta)$. The model parameters θ are comprised of:

- The set of MUL-trees with both topologies and branch lengths;
- The set of local gene trees with both topologies and branch lengths;
- DNA substitution model parameter ϕ;
- MUL-tree and gene tree switching probabilities Δ_T and Δ_G

While the model likelihood for a fixed θ can be calculated efficiently using the dynamic programming [124], it is computationally difficult to optimize the model likelihood by learning θ. For this reason, local search heuristics, such as the Baum-Welch algorithm and the expectation-maximization algorithm [124], are typically used for HMM learning. In the PhyloNet version 3.6 implementation, the PhyloNet-HMM framework utilizes the BOBYQA algorithm [125] to iteratively perform multi-variate optimization as part of a hill-climbing search (whereas in the initial implementation, PhyloNet-HMM utilizes Brent's method for univariate optimization [126]).

Given an optimized model θ^*, the forward and backward algorithms are used to calculate the posterior decoding probability:

$$P(\pi_t = (T_i, G_j)|A, \theta^*) = \frac{f_t(i, j)b_t(i, j)}{P(A|\theta^*)}$$

where A is the observed multiple sequence alignment with L columns, and each aligned columns of A is $a_t \in A$ ($1 \leq t \leq L$); π_t is the t-th state of hidden state path π; (T_i, G_j) represents all possible hidden states ($1 \leq i \leq m$, $1 \leq j \leq n$) as shown in Figure 3.1; the forward probability $f_t(i, j) = P(a_1, a_2, ..., a_t, \pi_t = (T_i, G_j)|\theta^*)$ is calculated using the forward algorithm; the backward probability $b_t(i, j) = P(a_{t+1}, a_{t+2}, ..., a_L|\pi_t = (T_i, G_j), \theta^*)$ is calculated using the backward algorithm; $P(A|\theta^*)$ is the probability of the alignment, which can be computed by either the forward or backward algorithms.

Finally, a modified posterior decoding approach is used to assess the confidence of introgression inference by the PhyloNet-HMM algorithm. The modified posterior decoding probability that a site a_t ($1 \leq t \leq L$) in the alignment A has an introgressed origin is computed as follows:

$$p_t = \sum_{\substack{T_i \in \Omega_T \\ 1 \leq j \leq n}} P(\pi_t = (T_i, G_j)|A, \theta^*)$$

where Ω_T represents the set of MUL-trees having introgressive origins.

3.3 Methods

3.3.1 Problem Definition

Based on the description of PhyloNet-HMM, the inputs of the problem are the aligned DNA sequences A and a phylogenetic network N.

Consistent with the study of Liu et al. [12], the output of the problem is a sequence of modified posterior decoding probabilities for the columns of the input alignment A. At a site a_t ($1 \leq t \leq L$) in the alignment A, the probability of a_t having an introgressive origin is defined by:

$$p_t = \sum_{\substack{T_i \in \Omega_T \\ 1 \leq j \leq n}} P(\pi_t = (T_i, G_j)|A)$$

where Ω_T is the set of MUL-trees corresponding to introgression events.

3.3.2 PHiMM Algorithm

The PHiMM algorithm for statistical introgression mapping consists of a three-stage pipeline. The pseudocode of our algorithm is given in Algorithm 3.1 and Figure 3.2.

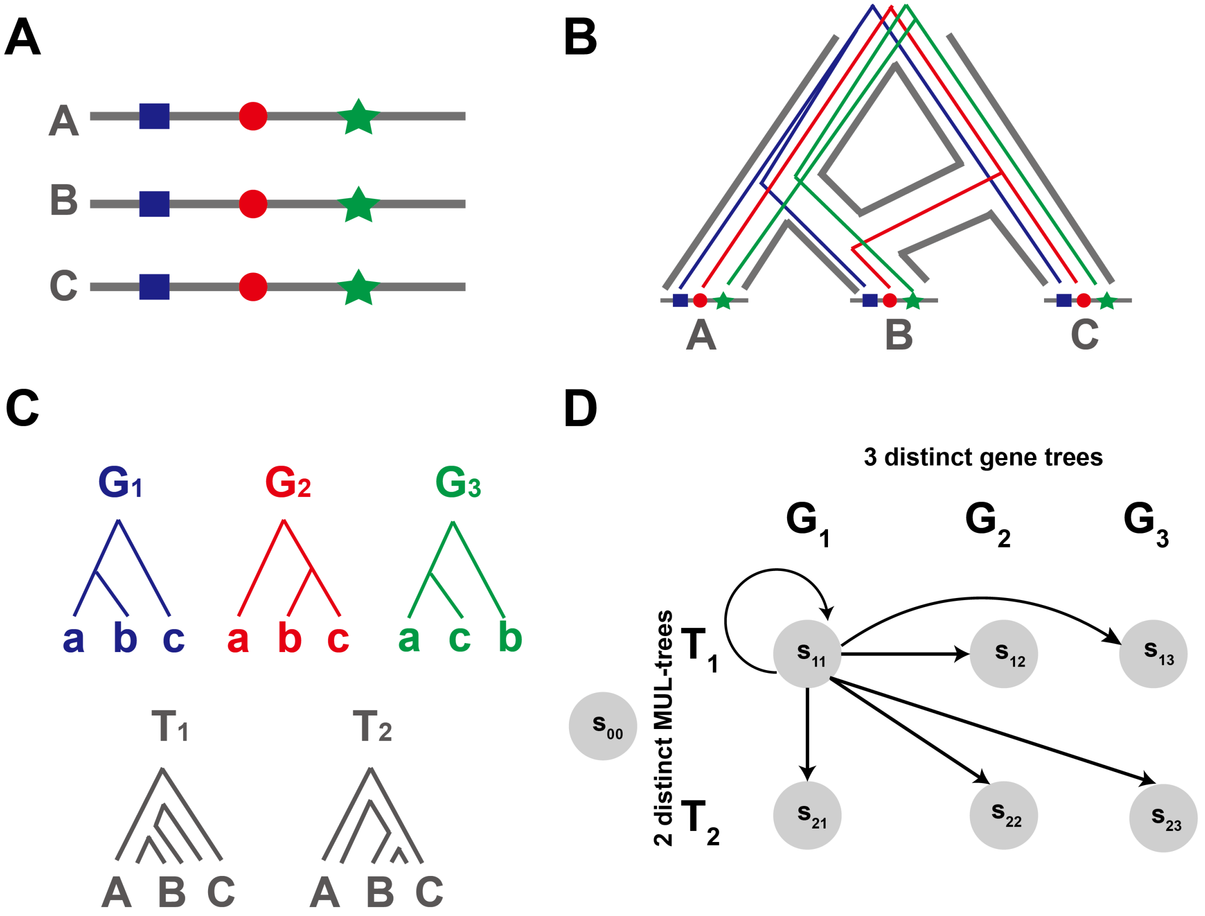


Figure 3.1 **An illustration of the PhyloNet-HMM framework.** (A) The genomes of three species A, B, and C. (B) The species network of A, B, and C, which has a reticulate evolutionary history, where individuals in B have some genetic material from the common ancestor of B and A, and other genetic material from C. (C) Corresponding gene trees and parental species trees. The blue, red, and green "locus" in the genomes have G_1, G_2, and G_3 as their local gene trees, respectively. Further, gene trees G_1 and G_3 for the blue and green loci evolved within the parental species tree T_1, whereas gene tree G_2 for the red locus evolved within the parental species tree T_2. (D) The structure of the HMM (only states are shown). The states s_{11}, s_{12}, and s_{13} correspond to three possible local gene trees G_1, G_2, and G_3 with evolution following the parental tree T_1, while states s_{21}, s_{22}, and s_{23} correspond to three possible local gene trees G_1, G_2, and G_3 with evolution following the parental species tree T_2. s_{00} is the start state. Note that only transitions outgoing from $s_{11} = (T_1, G_1)$ are shown to simplify the presentation. This figure is reproduced from Liu et al. [12].

The first stage consists of the following “truncation” algorithm:

(a) Under the input species network model $\mathcal{N}$, we conduct a Monte Carlo sampling of z local gene tree topologies from the gene tree topology distribution under the MSNC model. Here, z is set to 1000 in the simulation experiments and empirical analyses. The observed frequency distribution of local gene tree topologies is normalized to obtain an estimated probability distribution $\hat{f_{\mathcal{N}}}$.

(b) Topologies in the domain of $\hat{f_{\mathcal{N}}}$ are ranked based on their estimated probabilities. Let Δ be the top k_n topologies based on the topology ranking. In this study, k_n is set to 30.

(c) The distribution $\hat{f_{\mathcal{N}}}$ is truncated such that the domain consists only of topologies in Δ. Then, a truncated probability distribution $g_{\mathcal{N}}$ is obtained by normalizing the truncated distribution $\hat{f_{\mathcal{N}}}$.

The second stage of the PHiMM algorithm constructs a hidden Markov model (HMM) in a manner similar to the PhyloNet-HMM algorithm with a single modification. The set of MUL-trees T_i ($1 \leq i \leq m$, where m is the number of MUL-trees) in the MUL-tree representation of N is enumerated using the procedure described by Yu et al. [20]. HMM state construction utilizes the truncated distribution $\hat{f_{\mathcal{N}}}$ rather than the full theoretical distribution f, where a row of HMM states is instantiated for each distinct MUL-tree T_i and each state in a row corresponds to a distinct local gene tree topology G_j ($1 \leq j \leq k_n = |\Delta|$) in the domain of $g_{\mathcal{N}}$.

The final stage of the PHiMM algorithm performs model learning and statistical inference under the fitted model using the same procedures as the PhyloNet-HMM algorithm.

3.4 Materials

3.4.1 Simulation Data

The simulation study is used to evaluate the performance and applicability of the proposed PHiMM method, since we can track the true history of evolutionary events. The simulation data are constructed through various tools, such as r8s [28], ms [30], msmove [29], and seq-gen [32].

(1) Construction of model phylogenies

The model phylogenies are generated using the procedure of Hejase et al. [11].

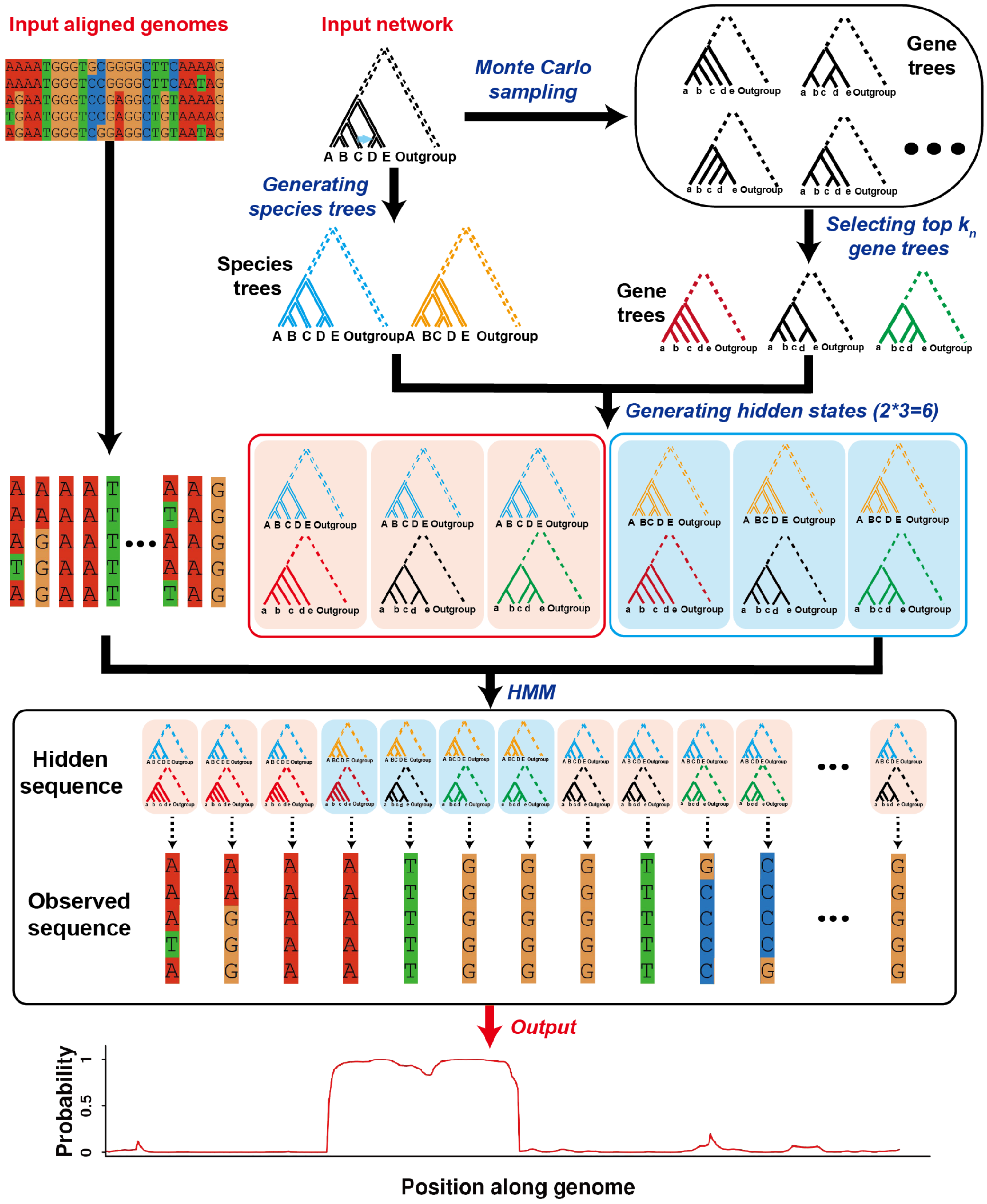


Figure 3.2 **The PHiMM pipeline.** This flowchart illustrates the three-stage PHiMM algorithm for statistical introgression mapping: **Stage 1 ("Truncation" Algorithm)** begins by performing Monte Carlo sampling of local gene tree topologies under the input species network and ranking these topologies according to their empirical frequencies. Only the top k_n topologies are retained, and a truncated probability distribution is formed. **Stage 2 (HMM Construction)** leverages the truncated set of topologies rather than the full gene tree distribution. Each MUL-tree derived from the species network is enumerated, and each state in the hidden Markov model corresponds to one of the retained local gene tree topologies attached to a specific MUL-tree. **Stage 3 (Model Learning and Inference)** applies maximum-likelihood or similar routines (as in PhyloNet-HMM) to the constructed HMM, producing a site-by-site probability of introgression along the input aligned genomes. Through these steps, PHiMM provides a computationally efficient framework to detect introgression signals while accounting for the complexity of network-like evolutionary histories.This figure is reproduced from Wuyun et al. [122].

Algorithm 3.1 PHiMM

```
procedure PHiMM($N, A$)
    $\mathcal{N} \leftarrow GetSpeciesNetworkModel(N)$    ▷ $N$: Phylogenetic network
                                                          ▷ $\mathcal{N}$: Species network model
    $\Delta_z \leftarrow \emptyset$    ▷ $\Delta_z$: Sampled gene tree topologies
    int $i \leftarrow 1$
    while $i \leq z$ do    ▷ $z$: Sampling size
        $\Delta_z \leftarrow \Delta_z + GeneTreeMonteCarloSampling(N, \mathcal{N})$
        $i \leftarrow i + 1$
    $\Delta_d \leftarrow GetDistinctGeneTrees(\Delta_z)$    ▷ $\Delta_d$: Distinct gene tree topologies in $\Delta_z$
    $\hat{f_{\mathcal{N}}} \leftarrow EstimateProbability(\Delta_d)$    ▷ $\hat{f_{\mathcal{N}}}$: Estimated probability distribution of $\Delta_d$
    $\hat{f_{\mathcal{N}}} \leftarrow RankTopology(\hat{f_{\mathcal{N}}})$
    $\Delta \leftarrow Truncate(\hat{f_{\mathcal{N}}}, \Delta_d, k_n)$    ▷ $k_n$: Truncation size; $\Delta$: Selected gene tree topologies
    $\hat{g_{\mathcal{N}}} \leftarrow EstimateTruncatedProbability(\Delta)$
                                   ▷ $\hat{g_{\mathcal{N}}}$: Estimated probability distribution of $\Delta$

    $\theta \leftarrow InitializeModelParameters(N, \Delta, \hat{g_{\mathcal{N}}})$    ▷ $\theta$: Model parameters
    while Not reaching the convergence criteria do
        $\theta \leftarrow HeuristicLearning(\theta, A)$
            ▷ $A$: Input multiple sequence alignment with $K$ aligned sequences and $L$ columns
    $\{p_t\}_{1 \leq t \leq L} \leftarrow ModifiedPosteriorDecoding(\theta, N, A)$
                     ▷ $p_t$: Introgression probability for each aligned site $t$ ($1 \leq t \leq L$)
    return $\{p_t\}_{1 \leq t \leq L}$
```

First, the r8s [28] tool is used to generate a random phylogenetic tree. The r8s version is 1.81, and the commands are shown below:

```
#nexus
begin r8s;
simulate diversemodel=bdback seed=<integer random seed>
charevol=yes ntaxa=<5, 6, 7, 8, 9, or 10> infinite=yes nreps=20;
end;
```

where “diversemodel” means the model used for tree generation, which is generally set to the birth-death model denoted by “bdback”. “seed” specifies a random seed for the tree generation. “ntaxa” represents the number of taxa, which is selected from 5 to 10 in this study. “nreps” indicates the number of generated repeats. “charevol=yes” indicates the model tree is output with branch lengths. “infinite=yes” represents that the branch lengths are set to the expected values based on rate and time.

Then, the resulting tree is scaled to a height h of 1 by multiplying the length of each edge in the model tree by h. Furthermore, an outgroup is added to the generated tree at 10.0 coalescent time.

In order to get the phylogenetic network for input, we finally add r reticulations ($r \in [1, 2]$) by iterating the following steps: a time t_M between 0 and the tree height is selected uniformly at random, two tree edges for which corresponding ancestral populations existed during a time interval $[t_A, t_B]$ such that $t_M \in [t_A, t_B]$ are randomly selected, and a reticulation at time t_M is added to connect the pair of tree edges. Similar to Leaché et al. [15], the model network can be further classified based upon whether gene flow is "deep" or "non-deep", which is defined by the topological placement of reticulations, i.e., non-deep reticulations are placed between two leaf edges, while deep reticulations include all other reticulations.

(2) Generation of local genealogies

Local genealogy at each locus is simulated using the msmove [29] or ms [30] following the above species network. The local gene trees are modeled for independent and identically distributed (i.i.d.) loci under a multi-species network coalescent with recombination (MSNCwR) model. msmove is modified from the Monte Carlo simulator ms [30] to allow the tracking of migration events, while ms does not provide this annotation. The following msmove or ms command is used:

```
msmove <number of samples> <number of repeats> -T
-r <crossover rate> <number of sites>
-I <number of populations> <n_1 n_2 ...  n_k>
-ej <t_1> i_1 j_1 -ej <t_2> i_2 j_2 ...  -ej <t_k> i_k j_k
-ev <t_m> i j <probability x>
```

or

```
ms <number of samples> <number of repeats> -T
```

```
-r <crossover rate> <number of sites>
-I <number of populations> <n_1 n_2 ...  n_k>
-ej <t_1> i_1 j_1 -ej <t_2> i_2 j_2 ...  -ej <t_k> i_k j_k
-em <t_m1> i j <x_1> -em <t_m2> i j <x_2>
```

where -T parameter indicates the gene trees representing the history of the sampled taxa are output. The -I parameter is followed by k that represents the number of populations. The list of integers (n_1 n_2 ... n_k) includes the number of taxa sampled in each population. In this study, one allele is sampled from each taxon. The -r parameter is used to set recombination under Hudson's finite-sites recombination model [30], where the crossover rate or recombination rate ρ is set to 1 for msmove or 10 for ms, and the number of sites between which recombination can occur is set to 100 for msmove or 1000 for ms. The -ej parameter specifies moving all lineages in population i to population j at time t. The -ev parameter is special for msmove, which sets migration at time t_m from population i to population j with the migration probability x. The migration rate is selected randomly for each replicate in this study. In contrast, the -em parameter for the ms tool sets migration from subpopulation j to subpopulation i at time t_m1 or t_m2 with migration rate as x_1 or x_2.

(3) Simulation of DNA sequences

The DNA sequences are generated using seq-gen [32] under the Jukes-Cantor substitution model [31] with mutation rate $\theta = 2$. The sequence simulation is performed under the procedure of multi-locus genomic sequence evolution. In detail, we simulate two different classes of loci: "query" loci that are sampled from the MSNC model with a recombination rate ρ of 1 based on msmove tool to produce shorter sequences with a length of 100 bp, and "non-query" loci that are sampled with a recombination rate ρ of 10 based on ms tool to generate longer sequences with a length of 1 kb. Loci from the two classes are interleaved, resulting in ten query loci and nine non-query loci sampled per dataset, where each locus is independent and identically distributed (i.i.d.). The following seq-gen command is used:

```
seq-gen -mHKY -l <sequence length> -p <number of partitions>
< genetreefile > seqfile
```

where -m parameter specifies the substitution model. Here, the Jukes-Cantor substitution model denoted by “HKY” is used. The -l parameter specifies the sequence length. The -p parameter sets the number of partitions. The genetreefile is the input file providing the gene trees. The seqfile is the output file with simulated sequences under the given gene trees.

The sampling design is carefully designed to ensure that the query loci are sufficiently separated by sequence length. This allows us to make the assumption of free recombination between the query loci based on the observed decay of linkage disequilibrium in previous empirical studies [127].

In the simulation study, we conduct 20 repetitions of the simulation procedure for each model condition. To assess the performance of the methods, we calculate the average and standard error of the performance measures across these 20 replicates. The statistical summary of the simulation dataset is presented in Table 3.1.

3.4.2 Performance Assessments

To evaluate and compare the performance of different approaches on the simulation dataset where the true history of evolutionary events can be tracked, we use two types of area under the curve (AUC): the area under the receiver-operating characteristic (ROC) curve, and the area under the Precision-Recall curve, referred to as simply ROC-AUC and PR-AUC, respectively. ROC-AUC is plotted by the true positive rate ($\frac{TP}{TP+FN}$) as a function of the false positive rate ($\frac{FP}{FP+TN}$) at various threshold settings, while PR-AUC similarly plots the precision ($\frac{TP}{TP+FP}$) against the recall ($\frac{TP}{TP+FN}$) at different thresholds, where TP, FP, TN, and FN represent the numbers of true positives, false positives, true negatives, and false negatives, respectively. The ROC-AUC and PR-AUC measures represent the tradeoff between type I errors and type II errors under different threshold values. The measures are calculated only on the “query” loci in the simulation dataset, where the true migration events are annotated by msmove. After concatenating all query loci in one simulation replicate, the ROC-AUC and PR-AUC of a method are assessed based on the probability of a particular site

Table 3.1 **Statistics for the simulation dataset.** The reticulation scenarios include one non-deep reticulation, two non-deep reticulations, and one deep reticulation. “Ntaxa” means the number of taxa. “Total Sites” indicates the number of total sites in the simulation genomes for running. “Sites for analysis” represents the number of “query” sites for analyzing the performance of introgression detection methods. The p-distance of a pair of aligned sequences is calculated by dividing the number of sites where the two sequences had different nucleotides by the number of sites in which both sequences had nucleotides. “Average/Maximum p-dist” is the average/maximum p-distance of all pairs of aligned sequences in the simulation dataset. “Introgression (%)” shows the percent of introgressed sites over the genome. “Network Height” gives the height of the model network. “Average Branch Length” is the average branch length of the model network. The average and standard error (SE) are shown based on 20 replicates.

Reticulation Scenario	**Ntaxa**	**Total Sites**	**Sites for analysis**	**Average p-dist (%)**		**Maximum p-dist (%)**		**Introgression (%)**		**Network Height**		**Average Branch Length**	
				Average	**SE**	**Average**	**SE**	**Average**	**SE**	**Average**	**SE**	**Average**	**SE**
one non-deep	5	10kb	1kb	68.307	2.339	73.317	0.624	54.000	13.565	1.000	0.000	0.337	0.052
	6	10kb	1kb	67.474	2.251	73.254	0.759	51.000	13.379	1.000	0.000	0.288	0.039
	7	10kb	1kb	68.039	2.434	73.403	0.386	47.500	12.600	1.000	0.000	0.297	0.050
	8	10kb	1kb	68.673	1.368	73.538	0.442	55.000	15.330	1.000	0.000	0.285	0.036
	9	10kb	1kb	68.327	1.493	73.446	0.310	60.500	14.654	1.000	0.000	0.270	0.041
	10	10kb	1kb	68.878	1.196	73.630	0.330	52.500	13.370	1.000	0.000	0.275	0.037
two non-deep	5	10kb	1kb	66.634	2.567	72.763	0.638	70.000	13.416	1.000	0.000	0.273	0.046
	6	10kb	1kb	66.945	3.350	72.645	1.560	79.000	11.790	1.000	0.000	0.272	0.040
	7	10kb	1kb	67.551	1.874	73.136	0.346	78.500	11.079	1.000	0.000	0.253	0.035
	8	10kb	1kb	68.882	1.270	73.468	0.378	72.500	16.086	1.000	0.000	0.264	0.034
	9	10kb	1kb	68.490	1.522	73.325	0.449	75.500	15.322	1.000	0.000	0.246	0.038
	10	10kb	1kb	68.256	1.683	73.535	0.312	74.500	11.169	1.000	0.000	0.241	0.033
one deep	5	10kb	1kb	66.605	2.501	72.330	1.996	72.500	16.086	1.000	0.000	0.300	0.042
	6	10kb	1kb	67.891	2.168	72.584	1.360	74.000	10.198	1.000	0.000	0.293	0.052
	7	10kb	1kb	67.262	2.224	73.072	0.503	80.000	10.000	1.000	0.000	0.272	0.054
	8	10kb	1kb	68.372	2.288	73.238	0.546	72.000	13.638	1.000	0.000	0.261	0.042
	9	10kb	1kb	68.056	1.736	73.456	0.400	79.500	13.219	1.000	0.000	0.247	0.048
	10	10kb	1kb	68.936	1.078	73.535	0.404	75.000	12.450	1.000	0.000	0.258	0.032

involving an introgressive origin.

Additionally, we report the memory usage and runtime in order to comprehensively evaluate the scalability of our framework.

3.4.3 Software and Data

Our PHiMM algorithm is implemented as custom software, which includes a tailored version of the MSNC-based Monte Carlo algorithm and custom adaptations of the PhyloNet software package [13, 123]. All of our software and study datasets are open-source and publicly available for access at https://gitlab.msu.edu/liulab/phimm-dataset.

3.5 Results

We present a comprehensive evaluation of the PHiMM framework using simulated datasets designed to capture a variety of factors that could influence introgression detection accuracy. Specifically, we focus on (1) the effect of varying the gene tree truncation size k_n, (2) the impact of increasing the number of taxa, and (3) the influence of different numbers and depths of reticulations. We further compare the performance of PHiMM with PhyloNet-HMM in terms of accuracy, runtime, and memory usage.

3.5.1 Effect of gene tree truncation size

An initial step in implementing PHiMM involves determining the optimal gene tree truncation size k_n, which directly affects the number of hidden states in the HMM. As depicted in Figure 3.3, both memory usage and runtime exhibit an increasing trend with the number of gene trees, particularly for networks involving 10 taxa. This outcome is expected because more gene trees translate into a larger state space for the HMM, thereby necessitating greater computational resources.

Despite the rise in computational cost, Figure 3.3 also shows that detection accuracy remains relatively stable across a wide range of gene tree counts. This stability implies that many gene trees may be redundant with respect to introgression detection, allowing us to choose a moderate truncation size without sacrificing predictive performance. Based on these findings, we set $k_n = 30$ as a default, thereby balancing accuracy with memory and runtime efficiency.

3.5.2 Impact of the number of taxa and reticulation number and depth

To further explore PHiMM's performance, we assess how the number of taxa and various reticulation scenarios affect both computational requirement (memory and runtime) and detection accuracy.

Figure 3.4 gives the performance of the PHiMM approach on different numbers of taxa and different reticulation scenarios. We find that the memory and runtime increase as the function of the number of taxa grows from 5 to 10. But overall, the memory and runtime are under 10 GB or 10 hours. Furthermore, the ROC-AUC of PHiMM remains above 0.75 on different numbers of taxa with non-deep reticulations. However, ROC-AUC on more than 7 taxa involving deep reticulations is reduced below 0.7.

Furthermore, we analyze the influence of different numbers or types of reticulations. There are three scenarios: one non-deep reticulation, two non-deep reticulations, and one deep reticulation, because the memory and runtime would rise as the number of reticulations increases. As shown in

Figure 3.4, we find that the memory usage grows as the number of reticulations increases, while the runtime remains relatively unchanged by comparing the results of one non-deep reticulation with those of two non-deep reticulations. However, the ROC-AUC is between 0.85 and 0.9 for two non-deep reticulations, which is significantly higher than that for one non-deep reticulation. On the other hand, we compare the difference between non-deep and deep reticulations. We noticed that the memory and runtime are approximately similar for these two scenarios. However, the ROC-AUC is comparatively lower for one deep reticulation (below 0.7 for more than 7 taxa). This is straightforward because the deep reticulation is more complex, thus resulting in hard calculations and optimizations in PHiMM.

3.5.3 Comparison with PhyloNet-HMM

Finally, we compare the PHiMM framework with PhyloNet-HMM. Since PhyloNet-HMM is time and memory-intensive, we only make the evaluation based on the 5-taxon phylogenetic network with one non-deep reticulation as input. In Table 3.2, the memory usage and runtime for PHiMM are only 2.4 GB and 0.1 hours, which are 0.3% and 0.7% of those for PhyloNet-HMM (318.6 GB and 40.9 hours), respectively, indicating that the PHiMM framework can dramatically reduce the memory usage and runtime for introgression detection when compared with the PhyloNet-HMM. However, PHiMM's ROC-AUC and PR-AUC are comparable with PhyloNet-HMM's, even though the ROC-AUC of PHiMM (0.7653) is 2% lower than that of PhyloNet-HMM (0.7806).

Overall, these results demonstrate that PHiMM significantly reduces computational overhead without severely compromising accuracy. Consequently, PHiMM represents a compelling alternative for large-scale introgression mapping, especially in datasets with dozens of DNA sequences and complex reticulation patterns.

3.6 Discussion and Conclusion

In this study, we introduced PHiMM, a novel introgression detection framework that employs a coalescent-based approximation strategy to reduce model complexity and enhance detection performance in large-scale genomic datasets. Through extensive simulation experiments, we demonstrated that PHiMM achieves significant improvements in runtime and memory usage —

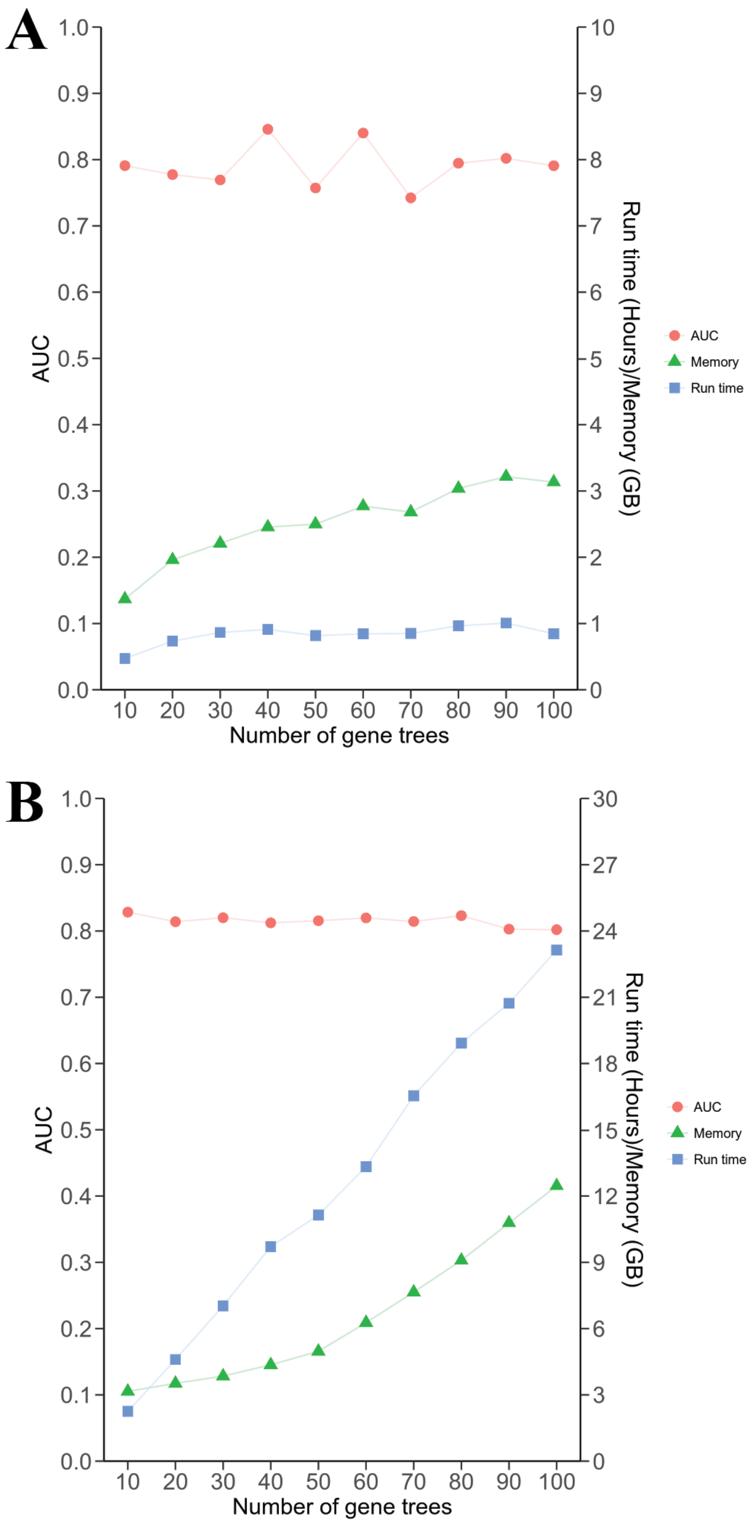


Figure 3.3 **Results for different numbers of gene trees for (A) 5 and (B) 10 taxa input network.** The simulation data is generated under the model conditions: one non-deep reticulation. The number of gene trees used ranges from 10 to 100. Performance measures include the area under the receiver operating characteristic curve (denoted by AUC) (red points), runtime (blue rectangles), and memory usage (green triangles). Averages and standard error bars are depicted based on 20 replicates. This figure comes from Wuyun et al. [122].

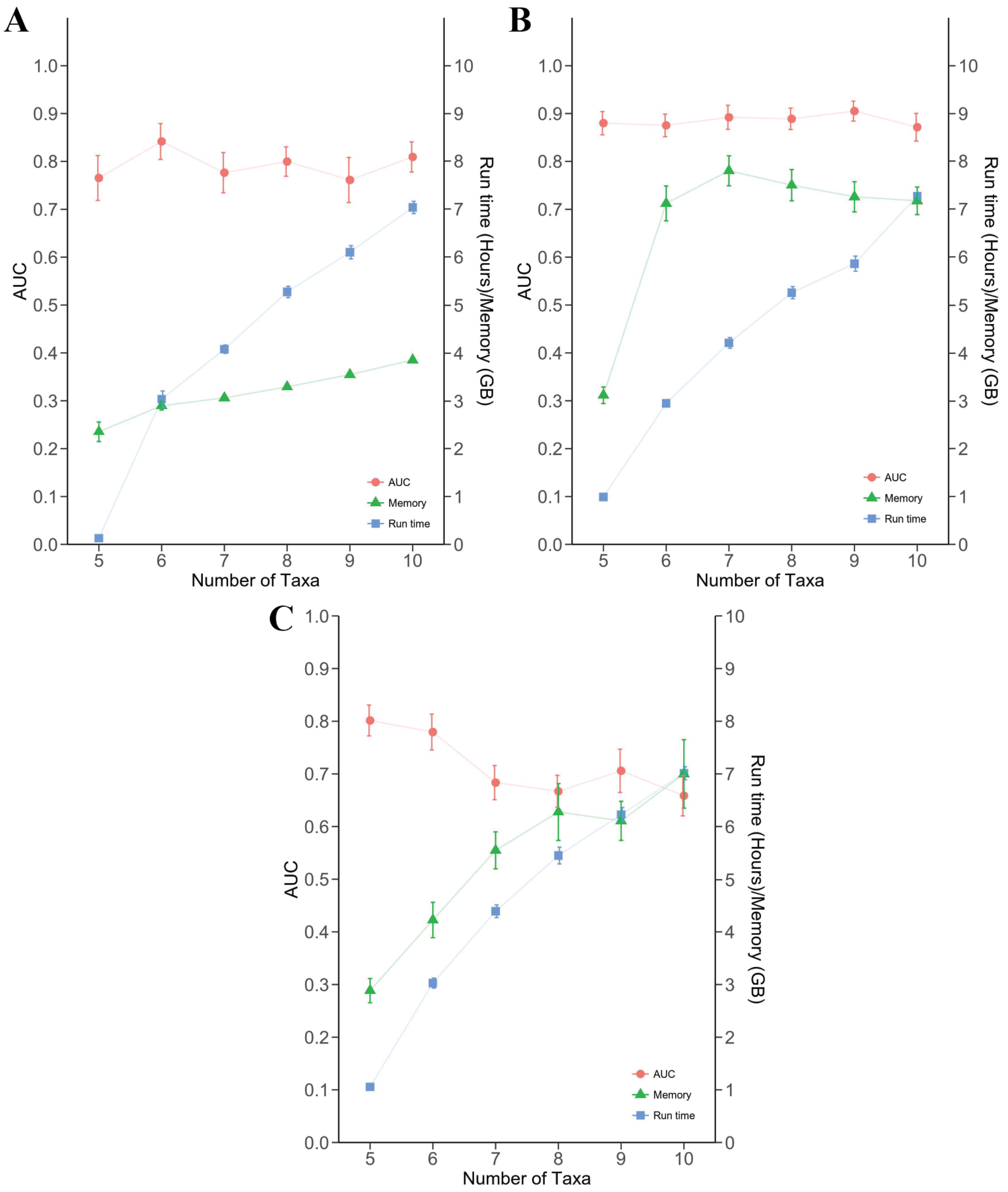


Figure 3.4 **The performance of PHiMM on model conditions with (A) one non-deep reticulation, (B) two non-deep reticulations, and (C) one deep reticulation.** Performance measures include the area under the receiver operating characteristic curve (denoted by AUC) (red points), runtime (blue rectangles), and memory usage (green triangles). Averages and standard error bars are depicted based on 20 replicates. This figure comes from Wuyun et al. [122].

Table 3.2 **The performance comparison between PHiMM and PhyloNet-HMM on the 5-taxon model condition with one non-deep reticulation.** Performance measures include the area under the receiver-operating characteristic curve (ROC-AUC), the area under the precision-recall curve (PR-AUC), runtime, and memory usage. The average and standard error (SE) are calculated based on 20 replicates. This table comes from Wuyun et al. [122].

	PhyloNet-HMM		**PHiMM**	
	Average	**SE**	**Average**	**SE**
ROC-AUC	0.7806	0.0534	0.7653	0.0523
PR-AUC	0.7197	0.0871	0.7305	0.0726
Runtime (hour)	40.8968	0.5607	0.1291	0.0035
Memory (GB)	318.6493	4.3099	2.3527	0.2271

often by several orders of magnitude — when compared to the state-of-the-art PhyloNet-HMM. This exceptional scalability is primarily attributable to the truncation strategy built into PHiMM, wherein the model approximation procedure effectively curtails the number of parameters involved in local optimizations.

Although model approximations typically raise concerns regarding trade-offs between computational efficiency and inference precision, our results show that PHiMM maintains accuracy levels comparable to PhyloNet-HMM, with only a marginal decrease in ROC-AUC and PR-AUC. One plausible explanation for this robust performance is that discarding redundant gene trees or parameters through truncation strategically simplifies the model without forfeiting critical genetic signals required for introgression inference.

Our findings further indicate that PHiMM's detection accuracy is generally resilient across variations in the number of taxa and the complexity of reticulation patterns. Nonetheless, a modest drop in ROC-AUC is observed in datasets containing a large number of taxa and deep reticulations. These observations align with recent studies suggesting that deep gene flow or ancient evolutionary events impose greater challenges for phylogenetic inference than more recent divergence scenarios [11, 14, 15]. Moreover, although the computational demands of PHiMM increase in proportion to the number and depth of reticulations, as well as the overall number of taxa, these requirements remain within manageable limits for modern high-performance computing systems.

In conclusion, PHiMM represents a powerful alternative for introgression mapping, combining computational efficiency with reliable inference of hybridization events. The framework's scalability, coupled with its robust accuracy, renders it particularly well-suited to the burgeoning volume and complexity of contemporary genomic datasets. Future endeavors could focus on refining the approximation strategies to further enhance performance on data characterized by deep reticulations, as well as extending PHiMM to incorporate additional sources of genomic heterogeneity such as gene duplication and loss. Through these developments, PHiMM has the potential to offer a versatile platform for unraveling the evolutionary processes that shape species diversity.

CHAPTER 4

APPLICATION OF PHIMM INTROGRESSION DETECTION METHOD TO EMERGING MODEL SYSTEMS

4.1 Introduction

In our previous work, we introduced the PHiMM approach as a scalable and accurate method for detecting introgression in simulated genomic data. Through a series of computational experiments, we demonstrated that PHiMM maintains high inference accuracy while dramatically enhancing the scalability of introgression detection. Encouraged by these promising simulation-based results, the next logical step is to validate the performance of PHiMM on empirical datasets that capture the complexity and heterogeneity inherent in real-world biological systems. Accordingly, this study applies PHiMM to two well-characterized cases of introgression, thereby evaluating its effectiveness and robustness under practical evolutionary scenarios.

The first empirical dataset is the widely-studied case of adaptive introgression among house mice, i.e., *Mus musculus dometicus* (*M. m. dometicus*) and *Mus spretus* (*M. spretus*). The adaptive significance of this introgression is evident in the acquired resistance to anticoagulant rodenticides by *M. m. dometicus*. This resistance is conferred by the introgression of *Vkorc1* (vitamin K 2,3-epoxide reductase subcomponent 1) gene from the gene pool of *M. spretus* to *M. m. dometicus* [120]. This case not only exemplifies how introgression can confer immediate and profound adaptive benefits but also provides a tractable system in which to assess the accuracy of PHiMM's introgression mapping under a known evolutionary outcome.

The second empirical dataset is from studies of mimicry in the butterfly genus *Limenitis*. The WntA gene was found to be responsible for wing mimicry among *Limenitis* [128]. This dataset allows us to explore the role of introgression in morphological diversification and ecological adaptation. By examining how introgression may have contributed to morphological diversification and ecological adaptation, this dataset allows for a broader evaluation of PHiMM's performance under selective pressures distinct from those observed in rodents. Moreover, these butterflies present an opportunity to interrogate introgression events that shape complex phenotypes, such as wing coloration and patterning.

By applying PHiMM to these two divergent empirical systems — an adaptive introgression conferring rodenticide resistance in mice and morphological mimicry in butterflies — we aim to (i) verify the consistency of our simulation-derived findings in real datasets, and (ii) explore the broader applicability of PHiMM to evolutionary processes involving diverse selection pressures. This study thereby not only extends PHiMM's utility beyond controlled simulation conditions but also contributes to a deeper understanding of the adaptive and ecological impacts of introgression in natural populations.

4.2 Materials

4.2.1 Mouse Data

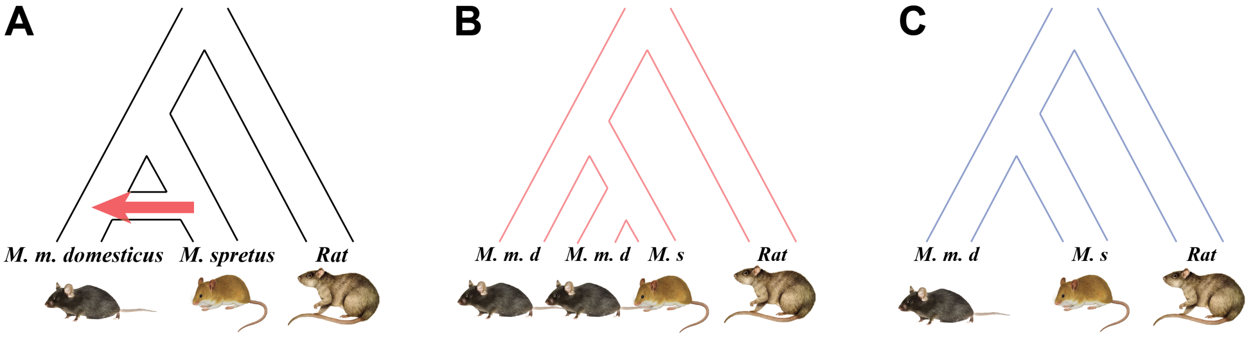


Figure 4.1 **The phylogenetic network and corresponding parental species trees used for evaluating PhyloNet-HMM and PHiMM on mouse data.** The phylogenetic network captures (A) introgression from *M. spretus* to *M. m. domesticus*. The parental tree in (B) captures genomic regions of introgressive descent, while the parental tree in (C) captures genomic regions with no introgression.

The data gathered from the mice include empirical genomic sequence datasets with positive and negative control loci. The datasets are sampled from wild-derived and wild mouse samples from *M. m. domesticus* and *M. spretus*. For comparison purposes, we replicate a subset of the PhyloNet-HMM analyses conducted in the research by Liu et al. [7], which utilized genomic sequence data from Didion et al. [131]. We briefly review relevant methodological details here (see [7] for more details). The data were sequenced using an SNP array designed by Yang et al. [129]; raw reads from the array were genotyped using MouseDivGeno software [131]. The genotypic sequence data

Table 4.1 **Mouse samples used in this study.** We used existing mouse samples from previous studies. Each row lists a sample's name, species, commonly used alias, the geographic origin of the population, and the corresponding reference source. The *M. m. domesticus* entries represent house mouse populations from various locations in Spain, Germany, Cyprus, and Georgia, while the *M. spretus* entry corresponds to a distinct mouse species from the Cadiz Province in Spain. References to original sources are also included for traceability.

Sample Name	Species	Alias	Origin	Source
Spain-Arenal	*M. m. domesticus*	MWN1279	Arenal, Mallorca Island, Spain	[129]
Spain-RocadelValles	*M. m. domesticus*	MWN1287	Roca del Valles, Catalunya, Spain	[129]
Germany-Hamm-A	*M. m. domesticus*	B9	Hamm, North Rhine-Westphalia, Germany	[7]
Germany-Hamm-B	*M. m. domesticus*	B10	Hamm, North Rhine-Westphalia, Germany	[7]
Germany-Hamm-C	*M. m. domesticus*	B11	Hamm, North Rhine-Westphalia, Germany	[7]
Germany-Hamm-D	*M. m. domesticus*	C1	Hamm, North Rhine-Westphalia, Germany	[7]
Germany-Hamm-E	*M. m. domesticus*	C2	Hamm, North Rhine-Westphalia, Germany	[7]
Germany-Hamm-F	*M. m. domesticus*	C3	Hamm, North Rhine-Westphalia, Germany	[7]
A-background	*M. m. domesticus*	DCA	Akotiri, Cyprus	[129, 130]
B-background	*M. m. domesticus*	DCP	Paphos, Cyprus	[129, 130]
C-background	*M. m. domesticus*	DGA	Adjaria, Georgia	[129, 130]
Spretus	*M. spretus*	SPRET/EiJ	Puerto Real, Cadiz Province, Spain	[129, 130]

Table 4.2 **Mouse datasets.** Each dataset comprises a subset of *Mus* samples selected for phylogenetic and comparative genomic analyses using PhyloNet-HMM or PHiMM. For PhyloNet-HMM, four haploid *Mus* genomes were included, while the extended PHiMM analysis incorporated five *Mus* genomes. The extended PHiMM datasets (noted in parentheses) specifically incorporate the additional *Mus* genome to deepen the inference of phylogenetic histories and evolutionary relationships across mouse populations. An additional rat genome (*Rattus norvegicus* reference genome RGSC Rnor_5.0) served as an outgroup.

Dataset	Set of samples included
Spain-Arenal	Spain-Arenal, A-background, B-background, C-background (for extended PHiMM analysis), Spretus
Spain-RocadelValles	Spain-RocadelValles, A-background, B-background, C-background (for extended PHiMM analysis), Spretus
Germany-Hamm-A	Germany-Hamm-A, A-background, B-background, C-background (for extended PHiMM analysis), Spretus
Germany-Hamm-B	Germany-Hamm-B, A-background, B-background, C-background (for extended PHiMM analysis), Spretus
Germany-Hamm-C	Germany-Hamm-C, A-background, B-background, C-background (for extended PHiMM analysis), Spretus
Germany-Hamm-D	Germany-Hamm-D, A-background, B-background, C-background (for extended PHiMM analysis), Spretus
Germany-Hamm-E	Germany-Hamm-E, A-background, B-background, C-background (for extended PHiMM analysis), Spretus
Germany-Hamm-F	Germany-Hamm-F, A-background, B-background, C-background (for extended PHiMM analysis), Spretus

Table 4.3 **Statistics for the mouse datasets.** "Haplotype" indicates the haplotype name in each dataset. "Nchrs" means the number of chromosomes in each dataset. "Sites" indicates the number of total sites from all chromosomes. The p-distance of a pair of aligned sequences is calculated by dividing the number of sites where the two sequences had different nucleotides by the number of sites in which both sequences had nucleotides. "Average/Maximum p-dist^a" is the average/maximum p-distance of all pairs of aligned sequences in the PhyloNet-HMM datasets with 4 ingroup taxa, while "Average/Maximum p-dist^b" is for the extended PHiMM datasets with 5 ingroup taxa. The average and standard error (SE) are shown based on all chromosomes.

Dataset	**Haplotype**	**Nchrs**	**Sites**	**Average p-dist^a**		**Maximum p-dist^a**		**Average p-dist^b**		**Maximum p-dist^b**	
				Average	**SE**	**Average**	**SE**	**Average**	**SE**	**Average**	**SE**
Spain-Arenal	1	20	414376	26.590	1.381	35.008	0.777	25.261	1.575	35.008	0.777
Spain-Arenal	2	20	414376	26.621	1.367	35.141	0.884	25.287	1.587	35.141	0.884
Spain-RocadelValles	1	20	414376	26.609	1.334	34.983	0.884	25.266	1.543	34.983	0.884
Spain-RocadelValles	2	20	414376	26.642	1.352	35.142	0.921	25.295	1.579	35.142	0.921
Germany-Hamm-A	1	20	414376	27.622	1.458	36.918	0.883	26.112	1.638	36.918	0.883
Germany-Hamm-A	2	20	414376	27.623	1.482	36.909	0.923	26.107	1.682	36.909	0.923
Germany-Hamm-B	1	20	414376	27.644	1.462	36.918	0.908	26.123	1.641	36.918	0.908
Germany-Hamm-B	2	20	414376	27.624	1.475	36.905	0.956	26.119	1.671	36.905	0.956
Germany-Hamm-C	1	20	414376	27.571	1.484	36.876	1.034	26.076	1.644	36.876	1.034
Germany-Hamm-C	2	20	414376	27.623	1.462	36.890	0.994	26.107	1.667	36.890	0.994
Germany-Hamm-D	1	20	414376	27.511	1.442	36.782	1.095	25.999	1.616	36.782	1.095
Germany-Hamm-D	2	20	414376	27.500	1.477	36.788	1.087	25.990	1.663	36.788	1.087
Germany-Hamm-E	1	20	414376	27.513	1.416	36.792	1.022	25.994	1.601	36.792	1.022
Germany-Hamm-E	2	20	414376	27.511	1.468	36.768	1.050	26.000	1.660	36.768	1.050
Germany-Hamm-F	1	20	414376	27.500	1.414	36.719	1.042	25.989	1.597	36.719	1.042
Germany-Hamm-F	2	20	414376	27.554	1.479	36.775	1.026	26.038	1.669	36.775	1.026

was phased into haploid genomic sequences using fastPHASE [132].

Comprehensive details about the data can be found in Table 4.1 and 4.2. Each dataset consists of genomic sequences from three *M. m. domesticus* samples, one *M. spretus* sample, and one *Rattus norvegicus* genome (RGSC Rnor_5.0/rn5) that is included as an outgroup (see Figure 4.1). Thus, each dataset consists of four ingroup taxa and one outgroup taxon. For the three *M. m. domesticus* samples, one originates from the region of sympatry between *M. m. domesticus* and *M. spretus*, and two are from far outside the sympatric region. Specifically, the *M. spretus* sample is also from the sympatric region. Therefore, we assume the network for this dataset involved a reticulation from *M. spretus* to the sympatric *M. m. domesticus*.

In addition to the datasets in Liu et al. [7], our study also incorporates larger "extended" datasets with a more extensive taxon sampling, which encompasses all the data from the original study as a strict superset. The larger size of the extended datasets necessitates the use of PHiMM for introgression mapping purposes. Other than the larger set of taxa in new datasets, all other aspects of empirical data are the same (i.e., genotyping, phasing, etc.). Notably, the extended datasets include an extra sample of *M. m. domesticus* from outside the region of sympatry between *M. m. domesticus* and *M. spretus* (Figure 4.1).

For the analyses of the extended datasets, PHiMM is run with the same settings used in the simulation study, except for two modifications. First, the number of iterations for model parameter learning is increased to 1000 (as opposed to 300). Second, model parameters are optimized using chromosome 7 from the extended Spain-Arenal dataset.

The summarized statistics of these datasets can be found in Table 4.3.

4.2.2 *Limenitis* Data

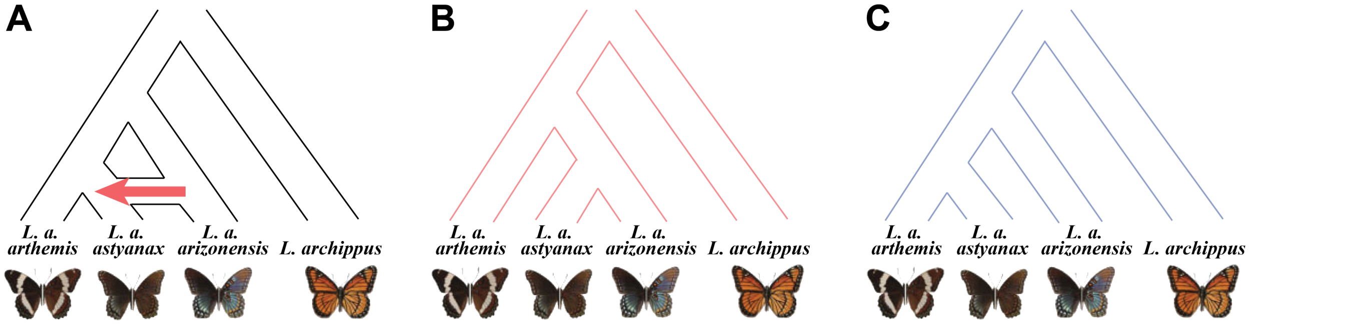


Figure 4.2 **The phylogenetic network and corresponding parental species trees used for evaluating PhyloNet-HMM and PHiMM on *Limenitis* data.** The phylogenetic network captures (A) introgression from *Limenitis arthemis arizonensis* to *Limenitis arthemis astyanax*. The parental tree in (B) captures genomic regions of introgressive descent, while the parental tree in (C) captures genomic regions with no introgression.

We also conduct a re-analysis of the *Limenitis* dataset described in the study by Gallant et al. [128]. For this re-analysis, we perform PhyloNet-HMM and PHiMM on the *Limenitis* AC scaffold, which contains the WntA gene. The dataset comprises four ingroup taxa, namely, *Limenitis arthemis arizonensis*, *Limenitis arthemis arthemis*, *Limenitis arthemis astyanax*, and *Limenitis archippus floridenesis*. Our assumption is that *Limenitis arthemis arthemis* and *Limenitis arthemis astyanax* first coalesce, followed by their ancestors coalescing with *Limenitis arthemis arizonensis*, and finally with *Limenitis archippus floridenesis*. In this 4-taxon network, a reticulation from *Limenitis arthemis arizonensis* to *Limenitis arthemis astyanax* is postulated (see Figure 4.2).

Due to the scalability limitations of PhyloNet-HMM, it is executed on the 4-taxon dataset, while PHiMM employs the extended dataset that includes an additional *Limenitis arthemis arthemis* sample (Figure 4.2).

4.3 Results

4.3.1 Mouse Data Re-analysis

In our study, we conduct a performance comparison between PHiMM and PhyloNet-HMM using mouse genomic sequence datasets previously analyzed in the works of Liu et al. [7] and Didion et al. [131]. However, due to the scalability constraints of PhyloNet-HMM, we run PhyloNet-HMM on a smaller dataset consisting of only four ingroup taxa. This smaller dataset is a proper subset of the larger dataset used for PHiMM analyses, which includes more samples of *M. m. domesticus* but remains otherwise identical. Additional information and detailed explanations of the datasets can be found in the Methods section.

Previous studies [7, 120] have reported instances of adaptive interspecific introgression involving the region around the *Vkorc1* gene, specifically the region on chromosome 7 between coordinates 123 Mb and 134 Mb. As depicted in Figure 4.3 and Figure A.1-A.20 of Appendix A, both PHiMM and PhyloNet-HMM methods infer the presence of multi-megabase-long introgressed tracts that are present in all eight samples from Spain and Germany, with the exception of the Arenal, Spain sample. Thus, both methods successfully detect interspecific introgression in this positive control.

Notably, the genomic region containing the *Vkorc1* gene shows the longest introgressed tracts detected in the mouse genome. Within this particular genomic region, it is worth mentioning that PHiMM infers a greater total sequence length of introgressed tracts compared to the inference made by PhyloNet-HMM.

Beyond *Vkorc1*, additional genomic regions spanning several hundred kilobases also exhibit substantial introgression signals, as reported by Liu et al. [7]. In many of these regions, PHiMM and PhyloNet-HMM converge on qualitatively similar inferences, reaffirming the presence of introgressed tracts across the genome.

However, when examining local inference patterns, two types of differences become evident. First, there are local differences in the pattern of introgressed and non-introgressed tracts, exemplified in regions such as the chromosome 7 region between coordinates 102 Mb and 108 Mb, and

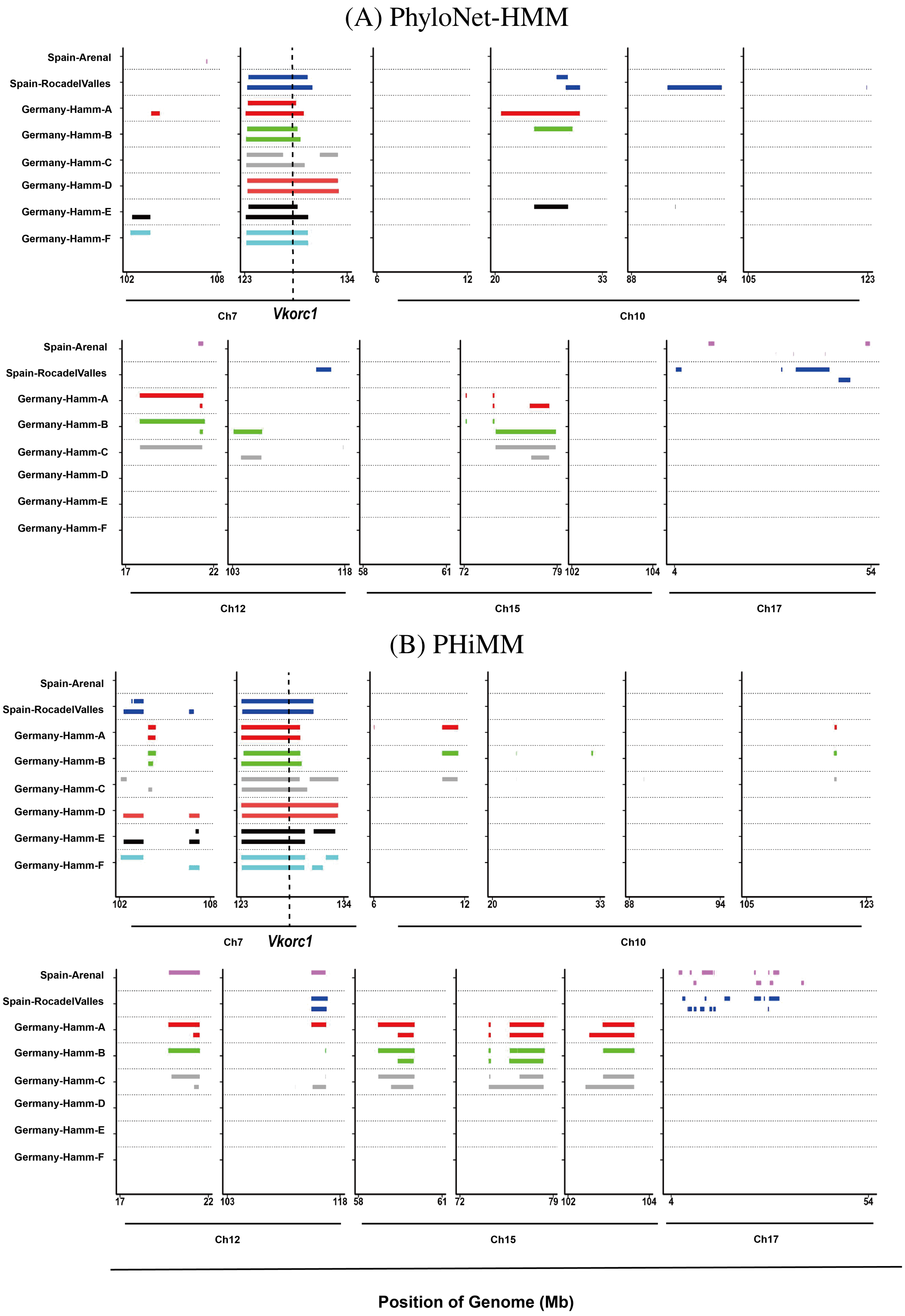


Figure 4.3 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on mouse data.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].

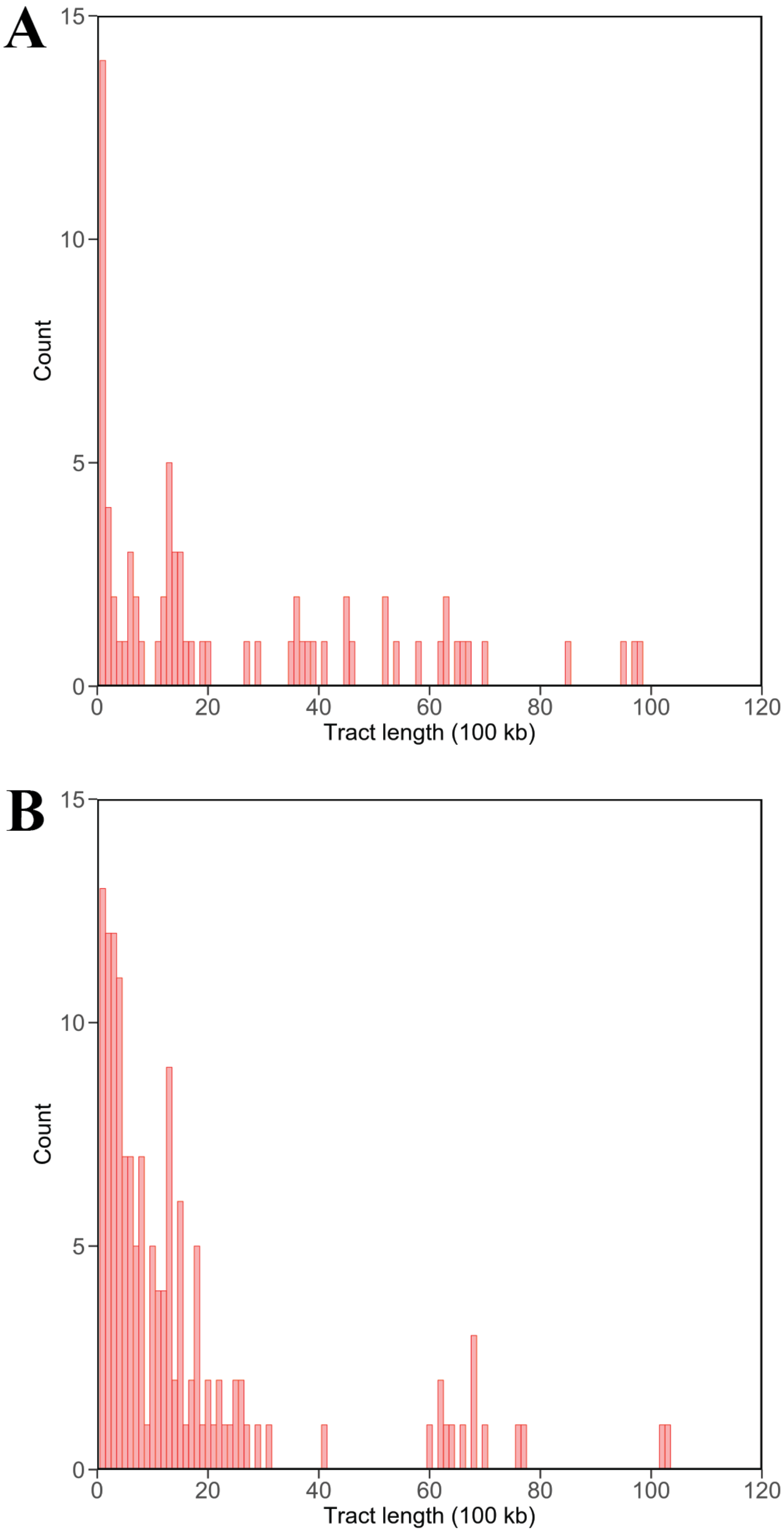


Figure 4.4 **The comparison of histograms of introgressed tract lengths between (A) PhyloNet-HMM and (B) PHiMM on mouse data.** Introgressed tract lengths are shown in 100 kilobases (Kb). Results are reported for all recipients *M. m. domesticus* samples from the region of sympatry between *M. m. domesticus* and *M. spretus*. This figure comes from Wuyun et al. [122].

the chromosome 17 region between coordinates 4 Mb and 54 Mb. Second, PHiMM infers longer and more numerous introgressed tracts compared to PhyloNet-HMM in certain regions, such as chromosome 10, 12, and 15.

Furthermore, we investigate the genome-wide histogram of introgressed tract lengths as shown in Figure 4.4. Qualitatively, PhyloNet-HMM and PHiMM return inferences with similar histogram shapes: a large peak over short tract lengths (kilobases or a few megabases of sequence length at most), followed by a long tail over longer sequence lengths. Two primary differences between the histograms of the two methods are noted. First, PHiMM's histogram has a greater total count of introgressed tracts and a larger total sequence length compared to the latter. Second, PHiMM's histogram appears to be bimodal: one part has tract lengths distributed between 0 and 3.5 Mb, and the other part consists of around a dozen much longer introgressed tracts with sequence lengths between 4 and 11 Mb. In comparison, PhyloNet-HMM's histogram has a long tail with a relatively uniform distribution of introgressed tracts between 2 and 10 Mb.

In summary, these comparisons illustrate that PHiMM and PhyloNet-HMM both accurately detect known adaptive introgression events in *M. m. domesticus*, although PHiMM appears more inclined to identify longer or subtler introgressed tracts. Future studies could delve into the biological relevance of such extended signals and refine methodological assumptions to better account for the complexities of natural populations, including varying recombination rates, incomplete lineage sorting, and the mosaic nature of introgressed genomes.

4.3.2 *Limenitis* Data Re-analysis

In addition to the mouse datasets, we further evaluated PHiMM and PhyloNet-HMM on genomic data from the butterfly genus *Limenitis*, which exhibits well-characterized mimicry traits. Due to computational constraints, PHiMM was applied to a larger five-taxon dataset, whereas PhyloNet-HMM could only be run on a smaller four-taxon subset (see the Methods section for detailed descriptions of both datasets). This setup facilitates a comparative assessment of the two approaches under varying data sizes and complexities.

As shown in Figure 4.5, the longest introgressed tract inferred by PHiMM closely corresponds

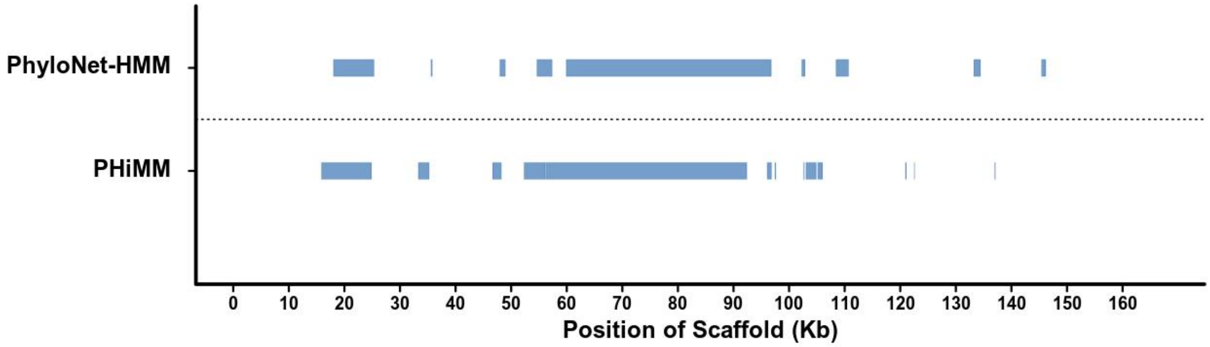


Figure 4.5 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on *Limenitis* data.** Introgressed tracts along the genome are shown in kilobases (Kb). Results are reported for the introgression from a donor *Limenitis arthemis arizonensis* sample to a recipient *Limenitis arthemis astyanax* sample. This figure comes from Wuyun et al. [122].

to the major tract identified by PhyloNet-HMM, albeit shifted slightly toward lower genomic coordinates. In principle, inferring introgression from datasets with more taxa can be more challenging owing to increased heterogeneity and potential discordance across gene regions. Consequently, the five-taxon analyses conducted by PHiMM exhibit a marginally higher level of noise relative to the smaller, four-taxon analysis by PhyloNet-HMM. This is reflected in the observation that PHiMM tends to detect a larger number of shorter introgressed tracts compared to PhyloNet-HMM.

Despite these differences, both methods consistently identify introgression within the *WntA* region, spanning approximately 27 to 101 kb, with a pronounced cluster of introgressed segments between 60 kb and 100 kb. These findings align with the results in Gallant et al. [128], who highlighted the importance of *WntA* for wing-pattern mimicry in *Limenitis*. The concordance of these inferences underscores each method's ability to capture critical genomic intervals, even when operating on datasets of different sizes and complexities. By confirming introgression signals in an ecologically and evolutionarily significant region, our analysis further validates the utility of both PHiMM and PhyloNet-HMM for dissecting genomic contributions to phenotypic adaptations.

4.4 Discussion and Conclusion

In this study, we evaluate the performance of PHiMM and another state-of-the-art method, PhyloNet-HMM, using two empirical genomic sequence datasets. Our analyses focus on the adaptive introgression in house mice and the mimicry in the butterfly genus *Limenitis*, providing a comprehensive evaluation of PHiMM's capabilities in real-world biological data.

On the mouse empirical datasets, both PHiMM and PhyloNet-HMM provide qualitatively similar inferences regarding the introgressed genomic regions, which align with the molecular hypotheses proposed by Liu et al. [7]. However, there are differences observed in the patterns of local inferences between the two methods.

In some genomic regions, such as the *Vkorc1*-containing region on chromosome 7, PHiMM infers longer and more numerous introgressed tracts compared to PhyloNet-HMM. Additionally, the distribution of introgressed tract lengths exhibited dissimilarities between the two methods. PHiMM detects more introgressed tracts and displays a clearer "separation" between two classes of tracts: megabases-long tracts, which are few in number, and shorter tracts, which are more numerous. The presence of the former "long" class of tracts is consistent with the hypothesis of adaptive introgression, where neutral recurrent backcrossing tends to shorten introgressed tracts over time, while positive selection and genetic hitchhiking have an opposing effect [7]. On the other hand, the latter "short" class of tracts aligns with Liu et al. [7]'s hypothesis regarding more ancient bouts of adaptive interspecific introgression; sympatry between *M. musculus* and *M. spretus* is understood to have predated the recent introduction of pesticides [7, 133].

On the *Limenitis* empirical datasets, both PhyloNet-HMM and PHiMM identify similar introgression tracts that exhibit significant overlap across much of the genomic region containing WntA. These results are in line with the experimental findings presented in the work of Gallant et al. [128].

The observed differences in our empirical study can be attributed to two factors: PHiMM's competitive statistical power and type I error control relative to PhyloNet-HMM, as well as the denser allele sampling made possible by PHiMM's improved scalability compared to PhyloNet-HMM.

In conclusion, our study demonstrates that the PHiMM method is a powerful tool for introgression detection, capable of handling large datasets and providing accurate inferences. By applying PHiMM to diverse empirical datasets, we have illustrated its robustness and utility in real-world biological research. These findings contribute to our understanding of the evolutionary processes underlying genetic diversity and adaptive introgression, and pave the way for future applications of the PHiMM approach in various genomic studies.

CHAPTER 5

BOOSTING PHIMM-BASED PHYLOGENETIC HMM INFERENCE AND LEARNING BASED ON RANDOM WALK RESAMPLING

5.1 Introduction

Non-parametric resampling techniques enable researchers to leverage empirical data to construct distributions for obtaining critical values, calculating p-values, or constructing confidence intervals. For the task of introgression mapping, non-parametric resampling methods can be employed to generate resampled replicates of a genome alignment. Inference/analysis can be thus performed and compared across replicates.

There are two primary classes of resampling methods: non-parametric and parametric. Among non-parametric methods, the bootstrap method is the most widely-used [134, 135]. Given an input set of observations, the bootstrap method resamples observations uniformly at random with replacements. Re-estimation is then conducted on these resampled replicates, and repeatability is assessed by comparing the re-estimated results. Other non-parametric methods include the jackknife and weighted bootstrap. In contrast, parametric methods resample directly from an explicit statistical model, often requiring the assumption of a hypothesis model due to the unavailability of the original model that generated the original inputs. Non-parametric methods are preferred in many cases as they do not necessitate assumptions where observations were generated under a specific model.

Despite their popularity, non-parametric resampling methods, such as the bootstrap, have significant limitations, particularly the assumption that input observations are independent and identically distributed (i.i.d.). This assumption does not hold in cases where inputs consist of sequences of observations, which is common in genomics and computational biology.

The SEquential RESampling (“SERES”) framework [136] consists of non-parametric or semi-parametric sequential resampling techniques that generalize the standard bootstrap method for non-parametric resampling [134] and the Heads-or-Tails method [137]. A critical feature of SERES is its “neighbor preservation property”, which ensures that neighboring bases within the original sequences are preserved during resampling. This generalized framework employs a random walk along a multiple sequence alignment (MSA) that is composed of either aligned or unaligned biomolecular sequences. In SERES, the random walk is conducted using the following procedure: A starting point and direction for the random walk are chosen uniformly at random across all sites. As the random walk proceeds, reversals occur with certainty at the start or end of the MSA; reversals can also occur during each step with probability γ. The random walk continues until the number of sampled characters equals the fixed MSA length. For each resampled replicate, re-estimation is performed, and repeatability is then measured by quantifying disagreement among re-estimations.

Initial studies of SERES focused on unaligned sequence inputs [136], rather than aligned inputs. Briefly, the SERES algorithm for unaligned sequence inputs also takes the form of a random walk, with one main difference: resampling "reads" along unaligned sequences occurs in an asynchronous fashion, and a set of anchors serves as synchronization "barriers" in the same way as they do in parallel computing. The SERES algorithm was applied to perform confidence interval placement for a classical problem in computational biology and bioinformatics — multiple sequence alignment (MSA) estimation. Results from synthetic and empirical data demonstrated that SERES random walks within a resampling/re-estimation pipeline yielded comparable or superior type I and type II error rates compared to state-of-the-art methods [136].

In subsequent research, we applied the SERES resampling algorithm on the aligned sequences for another classical problem in computational biology and bioinformatics: recombination-aware local genealogical inference [138]. We mainly focused on an HMM-based local genealogical inference method, recHMM [139], for the following reasons. RecHMM identifies local genealogy by applying heuristic searches on the space of all possible partitions, and is powerful in detecting local genealogy changes, especially when the regions involved in recombination are long and the dataset size is large. Thus, RecHMM reveals the most likely recombination breakpoint locations with high accuracy and fewer requirements on parameter settings such as window schemes, making it an ideal method to focus on. Another reason we focused on the recHMM method is that recHMM takes aligned sequences as input to annotate mosaic genome structures, making it possible to combine the SERES resampling approach. Simulation experiments indicated that combining SERES with recHMM significantly improved recombination detection and local genealogical inference. The SERES resampling also holds potential for applications in ancestral recombination inference problems, such as recombination rate estimation [140] and recombination hotspot or coldspot detection [141, 142].

In this study, we propose another application of SERES random walks on aligned sequences, and utilize SERES random walks as a means to “boost” HMM inference/learning performance. Similar to other non-parametric resampling methods, we demonstrate that SERES will serve as a data perturbation technique in addition to its use in confidence interval placement, as considered by earlier work [136, 138]. In the simulation study, we apply the SERES resampling algorithm on the aligned sequences for a classical problem in computational biology and bioinformatics — introgression mapping. We mainly focus on the PHiMM algorithm for introgression mapping. In detail, we compare the performance and scalability of the PHiMM method with and without the SERES resampling approach. Benchmark results suggest that the SERES resampling and re-estimation can significantly improve PHiMM’s inference accuracy.

5.2 Methods

5.2.1 Standalone PHiMM pipeline

PHiMM is an HMM-based introgression mapping method that combines inference and learning under a combined model of genetic drift, substitutions, recombination, and gene flow with a coalescent-based approximation technique. Benchmark analyses indicate that PHiMM offers better computational runtime and main memory usage by multiple orders of magnitude, while returning comparable inference accuracy.

We first run the PHiMM alone to establish a baseline performance on the introgression mapping in the simulation data. PHiMM analyses are run using default settings, e.g., the number of iterations for model parameter learning is 300, and the number of runs is set to 10. Users also need to assign a gene tree truncation size k_n for the HMM model of PHiMM. In our simulation study, we run PHiMM with the default setting, $k_n = 15$.

Algorithm 5.1 SERES+PHiMM

```
procedure SERES-BASED PHiMM($N, A, R$)
    $(\{p_t\}_{1 \le t \le L}, \theta) \leftarrow PHiMM(N, A)$    ▷ $N$: Phylogenetic network
        ▷ $A$: Input multiple sequence alignment with $K$ aligned sequences and $L$ columns
        ▷ $p_t$: Introgression probability for each aligned site $t$ $(1 \le t \le L)$
        ▷ $\theta$: Model parameters
    int $i \leftarrow 1$
    while $i \le R$ do    ▷ $R$: Number of replicates
        $startsite^{(i)}, direction^{(i)} \leftarrow SelectStartSite(A)$
            ▷ $startsite^{(i)}, direction^{(i)}$: Starting point and direction for SERES random walk
        $A^{(i)}, mapping^{(i)} \leftarrow \emptyset, \emptyset$    ▷ $A^{(i)}, mapping^{(i)}$: Resampled alignment and mapping
        while Length of $A^{(i)} \le L$ do
            $A^{(i)}, mapping^{(i)} \leftarrow A^{(i)}, mapping^{(i)} + RandomWalk(A, startsite^{(i)}, direction^{(i)}, \gamma)$
                ▷ $\gamma$: Reversal probability
        $\{p_t^{(i)}\}_{1 \le t \le L} \leftarrow PHiMM(N, A^{(i)}, \theta)$    ▷ Run PHiMM with fixed model parameters
        $i \leftarrow i + 1$
    $\{\bar{p}_t\} \leftarrow AverageProbability(\{p_t\}, \{p_t^{(1)}\}, mapping^{(1)} ..., \{p_t^{(R)}\}, mapping^{(R)})$
    return $\{\bar{p}_t\}_{1 \le t \le L}$
```

Consistent with the study by Wuyun et al. [122], the inputs of the PHiMM include: the aligned DNA sequences A, with K aligned sequences and L columns; and a phylogenetic network N on K taxa. The output is a sequence of modified posterior decoding probabilities for the columns of the input alignment A.

5.2.2 The SERES+PHiMM pipeline

The PHiMM framework can be augmented with SERES-based resampling and re-estimation to enhance inference and learning (Algorithm 5.1).

First, SERES random walks are conducted to perform resampling on the input alignment A. Detailed pseudocode for this procedure is shown in Algorithm 5.2 (reproduced from Wang et al. [136, 138]), and Figure 5.1 provides an illustrated example of a SERES random walk on an input MSA. The SERES resampling procedure in our simulation study utilizes a default reversal probability $\gamma = 0.0001$. We also conduct additional experiments with alternative reversal probability $\gamma \in \{0, 0.0001, 0.001, 0.005, 0.01, 0.1\}$. We use the SERES random walk to generate 10 resampled replicates for each dataset in our study.

Then, the PHiMM algorithm is run with default settings to perform optimization-based learning

Algorithm 5.2 SERES

```
procedure SERES Walk on Aligned Sequences($A$, $\gamma$, numReplicates)
                                        ▷ $A$: Input multiple sequence alignment
                                        ▷ $\gamma$: Walk reversal probability
                                        ▷ numReplicates: Number of SERES replicates
    replicates = < >
    for $i$ from 1 to numReplicates do
        direction = (rand() > 0.5) ? +1 : $-1$
                                        ▷ Uniformly at random choose a direction (right vs. left)
                                        ▷ rand() returns floating point number sampled uniformly at random from [0, 1)
        $i$ = $\lfloor$ length($A$) $*$ rand() $\rfloor$ + 1      ▷ Uniformly at random draw from [1, length($A$)]

        replicate = < >
        while length(replicate) < length($A$) do
            replicate .= $A_i$          ▷ Read $A_i$, which is the $i$-th character in alignment $A$
                                        ▷ Alignment characters $A_i$ are one-indexed
            $i$ += direction
            if ($i \leq 0$) or ($i >$ length($A$)) or (rand() < $\gamma$) then      ▷ Reflection of random walk
                direction $*= -1$
                if ($i \leq 0$) or ($i >$ length($A$)) then
                    $i$ += direction $*$ 2      ▷ Always reflect at start/end of alignment $A$
        replicates .= replicate
    return replicates
```

on the original input alignment A. The optimized model parameters are then used to perform fixed-parameter-value inference on resampled SERES replicates.

Finally, re-estimated posterior probability distributions are averaged across SERES replicate analyses to obtain a final inferred distribution. Consistent with the study of Wuyun et al. [122], the output of each SERES replicate is a sequence of modified posterior decoding probabilities for the columns of the input alignment A. For each site a_t ($1 \leq t \leq L$) in the alignment A, the posterior probability distributions are aggregated across all the replicates in which the site appeared. The aggregated distribution is then averaged to obtain a valid probability distribution.

5.3 Materials

5.3.1 Simulation Data

The simulation study is used to evaluate the performance and applicability of the standalone PHiMM method and SERES+PHiMM method, since we can track the true history of evolutionary

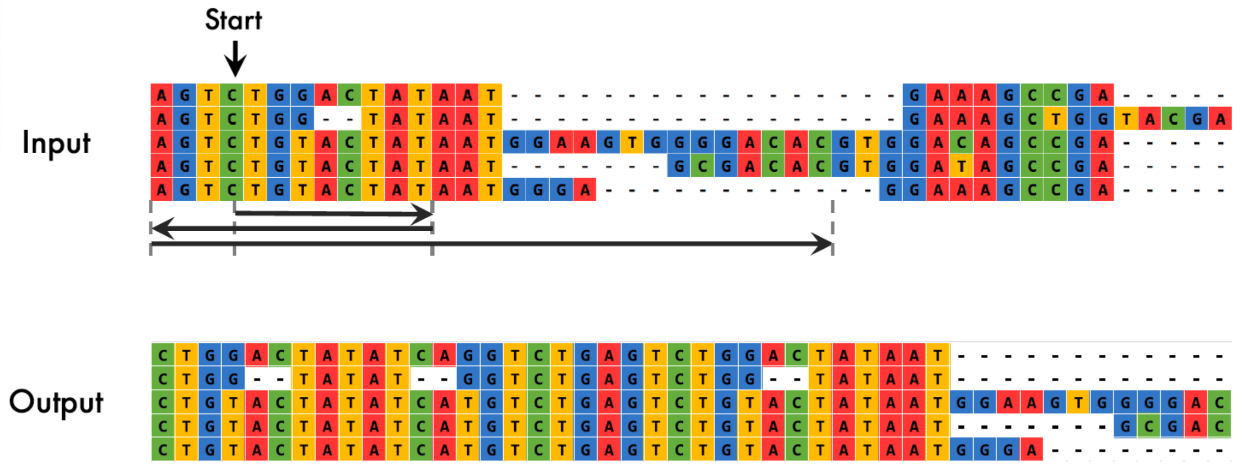

Start
Input
AGTCTGGACTATAAT-----------------GAAAGCCGA-----
AGTCTGG--TATAAT-----------------GAAAGCTGGTACGA
AGTCTGTACTATAATGGAAGTGGGGACACGTGGACAGCCGA-----
AGTCTGTACTATAAT-------GCGACACGTGGATAGCCGA-----
AGTCTGTACTATAATGGGA------------GGAAAGCCGA-----
Output
CTGGACTATATCAGGTCTGAGTCTGGACTATAAT------------
CTGG--TATAT--GGTCTGAGTCTGG--TATAAT------------
CTGTACTATATCATGTCTGAGTCTGTACTATAATGGAAGTGGGGAC
CTGTACTATATCATGTCTGAGTCTGTACTATAAT-------GCGAC
CTGTACTATATCATGTCTGAGTCTGTACTATAATGGGA--------

Figure 5.1 **Illustrated example of a SERES random walk on an input multiple sequence alignment (MSA).** The input of the SERES resampling algorithm is an MSA. The SERES resampling algorithm takes the form of a random walk. First, a start site and initial walk direction are chosen uniformly at random. In this example, the fourth site from the left is chosen as the start site, and the initial walk direction is rightward. The random walk proceeds, where an MSA site is sampled during each step of the walk. Walk reversals occur with certainty at the start and end of the input MSA and with probability γ at any other point of the walk. The walk concludes when the sampled replicate meets a sequence length criterion — namely, when the number of sites in the sampled replicate and input MSA are equal. In this example, a first reversal occurs in the interior of the input MSA and a second reversal occurs at the left boundary of the input MSA. The output consists of the sampled replicates. This figure comes from Wang et al. [138].

events. The simulation data are constructed through various tools, such as r8s [28], msmove [29], and seq-gen [32].

(1) Generation of model trees using r8s

A random rooted model tree can be generated using r8s [28] tool, which utilizes the birth-death model for the tree generation.

```
#nexus
begin r8s;
simulate diversemodel=bdback seed=<integer random seed> charevol=yes
ntaxa=<integer greater than 3> infinite=yes nreps=1;
end;
```

where "diversemodel" means the model used for tree generation, which is generally set to the birth-death model denoted by "bdback". "seed" specifies a random seed for the tree generation. “ntaxa” represents the number of taxa, which is set from 5 to 10 for comparison. “nreps” indicates the number of generated repeats. “charevol=yes” indicates the model tree is output with branch lengths. “infinite=yes” represents that branch lengths are set to the expected values based on rate and time.

The height of the resulting tree is scaled to h by multiplying the length of each edge in the model tree by h. Here, we set h to 5.0 coalescent units. Furthermore, an outgroup is added to the generated tree at 50.0 coalescent time.

(2) Generation of model networks

A model network can be generated by the steps listed in Hejase et al. [11]. With the model tree obtained from the above step, we then add r reticulations ($r \in [1, 2]$) by iterating the following steps: a time t_M between 0 and the tree height is selected uniformly at random, two tree edges for which corresponding ancestral populations exist during a time interval $[t_A, t_B]$ such that $t_M \in [t_A, t_B]$ are randomly selected, and a reticulation at time t_M is added to connect the pair of tree edges. Similar to Leaché et al. [15], the model network can be further classified based upon whether gene flow is “deep” or “non-deep”, which is defined by the topological placement of reticulations, i.e., non-deep reticulations are placed between two leaf edges, while deep reticulations include all other reticulations.

(3) Generation of local genealogies using msmove

Given a model species network, 100 local genealogies can be simulated using the msmove [29] for independent and identically distributed (i.i.d.) loci following a species network under a multi-species network coalescent with recombination (MSNCwR) model. msmove is a modified version of the Monte Carlo simulator ms [30] allowing the tracking of migration events, while ms does not provide this annotation. Recombination is modeled using Hudson’s finite-sites recombination model [30].

```
msmove <number of samples> <number of repeats> -T -r <crossover rate>
<number of sites> -I <number of populations> <n_1 n_2 ...  n_k>
-ej <t_1> i_1 j_1 -ej <t_2> i_2 j_2 ...  -ej <t_k> i_k j_k
```

```
-ev <t_m> i j <probability x>
```

where -T parameter indicates the gene trees representing the history of the sampled taxa are output. The -I parameter is followed by k that represents the number of populations. The list of integers (n_1 n_2 ... n_k) includes the number of taxa sampled in each population. In this study, one allele is sampled from each taxon. The -r parameter is used to set recombination by crossover rate and the number of sites between which recombination can occur, where the crossover rate or recombination rate ρ is set to 0.1, and the number of sites between which recombination can occur is set to 900. The -ej parameter specifies moving all lineages in population i to population j at time t. The -ev parameter is special for msmove, which sets migration at time t_m from population i to population j with the migration probability $x = 0.1$ in this study.

Then, the gene trees will be deviated away from ultrametricity by the following steps [143]. First, a deviation factor, c, that quantifies the deviation level is determined. Then, for each edge in the model tree, its branch length is multiplied by e^x, where x is uniformly and randomly chosen from the interval $[-\lg(c), \lg(c)]$. Similar to Liu et al. [144], c is set to 2.0 here.

(4) Simulation of sequences using seq-gen

The DNA sequences can be generated using seq-gen [32] under the general time-reversible (GTR) substitution model [67].

```
seq-gen -mGTR -f <base frequencies> -r <general reversible
rate matrix> -a <shape of Γ distribution>
-l <sequence length> -p <number of partitions>
< genetreefile > seqfile
```

where -m parameter specifies the general time-reversible (GTR) substitution model denoted by “GTR”. The -s parameter sets the mutation rate θ that scales the branch lengths to make them equal the expected number of substitutions per site for each branch. The mutation rate θ is set to 1.The -f parameter specifies the frequencies of the four nucleotides A, C, G, and

T. The -r parameter sets a relative rate of substitutions between nucleotides in a GTR model. The -a parameter specifies the shape of the Γ distribution. The -l parameter specifies the sequence length. Here, the simulated sequence length of each gene tree is set to 900 bp. The -p parameter sets the number of partitions. The genetreefile is the input file providing the gene trees. The seqfile is the output file with simulated sequences under the given gene trees.

The GTR substitution model parameter values were estimated based on empirical analyses of the mouse genomic sequence dataset from Liu et al. [7] study. We used all M. m. domesticus and M. spretus samples listed in "Table S1" of Liu et al. [7] study and concatenated all chromosomes to get the sequence data. RAxML was used to perform concatenated phylogenetic MLE under the GTR model. The estimated parameters used in seq-gen simulations include base frequencies: $\pi_A = 0.216$, $\pi_C = 0.284$, $\pi_G = 0.285$, $\pi_T = 0.215$; GTR matrix: 1.002820, 5.486349, 1.095939, 0.539209, 5.535556, and 1.000000 for $A \leftrightarrow C$, $A \leftrightarrow G$, $A \leftrightarrow T$, $C \leftrightarrow G$, $C \leftrightarrow T$, and $G \leftrightarrow T$, respectively; shape parameter α of the Γ distribution: 0.543225.

Then, coalescent times are converted into branch lengths using "Equation (3.1)" in Hein et al. [145].

For each model condition, the simulation procedure is repeated to obtain 30 replicate datasets. Model condition parameters and summary statistics for simulated datasets are shown in Table 5.1.

5.3.2 Performance Assessments

To evaluate and compare the performance of different approaches on the simulation dataset where the true history of evolutionary events can be tracked, we use two types of area under the curve (AUC): the area under the receiver-operating characteristic (ROC) curve, and the area under the Precision-Recall curve, referred to as simply ROC-AUC and PR-AUC, respectively. ROC-AUC is plotted by the true positive rate ($\frac{TP}{TP+FN}$) as a function of the false positive rate ($\frac{FP}{FP+TN}$) at various threshold settings, while PR-AUC similarly plots the precision ($\frac{TP}{TP+FP}$) against the recall ($\frac{TP}{TP+FN}$) at different thresholds, where TP, FP, TN, and FN represent the numbers of true positives, false

Table 5.1 **Statistics for the simulation dataset.** The reticulation scenarios include one non-deep reticulation, two non-deep reticulations, and one deep reticulation. “Ntaxa” means the number of taxa. “Network Height” gives the height of the model network. “Total Sites” indicates the number of total sites in the simulation genomes. The p-distance of a pair of aligned sequences is calculated by dividing the number of sites where the two sequences had different nucleotides by the number of sites in which both sequences had nucleotides. “Average/Maximum p-dist” is the average/maximum p-distance of all pairs of aligned sequences in the simulation dataset. “Introgression (%)” shows the percent of introgressed sites over the genome. “Number of Gene Trees” indicates the number of true gene trees. “Average Branch Length” is the average branch length of the model network. The average and standard error (SE) are shown based on 30 replicates.

Reticulation	**Ntaxa**	**Total**	**Network**	**Average p-dist (%)**		**Maximum p-dist (%)**		**Introgression (%)**		**Number of Gene Trees**		**Average Branch Length**	
Scenario		**Sites**	**Height**	**Average**	**SE**	**Average**	**SE**	**Average**	**SE**	**Average**	**SE**	**Average**	**SE**
one non-deep	5	90000	5	0.631	0.002	0.704	0.000	0.101	0.006	1196.800	14.416	2.014	0.049
	6	90000	5	0.626	0.002	0.704	0.000	0.100	0.008	1227.900	12.191	1.834	0.045
	7	90000	5	0.615	0.002	0.704	0.000	0.113	0.008	1256.800	16.283	1.532	0.047
	8	90000	5	0.612	0.002	0.704	0.000	0.120	0.010	1260.933	18.156	1.505	0.045
	9	90000	5	0.610	0.002	0.704	0.000	0.141	0.015	1270.967	10.680	1.441	0.047
	10	90000	5	0.608	0.002	0.704	0.000	0.126	0.012	1351.367	22.363	1.383	0.047
one deep	5	90000	5	0.632	0.002	0.704	0.000	0.208	0.022	1231.433	16.601	2.006	0.065
	6	90000	5	0.624	0.001	0.704	0.000	0.129	0.011	1219.767	16.295	1.757	0.044
	7	90000	5	0.617	0.002	0.704	0.000	0.173	0.017	1250.933	12.212	1.607	0.044
	8	90000	5	0.614	0.002	0.704	0.000	0.147	0.013	1304.167	12.799	1.510	0.049
	9	90000	5	0.609	0.002	0.705	0.000	0.177	0.019	1337.233	15.688	1.400	0.041
	10	90000	5	0.606	0.003	0.705	0.000	0.164	0.017	1289.033	19.507	1.350	0.047
two non-deep	5	90000	5	0.634	0.002	0.704	0.000	0.184	0.014	1208.633	13.822	2.082	0.055
	6	90000	5	0.626	0.002	0.704	0.000	0.184	0.014	1237.133	13.853	1.870	0.055
	7	90000	5	0.620	0.001	0.704	0.000	0.182	0.011	1258.067	15.582	1.658	0.036
	8	90000	5	0.612	0.002	0.704	0.000	0.182	0.015	1255.467	19.117	1.463	0.049
	9	90000	5	0.609	0.002	0.704	0.000	0.177	0.009	1290.900	18.244	1.436	0.041
	10	90000	5	0.603	0.002	0.704	0.000	0.152	0.009	1280.533	15.278	1.322	0.041

positives, true negatives, and false negatives, respectively. The ROC-AUC and PR-AUC measures represent the tradeoff between type I errors and type II errors under different threshold values. The measures are calculated on all loci in the simulation datasets, where the true migration events are annotated by msmove with an asterisk. After concatenating all query loci in one simulation replicate, the ROC-AUC and PR-AUC of a method are assessed based on the probability of a particular site involving an introgressive origin.

Additionally, we report the memory usage and runtime in order to comprehensively evaluate the scalability of our framework.

5.4 Results

In this study, we compare the performance of SERES-based PHiMM and PHiMM on datasets with different numbers of taxa (5 to 10) and different model conditions (one non-deep reticulation, one deep reticulation, and two non-deep reticulations), since the memory and runtime would rise as the number of taxa/reticulations and the complexity of reticulations increases.

Figure 5.2 shows the comparisons of the area under the receiver-operating characteristic (ROC) curve (ROC-AUC), and the area under the Precision-Recall curve (PR-AUC), runtime, and memory usage between PHiMM and SERES-based PHiMM on the 5- to 10-taxon model conditions with one non-deep reticulation, one deep reticulation, and two non-deep reticulations.

For predictable performance, both methods exhibit relatively high ROC-AUC and PR-AUC values for model conditions with one or two non-deep reticulations, suggesting strong prediction performance. The SERES+PHiMM method consistently demonstrates superior or comparable ROC-AUC and PR-AUC values across all taxa when compared to the PHiMM alone. Particularly for model conditions with one non-deep reticulation, the improvement brought by SERES-based PHiMM method is substantially larger than other model conditions with two non-deep reticulations or one deep reticulation, where the performance gap between the two methods narrows. The model conditions with one deep reticulation shows a different trend where the ROC-AUC and PR-AUC values for both methods tend to be lower compared to the "non-deep" model conditions, suggesting that the deeper reticulations might face challenges in maintaining high classification accuracy. Nevertheless, the SERES+PHiMM method still generally outperforms the PHiMM method, particularly for smaller numbers of taxa (5 to 8 taxa). Overall, the integration of SERES with PHiMM tends to improve or sustain predictable performance across varying numbers of taxa and reticulation scenarios, with the "non-deep" model conditions yielding more robust results compared to the "deep" model conditions.

The runtime for both methods increases as the number of taxa increases under model conditions with one non-deep reticulation. PHiMM demonstrates significantly lower runtimes compared to SERES+PHiMM across all taxa numbers, because SERES+PHiMM runs on multiple SERES resampling replicates to aggregate the results. For instance, with 10 taxa, PHiMM takes approximately 50 hours, whereas SERES+PHiMM requires about 255 hours. The discrepancy becomes more pronounced with an increasing number of taxa, indicating that SERES+PHiMM scales less efficiently in these model conditions. In model conditions with one deep reticulation, a similar trend is observed. PHiMM consistently outperforms SERES+PHiMM in terms of runtime. For 10 taxa, PHiMM's runtime is around 60 hours compared to over 320 hours for SERES+PHiMM. Again, the gap widens as the number of taxa increases, highlighting the inefficiency of SERES+PHiMM in handling larger datasets under the deep model conditions. The two non-deep model conditions also show PHiMM with substantially lower runtimes compared to SERES+PHiMM. The runtimes for PHiMM remain relatively low and stable, even as the number of taxa increases. In contrast, SERES+PHiMM's runtime escalates dramatically with the number of taxa. For example, with 10 taxa, PHiMM's runtime is about 55 hours, while SERES+PHiMM's runtime is close to 350 hours. Overall, the results indicate that PHiMM is significantly more efficient in runtime than SERES+PHiMM across all tested reticulation scenarios and numbers of taxa. The efficiency gap between the two methods becomes more substantial as the number of taxa increases, particularly under model conditions with one deep reticulation or two non-deep reticulations where both methods tend to use more runtimes. These findings suggest that PHiMM is a more scalable and time-efficient method for large datasets and complex model conditions.

In terms of memory usage, both methods exhibit low memory usage for 5 taxa, with memory consumption increasing as the number of taxa rises. In model conditions with one non-deep reticulation, PHiMM and SERES+PHiMM show comparable memory usage, particularly at higher numbers of taxa where both methods utilize approximately 60 GB at 10 taxa. In model conditions with one deep reticulation, memory consumption increases for both methods as the number of taxa grows, with PHiMM showing marginally higher memory usage than SERES+PHiMM for 6 to 10 taxa. In the two non-deep model conditions, both methods show a significant increase in memory consumption as the number of taxa rises. At 10 taxa, PHiMM reaches around 55 GB, while SERES+PHiMM is slightly higher, around 60 GB, indicating similar memory usage overall. Overall, PHiMM and SERES+PHiMM demonstrate similar memory consumption across all model conditions, with only slight differences that become more pronounced at higher numbers of taxa. This suggests that both methods have comparable memory efficiency, making them both viable options for large-scale phylogenetic studies.

Table 5.2 presents a detailed comparison of the area under the receiver-operating characteristic curve (ROC-AUC) values for two phylogenetic inference methods, PHiMM and SERES+PHiMM, under varying reversal probabilities γ for model conditions involving one non-deep reticulation, one deep reticulation, and two non-deep reticulations. The analysis spans 5- to 10-taxon model conditions. For model conditions with one non-deep reticulation, PHiMM consistently shows high ROC-AUC values across all γ values, with slight variations as the number of taxa increases. SERES+PHiMM generally demonstrates higher ROC-AUC values compared to PHiMM, particularly at lower γ values. When two non-deep reticulations are considered, both methods demonstrate a decrease in performance as the γ value increases. However, the performance of SERES+PHiMM remains superior, especially at lower γ values, showing consistent ROC-AUC improvements across most γ values. The trend continues for model conditions with one deep reticulation, where both methods maintain relatively high ROC-AUC values. SERES+PHiMM outperforms PHiMM across most γ values and different numbers of taxa, especially noticeable at γ values of 0, 0.0001, and 0.001. The consistency in performance across different reticulation scenarios and γ values highlights SERES+PHiMM's superior ability to manage reversal probabilities in phylogenetic model conditions. The Table 5.2 demonstrates the enhanced effectiveness of SERES+PHiMM in phylogenetic inference tasks across various taxa counts, reticulation complexities, and reversal probabilities, highlighting its robustness and reliability in evolutionary studies.

Table 5.3 presents the comparison of the area under the precision-recall curve (PR-AUC) between the PHiMM and SERES+PHiMM models across various reversal probabilities γ for each combination of reticulation scenario and number of taxa. For model conditions with one non-deep reticulation, SERES+PHiMM generally exhibits higher PR-AUC values compared to PHiMM, especially at lower γ levels. The improvement diminishes as γ increases. In scenarios with two non-deep reticulations, a similar trend is observed, but both methods display a notable decline in PR-AUC values. For model conditions with one deep reticulation, SERES+PHiMM consistently shows superior performance compared to PHiMM across most γ levels, but the performance gap narrows at higher γ levels.

Table 5.4 provides a detailed comparison of the runtime (measured in hours) between the

PHiMM and SERES+PHiMM methods under various reversal probabilities γ in 5- to 10-taxon model conditions. For scenarios with one non-deep reticulation, PHiMM generally exhibits shorter runtimes compared to SERES+PHiMM, with the difference becoming more pronounced as γ decreases. In scenarios involving two non-deep reticulations, SERES+PHiMM consistently shows significantly longer runtimes than PHiMM across all γ levels, especially for lower γ values, with the gap widening as the number of taxa increases. Similarly, in the one deep reticulation scenario, both methods tend to have increased runtimes due to the complexity of model conditions. SERES+PHiMM's runtime is substantially longer compared to PHiMM, particularly at lower γ levels. These results suggest that SERES+PHiMM may be computationally more intensive, especially in complex reticulation scenarios and with lower reversal probabilities, while PHiMM is more time-efficient without a significant loss in accuracy, particularly in scenarios with more taxa.

Table 5.5 shows a comparative analysis of memory usage (measured in gigabytes or GB) between the PHiMM and SERES+PHiMM methods under different reversal probabilities γ for 5- to 10-taxon model conditions. For all three scenarios, PHiMM exhibits consistently similar memory usage compared to SERES+PHiMM across all γ levels. Notably, for the SERES+PHiMM method, there is no significant difference of memory usage between different γ levels. These findings suggest that SERES+PHiMM requires similar memory resources than PHiMM across different reticulation scenarios and reversal probabilities, indicating that SERES+PHiMM is a more memory-efficient alternative while maintaining better performance in accuracy, particularly for lower γ values.

5.5 Discussion and Conclusion

This study introduces the application of SERES random walks on aligned sequences and shows SERES as a data perturbation technique to improve introgression inference and learning. The simulation experiments show that the combination of the SERES resampling approach with the PHiMM returns great improvement in the introgression inference compared to standalone PHiMM on different model conditions in our study, which suggests that the SERES resampling and re-estimation has the potential to "boost" PHiMM's inference accuracy.

To evaluate to what extent the SERES resampling approach boosts the introgression inference

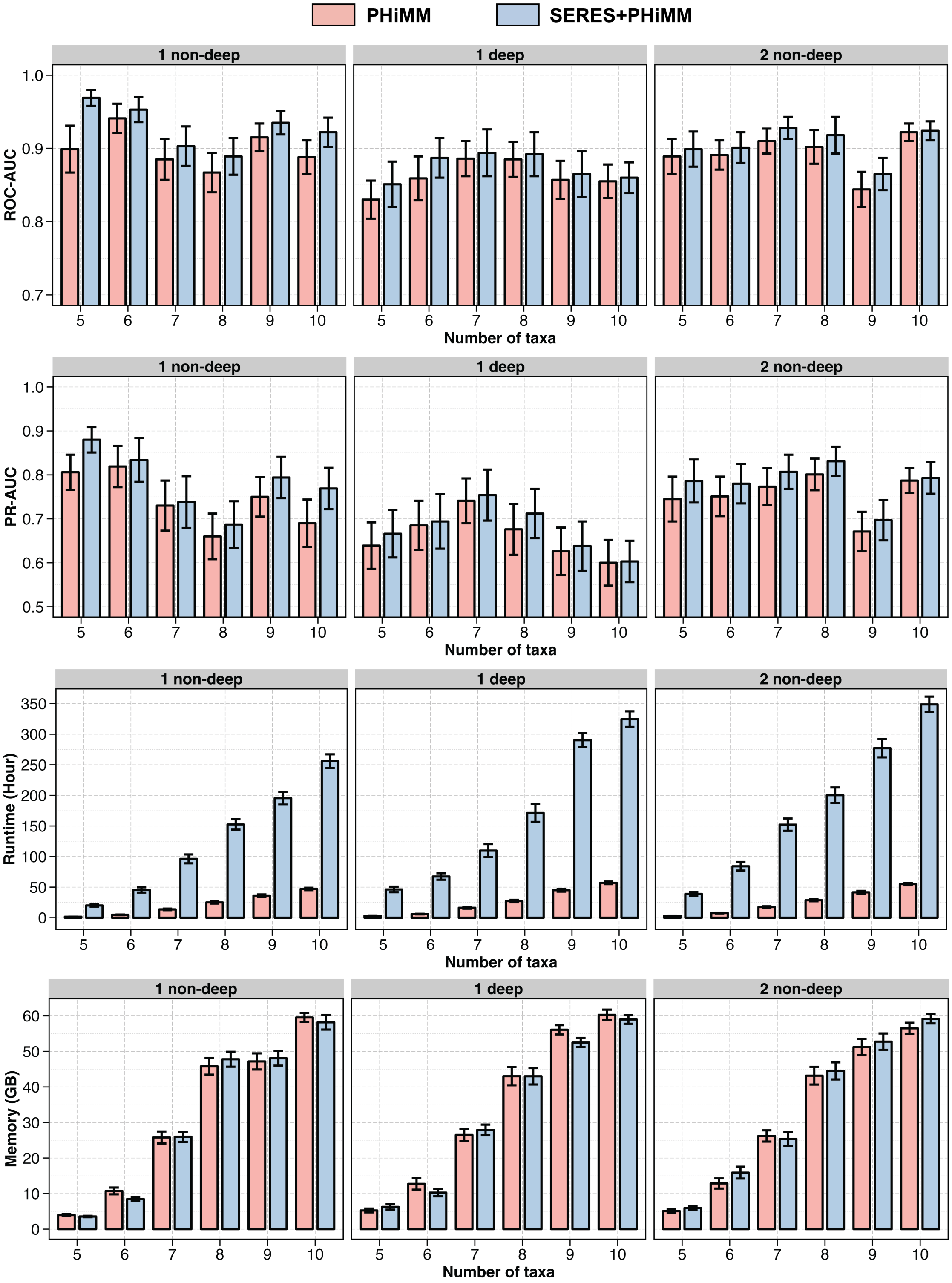


Figure 5.2 **The performance comparison between PHiMM and SERES+PHiMM on the 5- to 10-taxon model conditions with one non-deep reticulation, one deep reticulation, and two non-deep reticulations.** The measures include the area under the receiver-operating characteristic curve (ROC-AUC), the area under the precision-recall curve (PR-AUC), runtime (Hour), and memory usage (GB). The SERES+PHiMM method was run with reversal probabilities $\gamma = 0.0001$. The average and standard error (SE) are calculated based on 30 replicates.

Table 5.2 **The comparison of the area under the receiver-operating characteristic curve (ROC-AUC) between PHiMM and SERES+PHiMM among different reversal probabilities γ on the 5- to 10-taxon model conditions with one non-deep reticulation, one deep reticulation, and two non-deep reticulations.** The parameter γ represents the probability of reversals in SERES resampling and is tested at several levels (0, 0.0001, 0.001, 0.005, 0.01, and 0.1). The "no γ" column indicates results for a baseline method where reversals are not considered. "Ntaxa" means the number of taxa. The average and standard error (SE) are calculated based on 30 replicates (represented as "average value ± SE value").

Reticulation Scenario	**Ntaxa**	**PHiMM**	**SERES+PHiMM**					
		no γ	$\gamma = 0$	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.005$	$\gamma = 0.01$	$\gamma = 0.1$
one non-deep	5	0.899±0.032	0.965±0.012	0.969±0.011	0.958±0.015	0.958±0.012	0.966±0.010	0.923±0.022
	6	0.941±0.020	0.950±0.019	0.953±0.017	0.953±0.018	0.916±0.030	0.944±0.018	0.920±0.025
	7	0.885±0.028	0.925±0.019	0.903±0.027	0.888±0.034	0.880±0.031	0.886±0.027	0.885±0.027
	8	0.867±0.027	0.909±0.020	0.889±0.025	0.881±0.028	0.890±0.021	0.888±0.025	0.859±0.026
	9	0.915±0.019	0.945±0.011	0.935±0.016	0.916±0.025	0.919±0.021	0.937±0.016	0.915±0.021
	10	0.888±0.023	0.918±0.020	0.922±0.020	0.909±0.021	0.909±0.022	0.905±0.023	0.901±0.022
two non-deep	5	0.830±0.026	0.849±0.027	0.851±0.031	0.840±0.033	0.826±0.037	0.823±0.032	0.811±0.029
	6	0.859±0.030	0.882±0.032	0.887±0.027	0.868±0.029	0.871±0.027	0.877±0.027	0.865±0.026
	7	0.886±0.024	0.909±0.026	0.894±0.032	0.887±0.032	0.907±0.025	0.892±0.027	0.877±0.029
	8	0.885±0.024	0.907±0.020	0.892±0.030	0.879±0.029	0.880±0.025	0.890±0.023	0.858±0.038
	9	0.857±0.026	0.869±0.027	0.865±0.031	0.864±0.027	0.834±0.032	0.854±0.028	0.851±0.027
	10	0.855±0.023	0.875±0.019	0.860±0.021	0.851±0.025	0.861±0.019	0.859±0.019	0.852±0.021
one deep	5	0.889±0.024	0.897±0.024	0.899±0.024	0.891±0.026	0.884±0.026	0.892±0.022	0.881±0.024
	6	0.891±0.020	0.899±0.022	0.901±0.021	0.893±0.023	0.904±0.024	0.889±0.020	0.882±0.025
	7	0.910±0.017	0.924±0.015	0.928±0.015	0.926±0.015	0.924±0.016	0.921±0.015	0.904±0.017
	8	0.902±0.023	0.920±0.022	0.918±0.025	0.920±0.023	0.900±0.026	0.912±0.024	0.907±0.021
	9	0.844±0.024	0.870±0.021	0.865±0.022	0.859±0.022	0.855±0.022	0.847±0.021	0.841±0.023
	10	0.922±0.012	0.942±0.011	0.924±0.013	0.921±0.014	0.925±0.014	0.922±0.014	0.914±0.013

accuracy, we compare the performances of the standalone PHiMM method and the combined SERES+PHiMM method. Across all the simulation model conditions, the SERES+PHiMM method has a consistently better inference accuracy compared to the standalone PHiMM method. The improved performance obtained by combining PHiMM inference with SERES resampling and re-estimation is robust in different numbers of taxa and different reticulation scenarios.

We also compare the runtime and memory usage produced by these two methods. We find that both methods produce a similar memory usage for different numbers of taxa and different reticulation scenarios. The SERES+PHiMM method produces longer runtimes compared to the standalone PHiMM method.

We attribute these findings to two factors. First, the application of the SERES resampling and re-estimation appears to be conducive to the introgression inferences. The SERES resampling approach has the ability to retain the sequence dependence in the input alignment to the resampled

Table 5.3 **The comparison of the area under the precision-recall curve (PR-AUC) between PHiMM and SERES+PHiMM among different reversal probabilities γ on the 5- to 10-taxon model conditions with one non-deep reticulation, one deep reticulation, and two non-deep reticulations.** The parameter γ represents the probability of reversals in SERES resampling and is tested at several levels (0, 0.0001, 0.001, 0.005, 0.01, and 0.1). The "no γ" column indicates results for a baseline method where reversals are not considered. "Ntaxa" means the number of taxa. The average and standard error (SE) are calculated based on 30 replicates (represented as "average value ± SE value").

Reticulation	**Ntaxa**	**PHiMM**			**SERES+PHiMM**			
Scenario		**no γ**	$\gamma = 0$	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.005$	$\gamma = 0.01$	$\gamma = 0.1$
one non-deep	5	0.806±0.040	0.876±0.032	0.880±0.029	0.865±0.036	0.860±0.031	0.861±0.034	0.764±0.053
	6	0.819±0.047	0.865±0.046	0.834±0.050	0.862±0.046	0.808±0.054	0.825±0.050	0.788±0.052
	7	0.730±0.057	0.761±0.052	0.738±0.059	0.752±0.055	0.712±0.058	0.733±0.052	0.719±0.055
	8	0.660±0.052	0.711±0.051	0.687±0.053	0.678±0.061	0.690±0.051	0.682±0.053	0.612±0.056
	9	0.750±0.045	0.811±0.041	0.794±0.047	0.760±0.051	0.764±0.048	0.799±0.047	0.735±0.050
	10	0.690±0.054	0.784±0.045	0.769±0.047	0.733±0.053	0.740±0.048	0.741±0.049	0.740±0.048
two non-deep	5	0.639±0.053	0.647±0.053	0.666±0.054	0.681±0.049	0.645±0.057	0.636±0.053	0.625±0.052
	6	0.685±0.056	0.707±0.058	0.694±0.062	0.660±0.061	0.647±0.058	0.692±0.056	0.653±0.055
	7	0.741±0.051	0.781±0.052	0.754±0.058	0.731±0.059	0.777±0.050	0.737±0.056	0.720±0.053
	8	0.676±0.058	0.715±0.052	0.712±0.056	0.662±0.059	0.664±0.056	0.701±0.052	0.659±0.057
	9	0.626±0.054	0.658±0.051	0.638±0.056	0.637±0.049	0.604±0.056	0.628±0.048	0.632±0.051
	10	0.600±0.052	0.617±0.048	0.603±0.047	0.605±0.049	0.612±0.044	0.607±0.046	0.594±0.045
one deep	5	0.745±0.051	0.771±0.050	0.786±0.049	0.773±0.051	0.742±0.052	0.762±0.046	0.740±0.050
	6	0.751±0.045	0.771±0.046	0.780±0.045	0.768±0.047	0.798±0.045	0.753±0.042	0.756±0.046
	7	0.773±0.042	0.800±0.040	0.807±0.039	0.793±0.041	0.806±0.041	0.794±0.038	0.755±0.043
	8	0.801±0.036	0.834±0.031	0.831±0.033	0.838±0.036	0.801±0.035	0.827±0.033	0.801±0.034
	9	0.671±0.045	0.701±0.043	0.697±0.046	0.690±0.047	0.676±0.045	0.666±0.044	0.661±0.046
	10	0.787±0.028	0.830±0.028	0.793±0.036	0.781±0.033	0.786±0.033	0.769±0.036	0.761±0.032

Table 5.4 **The comparison of runtime (Hour) between PHiMM and SERES+PHiMM among different reversal probabilities γ on the 5- to 10-taxon model conditions with one non-deep reticulation, one deep reticulation, and two non-deep reticulations.** The parameter γ represents the probability of reversals in SERES resampling and is tested at several levels (0, 0.0001, 0.001, 0.005, 0.01, and 0.1). The "no γ" column indicates results for a baseline method where reversals are not considered. "Ntaxa" means the number of taxa. The average and standard error (SE) are calculated based on 30 replicates (represented as "average value ± SE value").

Reticulation	**Ntaxa**	**PHiMM**			**SERES+PHiMM**			
Scenario		**no γ**	$\gamma = 0$	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.005$	$\gamma = 0.01$	$\gamma = 0.1$
one non-deep	5	1.615±0.108	16.546±1.216	20.031±1.828	14.912±1.109	15.251±1.472	12.675±0.897	12.194±0.973
	6	4.715±0.454	49.991±4.406	45.421±4.291	31.086±2.952	26.044±2.399	23.157±2.099	18.694±1.883
	7	13.589±1.263	138.952±11.148	96.198±7.236	58.166±5.285	53.561±4.949	43.252±3.472	36.710±2.977
	8	25.038±1.869	234.342±14.003	152.545±8.533	99.570±5.828	70.428±4.312	66.534±3.835	53.501±3.225
	9	36.166±1.938	307.725±19.453	195.459±10.506	123.787±7.184	86.693±5.498	85.434±4.559	69.443±3.788
	10	47.028±1.958	387.345±22.982	255.875±11.187	157.106±6.896	119.800±4.739	109.674±4.363	90.982±3.310
two non-deep	5	3.213±0.341	35.666±3.726	46.201±4.475	33.429±4.173	29.445±3.340	30.117±3.287	27.399±3.204
	6	6.008±0.497	62.110±4.795	67.373±5.306	42.995±3.690	39.200±3.932	37.157±3.110	30.806±2.945
	7	16.251±1.480	159.016±13.826	109.736±10.768	91.010±8.737	78.998±7.451	70.492±6.905	65.597±6.562
	8	27.248±1.985	273.218±20.707	171.304±14.714	131.127±9.094	104.789±8.835	92.155±8.071	91.718±7.491
	9	44.984±2.276	424.090±17.552	290.139±11.489	195.802±9.559	168.118±7.621	139.561±7.637	141.756±6.827
	10	57.054±2.142	526.604±16.344	324.596±12.752	242.069±7.839	200.630±10.444	197.226±7.346	170.772±6.564
one deep	5	3.128±0.234	38.677±3.422	38.941±3.025	31.505±2.334	40.511±2.762	29.911±2.388	24.947±2.296
	6	7.639±0.560	83.086±6.566	84.079±6.956	62.756±4.682	64.538±4.975	56.337±5.096	50.446±5.096
	7	17.409±1.263	179.234±12.396	152.101±10.097	102.317±7.182	89.755±6.701	82.943±7.334	71.305±5.565
	8	28.617±1.689	277.617±17.329	200.300±12.639	131.271±9.287	125.695±9.284	109.313±8.128	102.464±8.426
	9	41.516±2.337	404.211±22.775	277.049±14.920	191.190±10.614	160.965±9.449	149.415±8.548	135.226±8.261
	10	55.080±1.916	508.594±26.932	348.707±12.763	231.784±8.474	196.986±7.220	186.991±7.114	170.549±6.458

Table 5.5 **The comparison of memory usage (GB) between PHiMM and SERES+PHiMM among different reversal probabilities γ on the 5- to 10-taxon model conditions with one non-deep reticulation, one deep reticulation, and two non-deep reticulations.** The parameter γ represents the probability of reversals in SERES resampling and is tested at several levels (0, 0.0001, 0.001, 0.005, 0.01, and 0.1). The "no γ" column indicates results for a baseline method where reversals are not considered. "Ntaxa" means the number of taxa. The average and standard error (SE) are calculated based on 30 replicates (represented as "average value ± SE value").

Reticulation Scenario	**Ntaxa**	**PHiMM**	**SERES+PHiMM**					
		no γ	γ = 0	γ = 0.0001	γ = 0.001	γ = 0.005	γ = 0.01	γ = 0.1
one non-deep	5	3.974±0.276	8.116±0.901	3.552±0.199	5.697±0.607	4.733±0.394	4.974±0.220	3.771±0.299
	6	10.749±0.953	14.667±1.367	8.464±0.603	10.828±0.672	10.873±0.829	11.137±0.820	10.043±0.790
	7	25.778±1.690	29.367±2.186	25.978±1.447	24.698±2.531	27.031±2.311	27.012±2.305	27.410±2.239
	8	45.796±2.347	48.084±2.383	47.791±2.098	42.071±2.391	45.293±2.299	41.971±2.167	46.498±2.653
	9	47.177±2.270	55.784±1.776	48.073±2.073	48.495±2.165	47.093±2.766	46.954±1.888	48.179±1.942
	10	59.551±1.269	58.761±2.496	58.189±2.046	54.531±1.372	56.675±1.390	55.667±1.521	58.262±1.212
two non-deep	5	5.249±0.523	8.481±1.018	6.254±0.762	7.913±0.843	6.739±0.698	6.147±0.742	4.994±0.506
	6	12.731±1.622	15.422±1.600	10.278±1.021	13.111±1.266	9.579±0.721	10.026±0.976	8.364±0.688
	7	26.489±1.727	26.709±2.116	27.908±1.500	24.325±1.920	22.528±1.806	24.571±1.610	23.876±1.749
	8	43.038±2.572	44.987±2.474	43.014±2.326	41.804±2.565	40.568±2.152	42.366±2.337	44.181±2.484
	9	56.088±1.308	55.065±1.511	52.504±1.271	53.388±1.730	54.329±1.430	55.747±1.586	57.242±2.377
	10	60.285±1.470	58.654±1.360	58.981±1.224	58.310±1.434	60.466±2.537	63.296±1.447	60.150±1.259
one deep	5	5.072±0.543	7.317±0.669	5.949±0.614	7.557±0.602	5.855±0.455	6.321±0.504	5.166±0.606
	6	12.845±1.433	19.203±2.126	15.900±1.665	16.421±1.690	12.690±1.370	14.750±1.922	12.010±1.584
	7	26.223±1.588	33.512±2.154	25.357±1.920	28.008±1.715	29.564±2.369	27.848±1.992	25.206±1.180
	8	43.157±2.486	43.338±2.153	44.520±2.404	43.925±2.811	48.347±2.414	44.036±2.373	44.941±2.765
	9	51.237±2.288	56.058±1.783	52.744±2.306	53.838±1.825	61.073±1.933	52.754±2.121	58.877±1.985
	10	56.510±1.548	59.617±1.799	59.161±1.292	56.849±1.327	58.594±1.345	58.694±1.494	60.472±1.506

replicates. In addition, the intra-sequence dependence among sites provides additional information on the historical evolutionary events, especially those that caused the dependence. Thus, the introgression inference greatly benefits from the SERES resampling and re-estimation process. Second, the SERES resampling algorithm reveals uncertainties in the introgression inference. Incorrect introgression inferences are less repeatable. The accuracy of introgression inference by SERES+PHiMM method is consistently better than the standalone PHiMM for all model conditions of the simulated datasets, which indicates that the SERES resampling and re-estimation process produces consistently more correct introgression inferences.

Additional experiments performed to evaluate how the choice of the reversal probability impacts the method performance indicate that the SERES+PHiMM is robust to the choice of reversal probability γ. The results are consistent with the original motivation for sequence-aware resampling and re-estimation. We note the smaller γ values mean that longer-distance sequential dependence is retained. Our results suggest that longer-distance sequential dependence is critical to the performance of resampling and re-estimation for sequence-based inference problems.

CHAPTER 6

DACS: FAST AND ACCURATE ULTRA LARGE-SCALE COESTIMATION OF PHYLOGENETIC NETWORKS AND INTROGRESSIONS USING DIVIDE-AND-CONQUER

6.1 Introduction

Uncovering introgression events among a large number of taxa is a challenging task that demands both efficient computational algorithms and accurate phylogenetic models. In prior studies [12, 122], we introduced the PHiMM framework and demonstrated its efficacy in mapping introgression signals when a known phylogenetic network is available. Specifically, PHiMM uses the topology of an input network to construct the hidden Markov model (HMM) structure, enabling site-by-site inference of introgression probabilities. When the true network is known, as often assumed in simulation studies, PHiMM has been shown to maintain high accuracy while achieving substantial reductions in runtime and memory usage compared to existing methods such as PhyloNet-HMM.

In empirical applications, however, the ground-truth phylogenetic network is rarely available a priori. Consequently, network inference is typically guided by existing knowledge or domain expertise [12, 122]. While this approach is feasible for well-studied, small taxon sets, it becomes intractable for large sets of taxa for which only limited biological information is known. In such cases, robust mathematical and computational methods are needed for scalable network inference.

Unfortunately, most current state-of-the-art phylogenetic network inference methods encounter severe computational bottlenecks with datasets containing more than 30 taxa [14]. As the number of taxa increases, the computational requirements grow rapidly, while the corresponding inference accuracy decreases. These constraints highlight the necessity for novel, scalable techniques that can recover phylogenetic networks efficiently while maintaining high accuracy.

FastNet [11] is a recently developed tool that addresses these challenges by employing a divide-and-conquer strategy to infer phylogenetic networks from large-scale genomic datasets. The key insight lies in partitioning the full taxon set into smaller, more closely related subsets, inferring sub-network topologies, and then combining these partial solutions into a final network. This partitioning not only alleviates computational demands but also improves inference accuracy by reducing the evolutionary divergence within each subset.

Building upon these advances, we propose DACS (Divide-And-Conquer and Subsampling), a new algorithm specifically designed to detect introgression events more efficiently in large phylogenomic datasets by integrating a divide-and-conquer approach with a robust subsampling strategy. Unlike the standard PHiMM pipeline, DACS does not require a predefined network as input. Instead, it leverages FastNet's network inference capabilities for large-scale analyses. The framework systematically addresses two fundamental challenges in introgression detection: (1) DACS reduces the computational overhead associated with analyzing a large number of taxa simultaneously; and (2) rather than relying on a single, potentially erroneous phylogenetic network, DACS integrates over multiple plausible topologies to reduce errors. As demonstrated by our simulation experiments and empirical analyses, this approach not only scales effectively to dozens or even hundreds of taxa but also maintains high accuracy in identifying introgressed genomic regions. By enabling large-scale, site-by-site introgression mapping without requiring prior knowledge of the true network or an optimal taxon sampling scheme, DACS opens new avenues for investigating complex evolutionary scenarios.

6.2 Methods

6.2.1 Problem Definition

The inputs of the problem must include the aligned DNA sequences A, which can be defined as $\{A, C, T, G\}^{K \times L}$, where K is the number of taxa, and L is the length of genomic sequence alignment. A phylogenetic network Ψ is optional. But if Ψ is not provided, the number of reticulations presented in the phylogenetic network Ψ should at least be specified, denoted as R. Using that information, FastNet method [11] can be used to infer a phylogenetic network Ψ.

To represent the species network Ψ, we use a collection of MUL-trees [19, 20]. Corresponding gene trees are assumed to be any rooted binary tree on K leaves. Let m and n be the number of MUL-trees and gene trees, respectively. Thus, the total number of HMM hidden states should be $m \times n$, each state represented as a pair (T_i, G_j), where T_i is the i-th MUL-tree ($1 \leq i \leq m$) and G_j

is the j-th gene tree ($1 \leq j \leq n$).

Consistent with the studies of Liu et al. [12] and Wuyun et al. [122], the output of the problem is a sequence of modified posterior decoding probabilities for the columns of the input alignment A. At a site a_t ($1 \leq t \leq L$) in the alignment A, the probability of a_t having an introgressive origin is defined by:

$$p_t = \sum_{\substack{T_i \in \Omega_T \\ 1 \leq j \leq n}} P(\pi_t = (T_i, G_j) | A)$$

where Ω_T is the set of MUL-trees corresponding to introgression events.

6.2.2 Proposed Method

Previously, we introduced the PHiMM approach for introgression mapping. Our simulation results indicated that PHiMM reduces runtime and memory usage significantly compared to PhyloNet-HMM, while maintaining comparable accuracy. However, two key challenges remain:

1. Although PHiMM has largely reduced the runtime and memory usage, it cannot be run efficiently on large numbers of taxa. For instance, more than 300GB of memory may be needed for just 30 taxa with sequence length >100kb.
2. PHiMM requires a phylogenetic network as input to constitute the structure of the HMM. But in real-world datasets, we generally do not know true phylogenetic network information.

To address these challenges, we propose DACS (Divide-And-Conquer and Subsampling), which extends PHiMM by (1) leveraging the divide-and-conquer and subsampling schemes to focus on local reticulation events within smaller taxa subsets, and (2) incorporating FastNet to systematically reduce uncertainty in phylogenetic network inference. The pseudocode of DACS is given in Algorithm 6.1, and an illustration of the pipeline is provided in Figure 6.1.

6.2.2.1 The divide-and-conquer approach with subsampling

To alleviate PHiMM's high computational demands on large datasets, we integrate a divide-and-conquer approach with subsampling.

Phylogenomic subsampling can be defined as a phylogenomic protocol in which loci are sampled at random to create different-sized locus-by-species matrices, with the goal of exploring the stability of a phylogenetic hypothesis [146]. Subsampling can also be performed on taxa, and although many studies have explored this approach [146, 147], often called jackknifing in past studies. Many kinds of subsampling have been employed throughout the history of phylogenetics and phylogenomics [146–149].

DACS begins with a phylogenetic network Ψ. If the true network Ψ is unknown, it can be estimated using a suitable inference method such as FastNet (see the next Chapter for details). Once a network Ψ is obtained, the algorithm identifies all reticulation edges, labeled $\Upsilon_1, ..., \Upsilon_R$ in the input network Ψ, where R is the total number of reticulations in the input network. For each non-sister reticulation $\Upsilon_i (1 \leq i \leq R)$, DACS randomly subsample a small subset of taxa containing the reticulation edge Υ_i multiple times. The subsampling process begins by identifying the most recent common ancestor (MRCA) of all leaves under the putative reticulation. From there, a specified subnetwork size C (e.g., C set to 7) is enforced by first retaining at least one "visible" [150, 151] leaf from the source node and one from the sink node of the reticulation. To fill out the rest of the subset to reach the size C, the remaining leaves are randomly sampled from the overall set, with the constraint that at least one of these sampled leaves must be under the MRCA but not under the reticulation nodes to ensure that the subnetwork captures the non-sister reticulation event. By limiting each subset to a maximum size C (e.g., 4 or 7 taxa), the method substantially reduces runtime and memory usage compared to analyzing the full dataset. This selective subsampling process is repeated M times for each reticulation Υ_i, resulting in multiple random subsamples that ensure the robustness of the final estimates to subsampling variation.

After subsampling, DACS applies the original PHiMM algorithm to each subset to infer site-by-site posterior decoding probability distribution (i.e., introgression probabilities). PHiMM analyses are run using default settings, e.g., the number of iterations for model parameter learning is 300, and the number of runs is set to 10. Users also need to assign a gene tree truncation size k_n for the HMM model of PHiMM. In our simulation study, we run PHiMM with the default setting, $k_n = 15$. Concretely, for the m-th subsample of reticulation Υ_i, DACS obtains an introgression probability

$p_{t,m}^{(\Upsilon_i)}$ at each alignment site t. These probabilities are then merged across all M subsamples for the same reticulation Υ_i using an "average" merge strategy:

$$p_t^{(\Upsilon_i)} = \frac{1}{M} \sum_{m=1}^{M} p_{t,m}^{(\Upsilon_i)}$$

This averaging step stabilizes the introgression probability estimates for each reticulation by reducing the influence of individual subsamples.

Next, DACS merges the introgression probabilities from different reticulations $\Upsilon_1, ..., \Upsilon_R$ via a "maximum" merge strategy:

$$p_t = \max_{1 \le i \le R} p_t^{(\Upsilon_i)}$$

This yields a final site-by-site introgression probability distribution p_t for each alignment column after considering all reticulation events in Ψ. This choice of merge function identifies the largest introgression signal across any of the detected reticulations at each site.

By focusing only on small subsets of taxa around each reticulation, we drastically reduce the memory and computational time. Meanwhile, repeated subsampling ensures that the resulting estimates are robust to sampling variance.

6.2.2.2 Bypassing the need for a predefined network input: Integration with FastNet

While the procedure above applies directly when a phylogenetic network Ψ is provided, real-world datasets often lack a ground-truth network. To account for this network uncertainty, FastNet [11] can be used to infer one or more candidate phylogenetic networks from sequence alignments.

FastNet provides a set of top-ranked network topologies $\Psi_1, ..., \Psi_N$. Each candidate network has a log pseudolikelihood score (defined by Equation (1) in the FastNet paper). When selecting candidate networks, one can either choose the top N scoring networks or choose all networks whose log pseudolikelihood score lies within a specified $\Delta\%$ range of the best score. To avoid including too few or too many networks, here we set a lower limit of 5 and an upper limit of 15 topologies. This procedure ensures that moderately well-supported network topologies are not excluded while preventing an unmanageably large set of candidates.

For each guide network $\Psi_j (1 \leq j \leq N)$ provided by FastNet, above introgression mapping procedure (PHiMM with the divide-and-conquer and subsampling strategy) is run to obtain a introgression probability distribution $\{p_{t,j}\}$. Since different networks may vary in quality, DACS assigns a weight ω_j to each guide network Ψ_j according to its relative log pseudolikelihood:

$$\omega_j = \frac{max_{1 \leq k \leq N} log(L(\Psi_k, \Gamma | G))}{log(L(\Psi_j, \Gamma | G))}$$

Here, $L(\Psi, \Gamma | G)$ means the pseudo-likelihood of phylogenetic network Ψ and inheritance probabilities Γ given a set of gene trees G (see Chapter 2.1.5). This ensures that networks with higher log pseudolikelihood are more influential.

Finally, DACS combines the introgression probabilities from all candidate networks into a single consensus distribution $\tilde{p_t}$ by adopting a weighted average approach:

$$\tilde{p_t} = \frac{\sum_j \omega_j \times p_{t,j}}{\sum_j \omega_j}$$

Users may alternatively adopt other combination functions (e.g., simple averaging, median, or maximum) in place of the weighted average as deemed appropriate.

This multi-network strategy greatly reduces the risk of relying on a single, potentially inaccurate network and captures reticulation signals that might be missed or incorrectly represented by any one topology, and therefore typically improves robustness, particularly for large-scale or complex datasets where reticulation signals can be subtle or where gene flow may be multilayered.

The end result is a site-by-site introgression probability distribution that integrates over: potential reticulation events, multiple subsamplings of large sets of taxa, and uncertainty in network inference. Therefore, these steps provide a scalable, memory-efficient, and robust pipeline for detecting introgression events in large phylogenomic datasets while explicitly accounting for phylogenetic network uncertainties.

6.3 Materials

6.3.1 Simulation Data

The simulation study is used to evaluate the performance and applicability of the proposed DACS method, since we can track the true history of evolutionary events. The simulation data are

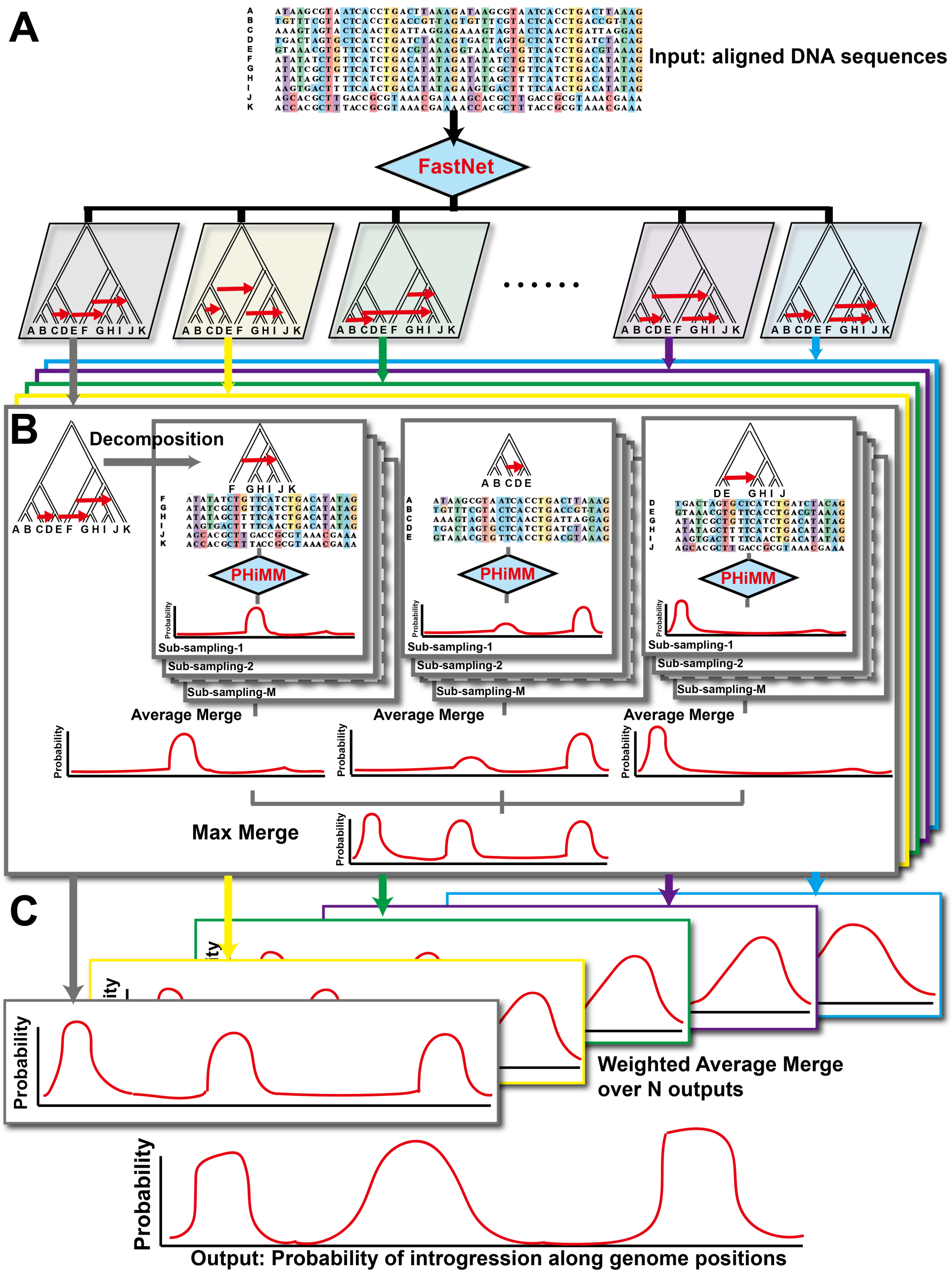


Figure 6.1 **The DACS pipeline.** (A) Starting with a multiple DNA sequence alignment, if a phylogenetic network is not already available, FastNet is used to infer a set of top-ranked candidate networks. (B) Each inferred network is then decomposed around its reticulation events and subjected to repeated subsampling (subsets of taxa around each reticulation) to reduce computational demands. PHiMM is run on each subsample, yielding site-by-site introgression probabilities, which are then averaged per reticulation (average merge). Subsequently, the results for all reticulations in a single network are combined (max merge) to produce an introgression distribution for that network. (C) Finally, to account for uncertainty in network inference, the site-by-site distributions from all candidate networks are combined via a weighted average to generate the final introgression probability profile along the genome alignment (bottom). The pipeline thus integrates the possibility of multiple network topologies, subsampling for computational efficiency, and statistical merging steps to yield robust introgression probability estimates.

Algorithm 6.1 DACS

```
procedure DACSwithInputNetwork($\Psi$, $A$)
                                              ▷ $\Psi$: Phylogenetic network
    ▷ $A^{K \times L}$: Multiple sequence alignment with $K$ aligned sequences and $L$ columns
    $\Upsilon_1, ..., \Upsilon_R \leftarrow IdentifyReticulations(\Psi)$    ▷ $R$: Total number of reticulations in $\Psi$
    for $\Upsilon_i$ from $\Upsilon_1$ to $\Upsilon_R$ do
        $\{p_t^{(\Upsilon_i)}\} \leftarrow 0$
        for $m$ from 1 to $M$ do                          ▷ $M$: Number of replicates
            $\Psi_m^{(\Upsilon_i)}, A_m^{(\Upsilon_i)} \leftarrow SubsampleTaxa(A, \Psi, \Upsilon_i, C)$    ▷ $C$: Maximum size of subsampling
            $\{p_{t,m}^{(\Upsilon_i)}\} \leftarrow PHiMM(\Psi_m^{(\Upsilon_i)}, A_m^{(\Upsilon_i)})$
            $\{p_t^{(\Upsilon_i)}\} \leftarrow \{p_t^{(\Upsilon_i)}\} + \{p_{t,m}^{(\Upsilon_i)}\}$
        $\{p_t^{(\Upsilon_i)}\} \leftarrow \{p_t^{(\Upsilon_i)}\}/M$
    $\{p_t\} \leftarrow \max_{1 \le i \le R}\{p_t^{(\Upsilon_i)}\}$
    return $\{p_t\}_{1 \le t \le L}$


procedure DACSwithoutInputNetwork($\Psi$, $R$)
                                ▷ $R$: Number of reticulations in the input network
    ▷ $A^{K \times L}$: Multiple sequence alignment with $K$ aligned sequences and $L$ columns
    $\Psi_1, L(\Psi_1, \Gamma|G), ..., \Psi_N, L(\Psi_N, \Gamma|G) \leftarrow FastNet(A, R, N)$
                                              ▷ $N$: Number of candidate networks
    ▷ $L(\Psi, \Gamma|G)$: Pseudo-likelihood of phylogenetic network $\Psi$ and inheritance probabilities $\Gamma$ given a set of gene trees $G$
    for $j$ from 1 to $N$ do
        $\{p_{t,j}\} \leftarrow DACSwithInputNetwork(\Psi_j, A)$
        $\omega_j \leftarrow max_{1 \le k \le N} log(L(\Psi_k, \Gamma|G))/log(L(\Psi_j, \Gamma|G))$
    $\{\tilde{p_t}\} \leftarrow \sum_j \omega_j \times \{p_{t,j}\}/\sum_j \omega_j$
    return $\{\tilde{p_t}\}_{1 \le t \le L}$
```

constructed through various tools, such as r8s [28], msmove [29], and seq-gen [32].

(1) Generation of model trees using r8s

A random rooted model tree can be generated using r8s [28] tool, which utilizes the birth-death model for the tree generation.

```
#nexus
begin r8s;
simulate diversemodel=bdback seed=<integer random seed> charevol=yes
ntaxa=<integer greater than 3> infinite=yes nreps=1;
```

Algorithm 6.2 Subsampling

```
procedure SubsampleTaxa($\Psi, A, \Upsilon, C$)
                                                        ▷ $\Psi$: Phylogenetic network
        ▷ $A^{K \times L}$: Multiple sequence alignment with $K$ aligned sequences and $L$ columns
                                                                        ▷ $\Upsilon$: Reticulation
                                                     ▷ $C$: Maximum size of subsampling
    $source\_leaves \leftarrow GetSourceVisibleLeaves(\Psi, \Upsilon)$
    $sink\_leaves \leftarrow GetSinkVisibleLeaves(\Psi, \Upsilon)$
    $MRCA\_leaves \leftarrow GetMRCAleaves(\Psi, \Upsilon)$
    $all\_leaves \leftarrow GetAllLeaves(\Psi)$

    $subset \leftarrow \{\}$
    $subset.Add(RandomChoice(source\_leaves))$
    $subset.Add(RandomChoice(sink\_leaves))$
    $subset.Add(RandomChoice(MRCA\_leaves - source\_leaves - sink\_leaves))$
    while $Size(subset) < C$ do
        $candidate\_leaf \leftarrow RandomChoice(all\_leaves)$
        if $candidate\_leaf \notin MRCA\_leaves + source\_leaves + sink\_leaves$ then
            $subset.Add(candidate\_leaf)$
    $\Psi^{(\Upsilon)} \leftarrow GetSubnetwork(\Psi, \Upsilon, subset)$
    $A^{(\Upsilon)} \leftarrow GetSubsequences(A, subset)$
    return $\Psi^{(\Upsilon)}, A^{(\Upsilon)}$
```

```
end;
```

where “diversemodel” means the model used for tree generation, which is generally set to the birth-death model denoted by “bdback”. “seed” specifies a random seed for the tree generation. “ntaxa” represents the number of taxa, which is set to 10, 20, or 100 for comparison. “nreps” indicates the number of generated repeats. “charevol=yes” indicates the model tree is output with branch lengths. “infinite=yes” represents that branch lengths are set to the expected values based on rate and time.

The height of the resulting tree is scaled to h by multiplying the length of each edge in the model tree by h. Here, we set h to 5.0 coalescent units. Furthermore, an outgroup is added to the generated tree at 50.0 coalescent time.

(2) Generation of model networks

A model network can be generated by the steps listed in Hejase et al. [11]. With the model tree obtained from the above step, we then add r reticulations ($r \in [1, 2, 3, 4, 5]$) by iterating the following steps: a time t_M between 0 and the tree height is selected uniformly at random, two tree edges for which corresponding ancestral populations exist during a time interval $[t_A, t_B]$ such that $t_M \in [t_A, t_B]$ are randomly selected, and a reticulation at time t_M is added to connect the pair of tree edges. Similar to Leaché et al. [15], the model network can be further classified based upon whether gene flow is “deep” or “non-deep”, which is defined by the topological placement of reticulations, i.e., non-deep reticulations are placed between two leaf edges, while deep reticulations include all other reticulations.

(3) Generation of local genealogies using msmove

Given a model species network, 100 local genealogies can be simulated using the msmove [29] for independent and identically distributed (i.i.d.) loci following a species network under a multi-species network coalescent with recombination (MSNCwR) model. msmove is a modified version of Monte Carlo simulator ms [30] allowing the tracking of migration events, while ms does not provide this annotation. Recombination is modeled using Hudson's finite-sites recombination model [30].

```
msmove <number of samples> <number of repeats> -T -r <crossover rate>
<number of sites> -I <number of populations> <n_1 n_2 ...  n_k>
-ej <t_1> i_1 j_1 -ej <t_2> i_2 j_2 ...  -ej <t_k> i_k j_k
-ev <t_m> i j <probability x>
```

where -T parameter indicates the gene trees representing the history of the sampled taxa are output. The -I parameter is followed by k that represents the number of populations. The list of integers (n_1 n_2 ... n_k) includes the number of taxa sampled in each population. In this study, one allele is sampled from each taxon. The -r parameter is used to set recombination by crossover rate and the number of sites between which recombination can occur, where the crossover rate or recombination rate ρ is set to 0.1, and the number of sites between which recombination can occur is set to 900. The -ej parameter specifies moving all lineages in population i to population j at time t. The -ev parameter is special for msmove, which sets migration at time t_m from population i to population j with the migration probability

$x = 0.1$ in this study.

Then, the gene trees will be deviated away from ultrametricity by the following steps [143]. First, a deviation factor, c, that quantifies the deviation level is determined. Then, for each edge in the model tree, its branch length is multiplied by e^x, where x is uniformly and randomly chosen from the interval $[-\lg(c), \lg(c)]$. Similar to Liu et al. [144], c is set to 2.0 here.

(4) Simulation of sequences using seq-gen

The DNA sequences can be generated using seq-gen [32] under the general time-reversible (GTR) substitution model [67].

```
seq-gen -mGTR -f <base frequencies> -r <general reversible
rate matrix> -a <shape of Γ distribution>
-l <sequence length> -p <number of partitions>
< genetreefile > seqfile
```

where -m parameter specifies the general time-reversible (GTR) substitution model denoted by "GTR". The -s parameter sets the mutation rate θ that scales the branch lengths to make them equal the expected number of substitutions per site for each branch. The mutation rate θ is set to 0.1, 0.5, 1, 2, 5, and 10 for comparison. The default setting is 0.5. The -f parameter specifies the frequencies of the four nucleotides A, C, G, and T. The -r parameter sets a relative rate of substitutions between nucleotides in a GTR model. The -a parameter specifies the shape of the Γ distribution. The -l parameter specifies the sequence length. Here, the simulated sequence length of each gene tree is set to 900 bp. The -p parameter sets the number of partitions. The genetreefile is the input file providing the gene trees. The seqfile is the output file with simulated sequences under the given gene trees.

The GTR substitution model parameter values were estimated based on empirical analyses of the mouse genomic sequence dataset from Liu et al. [7] study. We used all M. m. domesticus and M. spretus samples listed in "Table S1" of Liu et al. [7] study and concatenated all chromosomes to get the sequence data. RAxML was used to perform concatenated phylogenetic MLE under the GTR model. The estimated parameters used in seq-gen simulations include base frequencies: $\pi_A = 0.216$, $\pi_C = 0.284$, $\pi_G = 0.285$, $\pi_T = 0.215$; GTR matrix: 1.002820, 5.486349, 1.095939, 0.539209, 5.535556, and 1.000000 for $A \leftrightarrow C$, $A \leftrightarrow G$, $A \leftrightarrow T$, $C \leftrightarrow G$, $C \leftrightarrow T$, and $G \leftrightarrow T$, respectively; shape parameter α of the Γ distribution: 0.543225.

Then, coalescent times are converted into branch lengths using "Equation (3.1)" in Hein et al. [145].

For each model condition, the simulation procedure is repeated to obtain 20 replicate datasets. Model condition parameters and summary statistics for simulated datasets are shown in Table 6.1.

Table 6.1 **Statistics for the simulation dataset.** The reticulation scenarios include solely non-deep reticulations, solely deep reticulations, and a combination of both non-deep and deep reticulations. "Ntaxa" means the number of taxa. "Network Height" gives the height of the model network. "Total Sites" indicates the number of total sites in the simulation genomes. The p-distance of a pair of aligned sequences is calculated by dividing the number of sites where the two sequences had different nucleotides by the number of sites in which both sequences had nucleotides. "Average/Maximum p-dist" is the average/maximum p-distance of all pairs of aligned sequences in the simulation dataset. "Introgression (%)" shows the percent of introgressed sites over the genome. "Average Branch Length" is the average branch length of the model network. The average and standard error (SE) are shown based on 20 replicates.

Ntaxa	Reticulation Scenario	Total Sites	Network Height	Average p-dist (%)		Maximum p-dist (%)		Introgression (%)		Average Branch Length	
				Average	SE	Average	SE	Average	SE	Average	SE
10	1 non-deep	90000	5	55.625	0.209	68.500	0.020	13.450	1.488	1.381	0.041
10	1 deep	90000	5	55.912	0.276	68.403	0.026	14.850	1.875	1.432	0.050
10	2 non-deep	90000	5	55.540	0.368	68.519	0.026	16.900	1.525	1.366	0.054
10	1 non-deep + 1 deep	90000	5	56.094	0.210	68.452	0.023	21.000	1.515	1.469	0.049
20	1 non-deep	90000	5	53.671	0.329	68.522	0.029	9.550	0.872	1.055	0.047
20	1 deep	90000	5	53.187	0.221	68.548	0.024	19.250	1.745	0.941	0.036
20	2 non-deep	90000	5	53.288	0.291	68.514	0.020	15.350	1.024	0.987	0.048
20	1 non-deep + 1 deep	90000	5	53.304	0.208	68.529	0.016	17.750	1.710	0.968	0.032
100	1 non-deep	90000	5	52.657	0.136	68.595	0.025	9.150	1.057	0.596	0.019
100	2 non-deep	90000	5	52.519	0.175	68.597	0.027	11.550	1.077	0.593	0.021
100	3 non-deep	90000	5	52.501	0.302	68.572	0.015	11.700	0.995	0.624	0.023
100	4 non-deep	90000	5	52.438	0.249	68.573	0.025	15.250	0.820	0.625	0.028
100	5 non-deep	90000	5	52.535	0.252	68.599	0.024	13.737	0.670	0.588	0.025
100	1 non-deep + 1 deep	90000	5	52.245	0.303	68.590	0.025	15.350	1.044	0.596	0.025
100	2 deep	90000	5	52.278	0.278	68.605	0.029	15.350	0.844	0.605	0.020
100	3 deep	90000	5	52.403	0.178	68.576	0.026	18.100	1.013	0.606	0.020

6.3.2 Mosquito Data

We downloaded the MAF whole genome alignment from high-depth field samples of Anopheles species from Dryad (doi: 10.5061/dryad.f4114) [152]. The data consist of one genome from each of the species An. gambiae, An. coluzzii, An. arabiensis, An. quadriannulatus, An. merus, An.

melas, An christyi, and An epiroticus.

Since DACS method can be applied to dozens of taxa, we added more background mosquito species from the study of Neafsey et al. [153]. Following the alignment strategy in studies of Neafsey et al. [153] and Fontaine et al. [152], we use *An. gambiae* PEST (AgamP3) as reference, i.e., all other species assemblies are mapped to the *An. gambiae* PEST (AgamP3) reference assembly. The all-against-all pairwise LASTZ (https://lastz.github.io/lastz/) alignments are projected to ensure that the reference species is "single-coverage", i.e. in any pairwise alignment, regions of the reference species may only be present once. Subsequent projection steps are then performed as guided by the species dendrogram (Figure 6.2) to progressively combine the pairwise alignments, and then the multiple alignments, until they encompass the complete phylogeny of all 19 species.

To avoid the impact of input phylogenetic networks on introgression detection, we directly used the species network from studies of Neafsey et al. [153] and Fontaine et al. [152] for our introgression analyses (see Figure 6.2).

Finally, we applied DACS to the 2L, 2R, 3L, 3R, and X chromosomal arms of 19 mosquito species. DACS was run with the same settings used in the simulation study, except for a modification of the number of iterations for model parameter learning increased to 1000 (as opposed to 300).

6.3.3 Darwin's Finch Data

As part of our empirical study, we re-analyzed Darwin's finch genomic sequence data originally published by Lamichhaney et al. [154]. We began by downloading the original Illumina HiSeq2000 paired-end read data from the NCBI SRA database (accession number PRJNA263122). From these data, we randomly selected one sample per species, yielding a dataset of 20 samples. We also downloaded the assembled whole-genome sequence and gene annotations for the medium ground finch, *Geospiza fortis*, from the GigaDB database (http://gigadb.org/dataset/100040). Next-generation sequencing (NGS) read alignment, quality filtering, and variant calling steps were based on the study of [154]. First, paired-end reads for each sample were aligned to the *G. fortis* reference genome using BWA version 0.7.17 with default parameters [155]. The medium ground finch genome assembly contains 27,239 scaffolds unassigned to chromosomes. Quality filtering, post-

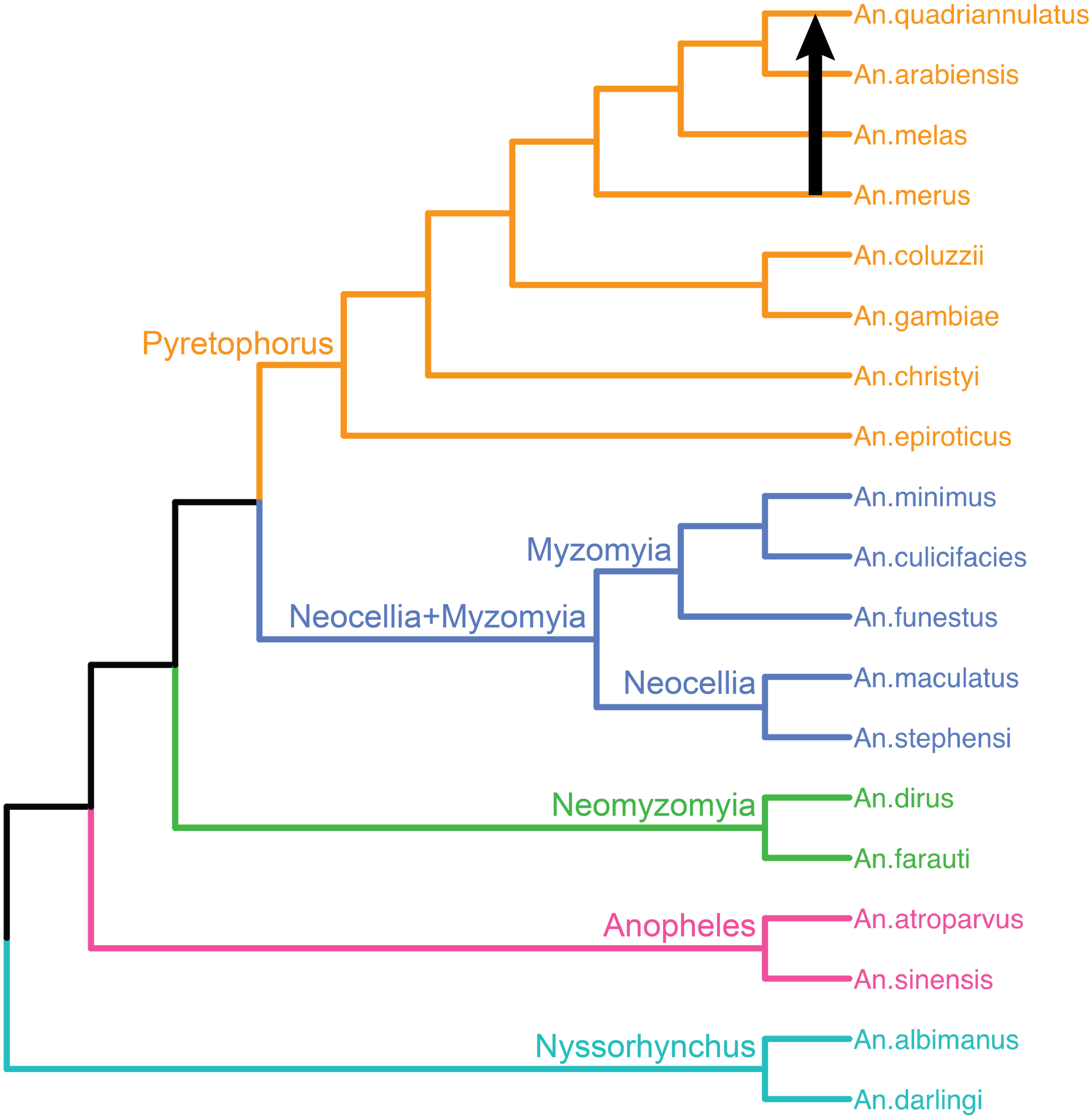


Figure 6.2 **The phylogenetic network used on mosquito data.** The phylogenetic network captures introgression from *An. merus* to *An. quadriannulatus*. Branch colors indicate genus information: orange for 8 *Pyretophorus* species, blue for 5 *Neocellia+Myzomyia* species, green for 2 *Neomyzomyia* species, pink for 2 *Anopheles* species, and cyan for 2 *Nyssorhynchus* species.

Table 6.2 **Mosquito species used in this study.** We used existing mosquito dataset from previous studies. Each row lists species name and the corresponding data sources. References to original sources are also included for traceability.

Species	Source	Reference
An. arabiensis	Dryad (doi: 10.5061/dryad.f4114)	[152]
An. christyi	Dryad (doi: 10.5061/dryad.f4114)	[152]
An. coluzzii	Dryad (doi: 10.5061/dryad.f4114)	[152]
An. epiroticus	Dryad (doi: 10.5061/dryad.f4114)	[152]
An. gambiae	Dryad (doi: 10.5061/dryad.f4114)	[152]
An. melas	Dryad (doi: 10.5061/dryad.f4114)	[152]
An. merus	Dryad (doi: 10.5061/dryad.f4114)	[152]
An. quadriannulatus	Dryad (doi: 10.5061/dryad.f4114)	[152]
An. albimanus	GenBank Assembly: GCA_000349125.2	[153]
An. atroparvus	GenBank Assembly: GCA_000473505.1	[153]
An. culicifacies	GenBank Assembly: GCA_000473375.1	[153]
An. darlingi	GenBank Assembly: GCA_943734745.1	[153]
An. dirus	GenBank Assembly: GCA_000349145.1	[153]
An. farauti	GenBank Assembly: GCA_000473445.2	[153]
An. funestus	GenBank Assembly: GCA_000349085.1	[153]
An. maculatus	GenBank Assembly: GCA_000473185.1	[153]
An. minimus	GenBank Assembly: GCA_000349025.1	[153]
An. sinensis	GenBank Assembly: GCA_000472065.2	[153]
An. stephensi	GenBank Assembly: GCA_000349045.1	[153]

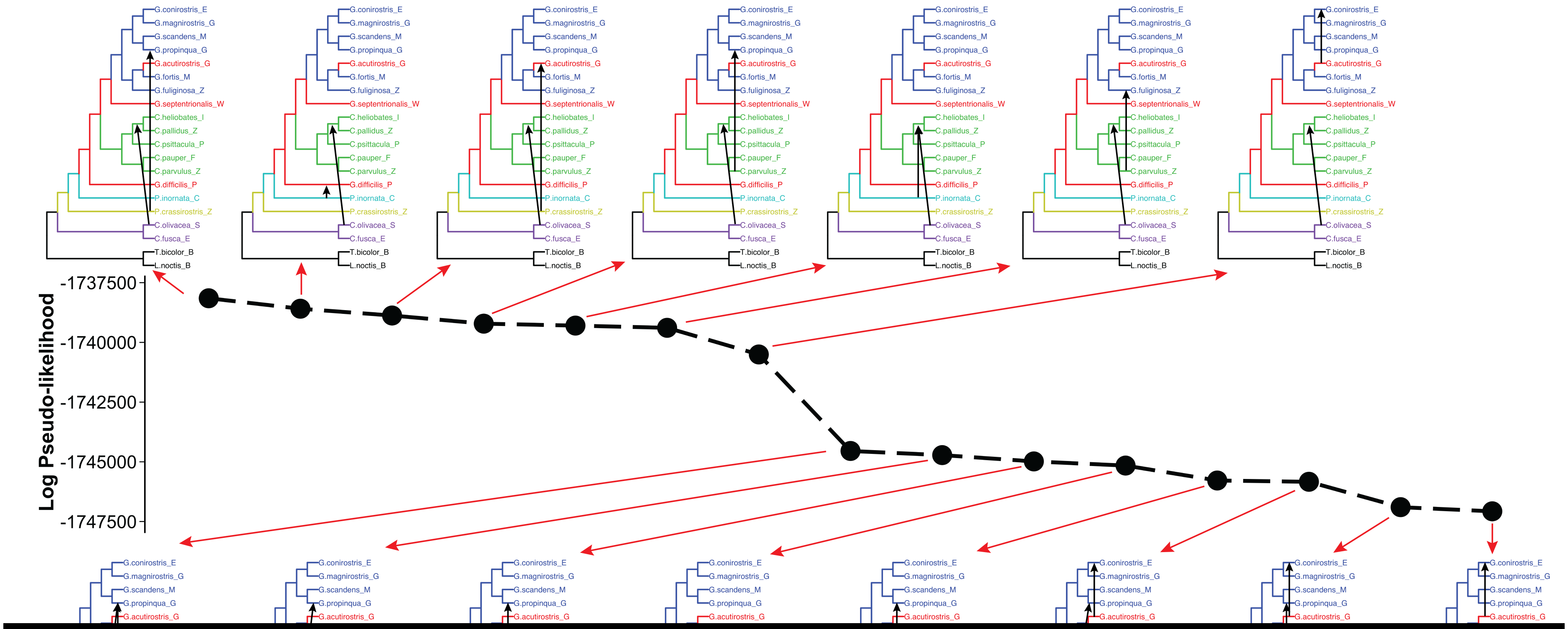
processing, and variant calling tasks were then performed using SAMtools [156]. Identifed variants consisted of SNPs and short indel polymorphisms.

To obtain haplotype sequences, bi-allelic SNPs were phased using fastPHASE version 1.4.8 [132]. The phased calls for bi-allelic SNPs were combined with the genotypic data of homozygous multi-allelic SNPs and homozygous indel polymorphisms. The heterozygous multi-allele SNPs and heterozygous indels, representing less than 1% of the input data, were treated as missing data.

To obtain a species phylogeny for our introgression analyses, we used FastNet [11] to infer species networks from the genomic alignment of all 27,239 scaffolds. The species tree listed in Lamichhaney et al. [154] was used as a starting tree and the number of reticulations was set ot 2 for FastNet's local search heuristics by performing maximum likelihood estimation (MLE) optimization using PhyloNet software package [13, 123]. As shown in Figure 6.3, we used top 15 two-reticulation species networks as input of DACS. These networks include introgression between warbler finches and non-warbler finches, introgression between *G. propinqua*_G and tree finches, and introgression between *G. propinqua*_G and *G. conirostris*_E, consistent with previous reports of extensive hybridization and gene flow among Darwin's finches [154, 157].

Finally, we applied DACS to the 469 largest scaffolds with sequence length greater than 100kb for computational efficiency. DACS was run with the same settings used in the simulation study, except for a modification of the number of iterations for model parameter learning increased to 1000 (as opposed to 300).

The summarized statistics of these datasets can be found in Table 6.3.



[redacted]

Figure 6.3 **The top 15 phylogenetic networks inferred by FastNet on Darwin’s finch data.** Branch colors indicate species groupings: purple for warbler finches, yellow for vegetarian finch, cyan for cocos finch, red for sharp-beaked ground finches, green for tree finches, blue for all other ground finches, and black for outgroups.

Table 6.3 **Summary of samples of Darwin's finches and outgroup species.** The table lists key metadata for Darwin's finches and their outgroups. "Classification" column indicates whether a sample belongs to Darwin's finches or an outgroup species. "Sample Name" provides a unique label assigned to each sample. "Common Name" and "Species (Alias)" detail the vernacular name (e.g., large ground finch) and the corresponding scientific name (e.g., *Geospiza magnirostris*). "Island (Abbreviation)" column identifies the sampling site for each species (e.g., Daphne Major, abbreviated "M"). "NCBI Accession" refers to the publicly available sequencing record associated with each sample.

Classification	**Sample Name**	**Common Name**	**Species (Alias)**	**Island (Abbreviation)**	**NCBI Accession**
Darwin's finches	*G. magnirostris*_G	Large ground finch	*Geospiza magnirostris*	Genovesa (G)	SRR1607485
	*G. fortis*_M	Medium ground finch	*Geospiza fortis*	Daphne Major (M)	SRR1607458
	*G. fuliginosa*_Z	Small ground finch	*Geospiza fuliginosa*	Santa Cruz (Z)	SRR1607462
	*G. propinqua*_G	Large cactus finch	*Geospiza propinqua*	Genovesa (G)	SRR1607365
	*G. conirostris*_E	Large cactus finch	*Geospiza conirostris*	Española (E)	SRR1607296
	*G. scandens*_M	Common cactus finch	*Geospiza scandens*	Daphne Major (M)	SRR1607547
	*G. difficilis*_P	Sharp-beaked ground finch	*Geospiza difficilis*	Pinta (P)	SRR1607420
	*G. septentrionalis*_W	Sharp-beaked ground finch	*Geospiza septentrionalis*	Wolf (W)	SRR1607440
	*G. acutirostris*_G	Sharp-beaked ground finch	*Geospiza acutirostris*	Genovesa (G)	SRR1607406
	*C. heliobates*_I	Mangrove finch	*Camarhynchus heliobates*	Isabela (I)	SRR1607472
	*C. pallidus*_Z	Woodpecker finch	*Camarhynchus pallidus*	Santa Cruz (Z)	SRR1607494
	*C. psittacula*_P	Large tree finch	*Camarhynchus psittacula*	Pinta (P)	SRR1607543
	*C. parvulus*_Z	Small tree finch	*Camarhynchus parvulus*	Santa Cruz (Z)	SRR1607504
	*C. pauper*_F	Medium tree finch	*Camarhynchus pauper*	Floreana (F)	SRR1607508
	*P. crassirostris*_Z	Vegetarian finch	*Platyspiza crassirostris*	Santa Cruz (Z)	SRR1607534
	*C. olivacea*_S	Green warbler finch	*Certhidea olivacea*	Santiago (S)	SRR1607385
	*C. fusca*_E	Grey warbler finch	*Certhidea fusca*	Española (E)	SRR1607359
	*P. inornata*_C	Cocos finch	*Pinaroloxias inornata*	Cocos Island (C)	SRR1607529
outgroup	*T. bicolor*_B	Black-faced grassquit	*Tiaris bicolor*	Barbados (B)	SRR1607551
	*L. noctis*_B	Lesser Antillean bullfinch	*Loxigilla noctis*	Barbados (B)	SRR1607480

6.3.4 Performance Assessments

To evaluate and compare the performance of different approaches on the simulation dataset where the true history of evolutionary events can be tracked, we use two types of area under the curve (AUC): the area under the receiver-operating characteristic (ROC) curve, and the area under the Precision-Recall curve, referred to as simply ROC-AUC and PR-AUC, respectively. ROC-AUC is plotted by the true positive rate ($\frac{TP}{TP+FN}$) as a function of the false positive rate ($\frac{FP}{FP+TN}$) at various threshold settings, while PR-AUC similarly plots the precision ($\frac{TP}{TP+FP}$) against the recall ($\frac{TP}{TP+FN}$) at different thresholds, where TP, FP, TN, and FN represent the numbers of true positives, false positives, true negatives, and false negatives, respectively. The ROC-AUC and PR-AUC measures represent the tradeoff between type I errors and type II errors under different threshold values. The measures are calculated on all loci in the simulation datasets, where the true migration events are annotated by msmove with an asterisk. After concatenating all query loci in one simulation replicate, the ROC-AUC and PR-AUC of a method are assessed based on the probability of a particular site involving an introgressive origin.

Additionally, we report the memory usage and runtime in order to comprehensively evaluate the scalability of our framework.

6.3.5 Software and Data

Our software implementation of the DACS algorithm includes the divide-and-conquer strategy to combine FastNet [11] and PHiMM [122] methods to realize the coestimation of phylogenetic network and introgression mapping. Open-source software and open data for all study datasets are publicly accessible at https://gitlab.msu.edu/liulab/phimm2.

6.4 Results

6.4.1 Simulation Studies

6.4.1.1 DACS's parameter setting on simulation data

To determine optimal parameter settings for DACS under 20-taxon model conditions with two non-deep reticulations, we systematically evaluate the impact of subsampling size, subsampling replicates, number of input networks, and merging strategies on both accuracy and runtime.

In Figure 6.4A and Table 6.4, we first investigate the impact of two key factors on model performance (the receiver operating characteristic area under the curve or ROC-AUC shown in the left y-axis) and computational cost (the runtime in CPU hours shown in the right y-axis): the sampling size and the number of subsampling replicates, on 20-taxon model conditions with 2 non-deep reticulations. We find that higher sampling size (blue markers) consistently yields a greater ROC-AUC than lower sampling size (red markers). This indicates that including more samples per subsample helps capture the underlying signal more effectively, thereby improving predictive performance. Furthermore, increasing the number of subsampling replicates—shown on the x-axis—improves the ROC-AUC for both sampling sizes, suggesting that more extensive subsampling helps the model generalize and reduces uncertainty in performance estimates. However, these gains in accuracy come with a cost in computational time. As the number of subsampling replicates increases, the runtime also rises, and this effect is more pronounced for the larger sampling

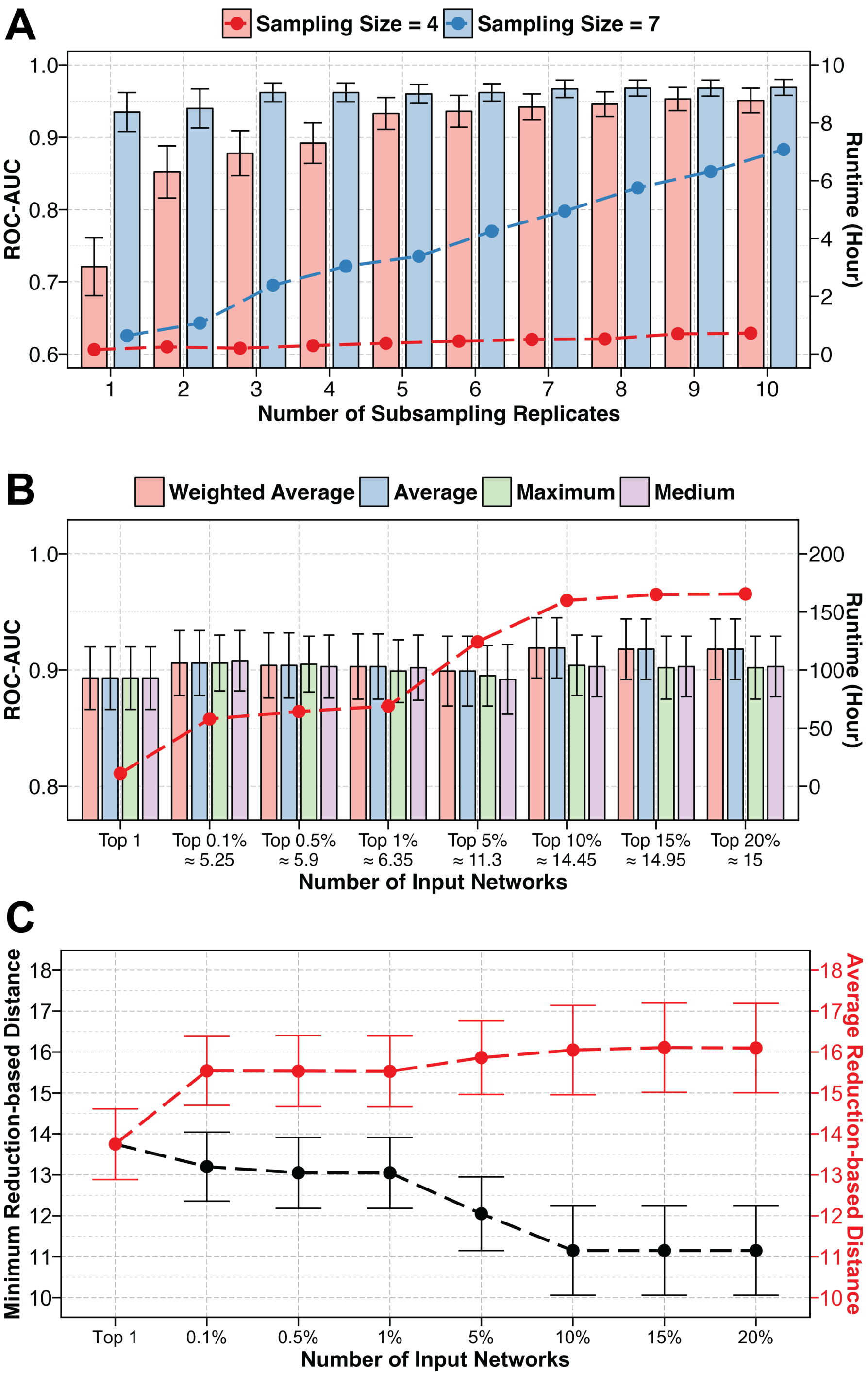


Figure 6.4 **The parameter setting on 20-taxon model conditions with 2 non-deep reticulations.** (A) ROC-AUC (left y-axis, bars) and runtime (right y-axis, dash lines) as functions of the number of subsampling replicates. Red bars/lines represent for a sampling size of 4, and blue bars/lines for a sampling size of 7. (B) ROC-AUC (left y-axis, bars) and runtime (right y-axis, dash lines) for four merging strategies of combining multiple input networks: Weighted Average (red), Average (blue), Maximum (green), and Medium (purple). The x-axis shows the number or the fraction of top-scoring candidate networks. (C) The minimum (left y-axis, black dash lines) and average (right y-axis, red dash lines) reduction-based distance between true and estimated networks (details see Chapter 2.2.3.2) for different numbers or fractions of top-scoring candidate networks.

Table 6.4 **The sampling parameter setting on 20-taxon model conditions with 2 non-deep reticulations.** The measures include the area under the receiver-operating characteristic curve (ROC-AUC) and runtime (in CPU Hour). DACS method was run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates (represented as "average value ± SE value" or only "average" for runtime).

Sampling Size	Number of subsampling	ROC-AUC	Runtime (CPU Hour)
Sampling Size = 4	1	0.721±0.040	0.157
	2	0.852±0.036	0.253
	3	0.878±0.031	0.206
	4	0.892±0.028	0.297
	5	0.933±0.022	0.380
	6	0.936±0.022	0.451
	7	0.942±0.018	0.508
	8	0.946±0.017	0.524
	9	0.953±0.016	0.702
	10	0.951±0.017	0.724
Sampling Size = 7	1	0.935±0.027	0.639
	2	0.940±0.027	1.075
	3	0.962±0.013	2.380
	4	0.962±0.013	3.041
	5	0.960±0.013	3.385
	6	0.962±0.012	4.253
	7	0.967±0.012	4.950
	8	0.968±0.011	5.751
	9	0.968±0.011	6.318
	10	0.969±0.011	7.078

size. Similar trend is found for sampling size where the sampling size of 7 reflects a higher overall computational burden. These findings highlight the importance of tuning both the sampling strategy and the number of replicates to achieve an acceptable balance of performance and resource usage for a given analysis. In the following sections, we set the sampling size to 7 and the subsampling replicates to 10 as default.

Figure 6.4B and Table 6.5 compare four different strategies—Weighted Average, Average, Maximum, and Medium—for combining multiple input networks, illustrating the trade-off between predictive performance (left y-axis) and computational runtime (right y-axis). The Weighted Average method (pink bars) exhibits lower ROC-AUC when only a small subset of networks is

Table 6.5 **The network parameter setting on 20-taxon model conditions with 2 non-deep reticulations.** The measures include the area under the receiver-operating characteristic curve (ROC-AUC) and runtime (in CPU Hour). DACS method was run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates (represented as "average value ± SE value" or only "average" for runtime).

Merging strategy	Number of networks	ROC-AUC	Runtime (CPU Hour)
Weighted Average	Top 1	0.893±0.027	11.043
	Top 0.1%≈5.25	0.906±0.028	57.912
	Top 0.5%≈5.9	0.904±0.028	64.274
	Top 1%≈6.35	0.903±0.028	68.933
	Top 5%≈11.3	0.899±0.030	124.232
	Top 10%≈14.45	0.919±0.026	159.871
	Top 15%≈14.95	0.918±0.026	164.963
	Top 20%≈15	0.918±0.026	165.452
Average	Top 1	0.893±0.027	11.043
	Top 0.1%≈5.25	0.906±0.028	57.912
	Top 0.5%≈5.9	0.904±0.028	64.274
	Top 1%≈6.35	0.903±0.028	68.933
	Top 5%≈11.3	0.899±0.030	124.232
	Top 10%≈14.45	0.919±0.026	159.871
	Top 15%≈14.95	0.918±0.026	164.963
	Top 20%≈15	0.918±0.026	165.452
Maximum	Top 1	0.893±0.027	11.043
	Top 0.1%≈5.25	0.906±0.024	57.912
	Top 0.5%≈5.9	0.905±0.024	64.274
	Top 1%≈6.35	0.899±0.027	68.933
	Top 5%≈11.3	0.895±0.026	124.232
	Top 10%≈14.45	0.904±0.026	159.871
	Top 15%≈14.95	0.902±0.027	164.963
	Top 20%≈15	0.902±0.027	165.452
Medium	Top 1	0.893±0.027	11.043
	Top 0.1%≈5.25	0.908±0.026	57.912
	Top 0.5%≈5.9	0.903±0.027	64.274
	Top 1%≈6.35	0.902±0.028	68.933
	Top 5%≈11.3	0.892±0.030	124.232
	Top 10%≈14.45	0.903±0.026	159.871
	Top 15%≈14.95	0.903±0.026	164.963
	Top 20%≈15	0.903±0.026	165.452

Table 6.6 **The network errors of different network parameter settings on 20-taxon model conditions with 2 non-deep reticulations.** The measures include the average and minimum reduction-based distance between true and estimated networks (details see Chapter 2.2.3.2). DACS method was run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates (represented as "average value ± SE value").

Number of networks	**Minimum reduction-based distance**	**Average eduction-based distance**
Top 1	13.750±0.864	13.750±0.864
Top 0.1%≈5.25	13.200±0.842	15.541±0.598
Top 0.5%≈5.9	13.050±0.866	15.534±0.617
Top 1%≈6.35	13.050±0.866	15.528±0.630
Top 5%≈11.3	12.050±0.899	15.863±0.554
Top 10%≈14.45	11.150±1.091	16.048±0.518
Top 15%≈14.95	11.150±1.091	16.107±0.498
Top 20%≈15	11.150±1.091	16.097±0.498

considered (e.g., "Top 1"), but its performance steadily improves as the number of input networks grows. Eventually, Weighted Average surpasses the other methods, indicating a notable advantage in predictive accuracy once enough candidate networks are incorporated. Meanwhile, Average (blue bars), Maximum (green bars), and Medium (purple bars) remain relatively close to one another, suggesting that these three aggregation methods yield comparable performance under different sampling fractions. However, these performance gains come with a pronounced rise in runtime (red dashed line) as the number of input networks increases. The runtime begins at a relatively modest level when only a single or a few top networks are selected but climbs sharply beyond 100 CPU hours as the subset of considered networks expands. This pattern underscores the computational costs associated with exploring a larger space of input networks, even though such exploration can improve predictive performance.

To further refine the selection of candidate networks used for merging, we evaluated the topological similarity between true and estimated networks using the reduction-based distance (Figure 6.4C and Table 6.6). As more top-scoring networks were incorporated, the minimum reduction-based distance decreased, indicating an improved best-case network topology match. However, the average reduction-based distance plateaued beyond the top 10%, implying diminishing gains from adding more networks beyond this threshold. These results suggest that while merging a greater number of candidate networks can slightly improve the chance of obtaining a network close to the true topology, the benefit saturates rapidly.

Overall, balancing accuracy and computational efficiency, we set the fraction of top-scoring candidate networks to 10% with weighted average method as default in the following sections.

6.4.1.2 Comparison of PHiMM and DACS with true phylogeny input

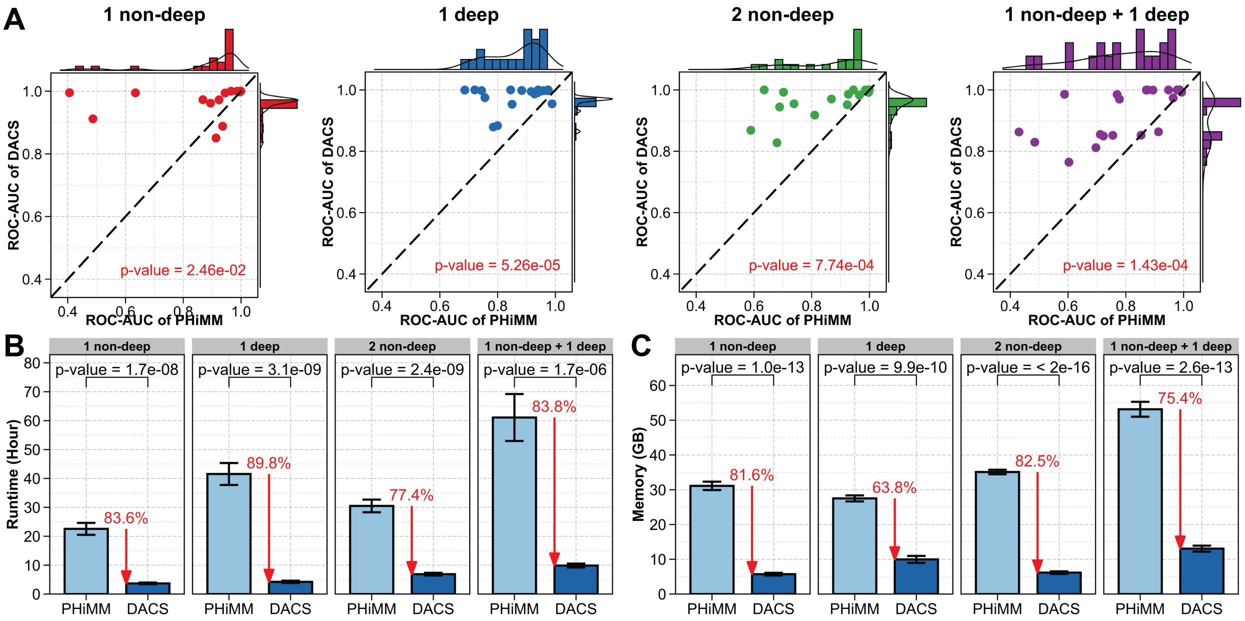


Figure 6.5 **The performance comparison between PHiMM and DACS with true network as input on 20-taxon model conditions.** Reticulation scenarios include one non-deep reticulation, one deep reticulation, two non-deep reticulations, and a combination of both non-deep and deep reticulations. (A) The head-to-head comparison of area under the receiver-operating characteristic curve (ROC-AUC) between PHiMM and DACS on different reticulation scenarios. Statistical significance of performance differences between PHiMM and DACS was evaluated using a one-tailed pairwise Student's t-test ($\alpha = 0.05$; $n = 20$). (B) The runtime (in CPU Hour) comparison between PHiMM and DACS. (C) The memory usage (in GB) comparison between PHiMM and DACS. The percentages and arrows (in red) shown in panels (B) and (C) indicate the improvement in DACS's performance compared to PHiMM's performance. Both PHiMM and DACS methods were run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates.

We make a comprehensive comparison of the PHiMM and DACS methods, using a true network as input under various 20-taxon model conditions, including different reticulation scenarios: one non-deep reticulation, one deep reticulation, two non-deep reticulations, and a combinations of both.

Figure 6.5A presents the head-to-head comparison of the area under the receiver operating characteristic curve (ROC-AUC) between PHiMM and DACS across the different reticulation scenarios. The results reveal a significant improvement in ROC-AUC for DACS over PHiMM across all conditions with p-values by one-tailed pairwise Student's t-test, all below 0.05.

Figure 6.5B indicates the comparison of runtime (in CPU hours) between PHiMM and DACS under the same conditions. It is evident that DACS not only improves accuracy but also reduces computational time significantly. For instance, DACS exhibits an 83.6%, 89.8%, 77.4%, and 83.8% reduction in runtime, respectively, in the case of one non-deep, one deep, two non-deep, and combined non-deep and deep reticulations with p-values by one-tailed pairwise Student's t-test all below 0.05. These substantial reductions in runtime underscore the efficiency gains achieved by DACS.

Figure 6.5C further examines memory usage (in GB) between the two methods. Here, DACS consistently demonstrates lower memory consumption across all tested conditions. The reductions range from 63.8% to 82.5%, with the highest memory savings observed in the non-deep reticulation scenario. These findings suggest that DACS not only enhances computational speed but also optimizes memory usage, making it a more resource-efficient method compared to PHiMM.

In summary, Figure 6.5 provides clear evidence that DACS is superior to PHiMM across multiple dimensions, including model accuracy, runtime efficiency, and memory consumption, under varying reticulation scenarios. The statistical significance of the results further validates the robustness of DACS, highlighting its potential for more effective and resource-efficient modeling in phylogenetic studies.

6.4.1.3 Comparison of PHiMM and DACS with inferred phylogeny by FastNet

We extend our comparison of PHiMM and DACS to estimated phylogenetic networks as input under various reticulation scenarios in 20-taxon model conditions. Similar trends are seen when estimated networks are used as input under various 20-taxon model conditions (Figure 6.6).

The ROC-AUC values in DACS continue to outperform those in PHiMM when estimated networks are used (Figure 6.6A). The improvement is statistically significant across all scenarios, as indicated by the p-values being below the 0.05 threshold. These results underscore that DACS

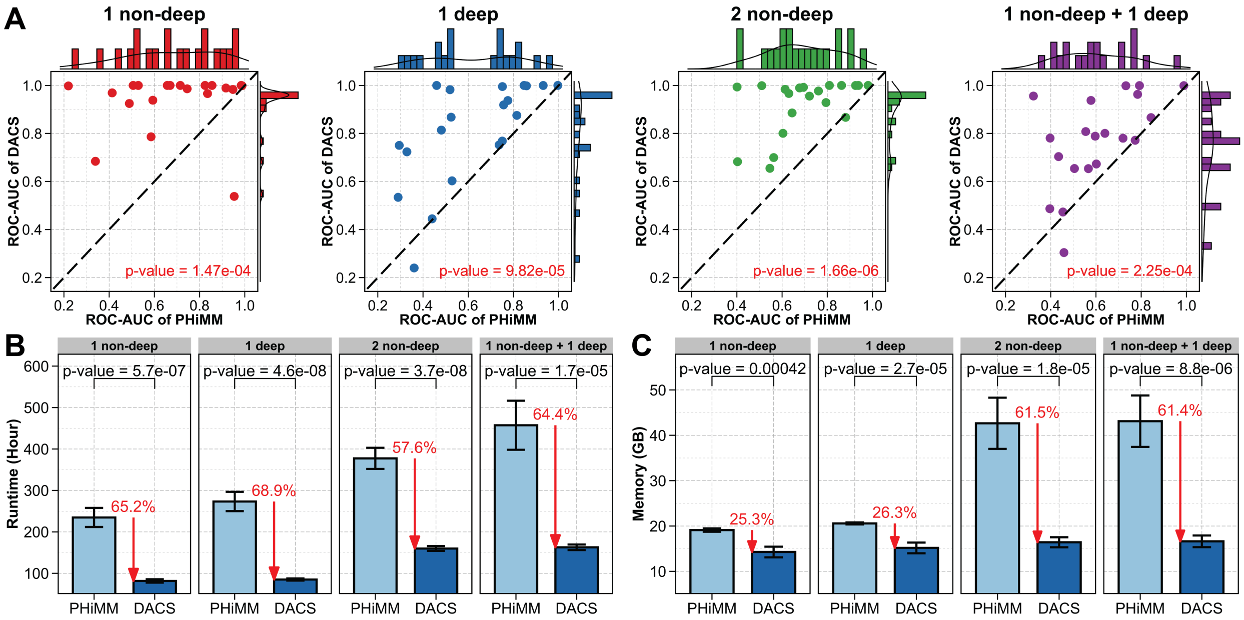


Figure 6.6 **The performance comparison between PHiMM and DACS with estimated networks as input on 20-taxon model conditions.** Reticulation scenarios include one non-deep reticulation, one deep reticulation, two non-deep reticulations, and a combination of both non-deep and deep reticulations. (A) The head-to-head comparison of area under the receiver-operating characteristic curve (ROC-AUC) between PHiMM and DACS on different reticulation scenarios. Statistical significance of performance differences between PHiMM and DACS was evaluated using a one-tailed pairwise Student's t-test ($\alpha = 0.05$; $n = 20$). (B) The runtime (in CPU Hour) comparison between PHiMM and DACS. (C) The memory usage (in GB) comparison between PHiMM and DACS. The percentages and arrows (in red) shown in panels (B) and (C) indicate the improvement in DACS's performance compared to PHiMM's performance. Both PHiMM and DACS methods were run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates.

maintains a robust performance advantage over PHiMM in different reticulation scenarios, even with the potential inaccuracies introduced by estimated networks.

Figure 6.6B shows the comparison of runtime (in CPU hours) between PHiMM and DACS under these conditions. The efficiency improvements seen with true networks (as in Figure 6.5) are also observed when using estimated networks, suggesting that DACS offers substantial computational benefits across different input conditions.

Figure 6.6C presents the memory usage (in GB) comparison between the two methods. Once again, DACS demonstrates lower memory consumption across all tested conditions. These results indicate that DACS not only accelerates computational performance but also optimizes memory utilization, even when estimated networks are used as input.

Overall, Figure 6.6 confirms that DACS significantly outperforms PHiMM with substantial improvements in ROC-AUC, runtime, and memory usage, even when the networks are estimated rather than true. This reinforces the reliability and resource efficiency of DACS in handling complex phylogenetic scenarios.

The consistency of these findings are also seen across various 10-taxon model conditions in Table 6.7 and 6.8. These tables illustrate comparisons of the PHiMM and DACS methods, using either a true network or estimated networks as input across diverse reticulation scenarios. The data consistently demonstrate the superior performance of DACS over PHiMM in terms of accuracy, computational runtime, and memory efficiency, irrespective of the input network utilized.

6.4.1.4 Impact of network inference on DACS's introgression detection

To quantify how inaccuracies in the input phylogenetic network influence DACS, we compare ROC-AUC values with the reduction-based distances between true and estimated networks under 20-taxon model conditions with four reticulation scenarios: (i) one non-deep reticulation, (ii) one deep reticulation, (iii) two non-deep reticulations, and (iv) one non-deep plus one deep reticulation (Table 6.9 and Figure 6.7).

In the simplest scenario involving a single non-deep reticulation, DACS remains highly robust with average ROC-AUC of 0.938, and both the minimum and average reduction-based distances between true and estimated networks are relatively small (6.55 and 12.79, respectively). The weak Pearson correlation coefficient (PCC) between minimum distance and ROC-AUC (PCC=0.088) indicates that minimal impact of minor network inference errors on downstream detection performance. With two non-deep reticulations, ROC-AUC stays high (0.919), yet the correlation between ROC-AUCs and reduction-based distances becomes strongly negative (PCC=-0.419), indicating that larger topological errors directly reduced DACS's detection accuracy.

As the reticulation complexity increases, particularly in scenarios involving deep reticulations, DACS's performance declines rapidly. In the single-deep scenario the minimum reduction-based distance nearly doubles (13.05) and ROC-AUC falls to 0.810. This trend is most pronounced in the hybrid scenario combining one non-deep and one deep reticulation, where ROC-AUC is the lowest

Table 6.7 **The performance comparison between PHiMM and DACS with true network as input on 10- and 20-taxon model conditions.** Reticulation scenarios include one non-deep reticulation, one deep reticulation, two non-deep reticulations, and a combination of both non-deep and deep reticulations. The measures include the area under the receiver-operating characteristic curve (ROC-AUC), the area under the precision-recall curve (PR-AUC), runtime (in CPU Hour), and memory usage (in GB). Statistical significance of performance differences between PHiMM and DACS was evaluated using a one-tailed pairwise Student's t-test (α = 0.05; n = 20). Both PHiMM and DACS methods were run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates (represented as "average value ± SE value").

Measure	**Ntaxa**	**Reticulation Scenario**	**PHiMM**	**DACS**	***p*-value**
ROC-AUC	10	1 non-deep	0.917±0.024	0.937±0.027	5.18e-02
	10	1 deep	0.880±0.027	0.933±0.020	3.55e-04
	10	2 non-deep	0.881±0.032	0.942±0.020	1.16e-02
	10	1 non-deep + 1 deep	0.849±0.032	0.874±0.028	3.43e-03
	20	1 non-deep	0.896±0.039	0.977±0.010	2.46e-02
	20	1 deep	0.877±0.022	0.980±0.008	5.26e-05
	20	2 non-deep	0.871±0.032	0.969±0.011	7.74e-04
	20	1 non-deep + 1 deep	0.791±0.037	0.922±0.018	1.43e-04
PR-AUC	10	1 non-deep	0.734±0.059	0.861±0.045	1.62e-04
	10	1 deep	0.689±0.051	0.786±0.051	9.03e-03
	10	2 non-deep	0.731±0.063	0.852±0.047	7.01e-03
	10	1 non-deep + 1 deep	0.721±0.048	0.751±0.051	3.60e-02
	20	1 non-deep	0.720±0.071	0.885±0.045	8.64e-03
	20	1 deep	0.680±0.050	0.951±0.019	2.11e-05
	20	2 non-deep	0.731±0.060	0.914±0.027	5.36e-04
	20	1 non-deep + 1 deep	0.596±0.063	0.813±0.045	1.06e-04
Runtime (CPU Hour)	10	1 non-deep	2.120±0.126	2.668±0.222	3.96e-03
	10	1 deep	2.400±0.149	3.448±0.318	4.85e-04
	10	2 non-deep	2.534±0.129	6.994±0.589	5.85e-08
	10	1 non-deep + 1 deep	3.780±0.253	7.572±0.574	5.18e-08
	20	1 non-deep	22.558±2.063	3.690±0.309	1.75e-08
	20	1 deep	41.541±3.796	4.223±0.407	3.12e-09
	20	2 non-deep	30.488±2.192	6.876±0.478	2.45e-09
	20	1 non-deep + 1 deep	61.081±8.119	9.873±0.673	1.65e-06
Memory (GB)	10	1 non-deep	25.522±2.893	4.091±0.230	1.82e-07
	10	1 deep	23.349±1.351	6.588±0.772	3.93e-10
	10	2 non-deep	41.691±4.001	4.988±0.325	1.01e-08
	10	1 non-deep + 1 deep	44.740±3.229	11.290±1.567	2.80e-09
	20	1 non-deep	31.112±1.206	5.715±0.379	1.03e-13
	20	1 deep	27.511±0.864	9.966±0.997	9.87e-10
	20	2 non-deep	35.099±0.650	6.155±0.371	1.76e-20
	20	1 non-deep + 1 deep	53.147±2.144	13.056±0.860	2.61e-13

Table 6.8 **The performance comparison between PHiMM and DACS with estimated networks as input on 10- and 20-taxon model conditions.** Reticulation scenarios include one non-deep reticulation, one deep reticulation, two non-deep reticulations, and a combination of both non-deep and deep reticulations. The measures include the area under the receiver-operating characteristic curve (ROC-AUC), the area under the precision-recall curve (PR-AUC), runtime (in CPU Hour), and memory usage (in GB). Statistical significance of performance differences between PHiMM and DACS was evaluated using a one-tailed pairwise Student's t-test ($\alpha = 0.05$; $n = 20$). Both PHiMM and DACS methods were run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates (represented as "average value ± SE value").

Measure	**Ntaxa**	**Reticulation Scenario**	**PHiMM**	**DACS**	***p*-value**
ROC-AUC	10	1 non-deep	0.844±0.037	0.865±0.056	3.20e-01
	10	1 deep	0.726±0.044	0.901±0.034	1.21e-04
	10	2 non-deep	0.746±0.051	0.853±0.035	7.99e-03
	10	1 non-deep + 1 deep	0.684±0.039	0.779±0.037	8.29e-03
	20	1 non-deep	0.688±0.051	0.938±0.028	1.47e-04
	20	1 deep	0.622±0.050	0.810±0.048	9.82e-05
	20	2 non-deep	0.698±0.038	0.919±0.026	1.66e-06
	20	1 non-deep + 1 deep	0.607±0.039	0.770±0.043	2.25e-04
PR-AUC	10	1 non-deep	0.586±0.071	0.768±0.075	5.30e-03
	10	1 deep	0.424±0.065	0.772±0.059	2.42e-05
	10	2 non-deep	0.511±0.068	0.700±0.062	1.44e-03
	10	1 non-deep + 1 deep	0.473±0.053	0.629±0.052	3.73e-03
	20	1 non-deep	0.312±0.067	0.795±0.069	2.05e-05
	20	1 deep	0.369±0.062	0.608±0.076	1.00e-03
	20	2 non-deep	0.409±0.061	0.839±0.046	5.38e-08
	20	1 non-deep + 1 deep	0.372±0.055	0.624±0.059	2.39e-05
Runtime (CPU Hour)	10	1 non-deep	23.723±1.916	65.806±5.201	7.31e-11
	10	1 deep	29.972±2.453	73.934±6.266	7.11e-10
	10	2 non-deep	31.926±2.414	123.099±9.094	1.67e-11
	10	1 non-deep + 1 deep	37.327±3.778	141.815±11.325	2.33e-11
	20	1 non-deep	234.787±23.018	81.629±4.147	5.69e-07
	20	1 deep	273.297±23.213	85.071±2.649	4.58e-08
	20	2 non-deep	377.342±25.539	159.871±5.853	3.73e-08
	20	1 non-deep + 1 deep	457.258±59.173	162.829±6.729	1.74e-05
Memory (GB)	10	1 non-deep	18.949±0.214	12.720±0.938	5.40e-08
	10	1 deep	19.091±0.249	14.348±1.147	3.71e-05
	10	2 non-deep	35.675±1.446	16.336±1.378	5.78e-22
	10	1 non-deep + 1 deep	38.317±4.360	15.584±1.445	7.04e-07
	20	1 non-deep	19.103±0.382	14.274±1.169	4.24e-04
	20	1 deep	20.576±0.221	15.173±1.182	2.67e-05
	20	2 non-deep	42.647±5.645	16.422±1.106	1.78e-05
	20	1 non-deep + 1 deep	43.110±5.673	16.628±1.290	8.81e-06

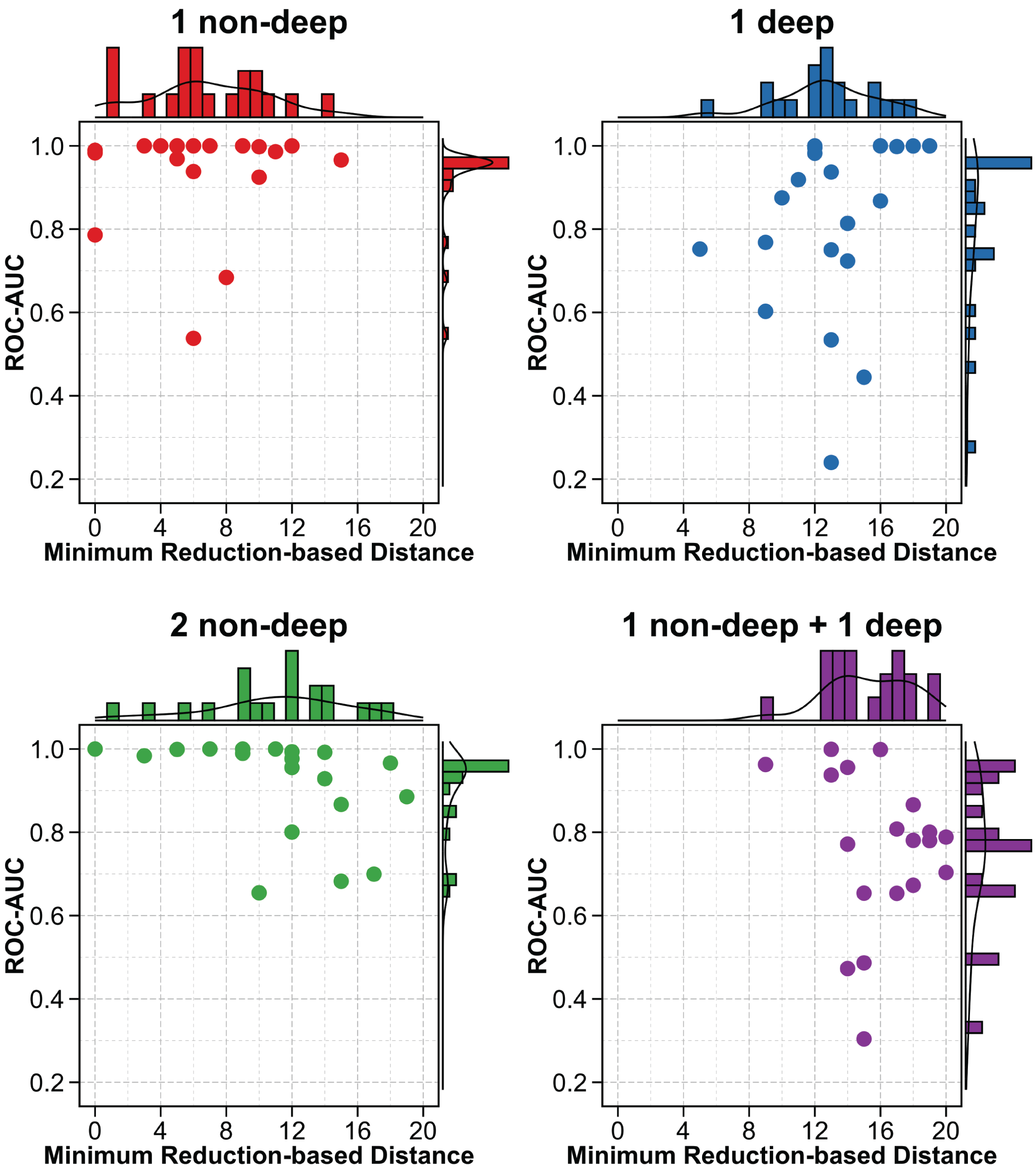


Figure 6.7 **The head-to-head performance comparison of network inference and introgression detection on 20-taxon model conditions.** Reticulation scenarios include one non-deep reticulation, one deep reticulation, two non-deep reticulations, and a combination of both non-deep and deep reticulations. X-axis shows minimum reduction-based distance between true and estimated networks; the Y-axis depicts the area under the receiver-operating characteristic curve (ROC-AUC) of DACS with estimated networks as input. DACS method was run using 20 CPUs.

Table 6.9 **The network inference errors for DACS on 10-, 20-, and 100-taxon model conditions.** Reticulation scenarios include solely non-deep reticulations, solely deep reticulations, and a combination of both non-deep and deep reticulations. The measures include the area under the receiver-operating characteristic curve (ROC-AUC), and the average and minimum reduction-based distance between true and estimated networks (details see Chapter 2.2.3.2). Pearson correlation coefficient (PCC) between each reduction-based distance and ROC-AUC was also calculated. DACS method was run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates (represented as "average value ± SE value").

Ntaxa	**Reticulation Scenario**	**ROC-AUC**	**Minimum reduction-based distance**		**Average eduction-based distance**	
			distance	PCC	distance	PCC
10	1 non-deep	0.865±0.056	2.300±0.726	-0.764	8.217±0.405	-0.482
10	1 deep	0.901±0.034	8.800±0.574	-0.148	10.146±0.278	-0.145
10	2 non-deep	0.853±0.035	6.800±0.756	-0.296	10.667±0.544	-0.299
10	1 non-deep + 1 deep	0.779±0.037	10.100±0.533	-0.136	12.268±0.356	-0.150
20	1 non-deep	0.938±0.028	6.550±0.910	0.088	12.790±0.512	0.083
20	1 deep	0.810±0.048	13.050±0.749	0.240	14.557±0.554	0.229
20	2 non-deep	0.919±0.026	11.150±1.091	-0.419	16.048±0.518	-0.348
20	1 non-deep + 1 deep	0.770±0.043	15.850±0.638	-0.219	18.257±0.475	-0.226
100	1 non-deep	0.857±0.047	30.050±1.725	-0.455	35.193±1.502	-0.326
100	2 non-deep	0.839±0.034	35.800±1.933	-0.253	40.627±1.605	-0.114
100	3 non-deep	0.855±0.031	39.450±2.498	-0.482	44.827±2.221	-0.417
100	4 non-deep	0.827±0.028	45.800±2.207	-0.146	52.520±1.901	-0.131
100	5 non-deep	0.812±0.029	49.950±2.367	-0.452	55.213±2.036	-0.454
100	1 non-deep + 1 deep	0.758±0.033	42.650±1.918	-0.244	45.777±1.649	-0.203
100	2 deep	0.710±0.037	45.400±1.895	-0.223	47.610±1.834	-0.252
100	3 deep	0.717±0.031	51.800±1.325	-0.077	53.120±1.326	-0.169

(0.770) and the minimun reduction-based distances reaches 15.85. The negative PCC (-0.219) further confirms that increased network inference error leads to consistent declines in introgression detection performance.

Overall, DACS demonstrates strong tolerance to moderate network-inference errors and maintains acceptable accuracy even for deep reticulations — ROC-AUC remains above 0.80 for the single deep reticulation case — underscoring its practicality for complex evolutionary histories. Nonetheless, achieving reliable introgression detection in highly reticulated or mixed-depth scenarios still benefits from the most accurate network estimates available.

6.4.1.5 Accuracy of DACS on ultra large datasets

To assess performance on ultra-large datasets, we analyze a 100-taxon model condition under multiple reticulation scenarios. Due to PHiMM's limited scalability, it was not feasible to run PHiMM on this dataset, preventing a direct comparison. Nevertheless, this analysis showcases DACS's capability to scale effectively to datasets of this size. Figure 6.8 and Table 6.10 presents

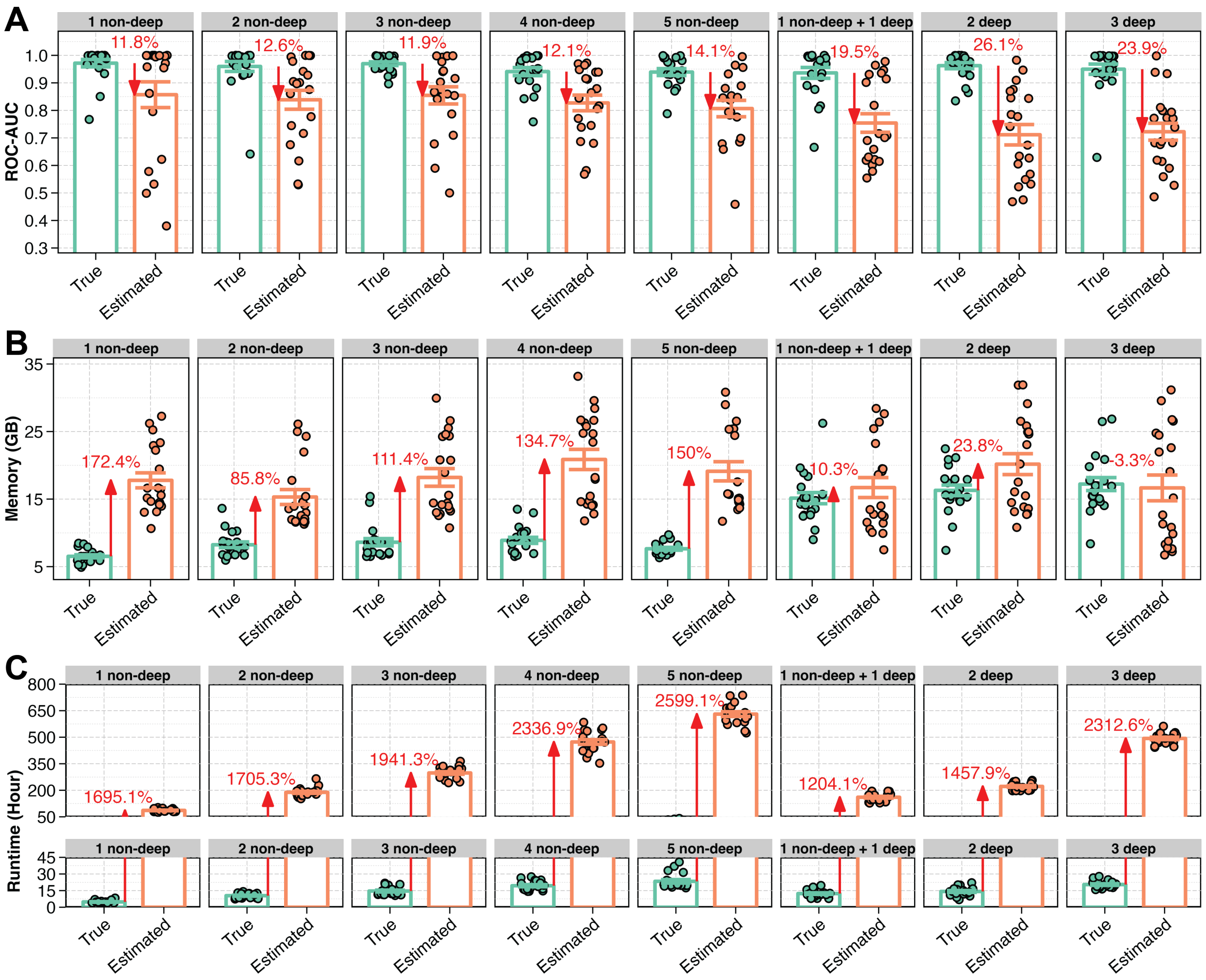


Figure 6.8 **The performance comparison between true network and estimated networks as input for DACS on ultra large datasets with 100 taxa.** The measures include (A) the area under the receiver-operating characteristic curve (ROC-AUC), (B) runtime (in CPU Hour), and (C) memory usage (in GB). Reticulation scenarios include solely non-deep reticulations, solely deep reticulations, and a combination of both non-deep and deep reticulations. DACS method was run using 20 CPUs. The percentages and arrows shown in red indicate the performance reduction with the estimated networks as input compared to true network as input. The average and standard error (SE) are calculated based on 20 replicates.

Table 6.10 **The performance comparison between true network and estimated networks as input for DACS on ultra large datasets with 100 taxa.** Reticulation scenarios include solely non-deep reticulations, solely deep reticulations, and a combination of both non-deep and deep reticulations. The measures include the area under the receiver-operating characteristic curve (ROC-AUC), the area under the precision-recall curve (PR-AUC), runtime (in CPU Hour), and memory usage (in GB). DACS method was run using 20 CPUs. The average and standard error (SE) are calculated based on 20 replicates (represented as "average value ± SE value").

Measure	Ntaxa	Reticulation Scenario	True	Estimated
ROC-AUC	100	1 non-deep	0.972±0.013	0.857±0.047
	100	2 non-deep	0.960±0.018	0.839±0.034
	100	3 non-deep	0.970±0.007	0.855±0.031
	100	4 non-deep	0.941±0.015	0.827±0.028
	100	5 non-deep	0.939±0.012	0.807±0.030
	100	1 non-deep + 1 deep	0.936±0.020	0.758±0.033
	100	2 deep	0.963±0.011	0.710±0.037
	100	3 deep	0.950±0.019	0.717±0.031
PR-AUC	100	1 non-deep	0.916±0.038	0.671±0.093
	100	2 non-deep	0.889±0.044	0.670±0.062
	100	3 non-deep	0.891±0.028	0.713±0.065
	100	4 non-deep	0.878±0.025	0.716±0.044
	100	5 non-deep	0.850±0.032	0.657±0.058
	100	1 non-deep + 1 deep	0.845±0.044	0.576±0.053
	100	2 deep	0.920±0.020	0.465±0.057
	100	3 deep	0.878±0.038	0.539±0.051
Runtime (CPU Hour)	100	1 non-deep	4.798±0.278	86.124±1.732
	100	2 non-deep	10.435±0.458	188.392±5.733
	100	3 non-deep	14.597±0.825	297.975±7.360
	100	4 non-deep	19.411±0.957	473.023±13.245
	100	5 non-deep	23.366±1.575	630.657±13.456
	100	1 non-deep + 1 deep	12.319±0.706	160.655±5.159
	100	2 deep	14.236±0.847	221.789±4.556
	100	3 deep	20.436±0.726	493.052±6.947
Memory (GB)	100	1 non-deep	6.525±0.228	17.776±1.110
	100	2 non-deep	8.239±0.421	15.309±1.105
	100	3 non-deep	8.613±0.538	18.208±1.309
	100	4 non-deep	8.892±0.446	20.868±1.493
	100	5 non-deep	7.646±0.208	19.116±1.422
	100	1 non-deep + 1 deep	15.141±0.825	16.707±1.477
	100	2 deep	16.294±0.783	20.176±1.547
	100	3 deep	17.224±0.959	16.658±1.905

detailed comparisons of the performance of DACS when using a true network versus estimated networks as input, on ultra-large datasets with 100 taxa.

Figure 6.8A displays the ROC-AUC values across various reticulation scenarios, including up to three deep and five non-deep reticulations. DACS maintains high accuracy when true networks are provided, while performance with estimated networks remains strong but shows a moderate decline. The percentage differences, indicated above each pair, range from an 11.8% decrease in the simplest scenario of one non-deep reticulation to a 25.7% decrease in the most complex scenario of three deep reticulations. These results highlight DACS's robustness but also underscore the challenges posed by input uncertainty as network complexity increases.

Figure 6.8B illustrates the memory usage (in GB) for the two types of inputs across the different scenarios. The results show a significant increase in memory consumption when estimated networks are used, with the differences ranging from 10.3% to a staggering 172.4% depending on the scenario. However, DACS remains operable even under the most demanding conditions. The largest increase, observed in the one non-deep reticulation case, reflects the extra overhead required to process less accurate input. Nevertheless, these results affirm that DACS can efficiently manage large-scale inputs even when input quality varies.

Figure 6.8C presents runtime (in CPU hours) required for running DACS method, which increases substantially when using estimated networks — by 1204.1% to 2339%, depending on the scenario. The most extreme increase in runtime is observed in the scenario involving three deep reticulations, where the computational burden more than doubles. This significant increase in runtime underscores the computational challenges posed by estimated networks, particularly in complex scenarios, and suggests that further optimizations or alternative approaches may be necessary to maintain computational efficiency.

Overall, these findings demonstrate that DACS is capable of handling ultra-large datasets with up to 100 taxa, even in the presence of complex reticulation patterns. While using estimated networks incurs costs in accuracy, memory, and runtime, DACS remains applicable and scalable, offering a practical solution for large-scale phylogenetic network analysis. When true networks are not available, researchers can still leverage DACS, with an understanding of the computational trade-offs, particularly in scenarios with high complexity.

6.4.2 Mosquito Data Re-analysis

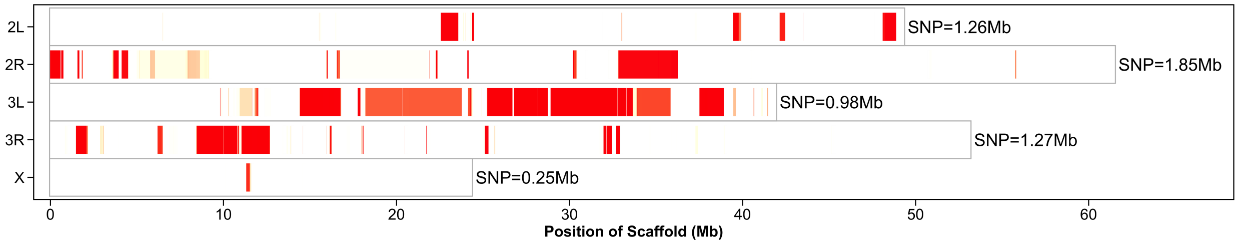


Figure 6.9 **The introgression patterns of DACS on mosquito data.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from *An. merus* to *An. quadriannulatus*.

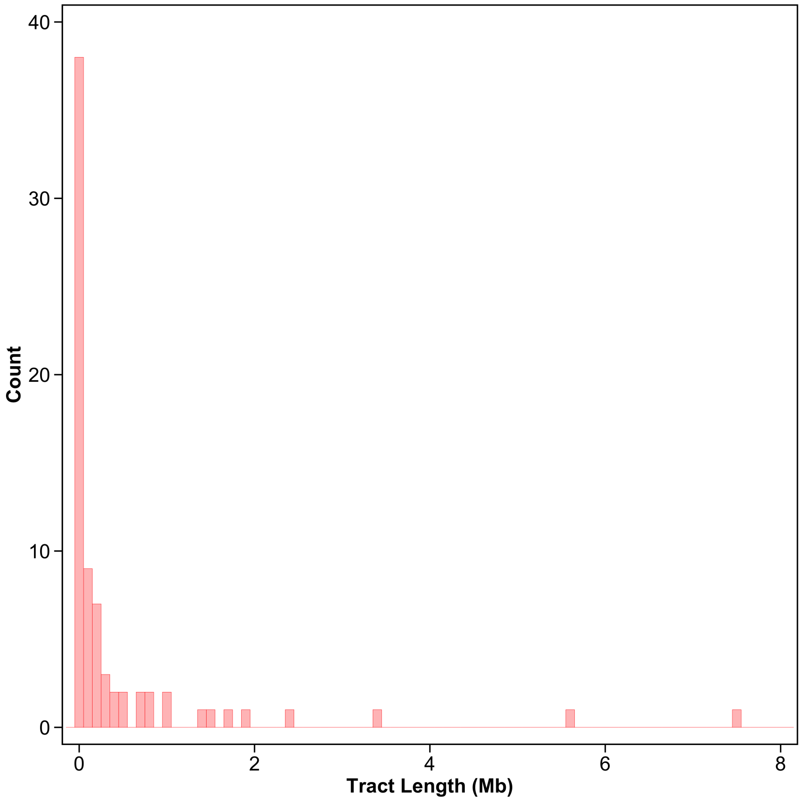


Figure 6.10 **The histograms of introgressed tract lengths by DACS on mosquito data.** Introgressed tract lengths are shown in megabases (Mb). Results are reported for the introgression from *An. merus* to *An. quadriannulatus*.

We applied DACS to mosquito whole-genome data to infer introgressed genomic tracts from *An.*

merus into *An. quadriannulatus*. The results revealed widespread introgression across all major autosomes (Figure 6.9), with a particularly dense signal on chromosome arm 3L. This observation is consistent with prior findings of pervasive autosomal introgression between *An. merus* and *An. quadriannulatus*, especially in the ~22 Mb 3La inversion region (Figure 4 from [152]). In our results, this region is marked by high probabilities of introgressed tracts, suggesting nearly complete replacement of the ancestral *An. quadriannulatus* sequence by its *An. merus* counterpart in this genomic interval.

Figure 6.10 shows the distribution of inferred tract lengths. Most tracts are relatively short (<1 Mb), but a small number of long tracts — up to ~8 Mb — were also observed, indicating potentially recent or large-scale introgression events. These patterns are consistent with a model of both ancient and recent gene flow, with the longer tracts possibly reflecting lower recombination rates, such as those expected in regions with historical inversions like 3La [152]. The extensive and contiguous replacement observed in 3L supports the hypothesis that historical recombination suppression facilitated the maintenance of a large introgressed haplotype.

6.4.3 Darwin's Finch Data Re-analysis

We applied DACS to the 469 largest scaffolds of the Darwin's finch genome, focusing on those exceeding 100 kb in length (Figure 6.11 and Figure B.1-B.8 of Appendix B). This subset captures the majority of genomic content relevant to introgression while excluding highly fragmented or low-information scaffolds. In total, these 469 scaffolds represent a substantial portion of the genome's informative content.

The inferred tract length distribution (Figure 6.12) was highly skewed, with most tracts under 100 kb but a subset extending beyond 500 kb. This heterogeneity suggests both ancient and recent introgression events: longer tracts are likely indicative of more recent gene flow or segments maintained by positive selection; in contrast, more fragmented or shorter tracts, scattered throughout various scaffolds, may reflect older events that have undergone increased recombination or simply represent the lower bound of detection accuracy [158].

To explore potential biological relevance, we examined the genomic context of inferred tracts

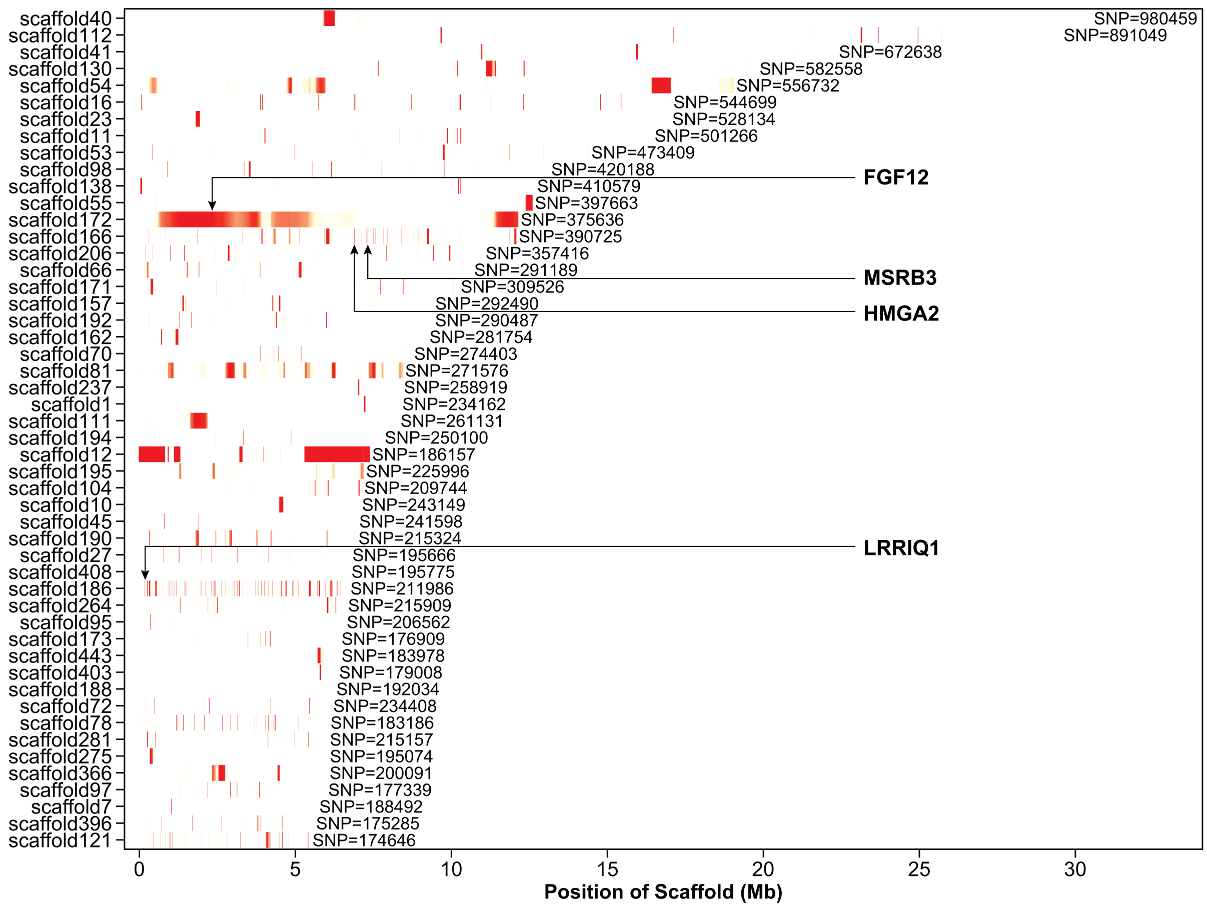


Figure 6.11 **The introgression patterns of DACS on on top 1-50 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from *An. merus* to *An. quadriannulatus*.

across species (Figure 6.11). Several introgressed regions overlapped genes previously implicated in finch morphology [154, 157]. Notably, we detected tracts overlapping 4 genes previously associated with craniofacial development, beak morphology, and/or body size variation in mammals or birds, including FGF12 (fibroblast growth factor 12), LRRIQ1 (leucine-rich repeats and IQ motif containing 1), MSRB3 (methionine-R-sulfoxide reductase B3), and HMGA2 (high mobility AT-hook 2). These signals were especially existed on scaffolds 172, 166, and 186, consistent with previous reports of extensive hybridization and gene flow among Darwin's finches [154, 157]. These findings exemplify how integrative approaches such as DACS can reveal biologically meaningful signals of introgression even in complex, reticulate clades.

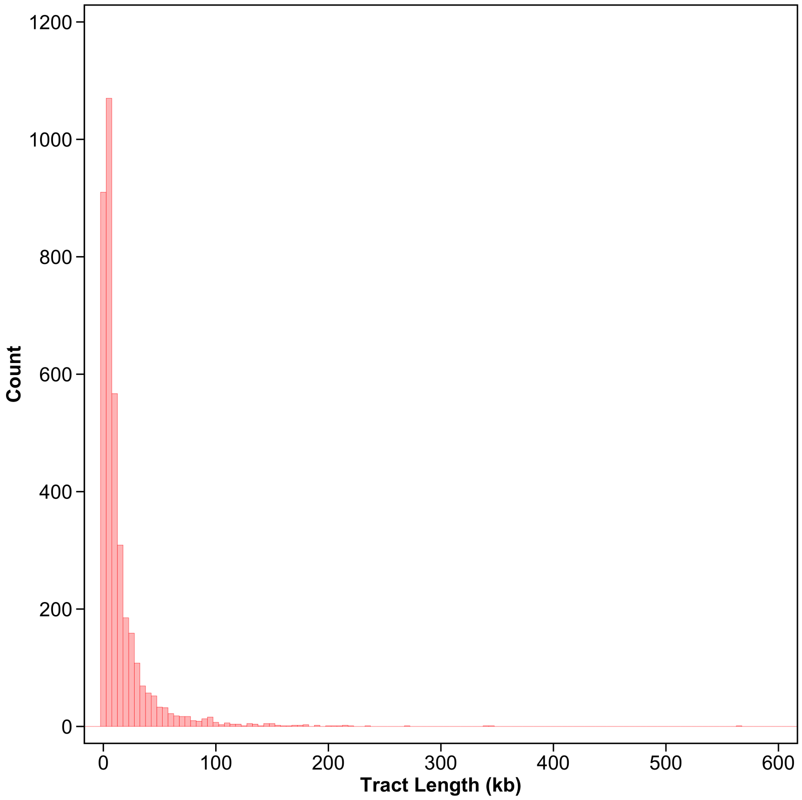


Figure 6.12 **The histograms of introgressed tract lengths by DACS on Darwin's finch data.** Introgressed tract lengths are shown in kilobases (Kb).

6.5 Discussion and Conclusion

In this study, we introduced a novel divide-and-conquer and subsampling-based algorithm, DACS, designed to detect introgression events in large phylogenomic datasets, even when a phylogenetic network is not initially available. Building upon the PHiMM framework, DACS integrates FastNet-based network inference, subsampling, and statistical merging strategies to address the scalability and accuracy limitations commonly encountered in introgression mapping.

Our simulation experiments demonstrated that DACS improves upon PHiMM both in terms of inference accuracy and computational efficiency. Even under challenging reticulation scenarios (e.g., multiple deep or non-deep reticulations), DACS consistently achieved higher ROC-AUC and PR-AUC values than PHiMM. Moreover, it substantially reduced memory usage and runtime, enabling analyses of datasets spanning up to 100 taxa—well beyond the practical limits of standard phylogenetic network inference tools reported in the literature [14].

These performance improvements can be attributed to two main algorithmic advances. First, the divide-and-conquer component selectively focuses on smaller taxa subsets around each reticulation, significantly lowering computational and memory overheads. Second, repeated subsampling further stabilizes the posterior decoding probabilities, reducing biases that can arise from single-run estimates. When coupled with FastNet's ability to infer multiple candidate networks, DACS also reduces the risk of conditioning on an inaccurate topology by weighting the final introgression probabilities according to network quality.

Our simulation results additionally highlight important trade-offs. For instance, while DACS achieves high accuracy when true networks are known, performance diminishes and computational costs rise when networks must be estimated. In ultra-large analyses (e.g., 100 taxa), runtime can increase substantially as the number of reticulation events grows. These findings indicate that users should balance the accuracy gains from broader network exploration (i.e., considering more candidate topologies) against the computational burdens that arise. Further methodological optimizations—such as more efficient sampling or heuristic selection of candidate networks—may help alleviate these challenges.

Beyond simulations, our empirical re-analyses of mosquito data and Darwin's finch genomic data underscore the practical utility of DACS. The method effectively processed real-world, large-scale sequencing data, detecting introgression signals in a biologically rich context where hybridization is known to play a significant role. By accommodating uncertain or unknown phylogenetic networks, DACS offers a robust framework for researchers investigating gene flow across diverse, understudied taxonomic groups.

Despite these advances, some limitations remain. The current pipeline relies on FastNet for network inference, which may itself become computationally expensive for extremely large datasets or highly reticulated evolutionary histories. Moreover, while repeated subsampling helps reduce variance, the choice of sampling size and number of replicates can affect both accuracy and runtime, necessitating careful parameter tuning. Addressing these issues—possibly via adaptive or dynamic sampling strategies—could further improve the method's efficiency.

Future work will concentrate on further optimizing both the subsampling and merging steps, broadening the framework to accommodate diverse reticulation types, and investigating refined strategies for network inference in highly reticulate evolutionary contexts. We anticipate that these developments will establish DACS as a powerful resource for evolutionary biologists and genomics researchers seeking to elucidate the dynamics of introgression across increasingly large and complex datasets.

CHAPTER 7

APPLICATION OF DACS-BASED METHOD TO HORIZONTAL GENE TRANSFER DETECTION IN METAGENOMICS

7.1 Introduction

In previous studies, we have developed DACS, an introgression detection tool that significantly enhances the scalability of introgression analysis while preserving high accuracy in inference. Building on this advancement, we propose to adapt and improve the DACS tool for Horizontal Gene Transfer (HGT) detection in microorganisms. This extension represents a more complex challenge due to the involvement of diverse taxa and intricate reticulations in HGT events, which require more robust methods to handle the increased biological complexity.

The advent of high-throughput metagenomic technologies has revolutionized biological research by generating vast amounts of sequencing data, with data volumes expanding exponentially. As a result, there is an urgent need to apply introgression and HGT detection methodologies to metagenomic datasets. However, this task is complicated by the inherent challenges of metagenomic data, including noisy and partial sequences. While shotgun read data can be directly employed for tasks such as taxon identification and functional annotation, the majority of metagenomic workflows focus on reconstructing metagenome-assembled genomes (MAGs) through processes such as filtering, assembly, and binning. MAGs, which exhibit high completeness, are considered close representations of actual microbial genomes and can be utilized for downstream genomic analyses.

Nevertheless, metagenomic datasets are often derived from heterogeneous microbial communities, sometimes containing over 10,000 distinct species [16]. This complexity, coupled with noisy and incomplete sequence data, presents significant challenges when applying traditional genomics-based methods to metagenomics. Existing tools and methods may perform suboptimally due to these quality issues. Furthermore, many phylogenetic-based approaches, which have proven valuable in genomic studies, have yet to be fully adapted or tested in the metagenomics domain.

In this study, we address the challenges associated with HGT detection in metagenomic datasets by applying our proposed method, DACS, to metagenomic studies. In this study, we aim to address the challenges associated with HGT detection in metagenomic datasets by applying our proposed method, DACS, to these data. After assembling sequencing reads into MAGs, we use DACS to identify reticulate evolutionary events, such as introgression and HGT, even under challenging model conditions characterized by noise, incomplete data, and large numbers of taxa. We also compare the performance of DACS in metagenomics with its application to genomics, discussing differences in terms of input requirements, accuracy, scalability, and the specific conditions under which each method is most applicable.

7.2 Background

7.2.1 Metagenomics Definition

Traditionally, microbiology has relied on cultivating pure cultures within laboratory settings, allowing for the sequencing of individual organisms' genomes one at a time. However, this approach only scratches the surface, as up to 99% of microorganisms elude cultivation, leaving us ignorant of their very existence. This limitation in cultivation has distorted our perception of microbial diversity and restricted our understanding of the vast microbial realm.

In contrast, metagenomics presents a refreshing perspective by directly sampling the genome sequences of entire communities of organisms within their natural habitats, thus circumventing the necessity for isolating and cultivating individual organisms in the lab. Metagenomics offers a comparatively unbiased outlook not only on the community's structure, encompassing species richness and distribution, but also on the functional and metabolic potential of the community as a whole.

Various habitats or environments, such as soil, sea, mud, forests, space, and even the human body, harbor distinct communities of microorganisms, comprising bacteria, viruses/phages, microbial eukaryotes (e.g., yeast), as well as worms (e.g., helminths and nematodes).

7.2.2 Metagenomics Analysis

The field of metagenomics raises two fundamental questions [159]:

1. Who is there? — This question pertains to identifying the various species of microorganisms

present in the sample.

2. What are they doing? — Here, the objective is to discern the functions and behaviors of the identified microorganisms.

To address these questions, numerous strategies can be employed in the analysis of metagenomics shotgun data [160] (Figure 7.1, Table 7.1, and Table 7.2).

7.2.2.1 Quality filtering

A customary initial step involves running a range of computational tools for quality control or filtering, including trimming sequencing adapters, discarding short and low-quality reads, and eliminating reads with low-quality extremes or "N" characters based on quality.

Following quality control, the reads can be subjected to two potential paths: either assembled into longer contiguous sequences known as contigs, or directly employed in taxonomic classifiers or functional annotations (see Figure 7.1).

7.2.2.2 Assembly

Assembly, the process of merging collinear metagenomic reads from the same genome into contiguous sequences called contigs, proves valuable in generating longer sequences, thereby simplifying bioinformatic analysis compared to working with unassembled short metagenomic reads.

Two primary approaches to assembly are available:

1. *De novo* assembly: This method involves joining sequenced fragments to generate contigs without relying on a previously sequenced reference genome. The De Bruijn graph method stands as the most popular *De novo* assembly paradigm and finds implementation in various assemblers.
2. Comparative assembly: This approach utilizes a previously sequenced, closely related organism as a guide to aid the assembly process.

Following assembly, the quality of the obtained contigs can be assessed to facilitate the evaluation and comparison of metagenome data, relying on alignments to close references as a basis for assessment. The resulting contigs obtained after assembly can serve multiple purposes, directly contributing to gene prediction and functional annotation or playing a crucial role in the binning process.

7.2.2.3 Binning

Binning algorithms undertake the task of grouping contigs or scaffolds that originate from the same or closely related organisms. Once these contigs are clustered into bins, subsequent analyses, such as taxonomic profiling and functional annotation, are performed on these bins rather than individual contigs. Binning has proven to be a powerful approach capable of clustering contigs even from rare species. Moreover, it has demonstrated the ability to recover draft genomes from previously uncultivated bacteria [161].

There are two primary binning methods: taxonomy-dependent methods (supervised methods) and taxonomy-independent methods (unsupervised methods).

1. Taxonomy-dependent methods (Supervised methods): These methods utilize known reference genomes or characteristics extracted directly from the nucleotide sequences (e.g., oligonucleotide frequencies, GC content) to map the contigs. They are based on aligning metagenomic sequences to a reference. However, a drawback of taxonomy-dependent methods is the limited number of available sequenced genomes in current databases, which can lead to challenges in the alignment process and result in longer computing times.
2. Taxonomy-independent methods (Unsupervised methods): In contrast, taxonomy-independent methods do not rely on a reference genome and have become more popular due to their versatility. These methods assume that contigs belonging to the same genome should exhibit similar abundance and/or oligonucleotide composition within the same sample.

Following the binning process, reads can be mapped back to the bins, allowing each bin to be reassembled. If the binning is successful, this step may produce longer contigs, enhancing the overall quality of the assembled genomes [160]. Once the bins have been filtered for contamination and completeness, they are referred to as metagenome-assembled genomes (MAGs) [162].

7.2.2.4 Gene prediction or annotation

Gene prediction determines which metagenomic reads contain coding sequences. Once identified, coding sequences can be functionally annotated. Gene prediction can be conducted on assembled or unassembled metagenomic sequences. One of the most straight forward ways of identifying coding sequenced in a metagenome is to map metagenomic reads or contigs to a database of gene sequences. *De novo* gene prediction, on the other hand, can potentially identify novel genes. Here, gene prediction models, which are trained by evaluating various properties of microbial genes (e.g., length, codon usage, GC bias), are used to assess whether a metagenomic read or contig contains a gene without relying on sequence similarity to a reference database.

7.2.2.5 Functional annotation

Functional annotation in metagenomics serves the purpose of addressing the question "what are they doing". By examining the collective functions encoded in the genomes of the microorganisms within a community, metagenomes shed light on the physiological activities of that community.

Once genome assembly, binning, and gene annotation are completed, numerous tools are available to carry out functional annotations. The conventional approach to identifying gene function involves similarity searches using well-established tools like BLAST. Predicted peptides that are classified as homologs of a protein family are annotated with the family's function. Conducting this analysis across all reads results in a community functional diversity profile. To refine the predicted function, it is essential to utilize comprehensive databases (Table 7.1).

7.2.2.6 Taxonomic profiling

In metagenomics, the question of "who is there" is addressed through taxonomic profiling, providing insights into the taxonomic composition of each analyzed sample. Taxonomic profiling not only identifies the detected taxa in a sample but also estimates their relative abundances and various diversity indices.

Traditionally, taxonomic profiling has been performed using marker gene analysis, a widely-adopted approach that relies on specific genes conserved across related organisms to infer taxonomic identities. This technique will be described in greater detail later.

Several software tools are available to carry out taxonomic profiling using either MAGs or contigs. These tools aid in the taxonomic classification and quantification of the microbial communities present in the samples (Table 7.1). In recent developments, methods have emerged to enable genome profiling of MAGs at the strain level, providing more detailed insights into the diversity within specific taxa. These tools offer a more detailed and precise understanding of the taxonomic structure and strain-level diversity within microbial communities (Table 7.1).

7.2.2.7 Assembly-free approach

Assembled genomes offer evident advantages for subsequent functional analyses. Nonetheless, obtaining accurate assemblies, bins, and even MAGs when dealing with metagenomic samples remains a challenging task. Consequently, a considerable portion of the reads acquired from metagenomic samples will remain non-assembled. Hence, approaches have been developed to enable the direct analysis of raw metagenomic data for both taxonomic classification and functional assignments.

Marker gene analysis represents one of the most straightforward and computationally efficient methods for assessing the taxonomic diversity of a metagenome. This approach involves comparing metagenomic reads to a database of taxonomically informative gene families, commonly known as marker genes. By identifying metagenomic reads that exhibit homology to these marker genes, the sequences are taxonomically annotated through sequence or phylogenetic similarity to the marker gene database. The frequently employed marker genes are those found in ribosomal RNA (rRNA) genes and protein-coding genes, which typically exist as single copies and are widespread across microbial genomes. Notably, the 16S rRNA, 18S rRNA, and ITS genes are particularly esteemed as excellent marker genes for bacteria, eukaryotes, and fungi, respectively.

Various methods have been developed for assesbly-free taxonomic classification in metagenomics, including both reference-based and reference-free approaches (Table 7.2). Reference-based methods rely on reference databases for classification. On the other hand, reference-free methods do not depend on reference databases and offer alternative approaches for taxonomic profiling in metagenomic samples.

When working directly with reads for annotation, conventional tools like BLAST often suffer from slowness due to the substantial amount of data being processed. As a result, novel methods employing optimized strategies have emerged to expedite the process. Additionally, several pipelines have been established to integrate some of these tools, enabling direct annotation from raw sequencing data. These pipelines streamline the annotation process, providing efficient ways to analyze metagenomic data directly from raw reads (Table 7.2).

7.3 Methods

7.3.1 Standalone DACS pipeline

DACS is an advanced algorithm specifically designed to detect introgression events in large phylogenomic datasets efficiently by integrating a divide-and-conquer approach with a robust subsampling strategy. Unlike the standard PHiMM pipeline, which requires a predefined network as input, DACS leverages FastNet's network-inference capabilities for large-scale analyses. DACS aims to address two fundamental challenges in introgression detection. First, DACS reduces the computational overhead associated with analyzing a large number of taxa simultaneously. Second, rather than relying on a single, potentially erroneous phylogenetic network, DACS integrates over multiple plausible topologies to minimize errors and improve the robustness of the results. Benchmark analyses have demonstrated that DACS not only scales effectively to handle dozens or even hundreds of taxa but also maintains high accuracy in identifying introgressed genomic regions.

To apply DACS to metagenomic datasets, the process begins after assembling sequencing reads into metagenome-assembled genomes (MAGs) (as described in Chapter 7.4). The DACS pipeline is then run to perform introgression mapping on these MAGs under different model conditions, which account for noise, incomplete data, and large numbers of taxa.

The divide-and-conquer and subsampling procedures in DACS require the user to set the subsampling size and the number of subsampling replicates. By default, we use a subsampling size of 7 and perform 10 subsampling replicates. During the subproblem-solving phase of DACS, the PHiMM tool is used for introgression analysis. PHiMM runs with its default settings, which include

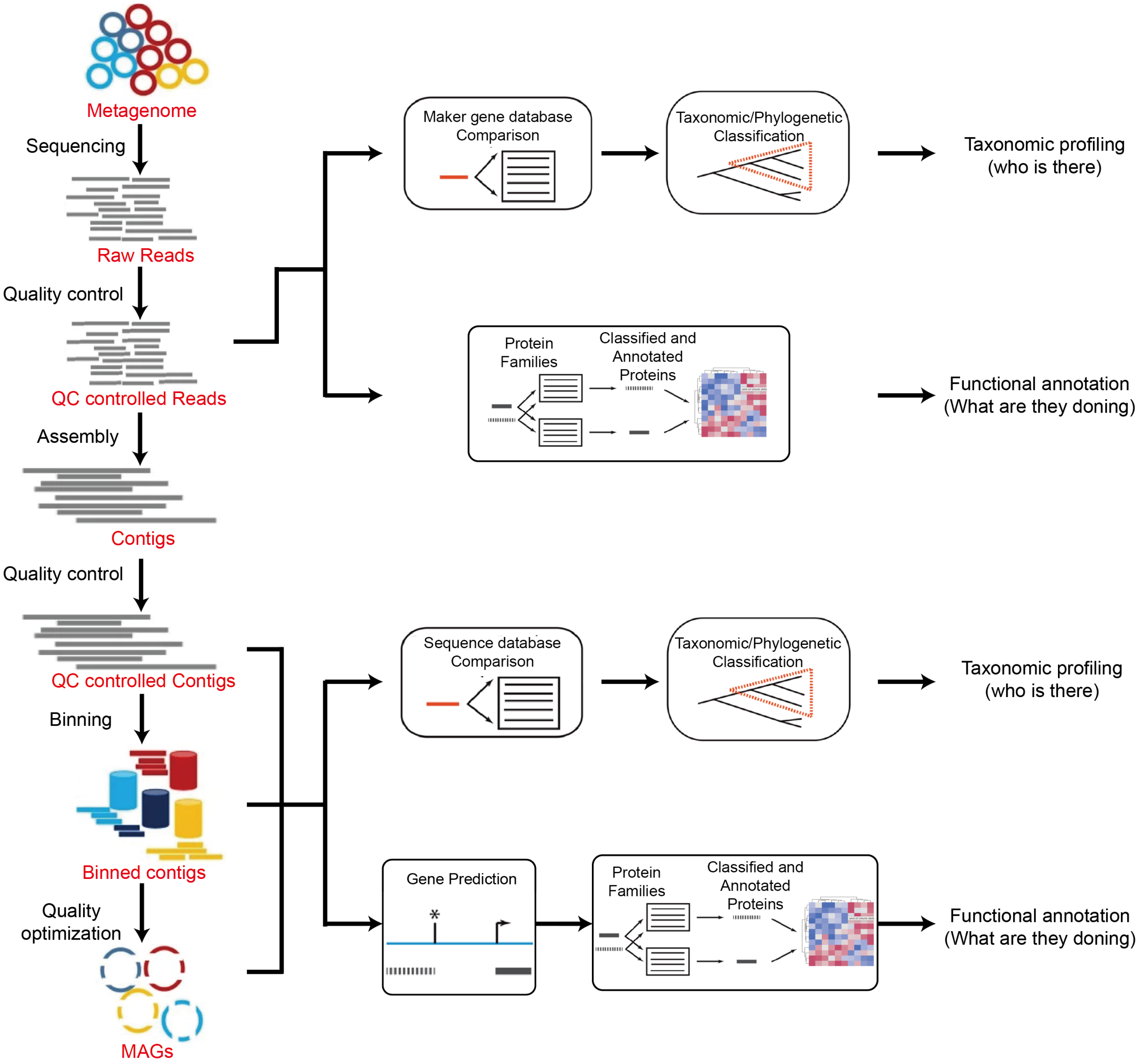


Figure 7.1 **Common analysis procedures for metagenomics data.** First, raw reads undergo quality control (QC) to remove low-quality bases and contaminants. The QC-controlled reads are then assembled into longer contiguous sequences (contigs), which may be further evaluated and filtered. Next, binned contigs are formed by grouping contigs that likely originate from the same organism, yielding metagenome-assembled genomes (MAGs). Two major downstream analyses follow: **Taxonomic Profiling ("Who is there?")** involves comparing reads or contigs to reference databases—either marker-gene based or more comprehensive sequence repositories—to classify the taxa present and infer their evolutionary relationships. **Functional Annotation ("What are they doing?")** entails gene prediction and mapping predicted proteins to known families and pathways, thereby identifying the biological roles and metabolic capabilities of the microbial community.

Table 7.1 **Common assembly-based tools for metagenomics data.** This table compiles a broad range of bioinformatics software used throughout the metagenomics workflow. Tools are organized by major analytical steps: (1) *Quality filtering*—covering low-quality read trimming, contig assessment, and bin validation; (2) *Assembly*—including traditional *de novo* strategies and ensemble methods; (3) *Binning*—encompassing taxonomy-dependent (supervised), taxonomy-independent (unsupervised), and ensemble approaches; (4) *Gene prediction*—distinguishing between prokaryotic and eukaryotic prediction tools; (5) *Functional annotation*—highlighting commonly used databases, aligners, and tools; and (6) *Taxonomic profiling*—covering broad (basic) as well as strain-level taxonomic resolution. References are provided for each method/tool to offer further guidance on specific algorithms, use cases, implementation details, performance characteristics, and potential applications.

Category	Subcategory	Methods & Reference
Quality filtering	**Quality filtering for reads**	PRINSEQ [163], Trimmomatic [164], FastQC [165], SeqKit [166], Cutadapt [167], BBDuk [168], Sickle (v.1.33) [169]
	Quality filtering for contigs	MetaQUAST [170]
	Quality filtering for bins	CheckM [171], AMBER [172]
Assembly	***De novo* assembly (Eg. De Bruijn graph methods)**	Minia [173], MetaVelvet [174], MetaVelvet-SL [175], Ray Meta [176], IDBA [177], IDBA-UD [178], MetaSPAdes [179], Megahit [180]
	Ensemble assembly	MetAMOS [181], MeGAMerge [182], GAM-NGS [183]
Binning	**Taxonomy-dependent or Supervised**	CARMA3 [184], MetaPhyler [185], SOrt-ITEMS [186], PhyloPythiaS+ [187], Phymm/PhymmBL [188], MG-RAST v.4 [189], MEGAN6 [190], TACOA [191], IMG/M v.5.0 [192]
	Taxonomy-independent or Unsupervised	Metawatt [193], AbundanceBin [194], LikelyBin [195], MBBC [196], Canopy [197], MetaCluster4 [198], MaxBin2 [199], MetaBAT2 [200], CONCOCT [201], COCACOLA [202], SCIMM [203], CompostBin [204]
	Ensemble binning	Binning-refiner [205], DAS Tool [206], ICoVeR [207], MetaWRAP [208]
Gene prediction	**Prokaryotes**	MetaGeneMark [209], MetageneAnnotator [210], Glimmer-MG [211], JGI (Joint Genome Institute) [212], MetaProdigal [213], FragGeneScan [214]
	Eukaryotes	GeneMark.hmm [215, 216], AUGUSTUS [217], Gnomon [218], SNAP [219]
Functional annotation	**Aligners**	BLAST [220]
	Databases	Pfam [221], Interpro [222], PRIAM [223], KEGG [224], Metacyc [225], NCBI taxonomy [226]
	Tools	Prokka [227], DFAST [228], NCBI's PGAP [229], MetaErg [230]
Taxonomic profiling	**Basic**	MEGAN6 [190], DIAMOND [231], MAGpy [232], MG-RAST v.4 [189, 233]
	Strain level	MetaMLST [234], StrainPhlAn [235], DESMAN [236], MetaSVN [237]

Table 7.2 **Common assembly-free tools for metagenomics data.** This table summarizes frequently used software tools that bypass the need for assembly when analyzing metagenomic samples. Tools are grouped into two main categories: (1) *Functional annotation*, which leverages fast aligners (e.g., USEARCH, DIAMOND) and integrated pipelines (e.g., HUMAnN2, MGS-Fast) to directly map reads to known databases for functional insights; and (2) *Taxonomic profiling*, subdivided into *reference-based* methods (e.g., Kraken2, Centrifuge) that match reads to comprehensive taxonomic databases, and *reference-free* methods (e.g., MetaPhlAn, PhymmBL) that infer taxonomy without relying on external reference genomes. References are provided for each method/tool to offer further guidance on specific algorithms, use cases, implementation details, performance characteristics, and potential applications.

Category		**Methods**	**Reference**
Functional annotation	**Aligners**	USEARCH	[238]
		BLAT5	[239]
		RAPSearch2	[240]
		DIAMOND	[231]
		PALADIN	[241]
		GRASP2	[242]
	Tools	FUN4ME	[243]
		MOCAT2	[244]
		MGS-Fast	[245]
		HUMAnN2	[246]
		MetaStorm	[247]
		MG-RASTv.4	[189]
		IMG/M v.5.0	[192]
Taxonomic profiling	**Reference-based**	MG-RASTv.4	[189]
		MEGAN6	[190]
		CARMA3	[184]
		Kraken	[248]
		Clark	[249]
		Kraken2	[250]
		Taxonomer	[251]
		Centrifuge	[252]
		Kaiju	[253]
		taxMaps	[254]
		Taxator-tk	[255]
	Reference-free	PhylopythiaS+	[187]
		PhymmBL	[188]
		MetaPhlAn	[256]
		MetaPhlAn2	[257]

300 iterations for model parameter learning and 10 seperate runs for robustness. Additionally, users need to assign a gene tree truncation size, denoted as k_n, for the hidden Markov model (HMM) in PHiMM. Here, we set $k_n = 15$, which corresponds to the default setting in PHiMM.

First of all, to make a fair comparison across different metagenomic scenarios and avoid potential confounding effects from network uncertainty, we opt to use the true phylogenetic network as input for DACS (obtained from Chapter 6.3.1). The required inputs for DACS thus include the aligned MAG sequences A, where K represents the number of sequences and L is the number of alignment columns, as well as a true phylogenetic network Ψ on the K taxa. The input MAG alignment A is constructed based on steps described in Chapter 7.4, while the true phylogenetic network Ψ is directly from the DACS simulation studies in Chapter 6.3.1.

However, the ground-truth phylogenetic network is rarely available a priori in real-world data. One of the key advantages of DACS is its ability to bypass the need for a predefined network by employing FastNet [11] to infer one or more candidate phylogenetic networks from the input sequence alignments. Therefore, we also run experiments with estimated phylogenetic networks by FastNet as input for DACS (see steps described in Chapter 6.2.2.2). The required inputs for DACS thus include the aligned MAG sequences A, where K represents the number of sequences and L is the number of alignment columns, as well as a the number of reticulations R (obtained from dataset construction step in Chapter 7.4).

The output of DACS is a sequence of modified posterior decoding probabilities for the columns of the input MAG alignment. These probabilities are crucial for identifying potential introgression events and understanding the evolutionary dynamics within the dataset.

7.4 Materials

7.4.1 Simulation of reads

Preliminary results presented in this Chapter are derived entirely from fully synthetic reference genomes; the 100-taxon datasets used for DACS analyses (see Chapter 6.3.1) do not correspond to any real-world species. These datasets were generated in silico solely to explore method behavior under extreme evolutionary complexity and should be interpreted as proof-of-concept rather than empirically validated findings.

First, we prepared reference genomes for simulating raw reads from two 100-taxon datasets used in the DACS simulation analysis (see Chapter 6.3.1). Each dataset includes 100 distinct genomes corresponding to different taxa or species, and each dataset consists of 20 replicates. The first dataset was constructed under model condition involving 3 deep reticulations, and the second dataset was constructed under model condition with 5 non-deep reticulations. These reticulation conditions represent highly complex evolutionary scenarios, making them suitable for mimicking real-world data. As a result, they present a challenging and robust foundation for our downstream analyses.

For the 100 reference genomes from each replicate of each dataset, we simulated metagenomic sequence reads using wgsim (available at https://github.com/lh3/wgsim), a widely-used simulator initially developed as part of SAMtools [156]. This tool generates error-containing paired-end reads. We simulated both short reads with a length of 250 bp as well as long reads with a length of 1000 bp, applying the default Illumina sequencing noise with a 2% error rate. All other parameters were left at their default settings.

Since the reconstruction of genes involved in HGT is highly sensitive to sequencing depth and the choice of assembler [258], we set the wgsim parameters to produce target DNA fragments by randomly sampling from the reference sequences. We generated reads with varying coverage depth levels: 5x, 10x, 20x, 30x, 50x, and 100x, to explore different levels of genetic divergence and their effects on metagenomic assembly and downstream introgression/HGT detection. Coverage depth (or read depth, sequencing depth) refers to the number of times a specific base (nucleotide) in the DNA is read during the sequencing process.

Thus, we simulated both 250-bp and 1000-bp paired-end reads with coverage depths of 5x, 10x, 20x, 30x, 50x, and 100x for the two 100-taxon datasets under 3 deep reticulation condition or 5 non-deep reticulation condition.

7.4.2 Metagenomic assembly and MAG construction

For each simulated read dataset, BBDuk [168] was used to remove Illumina adapters as well as PhiX and other Illumina trace contaminants. After adapter removal, reads were trimmed using Sickle (v.1.33) [169] with a default quality threshold of 20 (quality type set to Sanger, which corresponds to CASAVA v.1.8 or higher). Each physical filter was treated as an independent sample. Specifically, metagenomic reads from a single filter were assembled together, rather than co-assembling reads from all filters or size fractions.

Assembly was performed using Megahit (v.1.2.9) [180] with the default parameters. with a k-mer size cascade of 21, 29, 39, 59, 79, 99, 119, 141, 159, 179, 199, 219, 239, and 255 to assemble quality controlled reads into contigs. Assembled contigs were then scaffolded using the scaffolding function from IDBA-UD [178]. Only scaffolds greater than 1 kb in length were retained for downstream gene prediction and MAG construction.

These scaffolds were aligned to corresponding reference genomes with BLASTN [220] using the default setting. The BLASTN results were then filtered with an E-value cut-off of $< 10^{-3}$, an identity cut-off of > 90% and an aligned length cut-off of > 100 bp for the scaffolds. Note that scaffolds can only be mapped to one reference genome (i.e., one species) with best E-value. These stringent criteria ensured the high quality and relevance of the identified genes.

Subsequently, we derived MAGs by extracting all aligned scaffold segments corresponding to reference genomes for simulated read datasets with different read length and coverage levels. Given that our reference genomes were well-aligned, the resulting MAGs were also well-aligned, with gaps representing missing DNA sites. To prepare the MAGs for input into the DACS method, we removed these gap regions, as DACS requires a complete alignment with no gaps.

7.5 Results

Figure 7.2, Figure 7.3, and Table 7.3 summarize the effects of read length (250-bp vs. 1000-bp) and read sequencing depth (5×, 10×, 20×, 30×, 50×, and 100×) on the accuracy of detecting reticulate events using DACS. Performance was evaluated across two simulated 100-taxon metagenomic datasets, each modeled under either 5 non-deep or 3 deep reticulations, using both true and estimated phylogenetic networks as input.

Across all conditions, both the area under the receiver-operating characteristic curve (ROC-AUC) and the completeness of the reconstructed metagenome-assembled genomes (MAG coverage) improved steadily with increasing read coverage depth, regardless of read length or reticulation complexity. At low read depths (5×–10×), assemblies from short reads (250-bp) generally produced slightly higher ROC-AUC values than those from long reads (1000-bp). For example, in the 5 non-deep reticulation scenario with true network as input (Figure 7.2A), the mean ROC-AUC at 5x read depth was 0.926 for 250-bp reads and 0.915 for 1000-bp reads. However, as read depth increased beyond 20×, the differences between the two read lengths diminished, with both read-length assemblies converging on ROC-AUC values around 0.933-0.935. By 100x read depth, short- and long-read assemblies performed nearly identically (both at approximately 0.933-0.934 ROC-AUC), and both approached the "True" reference condition (0.939), in which DACS was run directly on the original reference genomes without any assembly. This same trend was reflected in MAG coverage, which neared 100% for all assemblies once read coverage depth reached 50x and above.

When estimated networks were used instead of true networks, a consistent drop in ROC-AUC was observed across all settings (Figure 7.3), highlighting DACS's sensitivity to phylogenetic inference accuracy. For the 5 non-deep reticulation model at 5× depth, ROC-AUC dropped to 0.760 for 250-bp reads and 0.749 for 1000-bp reads. The decline was more pronounced in the 3 deep reticulation model, where ROC-AUC at 5× depth fell to 0.671 (250-bp) and 0.661 (1000-bp). Although performance improved with increasing depth, the gap between true and estimated network inputs remained evident. Even at 100× coverage, the ROC-AUC under estimated networks remained ~0.717–0.807 — substantially lower than ~0.939–0.950 observed with true networks.

The 3 deep reticulation model posed greater challenges overall, with slightly reduced ROC-AUC and MAG coverage at low to moderate depths compared to the 5 non-deep reticulation model. This suggests that deeper reticulate events are more difficult to resolve, particularly when assemblies are incomplete or networks are inferred. Nonetheless, once coverage reached 30× or more, DACS was able to recover high-quality MAGs and achieve reliable HGT detection, even in the presence of complex evolutionary histories.

Overall, these findings demonstrate that DACS maintains robust performance in identifying HGT/introgression events under both shallow and deep reticulation models as well as under both true and estimated networks as input once moderate (≥20-30x) coverage depths are reached. The 250-bp assemblies often offered slightly better early-stage performance at 5-10x coverage, because the long-read assembly is more difficult and challenging. But the gap between short- and long-read assemblies closed quickly with increasing coverage, indicating that both read lengths can achieve accurate HGT detection at moderate to high coverage depths.

7.6 Discussion and Conclusion

In this study, we investigated the performance of our DACS-based approach for detecting horizontal gene transfer (HGT) in metagenomic datasets under a variety of read lengths (250-bp vs. 1000-bp), coverage depths (5x, 10x, 20x, 30x, 50x, and 100x), input phylogenies (true vs. estimated networks) and reticulation complexities (3 deep vs. 5 non-deep reticulations). Our results show that coverage depth is a crucial determinant of both reconstructed MAG completeness and accuracy in detecting reticulate events. While short reads exhibited slightly better in ROC-AUC at low coverage (5-10x), the gap between short- and long-read assemblies diminished as coverage increased, and both achieved near-reference-level accuracy at coverage depths of 20-30x and higher.

One potential explanation for the early lead of short reads under low coverage is that shorter reads may assemble more robustly when read coverage is sparse, possibly because the fragmented sequence data are less prone to errors in contig assembly. Conversely, long-read assemblies require sufficient coverage to resolve repetitive regions, complex structural variations, and other complications that can prevent accurate contig construction. Once coverage exceeds a certain threshold (in this study, about 20-30x), both short and long reads form sufficiently complete and accurate assemblies, yielding nearly indistinguishable performance in HGT detection.

We also observed that DACS consistently achieved higher accuracy when using true networks as input, highlighting the method's sensitivity to the quality of phylogenetic inference. Even at high

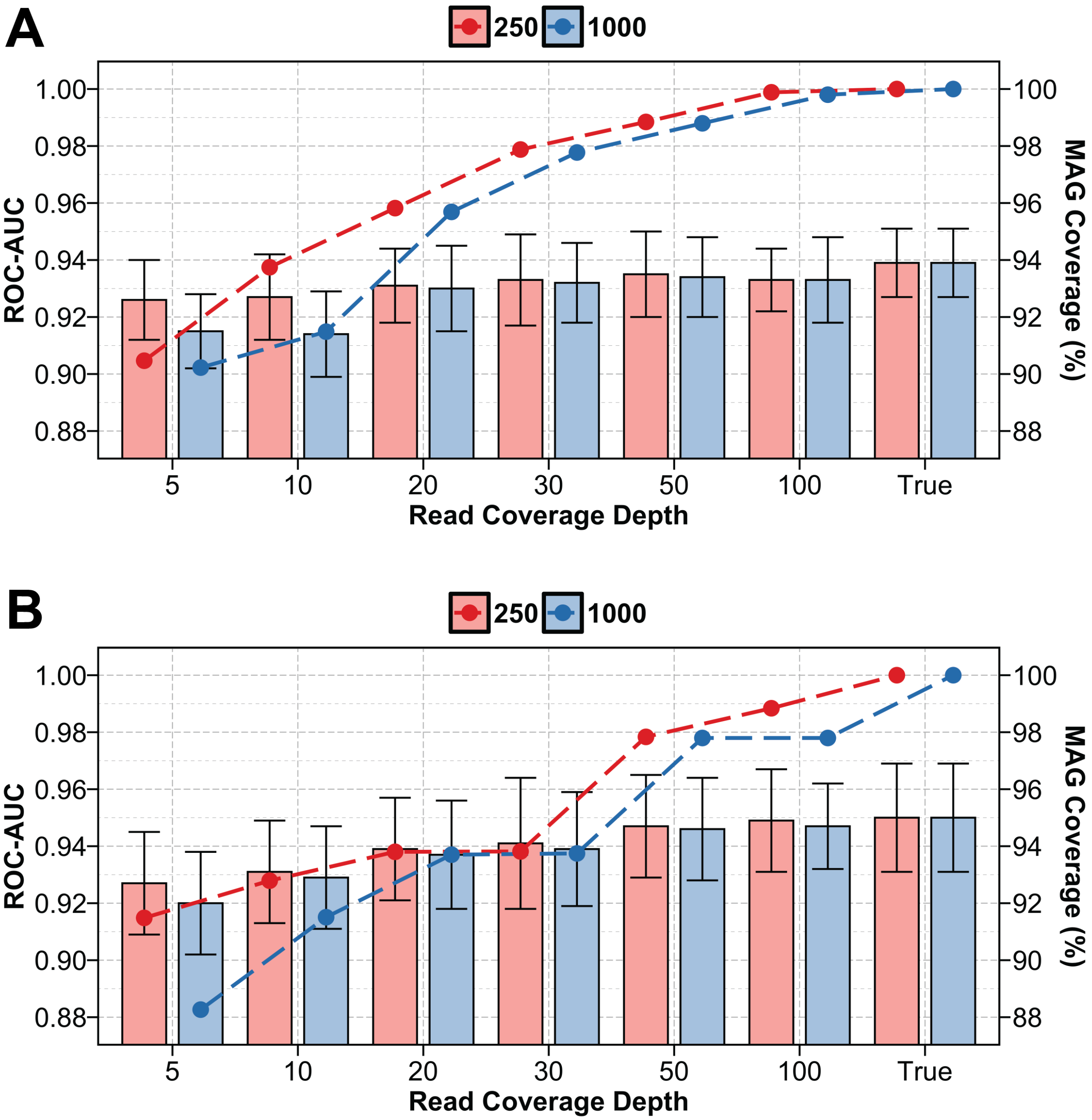


Figure 7.2 **The comparison of different read lengths of 250-bp and 1000-bp and coverage depths of 5x, 10x, 20x, 30x, 50x, and 100x when running DACS with true network as input on both 100-taxon datasets under model conditions with (A) 5 non-deep reticulations and (B) 3 deep reticulations.** "True" means results directly on the reference genomes. ROC-AUC (left y-axis, bars) and MAG coverage (%) (right y-axis, dash lines) as functions of read coverage depths. Red bars/lines represent for read length of 250-bp, and blue bars/lines for read length of 1000-bp.

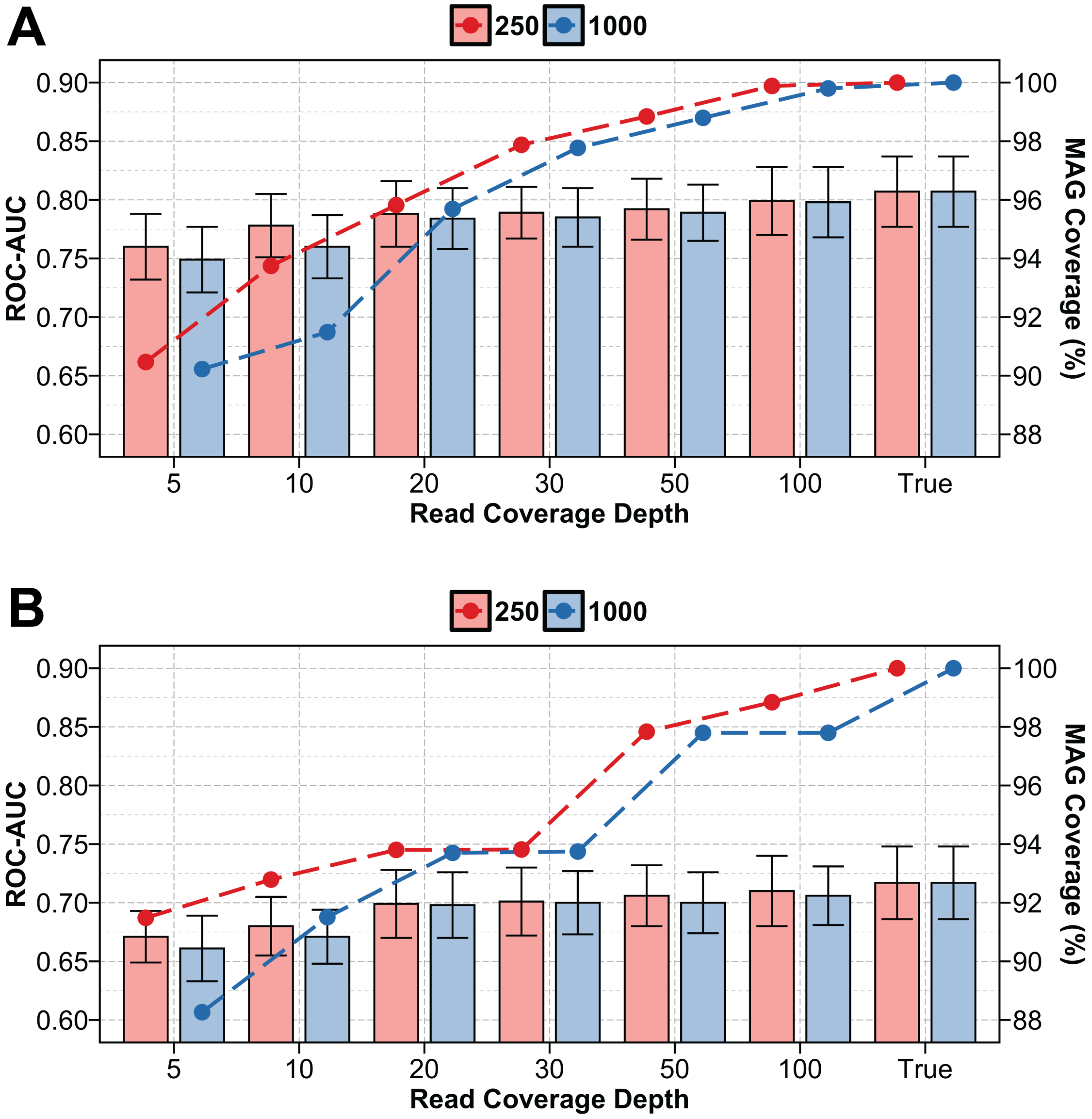


Figure 7.3 **The comparison of different read lengths of 250-bp and 1000-bp and coverage depths of 5x, 10x, 20x, 30x, 50x, and 100x when running DACS with estimated networks as input on both 100-taxon datasets under model conditions with (A) 5 non-deep reticulations and (B) 3 deep reticulations.** "True" means results directly on the reference genomes. ROC-AUC (left y-axis, bars) and MAG coverage (%) (right y-axis, dash lines) as functions of read coverage depths. Red bars/lines represent for read length of 250-bp, and blue bars/lines for read length of 1000-bp.

Table 7.3 **The comparison of different read lengths of 250-bp and 1000-bp and coverage depths of 5x, 10x, 20x, 30x, 50x, and 100x on both 100-taxon datasets under model conditions with 5 non-deep reticulations and 3 deep reticulations.** The measures include the area under the receiver-operating characteristic curve (ROC-AUC) and MAG coverage (%). DACS was run twice with true network or estimated networks as input. "True" means results directly on the reference genomes. The average and standard error (SE) are calculated based on 20 replicates (represented as "average value ± SE value" or only "average" for runtime).

Dataset	**Read length**	**Read Coverage Depth**	**MAG coverage (%)**	**ROC-AUC**	
				True Network	Estimated Networks
5 non-deep	250-bp	5x	90.467	0.926±0.014	0.760±0.028
		10x	93.747	0.927±0.015	0.778±0.027
		20x	95.821	0.931±0.013	0.788±0.028
		30x	97.877	0.933±0.016	0.789±0.022
		50x	98.845	0.935±0.015	0.792±0.026
		100x	99.885	0.933±0.011	0.799±0.029
		True	100.000	0.939±0.012	0.807±0.030
	1000-bp	5x	90.225	0.915±0.013	0.749±0.028
		10x	91.486	0.914±0.015	0.760±0.027
		20x	95.689	0.930±0.015	0.784±0.026
		30x	97.773	0.932±0.014	0.785±0.025
		50x	98.797	0.934±0.014	0.789±0.024
		100x	99.801	0.933±0.015	0.798±0.030
		True	100.000	0.939±0.012	0.807±0.030
3 deep	250-bp	5x	91.484	0.927±0.018	0.671±0.022
		10x	92.790	0.931±0.018	0.680±0.025
		20x	93.803	0.939±0.018	0.699±0.029
		30x	93.816	0.941±0.023	0.701±0.029
		50x	97.836	0.947±0.018	0.706±0.026
		100x	98.840	0.949±0.018	0.710±0.030
		True	100.000	0.950±0.019	0.717±0.031
	1000-bp	5x	88.266	0.920±0.018	0.661±0.028
		10x	91.508	0.929±0.018	0.671±0.023
		20x	93.702	0.937±0.019	0.698±0.028
		30x	93.747	0.939±0.020	0.700±0.027
		50x	97.797	0.946±0.018	0.700±0.026
		100x	97.797	0.947±0.015	0.706±0.025
		True	100.000	0.950±0.019	0.717±0.031

read depth, the use of estimated networks introduced noticeable performance degradation. This observation suggests that improvements in phylogenetic network inference methods could further enhance the predictive accuracy of DACS and related downstream analyses.

Despite the increased complexity of the 3 deep reticulation model compared to the 5 non-deep reticulation model, DACS demonstrated robustness across varying levels of reticulation complexity if the assembly quality is adequate. In both scenarios, coverage depth remained the key driver of improved MAG coverage and ROC-AUC, underscoring the importance of sufficient sequencing effort when detecting subtle evolutionary events such as introgression or HGT. Our results also highlight that even complex reticulate events are tractable with a carefully assembled metagenome, provided there is adequate depth of coverage.

Although this study used simulated reads from known reference genomes, real-world metagenomic datasets often introduce additional complexities such as uneven taxon abundances, incomplete reference databases, and the possibility of assembling novel microbial lineages. Future efforts should further evaluate the DACS method on real metagenomic datasets that capture the full diversity and uneven coverage typically encountered in microbial communities. Additional refinements to assembly strategies, binning algorithms, and data integration approaches may further enhance the accuracy of HGT detection by improving the resolution and completeness of MAGs.

Our application of the DACS-based framework to simulated metagenomic datasets demonstrates its robustness in identifying HGT and introgression events under a wide range of coverage depths and read lengths. Despite the initial advantage of shorter reads under very low coverage, both read lengths can ultimately achieve similar and high accuracy in detecting complex reticulate events once coverage depth reaches approximately 20-30x. These findings emphasize the critical role of coverage depth in assembling high-quality MAGs, which in turn drives more reliable HGT detection. The results further underscore the value of DACS for metagenomic research, offering a scalable and accurate tool that can be adapted to diverse microbial communities characterized by extensive reticulation and genomic complexity.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

The dissertation have introduced innovative techniques that push the boundaries of what is achievable in introgression studies. The research has laid a solid foundation for advancing the study of introgression by offering novel methods and insights that effectively address the limitations of existing approaches. In particular, our work tackles critical scalability issues and enhances the accuracy of phylogenetic analyses. These advancements not only make significant strides towards understanding genetics and biology, but also provide valuable tools for a wide range of biological and medical applications.

A central achievement of this research is the development of PHiMM, an introgression detection algorithm designed to handle genomic datasets containing dozens of DNA sequences. PHiMM employs a novel coalescent-based approximation strategy combined with a Hidden Markov Model (HMM) framework. This integration effectively reduces model complexity while preserving detection accuracy. Comparative studies with existing state-of-the-art methods demonstrated that PHiMM achieves substantially better runtime and memory usage than PhyloNet-HMM, maintaining inference accuracies on par with or surpassing established benchmarks.

We also employed SERES as a data perturbation technique to enhance introgression inference and learning. Simulation experiments demonstrated that combining the SERES resampling approach with PHiMM substantially improves introgression inference accuracy under various model conditions compared to standalone PHiMM. Although the combined SERES+PHiMM method results in longer runtimes, it maintains similar memory usage compared to standalone PHiMM, demonstrating its potential to “boost‘’ PHiMM’s inference accuracy.

Recognizing that reliable introgression detection is intrinsically linked to accurate phylogenetic network inference, we extended PHiMM to DACS. Unlike traditional methods that require a known network topology, DACS integrates *de novo* network inference with introgression detection, enabling more flexible and automated analysis. We further improved scalability by leveraging divide-and-conquer and subsampling techniques, allowing large-scale genomic datasets to be partitioned into smaller, more manageable subproblems. These enhancements collectively ensure that DACS is both computationally tractable and capable of producing highly accurate results in ultra-large data contexts.

Given the rising importance of metagenomic studies — particularly those involving complex microbial communities — our work underscores the potential of applying DACS to metagenomic datasets. Through the construction of metagenome-assembled genomes (MAGs), we demonstrated that our methods can be adapted to identify reticulate evolutionary events, such as introgression or horizontal gene transfer (HGT), even in the presence of substantial noise and partial data. These advancements underscore the growing relevance of network-based phylogenetics in metagenomics and pave the way for future research that further refines these approaches for real-world applications.

8.2 Future Work

While the methodologies developed in this dissertation provide a robust framework for introgression detection, several key avenues remain open for further investigation.

An important extension involves distinguishing adaptive introgression from the broader spectrum of introgression events. In adaptive introgression, relatively small genomic regions — transferred from a donor species — confers a selective advantage to the recipient species. This investigation is of great importance because adaptive introgression can broaden genetic diversity and promote species adaptations to various environments. Many methods have been developed to detect adaptive introgression [259–261]. With the rapid progress in deep learning techniques, Gower et al. [262] proposed a convolutional neural network (CNN)-based approach to jointly model introgression and positive selection. However, this method requires the use of simulation data to train the network, with certain parameters and models needing to be specified a priori. This necessitates the experience and expertise of users to avoid CNN performance declines caused by misspecifications. Inspired by this method, integrating generative adversarial networks (GANs) [263] into the adaptive introgression detection process may help allow the model to learn in a more unsupervised or semi-supervised manner. Attention-based mechanisms could also be introduced to incorporate expert knowledge without significantly constraining model flexibility.

Furthermore, making these methods more accessible to a wider audience is a key priority. Developing open-source software packages with user-friendly interfaces and thorough documentation will enable researchers from different fields—such as ecology, epidemiology, and evolutionary biology—to apply these approaches in their own work. Improved support for reproducible research practices, including containerized computing environments and version-controlled workflows, would further promote transparency and foster broader use of these tools.

Overall, these prospective directions aim to deepen our understanding of reticulate evolution across myriad biological contexts. By harnessing methodological advances in non-tree-like phylogenetics, machine learning, and large-scale data analysis, future work has the potential to further elucidate the complex processes that shape biodiversity and drive adaptation in both eukaryotic and prokaryotic lineages.

BIBLIOGRAPHY

[1] Bryan T. Grenfell, Oliver G. Pybus, Julia R. Gog, James L. N. Wood, Janet M. Daly, Jenny A. Mumford, and Edward C. Holmes. Unifying the epidemiological and evolutionary dynamics of pathogens. *Science*, 303(5656):327–332, 2004. doi: 10.1126/science.1090727. URL https://www.science.org/doi/abs/10.1126/science.1090727.

[2] Sean B Carroll, Jennifer K Grenier, and Scott D Weatherbee. *From DNA to diversity: molecular genetics and the evolution of animal design*. John Wiley & Sons, 2013.

[3] Christopher M Thomas and Kaare M Nielsen. Mechanisms of, and barriers to, horizontal gene transfer between bacteria. *Nature reviews microbiology*, 3(9):711–721, 2005.

[4] Manolis Kellis, Nick Patterson, Matthew Endrizzi, Bruce Birren, and Eric S Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423(6937):241–254, 2003.

[5] Jian Ma. Reconstructing the history of large-scale genomic changes: Biological questions and computational challenges. *Journal of Computational Biology*, 18(7):879–893, 2011. doi: 10.1089/cmb.2010.0189. URL https://doi.org/10.1089/cmb.2010.0189. PMID: 21563973.

[6] Richard E. Green, Johannes Krause, Adrian W. Briggs, Tomislav Maricic, Udo Stenzel, Martin Kircher, Nick Patterson, Heng Li, Weiwei Zhai, Markus Hsi-Yang Fritz, Nancy F. Hansen, Eric Y. Durand, Anna-Sapfo Malaspinas, Jeffrey D. Jensen, Tomas Marques-Bonet, Can Alkan, Kay Prüfer, Matthias Meyer, Hernán A. Burbano, Jeffrey M. Good, Rigo Schultz, Ayinuer Aximu-Petri, Anne Butthof, Barbara Höber, Barbara Höffner, Madlen Siegemund, Antje Weihmann, Chad Nusbaum, Eric S. Lander, Carsten Russ, Nathaniel Novod, Jason Affourtit, Michael Egholm, Christine Verna, Pavao Rudan, Dejana Brajkovic, Željko Kucan, Ivan Gušic, Vladimir B. Doronichev, Liubov V. Golovanova, Carles Lalueza-Fox, Marco de la Rasilla, Javier Fortea, Antonio Rosas, Ralf W. Schmitz, Philip L. F. Johnson, Evan E. Eichler, Daniel Falush, Ewan Birney, James C. Mullikin, Montgomery Slatkin, Rasmus Nielsen, Janet Kelso, Michael Lachmann, David Reich, and Svante Pääbo. A Draft Sequence of the Neandertal Genome. *Science*, 328(5979):710–722, 2010. doi: 10.1126/science.1188021. URL https://www.science.org/doi/abs/10.1126/science.1188021.

[7] Kevin J. Liu, Ethan Steinberg, Alexander Yozzo, Ying Song, Michael H. Kohn, and Luay Nakhleh. Interspecific introgressive origin of genomic diversity in the house mouse. *Proceedings of the National Academy of Sciences*, 112(1):196–201, 2015. doi: 10.1073/pnas.1406298111. URL https://www.pnas.org/doi/abs/10.1073/pnas.1406298111.

[8] Kanchon K. Dasmahapatra, James R. Walters, Adriana D. Briscoe, John W. Davey, Annabel Whibley, Nicola J. Nadeau, Aleksey V. Zimin, Daniel S. T. Hughes, Laura C. Ferguson, Simon H. Martin, Camilo Salazar, James J. Lewis, Sebastian Adler, Seung-Joon Ahn, Dean A. Baker, Simon W. Baxter, Nicola L. Chamberlain, Ritika Chauhan, Brian A. Counterman, Tamas Dalmay, Lawrence E. Gilbert, Karl Gordon, David G. Heckel, Heather M. Hines,

Katharina J. Hoff, Peter W. H. Holland, Emmanuelle Jacquin-Joly, Francis M. Jiggins, Robert T. Jones, Durrell D. Kapan, Paul Kersey, Gerardo Lamas, Daniel Lawson, Daniel Mapleson, Luana S. Maroja, Arnaud Martin, Simon Moxon, William J. Palmer, Riccardo Papa, Alexie Papanicolaou, Yannick Pauchet, David A. Ray, Neil Rosser, Steven L. Salzberg, Megan A. Supple, Alison Surridge, Ayse Tenger-Trolander, Heiko Vogel, Paul A. Wilkinson, Derek Wilson, James A. Yorke, Furong Yuan, Alexi L. Balmuth, Cathlene Eland, Karim Gharbi, Marian Thomson, Richard A. Gibbs, Yi Han, Joy C. Jayaseelan, Christie Kovar, Tittu Mathew, Donna M. Muzny, Fiona Ongeri, Ling-Ling Pu, Jiaxin Qu, Rebecca L. Thornton, Kim C. Worley, Yuan-Qing Wu, Mauricio Linares, Mark L. Blaxter, Richard H. ffrench Constant, Mathieu Joron, Marcus R. Kronforst, Sean P. Mullen, Robert D. Reed, Steven E. Scherer, Stephen Richards, James Mallet, W. Owen McMillan, Chris D. Jiggins, and The Heliconius Genome Consortium. Butterfly genome reveals promiscuous exchange of mimicry adaptations among species. *Nature*, 487(7405):94–98, 2012. doi: 10.1038/nature11041. URL https://doi.org/10.1038/nature11041.

[9] Emile Gluck-Thaler and Jason C. Slot. Dimensions of Horizontal Gene Transfer in Eukaryotic Microbial Pathogens. *PLOS Pathogens*, 11(10):1–7, 10 2015. doi: 10.1371/journal.ppat.1005156. URL https://doi.org/10.1371/journal.ppat.1005156.

[10] C. Gyles and P. Boerlin. Horizontally Transferred Genetic Elements and Their Role in Pathogenesis of Bacterial Disease. *Veterinary Pathology*, 51(2):328–340, 2014. doi: 10.1177/0300985813511131. URL https://doi.org/10.1177/0300985813511131. PMID: 24318976.

[11] Hussein A. Hejase, Natalie VandePol, Gregory M. Bonito, and Kevin J. Liu. FastNet: Fast and Accurate Statistical Inference of Phylogenetic Networks Using Large-Scale Genomic Sequence Data. In Mathieu Blanchette and Aïda Ouangraoua, editors, *Comparative Genomics*, pages 242–259, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00834-5.

[12] Kevin J. Liu, Jingxuan Dai, Kathy Truong, Ying Song, Michael H. Kohn, and Luay Nakhleh. An HMM-Based Comparative Genomic Framework for Detecting Introgression in Eukaryotes. *PLOS Computational Biology*, 10(6):1–13, 06 2014. doi: 10.1371/journal.pcbi.1003649. URL https://doi.org/10.1371/journal.pcbi.1003649.

[13] Dingqiao Wen, Yun Yu, Jiafan Zhu, and Luay Nakhleh. Inferring Phylogenetic Networks Using PhyloNet. *Systematic Biology*, 67(4):735–740, 03 2018. ISSN 1063-5157. doi: 10.1093/sysbio/syy015. URL https://doi.org/10.1093/sysbio/syy015.

[14] Hussein A. Hejase and Kevin J. Liu. A scalability study of phylogenetic network inference methods using empirical datasets and simulations involving a single reticulation. *BMC Bioinformatics*, 17(1):422, 2016. doi: 10.1186/s12859-016-1277-1. URL https://doi.org/10.1186/s12859-016-1277-1.

[15] Adam D. Leaché, Rebecca B. Harris, Bruce Rannala, and Ziheng Yang. The Influence of Gene Flow on Species Tree Estimation: A Simulation Study. *Systematic Biology*, 63(1):

17–30, 09 2013. ISSN 1063-5157. doi: 10.1093/sysbio/syt049. URL https://doi.org/10.1093/sysbio/syt049.

[16] John C. Wooley, Adam Godzik, and Iddo Friedberg. A Primer on Metagenomics. *PLOS Computational Biology*, 6(2):1–13, 02 2010. doi: 10.1371/journal.pcbi.1000667. URL https://doi.org/10.1371/journal.pcbi.1000667.

[17] Ziheng Yang. *Computational molecular evolution*. OUP Oxford, 2006.

[18] B.M.E. Moret, L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun, and R. Timme. Phylogenetic networks: modeling, reconstructibility, and accuracy. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):13–23, 2004. doi: 10.1109/TCBB.2004.10.

[19] Katharina T. Huber, Bengt Oxelman, Martin Lott, and Vincent Moulton. Reconstructing the Evolutionary History of Polyploids from Multilabeled Trees. *Molecular Biology and Evolution*, 23(9):1784–1791, 06 2006. ISSN 0737-4038. doi: 10.1093/molbev/msl045. URL https://doi.org/10.1093/molbev/msl045.

[20] Yun Yu, James H. Degnan, and Luay Nakhleh. The Probability of a Gene Tree Topology within a Phylogenetic Network with Applications to Hybridization Detection. *PLOS Genetics*, 8(4):1–10, 04 2012. doi: 10.1371/journal.pgen.1002660. URL https://doi.org/10.1371/journal.pgen.1002660.

[21] Gabriel Cardona, Francesc Rosselló, and Gabriel Valiente. Extended Newick: it is time for a standard representation of phylogenetic networks. *BMC Bioinformatics*, 9(1):532, 2008. doi: 10.1186/1471-2105-9-532. URL https://doi.org/10.1186/1471-2105-9-532.

[22] Tanja Stadler and James H. Degnan. A polynomial time algorithm for calculating the probability of a ranked gene tree given a species tree. *Algorithms for Molecular Biology*, 7 (1):7, 2012. doi: 10.1186/1748-7188-7-7. URL https://doi.org/10.1186/1748-7188-7-7.

[23] J. F. C. Kingman. On the genealogy of large populations. *Journal of Applied Probability*, 19(A):27–43, 1982. doi: 10.2307/3213548.

[24] Yun Yu, Cuong Than, James H. Degnan, and Luay Nakhleh. Coalescent Histories on Phylogenetic Networks and Detection of Hybridization Despite Incomplete Lineage Sorting. *Systematic Biology*, 60(2):138–149, 01 2011. ISSN 1063-5157. doi: 10.1093/sysbio/syq084. URL https://doi.org/10.1093/sysbio/syq084.

[25] Joseph Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981. doi: 10.1007/BF01734359. URL https://doi.org/10.1007/BF01734359.

[26] Yun Yu, Jianrong Dong, Kevin J. Liu, and Luay Nakhleh. Maximum likelihood inference

of reticulate evolutionary histories. *Proceedings of the National Academy of Sciences*, 111 (46):16448–16453, 2014. doi: 10.1073/pnas.1407950111. URL https://www.pnas.org/doi/abs/10.1073/pnas.1407950111.

[27] Yun Yu and Luay Nakhleh. A maximum pseudo-likelihood approach for phylogenetic networks. *BMC Genomics*, 16(10):S10, 2015. doi: 10.1186/1471-2164-16-S10-S10. URL https://doi.org/10.1186/1471-2164-16-S10-S10.

[28] Michael J. Sanderson. r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics*, 19(2):301–302, 01 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/19.2.301. URL https://doi.org/10.1093/bioinformatics/19.2.301.

[29] D. Garrigan and A. J. Geneva. msmove, 2014. URL http://dx.doi.org/10.6084/m9.figshare.1060474.

[30] Richard R. Hudson. Generating samples under a Wright–Fisher neutral model of genetic variation . *Bioinformatics*, 18(2):337–338, 02 2002. ISSN 1367-4803. doi: 10.1093/bioinformatics/18.2.337. URL https://doi.org/10.1093/bioinformatics/18.2.337.

[31] T.H. Jukes and C.R. Cantor. *Evolution of Protein Molecules*, volume 3, pages 21–132. Academic Press, New York, NY, USA, 1969.

[32] Andrew Rambaut and Nicholas C. Grass. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Bioinformatics*, 13(3): 235–238, 06 1997. ISSN 1367-4803. doi: 10.1093/bioinformatics/13.3.235. URL https://doi.org/10.1093/bioinformatics/13.3.235.

[33] William Fletcher and Ziheng Yang. INDELible: A Flexible Simulator of Biological Sequence Evolution. *Molecular Biology and Evolution*, 26(8):1879–1888, 05 2009. ISSN 0737-4038. doi: 10.1093/molbev/msp098. URL https://doi.org/10.1093/molbev/msp098.

[34] Robert R Sokal. A statistical method for evaluating systematic relationships. *Univ. Kansas, Sci. Bull.*, 38:1409–1438, 1958.

[35] N Saitou and M Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 07 1987. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040454. URL https://doi.org/10.1093/oxfordjournals.molbev.a040454.

[36] Walter M. Fitch. Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Biology*, 20(4):406–416, 12 1971. ISSN 1063-5157. doi: 10.1093/sysbio/20.4.406. URL https://doi.org/10.1093/sysbio/20.4.406.

[37] Bruce Rannala and Ziheng Yang. Probability distribution of molecular evolutionary trees:

A new method of phylogenetic inference. *Journal of Molecular Evolution*, 43(3):304–311, 1996. doi: 10.1007/BF02338839. URL https://doi.org/10.1007/BF02338839.

[38] Ziheng Yang et al. Paml: a program package for phylogenetic analysis by maximum likelihood. *Computer applications in the biosciences*, 13(5):555–556, 1997.

[39] Stéphane Guindon, Jean-François Dufayard, Vincent Lefort, Maria Anisimova, Wim Hordijk, and Olivier Gascuel. New algorithms and methods to estimate maximum-likelihood phylogenies: Assessing the performance of phyml 3.0. *Systematic Biology*, 59(3):307–321, 05 2010. ISSN 1063-5157. doi: 10.1093/sysbio/syq010. URL https://doi.org/10.1093/sysbio/syq010.

[40] Alexandros Stamatakis. Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 01 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu033. URL https://doi.org/10.1093/bioinformatics/btu033.

[41] Alexey M Kozlov, Diego Darriba, Tomáš Flouri, Benoit Morel, and Alexandros Stamatakis. Raxml-ng: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics*, 35(21):4453–4455, 05 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz305. URL https://doi.org/10.1093/bioinformatics/btz305.

[42] Lam-Tung Nguyen, Heiko A. Schmidt, Arndt von Haeseler, and Bui Quang Minh. Iq-tree: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32(1):268–274, 11 2014. ISSN 0737-4038. doi: 10.1093/molbev/msu300. URL https://doi.org/10.1093/molbev/msu300.

[43] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. Fasttree 2 – approximately maximum-likelihood trees for large alignments. *PLOS ONE*, 5(3):1–10, 03 2010. doi: 10.1371/journal.pone.0009490. URL https://doi.org/10.1371/journal.pone.0009490.

[44] John P Huelsenbeck and Fredrik Ronquist. Mrbayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001.

[45] Sebastian Höhna, Michael J. Landis, Tracy A. Heath, Bastien Boussau, Nicolas Lartillot, Brian R. Moore, John P. Huelsenbeck, and Fredrik Ronquist. Revbayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Systematic Biology*, 65(4):726–736, 05 2016. ISSN 1063-5157. doi: 10.1093/sysbio/syw021. URL https://doi.org/10.1093/sysbio/syw021.

[46] Marc A Suchard, Philippe Lemey, Guy Baele, Daniel L Ayres, Alexei J Drummond, and Andrew Rambaut. Bayesian phylogenetic and phylodynamic data integration using beast 1.10. *Virus evolution*, 4(1):vey016, 2018.

[47] Remco Bouckaert, Timothy G. Vaughan, Joëlle Barido-Sottani, Sebastián Duchêne, Mathieu Fourment, Alexandra Gavryushkina, Joseph Heled, Graham Jones, Denise Kühnert, Nicola

De Maio, Michael Matschiner, Fábio K. Mendes, Nicola F. Müller, Huw A. Ogilvie, Louis du Plessis, Alex Popinga, Andrew Rambaut, David Rasmussen, Igor Siveroni, Marc A. Suchard, Chieh-Hsi Wu, Dong Xie, Chi Zhang, Tanja Stadler, and Alexei J. Drummond. Beast 2.5: An advanced software platform for bayesian evolutionary analysis. *PLOS Computational Biology*, 15(4):1–28, 04 2019. doi: 10.1371/journal.pcbi.1006650. URL https://doi.org/10.1371/journal.pcbi.1006650.

[48] Nicolas Lartillot, Thomas Lepage, and Samuel Blanquart. Phylobayes 3: a bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics*, 25(17): 2286–2288, 06 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp368. URL https://doi.org/10.1093/bioinformatics/btp368.

[49] Nicolas Lartillot, Nicolas Rodrigue, Daniel Stubbs, and Jacques Richer. PhyloBayes MPI: Phylogenetic Reconstruction with Infinite Mixtures of Profiles in a Parallel Environment. *Systematic Biology*, 62(4):611–615, 04 2013. ISSN 1063-5157. doi: 10.1093/sysbio/syt022. URL https://doi.org/10.1093/sysbio/syt022.

[50] Paschalia Kapli, Ziheng Yang, and Maximilian J Telford. Phylogenetic tree building in the genomic age. *Nature Reviews Genetics*, 21(7):428–444, 2020.

[51] Joseph Felsenstein. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic zoology*, 27(4):401–410, 1978.

[52] Md Shamsuzzoha Bayzid and Tandy Warnow. Naive binning improves phylogenomic analyses. *Bioinformatics*, 29(18):2277–2284, 07 2013. ISSN 1367-4803. doi: 10.1093/bioinformatics/btt394. URL https://doi.org/10.1093/bioinformatics/btt394.

[53] Joseph Heled and Alexei J. Drummond. Bayesian Inference of Species Trees from Multilocus Data. *Molecular Biology and Evolution*, 27(3):570–580, 11 2009. ISSN 0737-4038. doi: 10.1093/molbev/msp274. URL https://doi.org/10.1093/molbev/msp274.

[54] Huw A. Ogilvie, Remco R. Bouckaert, and Alexei J. Drummond. Starbeast2 brings faster species tree inference and accurate estimates of substitution rates. *Molecular Biology and Evolution*, 34(8):2101–2114, 04 2017. ISSN 0737-4038. doi: 10.1093/molbev/msx126. URL https://doi.org/10.1093/molbev/msx126.

[55] Ziheng Yang and Bruce Rannala. Unguided species delimitation using dna sequence data from multiple loci. *Molecular Biology and Evolution*, 31(12):3125–3135, 10 2014. ISSN 0737-4038. doi: 10.1093/molbev/msu279. URL https://doi.org/10.1093/molbev/msu279.

[56] Tomáš Flouri, Xiyun Jiao, Bruce Rannala, and Ziheng Yang. Species tree inference with bpp using genomic sequences and the multispecies coalescent. *Molecular Biology and Evolution*, 35(10):2585–2593, 07 2018. ISSN 0737-4038. doi: 10.1093/molbev/msy147. URL https://doi.org/10.1093/molbev/msy147.

[57] Wayne P. Maddison. Gene Trees in Species Trees. *Systematic Biology*, 46(3):523–536, 09 1997. ISSN 1063-5157. doi: 10.1093/sysbio/46.3.523. URL https://doi.org/10.1093/sysbio/46.3.523.

[58] Cuong Than and Luay Nakhleh. Species Tree Inference by Minimizing Deep Coalescences. *PLOS Computational Biology*, 5(9):1–12, 09 2009. doi: 10.1371/journal.pcbi.1000501. URL https://doi.org/10.1371/journal.pcbi.1000501.

[59] Yun Yu, Tandy Warnow, and Luay Nakhleh. Algorithms for MDC-Based Multi-Locus Phylogeny Inference: Beyond Rooted Binary Gene Trees on Single Alleles. *Journal of Computational Biology*, 18(11):1543–1559, 2011. doi: 10.1089/cmb.2011.0174. URL https://doi.org/10.1089/cmb.2011.0174. PMID: 22035329.

[60] S. Mirarab, R. Reaz, Md. S. Bayzid, T. Zimmermann, M. S. Swenson, and T. Warnow. ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics*, 30(17): i541–i548, 08 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu462. URL https://doi.org/10.1093/bioinformatics/btu462.

[61] Siavash Mirarab and Tandy Warnow. ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. *Bioinformatics*, 31(12):i44–i52, 06 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv234. URL https://doi.org/10.1093/bioinformatics/btv234.

[62] Liang Liu, Lili Yu, and Scott V Edwards. A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. *BMC evolutionary biology*, 10:1–18, 2010.

[63] Liang Liu. BEST: Bayesian estimation of species trees under the coalescent model. *Bioinformatics*, 24(21):2542–2543, 09 2008. ISSN 1367-4803. doi: 10.1093/bioinformatics/btn484. URL https://doi.org/10.1093/bioinformatics/btn484.

[64] D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1):131–147, 1981. ISSN 0025-5564. doi: https://doi.org/10.1016/0025-5564(81)90043-2. URL https://www.sciencedirect.com/science/article/pii/0025556481900432.

[65] Jiafan Zhu, Dingqiao Wen, Yun Yu, Heidi M. Meudt, and Luay Nakhleh. Bayesian inference of phylogenetic networks from bi-allelic genetic markers. *PLOS Computational Biology*, 14(1):1–32, 01 2018. doi: 10.1371/journal.pcbi.1005932. URL https://doi.org/10.1371/journal.pcbi.1005932.

[66] Luay Nakhleh. A Metric on the Space of Reduced Phylogenetic Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(2):218–222, 2010. doi: 10.1109/TCBB.2009.2.

[67] F. Rodríguez, J.L. Oliver, A. Marín, and J.R. Medina. The general stochastic model of nucleotide substitution. *Journal of Theoretical Biology*, 142(4):485–501, 1990. ISSN 0022-5193. doi: https://doi.org/10.1016/S0022-5193(05)80104-3. URL https://www.sciencedirect.com/science/article/pii/S0022519305801043.

[68] Motoo Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of molecular evolution*, 16: 111–120, 1980.

[69] Joseph Felsenstein and Joseph Felenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

[70] Z Yang. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular Biology and Evolution*, 10(6):1396–1401, 11 1993. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040082. URL https://doi.org/10.1093/oxfordjournals.molbev.a040082.

[71] Jente Ottenburghs. Ghost Introgression: Spooky Gene Flow in the Distant Past. *BioEssays*, 42(6):2000012, 2020. doi: https://doi.org/10.1002/bies.202000012. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/bies.202000012.

[72] Adriana Suarez-Gonzalez, Christian Lexer, and Quentin C. B. Cronk. Adaptive introgression: a plant perspective. *Biology Letters*, 14(3):20170688, 2018. doi: 10.1098/rsbl.2017.0688. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsbl.2017.0688.

[73] Tao Sang and Yang Zhong. Testing Hybridization Hypotheses Based on Incongruent Gene Trees. *Systematic Biology*, 49(3):422–434, 09 2000. ISSN 1063-5157. doi: 10.1080/10635159950127321. URL https://doi.org/10.1080/10635159950127321.

[74] Mark T. Holder, Jennifer A. Anderson, and Alisha K. Holloway. Difficulties in Detecting Hybridization. *Systematic Biology*, 50(6):978–982, 2001. ISSN 10635157, 1076836X. URL http://www.jstor.org/stable/3070875.

[75] Nick Patterson, Daniel J. Richter, Sante Gnerre, Eric S. Lander, and David Reich. Genetic evidence for complex speciation of humans and chimpanzees. *Nature*, 441(7097):1103–1108, 2006. doi: 10.1038/nature04789. URL https://doi.org/10.1038/nature04789.

[76] Asger Hobolth, Ole F Christensen, Thomas Mailund, and Mikkel H Schierup. Genomic Relationships and Speciation Times of Human, Chimpanzee, and Gorilla Inferred from a Coalescent Hidden Markov Model. *PLOS Genetics*, 3(2):1–11, 02 2007. doi: 10.1371/journal.pgen.0030007. URL https://doi.org/10.1371/journal.pgen.0030007.

[77] Julien Y Dutheil, Ganesh Ganapathy, Asger Hobolth, Thomas Mailund, Marcy K Uyenoyama, and Mikkel H Schierup. Ancestral Population Genomics: The Coalescent Hidden Markov Model Approach. *Genetics*, 183(1):259–274, 09 2009. ISSN 1943-2631.

doi: 10.1534/genetics.109.103010. URL https://doi.org/10.1534/genetics.109.103010.

[78] Thomas Mailund, Julien Y. Dutheil, Asger Hobolth, Gerton Lunter, and Mikkel H. Schierup. Estimating Divergence Time and Ancestral Effective Population Size of Bornean and Sumatran Orangutan Subspecies Using a Coalescent Hidden Markov Model. *PLOS Genetics*, 7(3): 1–15, 03 2011. doi: 10.1371/journal.pgen.1001319. URL https://doi.org/10.1371/journal.pgen.1001319.

[79] Thomas Mailund, Anders E. Halager, Michael Westergaard, Julien Y. Dutheil, Kasper Munch, Lars N. Andersen, Gerton Lunter, Kay Prüfer, Aylwyn Scally, Asger Hobolth, and Mikkel H. Schierup. A New Isolation with Migration Model along Complete Genomes Infers Very Different Divergence Processes among Closely Related Great Ape Species. *PLOS Genetics*, 8(12):1–19, 12 2012. doi: 10.1371/journal.pgen.1003125. URL https://doi.org/10.1371/journal.pgen.1003125.

[80] Simon Joly, Patricia A. McLenachan, and Peter J. Lockhart. A Statistical Approach for Distinguishing Hybridization and Incomplete Lineage Sorting. *The American Naturalist*, 174(2):E54–E70, 2009. doi: 10.1086/600082. URL https://doi.org/10.1086/600082. PMID: 19519219.

[81] Simon Joly. JML: testing hybridization from species trees. *Molecular Ecology Resources*, 12(1):179–184, 2012. doi: https://doi.org/10.1111/j.1755-0998.2011.03065.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1755-0998.2011.03065.x.

[82] Cedric Chauve, Jingxue Feng, and Liangliang Wang. Detecting Introgression in Anopheles Mosquito Genomes Using a Reconciliation-Based Approach. In Mathieu Blanchette and Aïda Ometricuangraoua, editors, *Comparative Genomics*, pages 163–178, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00834-5.

[83] Daniel R. Schrider, Julien Ayroles, Daniel R. Matute, and Andrew D. Kern. Supervised machine learning reveals introgressed loci in the genomes of Drosophila simulans and D. sechellia. *PLOS Genetics*, 14(4):1–29, 04 2018. doi: 10.1371/journal.pgen.1007341. URL https://doi.org/10.1371/journal.pgen.1007341.

[84] Lex Flagel, Yaniv Brandvain, and Daniel R Schrider. The unreasonable effectiveness of convolutional neural networks in population genetic inference. *Molecular Biology and Evolution*, 36(2):220–238, 12 2018. ISSN 0737-4038. doi: 10.1093/molbev/msy224. URL https://doi.org/10.1093/molbev/msy224.

[85] Dylan D. Ray, Lex Flagel, and Daniel R. Schrider. Introunet: Identifying introgressed alleles via semantic segmentation. *PLOS Genetics*, 20(2):1–37, 02 2024. doi: 10.1371/journal.pgen.1010657. URL https://doi.org/10.1371/journal.pgen.1010657.

[86] Daniel H. Huson, Tobias Klöpper, Pete J. Lockhart, and Mike A. Steel. Reconstruction of Reticulate Networks from Gene Trees. In Satoru Miyano, Jill Mesirov, Simon Kasif,

Sorin Istrail, Pavel A. Pevzner, and Michael Waterman, editors, *Research in Computational Molecular Biology*, pages 233–249, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31950-4.

[87] Eric Y. Durand, Nick Patterson, David Reich, and Montgomery Slatkin. Testing for Ancient Admixture between Closely Related Populations. *Molecular Biology and Evolution*, 28 (8):2239–2252, 02 2011. ISSN 0737-4038. doi: 10.1093/molbev/msr048. URL https://doi.org/10.1093/molbev/msr048.

[88] James B. Pease and Matthew W. Hahn. Detection and Polarization of Introgression in a Five-Taxon Phylogeny. *Systematic Biology*, 64(4):651–662, 04 2015. ISSN 1063-5157. doi: 10.1093/sysbio/syv023. URL https://doi.org/10.1093/sysbio/syv023.

[89] Ryan A. Leo Elworth, Chabrielle Allen, Travis Benedict, Peter Dulworth, and Luay Nakhleh. DGEN: A Test Statistic for Detection of General Introgression Scenarios. In Laxmi Parida and Esko Ukkonen, editors, *18th International Workshop on Algorithms in Bioinformatics (WABI 2018)*, volume 113 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-082-8. doi: 10.4230/LIPIcs.WABI.2018.19. URL http://drops.dagstuhl.de/opus/volltexte/2018/9321.

[90] David Reich, Kumarasamy Thangaraj, Nick Patterson, Alkes L. Price, and Lalji Singh. Reconstructing Indian population history. *Nature*, 461(7263):489–494, 2009. doi: 10.1038/nature08365. URL https://doi.org/10.1038/nature08365.

[91] David Reich, Nick Patterson, Martin Kircher, Frederick Delfin, Madhusudan R. Nandineni, Irina Pugach, Albert Min-Shan Ko, Ying-Chin Ko, Timothy A. Jinam, Maude E. Phipps, Naruya Saitou, Andreas Wollstein, Manfred Kayser, Svante Pääbo, and Mark Stoneking. Denisova Admixture and the First Modern Human Dispersals into Southeast Asia and Oceania. *The American Journal of Human Genetics*, 89(4):516–528, 2011. ISSN 0002-9297. doi: https://doi.org/10.1016/j.ajhg.2011.09.005. URL https://www.sciencedirect.com/science/article/pii/S0002929711003958.

[92] Nick Patterson, Priya Moorjani, Yontao Luo, Swapan Mallick, Nadin Rohland, Yiping Zhan, Teri Genschoreck, Teresa Webster, and David Reich. Ancient Admixture in Human History. *Genetics*, 192(3):1065–1093, 11 2012. ISSN 1943-2631. doi: 10.1534/genetics.112.145037. URL https://doi.org/10.1534/genetics.112.145037.

[93] Simon H. Martin, John W. Davey, and Chris D. Jiggins. Evaluating the Use of ABBA–BABA Statistics to Locate Introgressed Loci. *Molecular Biology and Evolution*, 32(1):244–257, 09 2014. ISSN 0737-4038. doi: 10.1093/molbev/msu269. URL https://doi.org/10.1093/molbev/msu269.

[94] Benjamin M Peter. Admixture, Population Structure, and F-Statistics. *Genetics*, 202(4):1485–1501, 02 2016. ISSN 1943-2631. doi: 10.1534/genetics.115.183913. URL https:

//doi.org/10.1534/genetics.115.183913.

[95] Paul D Blischak, Julia Chifman, Andrea D Wolfe, and Laura S Kubatko. HyDe: A Python Package for Genome-Scale Hybridization Detection. *Systematic Biology*, 67(5):821–829, 03 2018. ISSN 1063-5157. doi: 10.1093/sysbio/syy023. URL https://doi.org/10.1093/sysbio/syy023.

[96] Laura S. Kubatko and Julia Chifman. An invariants-based method for efficient identification of hybrid species from large-scale genomic data. *BMC Evolutionary Biology*, 19(1):112, 2019. doi: 10.1186/s12862-019-1439-7. URL https://doi.org/10.1186/s12862-019-1439-7.

[97] Bastian Pfeifer and Durrell D. Kapan. Estimates of introgression as a function of pairwise distances. *BMC Bioinformatics*, 20(1):207, 2019. doi: 10.1186/s12859-019-2747-z. URL https://doi.org/10.1186/s12859-019-2747-z.

[98] Howard Ochman, Jeffrey G. Lawrence, and Eduardo A. Groisman. Lateral gene transfer and the nature of bacterial innovation. *Nature*, 405(6784):299–304, 2000. doi: 10.1038/35012500. URL https://doi.org/10.1038/35012500.

[99] Brad Spellberg, Robert Guidos, David Gilbert, John Bradley, Helen W. Boucher, W. Michael Scheld, John G. Bartlett, Jr Edwards, John, and the Infectious Diseases Society of America. The Epidemic of Antibiotic-Resistant Infections: A Call to Action for the Medical Community from the Infectious Diseases Society of America. *Clinical Infectious Diseases*, 46(2):155–164, 01 2008. ISSN 1058-4838. doi: 10.1086/524891. URL https://doi.org/10.1086/524891.

[100] Alan G. Mathew, Robin Cissell, and S. Liamthong. Antibiotic Resistance in Bacteria Associated with Food Animals: A United States Perspective of Livestock Production. *Foodborne Pathogens and Disease*, 4(2):115–133, 2007. doi: 10.1089/fpd.2006.0066. URL https://doi.org/10.1089/fpd.2006.0066. PMID: 17600481.

[101] Rafael Szczepanowski, Burkhard Linke, Irene Krahn, Karl-Heinz Gartemann, Tim Gützkow, Wolfgang Eichler, Alfred Pühler, and Andreas Schlüter. Detection of 140 clinically relevant antibiotic-resistance genes in the plasmid metagenome of wastewater treatment plant bacteria showing reduced susceptibility to selected antibiotics. *Microbiology*, 155(7):2306–2319, 2009. ISSN 1465-2080. doi: https://doi.org/10.1099/mic.0.028233-0. URL https://www.microbiologyresearch.org/content/journal/micro/10.1099/mic.0.028233-0.

[102] Johan Bengtsson-Palme, Fredrik Boulund, Jerker Fick, Erik Kristiansson, and D. G. Joakim Larsson. Shotgun metagenomics reveals a wide array of antibiotic resistance genes and mobile elements in a polluted lake in India. *Frontiers in Microbiology*, 5, 2014. ISSN 1664-302X. doi: 10.3389/fmicb.2014.00648. URL https://www.frontiersin.org/articles/10.3389/fmicb.2014.00648.

[103] Lauren D. McDaniel, Elizabeth Young, Jennifer Delaney, Fabian Ruhnau, Kim B. Ritchie,

and John H. Paul. High Frequency of Horizontal Gene Transfer in the Oceans. *Science*, 330 (6000):50–50, 2010. doi: 10.1126/science.1192243. URL https://www.science.org/doi/abs/10.1126/science.1192243.

[104] James H. Degnan and Noah A. Rosenberg. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends in Ecology & Evolution*, 24(6):332–340, 2009. ISSN 0169-5347. doi: https://doi.org/10.1016/j.tree.2009.01.009. URL https://www.sciencedirect.com/science/article/pii/S0169534709000846.

[105] Ricard Albalat and Cristian Cañestro. Evolution by gene loss. *Nature Reviews Genetics*, 17 (7):379–391, 2016. doi: 10.1038/nrg.2016.39. URL https://doi.org/10.1038/nrg.2016.39.

[106] Shweta Meshram, Sunaina Bisht, and Robin Gogoi. Chapter 14 - Current development, application and constraints of biopesticides in plant disease management. In Amitava Rakshit, Vijay Singh Meena, P.C. Abhilash, B.K. Sarma, H.B. Singh, Leonardo Fraceto, Manoj Parihar, and Anand Kumar Singh, editors, *Biopesticides*, Advances in Bio-inoculant Science, pages 207–224. Woodhead Publishing, 2022. ISBN 978-0-12-823355-9. doi: https://doi.org/10.1016/B978-0-12-823355-9.00004-3. URL https://www.sciencedirect.com/science/article/pii/B9780128233559000043.

[107] Jeffrey G. Lawrence and Howard Ochman. Reconciling the many faces of lateral gene transfer. *Trends in Microbiology*, 10(1):1–4, 2002. ISSN 0966-842X. doi: https://doi.org/10.1016/S0966-842X(01)02282-X. URL https://www.sciencedirect.com/science/article/pii/S0966842X0102282X.

[108] Mohammad Shabbir Hasan, Qi Liu, Han Wang, John Fazekas, Bernard Chen, and Dongsheng Che. GIST: Genomic island suite of tools for predicting genomic islands in genomic sequences. *Bioinformation*, 8(4):203–205, 2012. ISSN 0973-2063 (Electronic); 0973-8894 (Print); 0973-2063 (Linking). doi: 10.6026/97320630008203.

[109] Claire Bertelli, Matthew R Laird, Kelly P Williams, Simon Fraser University Research Computing Group, Britney Y Lau, Gemma Hoad, Geoffrey L Winsor, and Fiona SL Brinkman. IslandViewer 4: expanded prediction of genomic islands for larger-scale datasets. *Nucleic Acids Research*, 45(W1):W30–W35, 05 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx343. URL https://doi.org/10.1093/nar/gkx343.

[110] Christophe Dessimoz, Daniel Margadant, and Gaston H. Gonnet. DLIGHT – Lateral Gene Transfer Detection Using Pairwise Evolutionary Distances in a Statistical Framework. In Martin Vingron and Limsoon Wong, editors, *Research in Computational Molecular Biology*, pages 315–330, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-78839-3.

[111] Qiyun Zhu, Michael Kosoy, and Katharina Dittmar. HGTector: an automated method facilitating genome-wide discovery of putative horizontal gene transfers. *BMC Genomics*, 15(1):717, 2014. doi: 10.1186/1471-2164-15-717. URL https://doi.org/10.1186/

1471-2164-15-717.

[112] Sheila Podell and Terry Gaasterland. DarkHorse: a method for genome-wide prediction of horizontal gene transfer. *Genome Biology*, 8(2):R16, 2007. doi: 10.1186/gb-2007-8-2-r16. URL https://doi.org/10.1186/gb-2007-8-2-r16.

[113] Lawrence A. David and Eric J. Alm. Rapid evolutionary innovation during an Archaean genetic expansion. *Nature*, 469(7328):93–96, 2011. doi: 10.1038/nature09649. URL https://doi.org/10.1038/nature09649.

[114] Mukul S Bansal, Manolis Kellis, Misagh Kordi, and Soumya Kundu. RANGER-DTL 2.0: rigorous reconstruction of gene-family evolution by duplication, transfer and loss. *Bioinformatics*, 34(18):3214–3216, 04 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty314. URL https://doi.org/10.1093/bioinformatics/bty314.

[115] Thomas W. Schoenfeld, Senthil K. Murugapiran, Jeremy A. Dodsworth, Sally Floyd, Michael Lodes, David A. Mead, and Brian P. Hedlund. Lateral Gene Transfer of Family A DNA Polymerases between Thermophilic Viruses, Aquificae, and Apicomplexa. *Molecular Biology and Evolution*, 30(7):1653–1664, 04 2013. ISSN 0737-4038. doi: 10.1093/molbev/mst078. URL https://doi.org/10.1093/molbev/mst078.

[116] Weizhi Song, Bernd Wemheuer, Shan Zhang, Kerrin Steensen, and Torsten Thomas. MetaCHIP: community-level horizontal gene transfer identification through the combination of best-match and phylogenetic approaches. *Microbiome*, 7(1):36, 2019. doi: 10.1186/s40168-019-0649-y. URL https://doi.org/10.1186/s40168-019-0649-y.

[117] Matt Ravenhall, Nives Škunca, Florent Lassalle, and Christophe Dessimoz. Inferring Horizontal Gene Transfer. *PLOS Computational Biology*, 11(5):1–16, 05 2015. doi: 10.1371/journal.pcbi.1004095. URL https://doi.org/10.1371/journal.pcbi.1004095.

[118] Richard G. Harrison and Erica L. Larson. Hybridization, Introgression, and the Nature of Species Boundaries. *Journal of Heredity*, 105(S1):795–809, 08 2014. ISSN 0022-1503. doi: 10.1093/jhered/esu033. URL https://doi.org/10.1093/jhered/esu033.

[119] James Mallet. Hybridization as an invasion of the genome. *Trends in Ecology & Evolution*, 20(5):229–237, 2005. ISSN 0169-5347. doi: https://doi.org/10.1016/j.tree.2005.02.010. URL https://www.sciencedirect.com/science/article/pii/S016953470500039X. Special issue: Invasions, guest edited by Michael E. Hochberg and Nicholas J. Gotelli.

[120] Ying Song, Stefan Endepols, Nicole Klemann, Dania Richter, Franz-Rainer Matuschka, Ching-Hua Shih, Michael W. Nachman, and Michael H. Kohn. Adaptive Introgression of Anticoagulant Rodent Poison Resistance by Hybridization between Old World Mice. *Current Biology*, 21(15):1296–1301, 2011. ISSN 0960-9822. doi: https://doi.org/10.1016/j.cub.2011.06.043. URL https://www.sciencedirect.com/science/article/pii/S0960982211007160.

[121] Luay Nakhleh. Computational approaches to species phylogeny inference and gene tree reconciliation. *Trends in Ecology & Evolution*, 28(12):719–728, 2013. ISSN 0169-5347. doi: https://doi.org/10.1016/j.tree.2013.09.004. URL https://www.sciencedirect.com/science/article/pii/S0169534713002139.

[122] Qiqige Wuyun, Nicholas W. VanKuren, Marcus Kronforst, Sean P. Mullen, and Kevin J. Liu. Scalable Statistical Introgression Mapping Using Approximate Coalescent-Based Inference. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '19, page 504–513, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366663. doi: 10.1145/3307339.3342165. URL https://doi.org/10.1145/3307339.3342165.

[123] Cuong Than, Derek Ruths, and Luay Nakhleh. PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, 9(1):322, 2008. doi: 10.1186/1471-2105-9-322. URL https://doi.org/10.1186/1471-2105-9-322.

[124] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. doi: 10.1109/5.18626.

[125] Michael JD Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, pages 26–46, 2009.

[126] R. P. Brent. *Algorithms for Minimization without Derivatives*. Dover Publications, Mineola, New York, 1973.

[127] Fabian Staubach, Anna Lorenc, Philipp W. Messer, Kun Tang, Dmitri A. Petrov, and Diethard Tautz. Genome Patterns of Selection and Introgression of Haplotypes in Natural Populations of the House Mouse (Mus musculus). *PLOS Genetics*, 8(8):1–13, 08 2012. doi: 10.1371/journal.pgen.1002891. URL https://doi.org/10.1371/journal.pgen.1002891.

[128] Jason R. Gallant, Vance E. Imhoff, Arnaud Martin, Wesley K. Savage, Nicola L. Chamberlain, Ben L. Pote, Chelsea Peterson, Gabriella E. Smith, Benjamin Evans, Robert D. Reed, Marcus R. Kronforst, and Sean P. Mullen. Ancient homology underlies adaptive mimetic diversity across butterflies. *Nature Communications*, 5(1):4817, 2014. doi: 10.1038/ncomms5817. URL https://doi.org/10.1038/ncomms5817.

[129] Hyuna Yang, Jeremy R Wang, John P Didion, Ryan J Buus, Timothy A Bell, Catherine E Welsh, François Bonhomme, Alex Hon-Tsen Yu, Michael W Nachman, Jaroslav Pialek, Priscilla Tucker, Pierre Boursot, Leonard McMillan, Gary A Churchill, and Fernando Pardo-Manuel de Villena. Subspecific origin and haplotype diversity in the laboratory mouse. *Nature Genetics*, 43(7):648–655, 2011. doi: 10.1038/ng.847. URL https://doi.org/10.1038/ng.847.

[130] Jean-Louis Guénet and François Bonhomme. Wild mice: an ever-increasing contribu-

tion to a popular mammalian model. *Trends in Genetics*, 19(1):24–31, 2003. ISSN 0168-9525. doi: https://doi.org/10.1016/S0168-9525(02)00007-0. URL https://www.sciencedirect.com/science/article/pii/S0168952502000070.

[131] John P. Didion, Hyuna Yang, Keith Sheppard, Chen-Ping Fu, Leonard McMillan, Fernando Pardo-Manuel de Villena, and Gary A. Churchill. Discovery of novel variants in genotyping arrays improves genotype retention and reduces ascertainment bias. *BMC Genomics*, 13(1): 34, 2012. doi: 10.1186/1471-2164-13-34. URL https://doi.org/10.1186/1471-2164-13-34.

[132] Paul Scheet and Matthew Stephens. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *The American Journal of Human Genetics*, 78(4):629–644, 2006.

[133] Bettina Harr, Emre Karakoc, Rafik Neme, Meike Teschke, Christine Pfeifle, Željka Pezer, Hiba Babiker, Miriam Linnenbrink, Inka Montero, Rick Scavetta, Mohammad Reza Abai, Marta Puente Molins, Mathias Schlegel, Rainer G. Ulrich, Janine Altmüller, Marek Franitza, Anna Büntge, Sven Künzel, and Diethard Tautz. Genomic resources for wild populations of the house mouse, Mus musculus and its close relative Mus spretus. *Scientific Data*, 3(1): 160075, 2016. doi: 10.1038/sdata.2016.75. URL https://doi.org/10.1038/sdata.2016.75.

[134] Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pages 569–593. Springer, 1992.

[135] Bradley Efron and Robert Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pages 54–75, 1986.

[136] Wei Wang, Jack Smith, Hussein A Hejase, and Kevin J Liu. Non-parametric and semi-parametric support estimation using sequential resampling random walks on biomolecular sequences. *Algorithms for Molecular Biology*, 15:1–15, 2020.

[137] Giddy Landan and Dan Graur. Heads or tails: a simple reliability check for multiple sequence alignments. *Molecular biology and evolution*, 24(6):1380–1383, 2007.

[138] Wei Wang, Qiqige Wuyun, and Kevin J Liu. An application of random walk resampling to phylogenetic hmm inference and learning. *IEEE Transactions on NanoBioscience*, 19(3): 506–517, 2020.

[139] Oscar Westesson and Ian Holmes. Accurate detection of recombinant breakpoints in whole-genome alignments. *PLOS Computational Biology*, 5(3):1–13, 03 2009. doi: 10.1371/journal.pcbi.1000318. URL https://doi.org/10.1371/journal.pcbi.1000318.

[140] Michael PH Stumpf and Gilean AT McVean. Estimating recombination rates from population-genetic data. *Nature Reviews Genetics*, 4(12):959–968, 2003.

[141] Simon Myers, Leonardo Bottolo, Colin Freeman, Gil McVean, and Peter Donnelly. A fine-

scale map of recombination rates and hotspots across the human genome. *Science*, 310 (5746):321–324, 2005. doi: 10.1126/science.1117196. URL https://www.science.org/doi/abs/10.1126/science.1117196.

[142] Adam Auton and Gil McVean. Recombination rate estimation in the presence of hotspots. *Genome Research*, 17(8):1219–1227, 2007. doi: 10.1101/gr.6386707. URL http://genome.cshlp.org/content/17/8/1219.abstract.

[143] Bernard M.E. Moret, Usman Roshan, and Tandy Warnow. Sequence-Length Requirements for Phylogenetic Methods. In Roderic Guigó and Dan Gusfield, editors, *Algorithms in Bioinformatics*, pages 343–356, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45784-8.

[144] Kevin Liu, Sindhu Raghavan, Serita Nelesen, C Randal Linder, and Tandy Warnow. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science*, 324(5934):1561–1564, 2009. doi: 10.1126/science.1171243. URL https://www.science.org/doi/abs/10.1126/science.1171243.

[145] Jotun Hein, Mikkel Schierup, and Carsten Wiuf. *Gene Genealogies, Variation and Evolution: a Primer in Coalescent Theory*. Oxford University Press, Oxford, 2004.

[146] Sen Song, Liang Liu, Scott V Edwards, and Shaoyuan Wu. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. *Proceedings of the National Academy of Sciences*, 109(37):14942–14947, 2012.

[147] Mark P Simmons, Daniel B Sloan, and John Gatesy. The effects of subsampling gene trees on coalescent methods applied to ancient divergences. *Molecular Phylogenetics and Evolution*, 97:76–89, 2016.

[148] Laura A Katz and Jessica R Grant. Taxon-rich phylogenomic analyses resolve the eukaryotic tree of life and reveal the power of subsampling by sites. *Systematic biology*, 64(3):406–415, 2015.

[149] Alexander Lalejini, Marcos Sanson, Jack Garbus, Matthew Andres Moreno, and Emily Dolson. Runtime phylogenetic analysis enables extreme subsampling for test-based problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 511–514, 2024.

[150] Andrew Francis, Daniel H. Huson, and Mike Steel. Normalising phylogenetic networks. *Molecular Phylogenetics and Evolution*, 163:107215, 2021. ISSN 1055-7903. doi: https://doi.org/10.1016/j.ympev.2021.107215. URL https://www.sciencedirect.com/science/article/pii/S1055790321001482.

[151] Andrew Francis, Daniele Marchei, and Mike Steel. Phylogenetic network classes through the lens of expanding covers. *Journal of Mathematical Biology*, 88(5):58, 2024.

[152] Michael C. Fontaine, James B. Pease, Aaron Steele, Robert M. Waterhouse, Daniel E. Neafsey, Igor V. Sharakhov, Xiaofang Jiang, Andrew B. Hall, Flaminia Catteruccia, Evdoxia Kakani, Sara N. Mitchell, Yi-Chieh Wu, Hilary A. Smith, R. Rebecca Love, Mara K. Lawniczak, Michel A. Slotman, Scott J. Emrich, Matthew W. Hahn, and Nora J. Besansky. Extensive introgression in a malaria vector species complex revealed by phylogenomics. *Science*, 347(6217):1258524, 2015. doi: 10.1126/science.1258524. URL https://www.science.org/doi/abs/10.1126/science.1258524.

[153] Daniel E. Neafsey, Robert M. Waterhouse, Mohammad R. Abai, Sergey S. Aganezov, Max A. Alekseyev, James E. Allen, James Amon, Bruno Arcà, Peter Arensburger, Gleb Artemov, Lauren A. Assour, Hamidreza Basseri, Aaron Berlin, Bruce W. Birren, Stephanie A. Blandin, Andrew I. Brockman, Thomas R. Burkot, Austin Burt, Clara S. Chan, Cedric Chauve, Joanna C. Chiu, Mikkel Christensen, Carlo Costantini, Victoria L. M. Davidson, Elena Deligianni, Tania Dottorini, Vicky Dritsou, Stacey B. Gabriel, Wamdaogo M. Guelbeogo, Andrew B. Hall, Mira V. Han, Thaung Hlaing, Daniel S. T. Hughes, Adam M. Jenkins, Xiaofang Jiang, Irwin Jungreis, Evdoxia G. Kakani, Maryam Kamali, Petri Kemppainen, Ryan C. Kennedy, Ioannis K. Kirmitzoglou, Lizette L. Koekemoer, Njoroge Laban, Nicholas Langridge, Mara K. N. Lawniczak, Manolis Lirakis, Neil F. Lobo, Ernesto Lowy, Robert M. MacCallum, Chunhong Mao, Gareth Maslen, Charles Mbogo, Jenny McCarthy, Kristin Michel, Sara N. Mitchell, Wendy Moore, Katherine A. Murphy, Anastasia N. Naumenko, Tony Nolan, Eva M. Novoa, Samantha O'Loughlin, Chioma Oringanje, Mohammad A. Oshaghi, Nazzy Pakpour, Philippos A. Papathanos, Ashley N. Peery, Michael Povelones, Anil Prakash, David P. Price, Ashok Rajaraman, Lisa J. Reimer, David C. Rinker, Antonis Rokas, Tanya L. Russell, N'Fale Sagnon, Maria V. Sharakhova, Terrance Shea, Felipe A. Simão, Frederic Simard, Michel A. Slotman, Pradya Somboon, Vladimir Stegniy, Claudio J. Struchiner, Gregg W. C. Thomas, Marta Tojo, Pantelis Topalis, José M. C. Tubio, Maria F. Unger, John Vontas, Catherine Walton, Craig S. Wilding, Judith H. Willis, Yi-Chieh Wu, Guiyun Yan, Evgeny M. Zdobnov, Xiaofan Zhou, Flaminia Catteruccia, George K. Christophides, Frank H. Collins, Robert S. Cornman, Andrea Crisanti, Martin J. Donnelly, Scott J. Emrich, Michael C. Fontaine, William Gelbart, Matthew W. Hahn, Immo A. Hansen, Paul I. Howell, Fotis C. Kafatos, Manolis Kellis, Daniel Lawson, Christos Louis, Shirley Luckhart, Marc A. T. Muskavitch, José M. Ribeiro, Michael A. Riehle, Igor V. Sharakhov, Zhijian Tu, Laurence J. Zwiebel, and Nora J. Besansky. Highly evolvable malaria vectors: The genomes of 16 <i>anopheles</i> mosquitoes. *Science*, 347(6217):1258522, 2015. doi: 10.1126/science.1258522. URL https://www.science.org/doi/abs/10.1126/science.1258522.

[154] Sangeet Lamichhaney, Jonas Berglund, Markus Sällman Almén, Khurram Maqbool, Manfred Grabherr, Alvaro Martinez-Barrio, Marta Promerová, Carl-Johan Rubin, Chao Wang, Neda Zamani, et al. Evolution of darwin's finches and their beaks revealed by genome sequencing. *Nature*, 518(7539):371–375, 2015.

[155] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 05 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp324. URL https://doi.org/10.1093/bioinformatics/btp324.

[156] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 06 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp352. URL https://doi.org/10.1093/bioinformatics/btp352.

[157] Sangeet Lamichhaney, Fan Han, Jonas Berglund, Chao Wang, Markus Sällman Almén, Matthew T. Webster, B. Rosemary Grant, Peter R. Grant, and Leif Andersson. A beak size locus in darwin's finches facilitated character displacement during a drought. *Science*, 352 (6284):470–474, 2016. doi: 10.1126/science.aad8786. URL https://www.science.org/doi/abs/10.1126/science.aad8786.

[158] Stepfanie M Aguillon, Tristram O Dodge, Gabriel A Preising, and Molly Schumer. Introgression. *Current Biology*, 32(16):R865–R868, 2022.

[159] Thomas J. Sharpton. An introduction to the analysis of shotgun metagenomic data. *Frontiers in Plant Science*, 5, 2014. ISSN 1664-462X. doi: 10.3389/fpls.2014.00209. URL https://www.frontiersin.org/articles/10.3389/fpls.2014.00209.

[160] Florian P Breitwieser, Jennifer Lu, and Steven L Salzberg. A review of methods and databases for metagenomic classification and assembly. *Briefings in Bioinformatics*, 20(4): 1125–1136, 09 2017. ISSN 1477-4054. doi: 10.1093/bib/bbx120. URL https://doi.org/10.1093/bib/bbx120.

[161] Mads Albertsen, Philip Hugenholtz, Adam Skarshewski, Kåre L Nielsen, Gene W Tyson, and Per H Nielsen. Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes. *Nature Biotechnology*, 31(6):533–538, 2013. doi: 10.1038/nbt.2579. URL https://doi.org/10.1038/nbt.2579.

[162] Luisa W. Hugerth, John Larsson, Johannes Alneberg, Markus V. Lindh, Catherine Legrand, Jarone Pinhassi, and Anders F. Andersson. Metagenome-assembled genomes uncover a global brackish microbiome. *Genome Biology*, 16(1):279, 2015. doi: 10.1186/s13059-015-0834-7. URL https://doi.org/10.1186/s13059-015-0834-7.

[163] Robert Schmieder and Robert Edwards. Quality control and preprocessing of metagenomic datasets. *Bioinformatics*, 27(6):863–864, 01 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr026. URL https://doi.org/10.1093/bioinformatics/btr026.

[164] Anthony M. Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 04 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu170. URL https://doi.org/10.1093/bioinformatics/btu170.

[165] Simon Andrews et al. FastQC: a quality control tool for high throughput sequence data, 2010.

[166] Wei Shen, Shuai Le, Yan Li, and Fuquan Hu. SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation. *PLOS ONE*, 11(10):1–10, 10 2016. doi: 10.1371/journal.pone.0163962. URL https://doi.org/10.1371/journal.pone.0163962.

[167] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17(1):10–12, 2011. ISSN 2226-6089. doi: 10.14806/ej.17.1.200. URL https://journal.embnet.org/index.php/embnetjournal/article/view/200.

[168] Brian Bushnell, Jonathan Rood, and Esther Singer. Bbmerge – accurate paired shotgun read merging via overlap. *PLOS ONE*, 12(10):1–15, 10 2017. doi: 10.1371/journal.pone.0185056. URL https://doi.org/10.1371/journal.pone.0185056.

[169] NA Joshi, JN2011 Fass, et al. Sickle: A sliding-window, adaptive, quality-based trimming tool for fastq files (version 1.33)[software], 2011. URL https://github.com/najoshi/sickle.

[170] Alla Mikheenko, Vladislav Saveliev, and Alexey Gurevich. MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics*, 32(7):1088–1090, 11 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv697. URL https://doi.org/10.1093/bioinformatics/btv697.

[171] Donovan H. Parks, Michael Imelfort, Connor T. Skennerton, Philip Hugenholtz, and Gene W. Tyson. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Research*, 25(7):1043–1055, 2015. doi: 10.1101/gr.186072.114. URL http://genome.cshlp.org/content/25/7/1043.abstract.

[172] Fernando Meyer, Peter Hofmann, Peter Belmann, Ruben Garrido-Oter, Adrian Fritz, Alexander Sczyrba, and Alice C McHardy. AMBER: Assessment of Metagenome BinnERs. *GigaScience*, 7(6), 06 2018. ISSN 2047-217X. doi: 10.1093/gigascience/giy069. URL https://doi.org/10.1093/gigascience/giy069. giy069.

[173] Rayan Chikhi and Guillaume Rizk. Space-efficient and exact de Bruijn graph representation based on a Bloom filter. *Algorithms for Molecular Biology*, 8(1):22, 2013. doi: 10.1186/1748-7188-8-22. URL https://doi.org/10.1186/1748-7188-8-22.

[174] Toshiaki Namiki, Tsuyoshi Hachiya, Hideaki Tanaka, and Yasubumi Sakakibara. MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads . *Nucleic Acids Research*, 40(20):e155–e155, 07 2012. ISSN 0305-1048. doi: 10.1093/nar/gks678. URL https://doi.org/10.1093/nar/gks678.

[175] Afiahayati, Kengo Sato, and Yasubumi Sakakibara. MetaVelvet-SL: an extension of the Velvet assembler to a de novo metagenomic assembler utilizing supervised learning. *DNA Research*, 22(1):69–77, 11 2014. ISSN 1340-2838. doi: 10.1093/dnares/dsu041. URL https://doi.org/10.1093/dnares/dsu041.

[176] Sébastien Boisvert, Frédéric Raymond, Élénie Godzaridis, François Laviolette, and Jacques Corbeil. Ray Meta: scalable de novo metagenome assembly and profiling. *Genome Biol-*

ogy, 13(12):R122, 2012. doi: 10.1186/gb-2012-13-12-r122. URL https://doi.org/10.1186/gb-2012-13-12-r122.

[177] Yu Peng, Henry C. M. Leung, S. M. Yiu, and Francis Y. L. Chin. IDBA – A Practical Iterative de Bruijn Graph De Novo Assembler. In Bonnie Berger, editor, *Research in Computational Molecular Biology*, pages 426–440, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-12683-3.

[178] Yu Peng, Henry C. M. Leung, S. M. Yiu, and Francis Y. L. Chin. IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, 28(11):1420–1428, 04 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts174. URL https://doi.org/10.1093/bioinformatics/bts174.

[179] Sergey Nurk, Dmitry Meleshko, Anton Korobeynikov, and Pavel A. Pevzner. metaSPAdes: a new versatile metagenomic assembler. *Genome Research*, 27(5):824–834, 2017. doi: 10.1101/gr.213959.116. URL http://genome.cshlp.org/content/27/5/824.abstract.

[180] Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, and Tak-Wah Lam. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, 31(10):1674–1676, 01 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv033. URL https://doi.org/10.1093/bioinformatics/btv033.

[181] Todd J. Treangen, Sergey Koren, Daniel D. Sommer, Bo Liu, Irina Astrovskaya, Brian Ondov, Aaron E. Darling, Adam M. Phillippy, and Mihai Pop. MetAMOS: a modular and open source metagenomic assembly and analysis pipeline. *Genome Biology*, 14(1):R2, 2013. doi: 10.1186/gb-2013-14-1-r2. URL https://doi.org/10.1186/gb-2013-14-1-r2.

[182] Matthew Scholz, Chien-Chi Lo, and Patrick S. G. Chain. Improved Assemblies Using a Source-Agnostic Pipeline for MetaGenomic Assembly by Merging (MeGAMerge) of Contigs. *Scientific Reports*, 4(1):6480, 2014. doi: 10.1038/srep06480. URL https://doi.org/10.1038/srep06480.

[183] Riccardo Vicedomini, Francesco Vezzi, Simone Scalabrin, Lars Arvestad, and Alberto Policriti. GAM-NGS: genomic assemblies merger for next generation sequencing. *BMC Bioinformatics*, 14(7):S6, 2013. doi: 10.1186/1471-2105-14-S7-S6. URL https://doi.org/10.1186/1471-2105-14-S7-S6.

[184] Wolfgang Gerlach and Jens Stoye. Taxonomic classification of metagenomic shotgun sequences with CARMA3. *Nucleic Acids Research*, 39(14):e91–e91, 05 2011. ISSN 0305-1048. doi: 10.1093/nar/gkr225. URL https://doi.org/10.1093/nar/gkr225.

[185] Bo Liu, Theodore Gibbons, Mohammad Ghodsi, and Mihai Pop. MetaPhyler: Taxonomic profiling for metagenomic sequences. In *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 95–100, 2010. doi: 10.1109/BIBM.2010.5706544.

[186] Monzoorul Haque Mohammed, Tarini Shankar Ghosh, Nitin Kumar Singh, and Sharmila S. Mande. SPHINX—an algorithm for taxonomic binning of metagenomic sequences. *Bioinformatics*, 27(1):22–30, 10 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq608. URL https://doi.org/10.1093/bioinformatics/btq608.

[187] Ivan Gregor, Johannes Dröge, Melanie Schirmer, Christopher Quince, and Alice C. McHardy. *PhyloPythiaS+*: a self-training method for the rapid reconstruction of low-ranking taxonomic bins from metagenomes. *PeerJ*, 4:e1603, February 2016. ISSN 2167-8359. doi: 10.7717/peerj.1603. URL https://doi.org/10.7717/peerj.1603.

[188] Arthur Brady and Steven L Salzberg. Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nature Methods*, 6(9):673–676, 2009. doi: 10.1038/nmeth.1358. URL https://doi.org/10.1038/nmeth.1358.

[189] Folker Meyer, Saurabh Bagchi, Somali Chaterji, Wolfgang Gerlach, Ananth Grama, Travis Harrison, Tobias Paczian, William L Trimble, and Andreas Wilke. MG-RAST version 4—lessons learned from a decade of low-budget ultra-high-throughput metagenome analysis. *Briefings in Bioinformatics*, 20(4):1151–1159, 09 2017. ISSN 1477-4054. doi: 10.1093/bib/bbx105. URL https://doi.org/10.1093/bib/bbx105.

[190] Daniel H. Huson, Sina Beier, Isabell Flade, Anna Górska, Mohamed El-Hadidi, Suparna Mitra, Hans-Joachim Ruscheweyh, and Rewati Tappu. MEGAN Community Edition - Interactive Exploration and Analysis of Large-Scale Microbiome Sequencing Data. *PLOS Computational Biology*, 12(6):1–12, 06 2016. doi: 10.1371/journal.pcbi.1004957. URL https://doi.org/10.1371/journal.pcbi.1004957.

[191] Naryttza N. Diaz, Lutz Krause, Alexander Goesmann, Karsten Niehaus, and Tim W. Nattkemper. TACOA –Taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach. *BMC Bioinformatics*, 10(1):56, 2009. doi: 10.1186/1471-2105-10-56. URL https://doi.org/10.1186/1471-2105-10-56.

[192] I-Min A Chen, Ken Chu, Krishna Palaniappan, Manoj Pillay, Anna Ratner, Jinghua Huang, Marcel Huntemann, Neha Varghese, James R White, Rekha Seshadri, Tatyana Smirnova, Edward Kirton, Sean P Jungbluth, Tanja Woyke, Emiley A Eloe-Fadrosh, Natalia N Ivanova, and Nikos C Kyrpides. IMG/M v.5.0: an integrated data management and comparative analysis system for microbial genomes and microbiomes. *Nucleic Acids Research*, 47(D1): D666–D677, 10 2018. ISSN 0305-1048. doi: 10.1093/nar/gky901. URL https://doi.org/10.1093/nar/gky901.

[193] Marc Strous, Beate Kraft, Regina Bisdorf, and Halina E Tegetmeyer. The binning of metagenomic contigs for microbial physiology of mixed cultures. *Frontiers in microbiology*, 3:410, 2012. ISSN 1664-302X. doi: 10.3389/fmicb.2012.00410. URL https://www.frontiersin.org/articles/10.3389/fmicb.2012.00410.

[194] Yu-Wei Wu and Yuzhen Ye. A Novel Abundance-Based Algorithm for Binning Metagenomic

Sequences Using l-tuples. *Journal of Computational Biology*, 18(3):523–534, 2011. doi: 10.1089/cmb.2010.0245. URL https://doi.org/10.1089/cmb.2010.0245. PMID: 21385052.

[195] Andrey Kislyuk, Srijak Bhatnagar, Jonathan Dushoff, and Joshua S. Weitz. Unsupervised statistical clustering of environmental shotgun sequences. *BMC Bioinformatics*, 10(1):316, 2009. doi: 10.1186/1471-2105-10-316. URL https://doi.org/10.1186/1471-2105-10-316.

[196] Ying Wang, Haiyan Hu, and Xiaoman Li. MBBC: an efficient approach for metagenomic binning based on clustering. *BMC Bioinformatics*, 16(1):36, 2015. doi: 10.1186/s12859-015-0473-8. URL https://doi.org/10.1186/s12859-015-0473-8.

[197] H Bjørn Nielsen, Mathieu Almeida, Agnieszka Sierakowska Juncker, Simon Rasmussen, Junhua Li, Shinichi Sunagawa, Damian R Plichta, Laurent Gautier, Anders G Pedersen, Emmanuelle Le Chatelier, Eric Pelletier, Ida Bonde, Trine Nielsen, Chaysavanh Manichanh, Manimozhiyan Arumugam, Jean-Michel Batto, Marcelo B Quintanilha dos Santos, Nikolaj Blom, Natalia Borruel, Kristoffer S Burgdorf, Fouad Boumezbeur, Francesc Casellas, Joël Doré, Piotr Dworzynski, Francisco Guarner, Torben Hansen, Falk Hildebrand, Rolf S Kaas, Sean Kennedy, Karsten Kristiansen, Jens Roat Kultima, Pierre Léonard, Florence Levenez, Ole Lund, Bouziane Moumen, Denis Le Paslier, Nicolas Pons, Oluf Pedersen, Edi Prifti, Junjie Qin, Jeroen Raes, Søren Sørensen, Julien Tap, Sebastian Tims, David W Ussery, Takuji Yamada, Agnieszka S Juncker, Pierre Leonard, Pierre Renault, Thomas Sicheritz-Ponten, Peer Bork, Jun Wang, Søren Brunak, S Dusko Ehrlich, Alexandre Jamet, Alexandre Mérieux, Antonella Cultrone, Antonio Torrejon, Benoit Quinquis, Christian Brechot, Christine Delorme, Christine M'Rini, Willem M de Vos, Emmanuelle Maguin, Encarna Varela, Eric Guedon, Falony Gwen, Florence Haimet, François Artiguenave, Gaetana Vandemeulebrouck, Gérard Denariaz, Ghalia Khaci, Hervé Blottière, Jan Knol, Jean Weissenbach, Johan E T van Hylckama Vlieg, Jørgensen Torben, Julian Parkhill, Keith Turner, Maarten van de Guchte, Maria Antolin, Maria Rescigno, Michiel Kleerebezem, Muriel Derrien, Nathalie Galleron, Nicolas Sanchez, Niels Grarup, Patrick Veiga, Raish Oozeer, Rozenn Dervyn, Séverine Layec, Thomas Bruls, Yohanan Winogradski, Zoetendal Erwin G, and MetaHIT Consortium. Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. *Nature Biotechnology*, 32 (8):822–828, 2014. doi: 10.1038/nbt.2939. URL https://doi.org/10.1038/nbt.2939.

[198] Yi Wang, Henry CM Leung, Siu-Ming Yiu, and Francis YL Chin. MetaCluster 4.0: a novel binning algorithm for NGS reads and huge number of species. *Journal of Computational Biology*, 19(2):241–249, 2012. doi: 10.1089/cmb.2011.0276. URL https://doi.org/10.1089/cmb.2011.0276. PMID: 22300323.

[199] Yu-Wei Wu, Blake A. Simmons, and Steven W. Singer. MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics*, 32(4): 605–607, 10 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv638. URL https://doi.org/10.1093/bioinformatics/btv638.

[200] Dongwan D Kang, Feng Li, Edward Kirton, Ashleigh Thomas, Rob Egan, Hong An, and

Zhong Wang. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ*, 7:e7359, 2019. ISSN 2167-8359. doi: 10.7717/peerj.7359. URL https://doi.org/10.7717/peerj.7359.

[201] Johannes Alneberg, Brynjar Smári Bjarnason, Ino de Bruijn, Melanie Schirmer, Joshua Quick, Umer Z Ijaz, Leo Lahti, Nicholas J Loman, Anders F Andersson, and Christopher Quince. Binning metagenomic contigs by coverage and composition. *Nature Methods*, 11(11):1144–1146, 2014. doi: 10.1038/nmeth.3103. URL https://doi.org/10.1038/nmeth.3103.

[202] Yang Young Lu, Ting Chen, Jed A Fuhrman, and Fengzhu Sun. COCACOLA: binning metagenomic contigs using sequence COmposition, read CoverAge, CO-alignment and paired-end read LinkAge. *Bioinformatics*, 33(6):791–798, 06 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw290. URL https://doi.org/10.1093/bioinformatics/btw290.

[203] David R. Kelley and Steven L. Salzberg. Clustering metagenomic sequences with interpolated Markov models. *BMC Bioinformatics*, 11(1):544, 2010. doi: 10.1186/1471-2105-11-544. URL https://doi.org/10.1186/1471-2105-11-544.

[204] Sourav Chatterji, Ichitaro Yamazaki, Zhaojun Bai, and Jonathan A Eisen. CompostBin: A DNA composition-based algorithm for binning environmental shotgun reads. In *Annual International Conference on Research in Computational Molecular Biology*, pages 17–28, Berlin, Heidelberg, 2008. Springer, Springer Berlin Heidelberg.

[205] Wei-Zhi Song and Torsten Thomas. Binning_refiner: improving genome bins through the combination of different binning programs. *Bioinformatics*, 33(12):1873–1875, 02 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx086. URL https://doi.org/10.1093/bioinformatics/btx086.

[206] Christian M. K. Sieber, Alexander J. Probst, Allison Sharrar, Brian C. Thomas, Matthias Hess, Susannah G. Tringe, and Jillian F. Banfield. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. *Nature Microbiology*, 3(7):836–843, 2018. doi: 10.1038/s41564-018-0171-1. URL https://doi.org/10.1038/s41564-018-0171-1.

[207] Bertjan Broeksema, Magdalena Calusinska, Fintan McGee, Klaas Winter, Francesco Bongiovanni, Xavier Goux, Paul Wilmes, Philippe Delfosse, and Mohammad Ghoniem. ICoVeR –an interactive visualization tool for verification and refinement of metagenomic bins. *BMC Bioinformatics*, 18(1):233, 2017. doi: 10.1186/s12859-017-1653-5. URL https://doi.org/10.1186/s12859-017-1653-5.

[208] Gherman V. Uritskiy, Jocelyne DiRuggiero, and James Taylor. MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. *Microbiome*, 6(1):158, 2018. doi: 10.1186/s40168-018-0541-1. URL https://doi.org/10.1186/s40168-018-0541-1.

[209] Wenhan Zhu, Alexandre Lomsadze, and Mark Borodovsky. Ab initio gene identification

in metagenomic sequences . *Nucleic Acids Research*, 38(12):e132–e132, 04 2010. ISSN 0305-1048. doi: 10.1093/nar/gkq275. URL https://doi.org/10.1093/nar/gkq275.

[210] Hideki Noguchi, Takeaki Taniguchi, and Takehiko Itoh. MetaGeneAnnotator: Detecting Species-Specific Patterns of Ribosomal Binding Site for Precise Gene Prediction in Anonymous Prokaryotic and Phage Genomes. *DNA Research*, 15(6):387–396, 10 2008. ISSN 1340-2838. doi: 10.1093/dnares/dsn027. URL https://doi.org/10.1093/dnares/dsn027.

[211] David R. Kelley, Bo Liu, Arthur L. Delcher, Mihai Pop, and Steven L. Salzberg. Gene prediction with Glimmer for metagenomic sequences augmented by classification and clustering. *Nucleic Acids Research*, 40(1):e9–e9, 11 2011. ISSN 0305-1048. doi: 10.1093/nar/gkr1067. URL https://doi.org/10.1093/nar/gkr1067.

[212] Marcel Huntemann, Natalia N. Ivanova, Konstantinos Mavromatis, H. James Tripp, David Paez-Espino, Kristin Tennessen, Krishnaveni Palaniappan, Ernest Szeto, Manoj Pillay, I-Min A. Chen, Amrita Pati, Torben Nielsen, Victor M. Markowitz, and Nikos C. Kyrpides. The standard operating procedure of the DOE-JGI Metagenome Annotation Pipeline (MAP v.4). *Standards in Genomic Sciences*, 11(1):17, 2016. doi: 10.1186/s40793-016-0138-x. URL https://doi.org/10.1186/s40793-016-0138-x.

[213] Doug Hyatt, Philip F. LoCascio, Loren J. Hauser, and Edward C. Uberbacher. Gene and translation initiation site prediction in metagenomic sequences. *Bioinformatics*, 28(17): 2223–2230, 07 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts429. URL https://doi.org/10.1093/bioinformatics/bts429.

[214] Mina Rho, Haixu Tang, and Yuzhen Ye. FragGeneScan: predicting genes in short and error-prone reads. *Nucleic Acids Research*, 38(20):e191–e191, 08 2010. ISSN 0305-1048. doi: 10.1093/nar/gkq747. URL https://doi.org/10.1093/nar/gkq747.

[215] Vardges Ter-Hovhannisyan, Alexandre Lomsadze, Yury O. Chernoff, and Mark Borodovsky. Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training. *Genome Research*, 18(12):1979–1990, 2008. doi: 10.1101/gr.081612.108. URL http://genome.cshlp.org/content/18/12/1979.abstract.

[216] Alexandre Lomsadze, Vardges Ter-Hovhannisyan, Yury O. Chernoff, and Mark Borodovsky. Gene identification in novel eukaryotic genomes by self-training algorithm. *Nucleic Acids Research*, 33(20):6494–6506, 01 2005. ISSN 0305-1048. doi: 10.1093/nar/gki937. URL https://doi.org/10.1093/nar/gki937.

[217] Mario Stanke and Burkhard Morgenstern. Augustus: a web server for gene prediction in eukaryotes that allows user-defined constraints. *Nucleic Acids Research*, 33(suppl_2): W465–W467, 07 2005. ISSN 0305-1048. doi: 10.1093/nar/gki458. URL https://doi.org/10.1093/nar/gki458.

[218] A Souvorov, Y Kapustin, B Kiryutin, V Chetvernin, T Tatusova, and D Lipman. Gnomon–

ncbi eukaryotic gene prediction tool. *National Center for Biotechnology Information*, pages 1–24, 2010.

[219] Ian Korf. Gene finding in novel genomes. *BMC bioinformatics*, 5:1–9, 2004.

[220] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990. ISSN 0022-2836. doi: https://doi.org/10.1016/S0022-2836(05)80360-2. URL https://www.sciencedirect.com/science/article/pii/S0022283605803602.

[221] Robert D. Finn, Alex Bateman, Jody Clements, Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, Andreas Heger, Kirstie Hetherington, Liisa Holm, Jaina Mistry, Erik L. L. Sonnhammer, John Tate, and Marco Punta. Pfam: the protein families database. *Nucleic Acids Research*, 42(D1):D222–D230, 11 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt1223. URL https://doi.org/10.1093/nar/gkt1223.

[222] Sarah Hunter, Rolf Apweiler, Teresa K Attwood, Amos Bairoch, Alex Bateman, David Binns, Peer Bork, Ujjwal Das, Louise Daugherty, Lauranne Duquenne, et al. InterPro: the integrative protein signature database. *Nucleic Acids Research*, 37(suppl_1):D211–D215, 10 2008. ISSN 0305-1048. doi: 10.1093/nar/gkn785. URL https://doi.org/10.1093/nar/gkn785.

[223] Clotilde Claudel-Renard, Claude Chevalet, Thomas Faraut, and Daniel Kahn. Enzyme-specific profiles for genome annotation: PRIAM. *Nucleic Acids Research*, 31(22):6633–6639, 11 2003. ISSN 0305-1048. doi: 10.1093/nar/gkg847. URL https://doi.org/10.1093/nar/gkg847.

[224] Minoru Kanehisa, Susumu Goto, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Research*, 42(D1):D199–D205, 11 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt1076. URL https://doi.org/10.1093/nar/gkt1076.

[225] Peter D. Karp, Monica Riley, Suzanne M. Paley, and Alida Pellegrini-Toole. The MetaCyc Database. *Nucleic Acids Research*, 30(1):59–61, 01 2002. ISSN 0305-1048. doi: 10.1093/nar/30.1.59. URL https://doi.org/10.1093/nar/30.1.59.

[226] Scott Federhen. The NCBI Taxonomy database. *Nucleic Acids Research*, 40(D1):D136–D143, 12 2011. ISSN 0305-1048. doi: 10.1093/nar/gkr1178. URL https://doi.org/10.1093/nar/gkr1178.

[227] Torsten Seemann. Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14): 2068–2069, 03 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu153. URL https://doi.org/10.1093/bioinformatics/btu153.

[228] Yasuhiro Tanizawa, Takatomo Fujisawa, and Yasukazu Nakamura. DFAST: a flexible prokaryotic genome annotation pipeline for faster genome publication. *Bioinformatics*,

34(6):1037–1039, 11 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx713. URL https://doi.org/10.1093/bioinformatics/btx713.

[229] Tatiana Tatusova, Michael DiCuccio, Azat Badretdin, Vyacheslav Chetvernin, Eric P. Nawrocki, Leonid Zaslavsky, Alexandre Lomsadze, Kim D. Pruitt, Mark Borodovsky, and James Ostell. NCBI prokaryotic genome annotation pipeline. *Nucleic Acids Research*, 44(14):6614–6624, 06 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw569. URL https://doi.org/10.1093/nar/gkw569.

[230] Xiaoli Dong and Marc Strous. An integrated pipeline for annotation and visualization of metagenomic contigs. *Frontiers in Genetics*, 10:999, 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.00999. URL https://www.frontiersin.org/articles/10.3389/fgene.2019.00999.

[231] Benjamin Buchfink, Chao Xie, and Daniel H Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12(1):59–60, 2015. doi: 10.1038/nmeth.3176. URL https://doi.org/10.1038/nmeth.3176.

[232] Robert D Stewart, Marc D Auffret, Timothy J Snelling, Rainer Roehe, and Mick Watson. MAGpy: a reproducible pipeline for the downstream analysis of metagenome-assembled genomes (MAGs). *Bioinformatics*, 35(12):2150–2152, 11 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty905. URL https://doi.org/10.1093/bioinformatics/bty905.

[233] Kevin P Keegan, Elizabeth M Glass, and Folker Meyer. MG-RAST, a metagenomics service for analysis of microbial community structure and function. In *Microbial environmental genomics (MEG)*, pages 207–233. Springer, New York, NY, 2016. ISBN 978-1-4939-3369-3. doi: 10.1007/978-1-4939-3369-3_13. URL https://doi.org/10.1007/978-1-4939-3369-3_13.

[234] Moreno Zolfo, Adrian Tett, Olivier Jousson, Claudio Donati, and Nicola Segata. MetaMLST: multi-locus strain-level bacterial typing from metagenomic samples. *Nucleic Acids Research*, 45(2):e7–e7, 09 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw837. URL https://doi.org/10.1093/nar/gkw837.

[235] Duy Tin Truong, Adrian Tett, Edoardo Pasolli, Curtis Huttenhower, and Nicola Segata. Microbial strain-level population structure and genetic diversity from metagenomes. *Genome Research*, 27(4):626–638, 2017. doi: 10.1101/gr.216242.116. URL http://genome.cshlp.org/content/27/4/626.abstract.

[236] Christopher Quince, Tom O. Delmont, Sébastien Raguideau, Johannes Alneberg, Aaron E. Darling, Gavin Collins, and A. Murat Eren. DESMAN: a new tool for de novo extraction of strains from metagenomes. *Genome Biology*, 18(1):181, 2017. doi: 10.1186/s13059-017-1309-9. URL https://doi.org/10.1186/s13059-017-1309-9.

[237] Paul Igor Costea, Robin Munch, Luis Pedro Coelho, Lucas Paoli, Shinichi Sunagawa, and Peer Bork. metaSNV: A tool for metagenomic strain level analysis. *PLOS ONE*, 12(7):1–9,

07 2017. doi: 10.1371/journal.pone.0182392. URL https://doi.org/10.1371/journal.pone.0182392.

[238] Robert C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 08 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq461. URL https://doi.org/10.1093/bioinformatics/btq461.

[239] W. James Kent. BLAT—The BLAST-Like Alignment Tool. *Genome Research*, 12(4): 656–664, 2002. doi: 10.1101/gr.229202. URL http://genome.cshlp.org/content/12/4/656.abstract.

[240] Yongan Zhao, Haixu Tang, and Yuzhen Ye. RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data. *Bioinformatics*, 28(1): 125–126, 10 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr595. URL https://doi.org/10.1093/bioinformatics/btr595.

[241] Anthony Westbrook, Jordan Ramsdell, Taruna Schuelke, Louisa Normington, R Daniel Bergeron, W Kelley Thomas, and Matthew D MacManes. PALADIN: protein alignment for functional profiling whole metagenome shotgun data. *Bioinformatics*, 33(10):1473–1478, 01 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx021. URL https://doi.org/10.1093/bioinformatics/btx021.

[242] Cuncong Zhong, Youngik Yang, and Shibu Yooseph. GRASP2: fast and memory-efficient gene-centric assembly and homolog search for metagenomic sequencing data. *BMC Bioinformatics*, 20(11):276, 2019. doi: 10.1186/s12859-019-2818-1. URL https://doi.org/10.1186/s12859-019-2818-1.

[243] Fatemeh Sharifi and Yuzhen Ye. From gene annotation to function prediction for metagenomics. In *Protein Function Prediction*, pages 27–34. Springer, New York, NY, 2017. ISBN 978-1-4939-7015-5. doi: 10.1007/978-1-4939-7015-5_3. URL https://doi.org/10.1007/978-1-4939-7015-5_3.

[244] Jens Roat Kultima, Luis Pedro Coelho, Kristoffer Forslund, Jaime Huerta-Cepas, Simone S. Li, Marja Driessen, Anita Yvonne Voigt, Georg Zeller, Shinichi Sunagawa, and Peer Bork. MOCAT2: a metagenomic assembly, annotation and profiling framework. *Bioinformatics*, 32(16):2520–2523, 04 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw183. URL https://doi.org/10.1093/bioinformatics/btw183.

[245] Stuart M Brown, Hao Chen, Yuhan Hao, Bobby P Laungani, Thahmina A Ali, Changsu Dong, Carlos Lijeron, Baekdoo Kim, Claudia Wultsch, Zhiheng Pei, and Konstantinos Krampis. MGS-Fast: Metagenomic shotgun data fast annotation using microbial gene catalogs. *GigaScience*, 8(4), 04 2019. ISSN 2047-217X. doi: 10.1093/gigascience/giz020. URL https://doi.org/10.1093/gigascience/giz020. giz020.

[246] Eric A. Franzosa, Lauren J. McIver, Gholamali Rahnavard, Luke R. Thompson, Melanie

Schirmer, George Weingart, Karen Schwarzberg Lipson, Rob Knight, J. Gregory Caporaso, Nicola Segata, and Curtis Huttenhower. Species-level functional profiling of metagenomes and metatranscriptomes. *Nature Methods*, 15(11):962–968, 2018. doi: 10.1038/s41592-018-0176-y. URL https://doi.org/10.1038/s41592-018-0176-y.

[247] Gustavo Arango-Argoty, Gargi Singh, Lenwood S. Heath, Amy Pruden, Weidong Xiao, and Liqing Zhang. MetaStorm: A Public Resource for Customizable Metagenomics Annotation. *PLOS ONE*, 11(9):1–13, 09 2016. doi: 10.1371/journal.pone.0162442. URL https://doi.org/10.1371/journal.pone.0162442.

[248] Derrick E. Wood and Steven L. Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):R46, 2014. doi: 10.1186/gb-2014-15-3-r46. URL https://doi.org/10.1186/gb-2014-15-3-r46.

[249] Rachid Ounit, Steve Wanamaker, Timothy J. Close, and Stefano Lonardi. CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, 16(1):236, 2015. doi: 10.1186/s12864-015-1419-2. URL https://doi.org/10.1186/s12864-015-1419-2.

[250] Derrick E. Wood, Jennifer Lu, and Ben Langmead. Improved metagenomic analysis with Kraken 2. *Genome Biology*, 20(1):257, 2019. doi: 10.1186/s13059-019-1891-0. URL https://doi.org/10.1186/s13059-019-1891-0.

[251] Steven Flygare, Keith Simmon, Chase Miller, Yi Qiao, Brett Kennedy, Tonya Di Sera, Erin H. Graf, Keith D. Tardif, Aurélie Kapusta, Shawn Rynearson, Chris Stockmann, Krista Queen, Suxiang Tong, Karl V. Voelkerding, Anne Blaschke, Carrie L. Byington, Seema Jain, Andrew Pavia, Krow Ampofo, Karen Eilbeck, Gabor Marth, Mark Yandell, and Robert Schlaberg. Taxonomer: an interactive metagenomics analysis portal for universal pathogen detection and host mRNA expression profiling. *Genome Biology*, 17(1):111, 2016. doi: 10.1186/s13059-016-0969-1. URL https://doi.org/10.1186/s13059-016-0969-1.

[252] Daehwan Kim, Li Song, Florian P. Breitwieser, and Steven L. Salzberg. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Research*, 26(12):1721–1729, 2016. doi: 10.1101/gr.210641.116. URL http://genome.cshlp.org/content/26/12/1721.abstract.

[253] Peter Menzel, Kim Lee Ng, and Anders Krogh. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications*, 7(1):11257, 2016. doi: 10.1038/ncomms11257. URL https://doi.org/10.1038/ncomms11257.

[254] André Corvelo, Wayne E. Clarke, Nicolas Robine, and Michael C. Zody. taxMaps: comprehensive and highly accurate taxonomic classification of short-read data in reasonable time. *Genome Research*, 28(5):751–758, 2018. doi: 10.1101/gr.225276.117. URL http://genome.cshlp.org/content/28/5/751.abstract.

[255] J. Dröge, I. Gregor, and A. C. McHardy. Taxator-tk: precise taxonomic assignment of metagenomes by fast approximation of evolutionary neighborhoods. *Bioinformatics*, 31(6): 817–824, 11 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu745. URL https://doi.org/10.1093/bioinformatics/btu745.

[256] Nicola Segata, Levi Waldron, Annalisa Ballarini, Vagheesh Narasimhan, Olivier Jousson, and Curtis Huttenhower. Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods*, 9(8):811–814, 2012. doi: 10.1038/nmeth.2066. URL https://doi.org/10.1038/nmeth.2066.

[257] Duy Tin Truong, Eric A Franzosa, Timothy L Tickle, Matthias Scholz, George Weingart, Edoardo Pasolli, Adrian Tett, Curtis Huttenhower, and Nicola Segata. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature Methods*, 12(10):902–903, 2015. doi: 10.1038/nmeth.3589. URL https://doi.org/10.1038/nmeth.3589.

[258] Weizhi Song, Kerrin Steensen, and Torsten Thomas. Hgtsim: a simulator for horizontal gene transfer (hgt) in microbial communities. *PeerJ*, 5:e4015, November 2017. ISSN 2167-8359. doi: 10.7717/peerj.4015. URL https://doi.org/10.7717/peerj.4015.

[259] Benjamin Vernot, Serena Tucci, Janet Kelso, Joshua G. Schraiber, Aaron B. Wolf, Rachel M. Gittelman, Michael Dannemann, Steffi Grote, Rajiv C. McCoy, Heather Norton, Laura B. Scheinfeldt, David A. Merriwether, George Koki, Jonathan S. Friedlaender, Jon Wakefield, Svante Pääbo, and Joshua M. Akey. Excavating Neandertal and Denisovan DNA from the genomes of Melanesian individuals. *Science*, 352(6282):235–239, 2016. doi: 10.1126/science.aad9416. URL https://www.science.org/doi/abs/10.1126/science.aad9416.

[260] Fernando Racimo, Davide Marnetto, and Emilia Huerta-Sánchez. Signatures of Archaic Adaptive Introgression in Present-Day Human Populations. *Molecular Biology and Evolution*, 34(2):296–317, 08 2016. ISSN 0737-4038. doi: 10.1093/molbev/msw216. URL https://doi.org/10.1093/molbev/msw216.

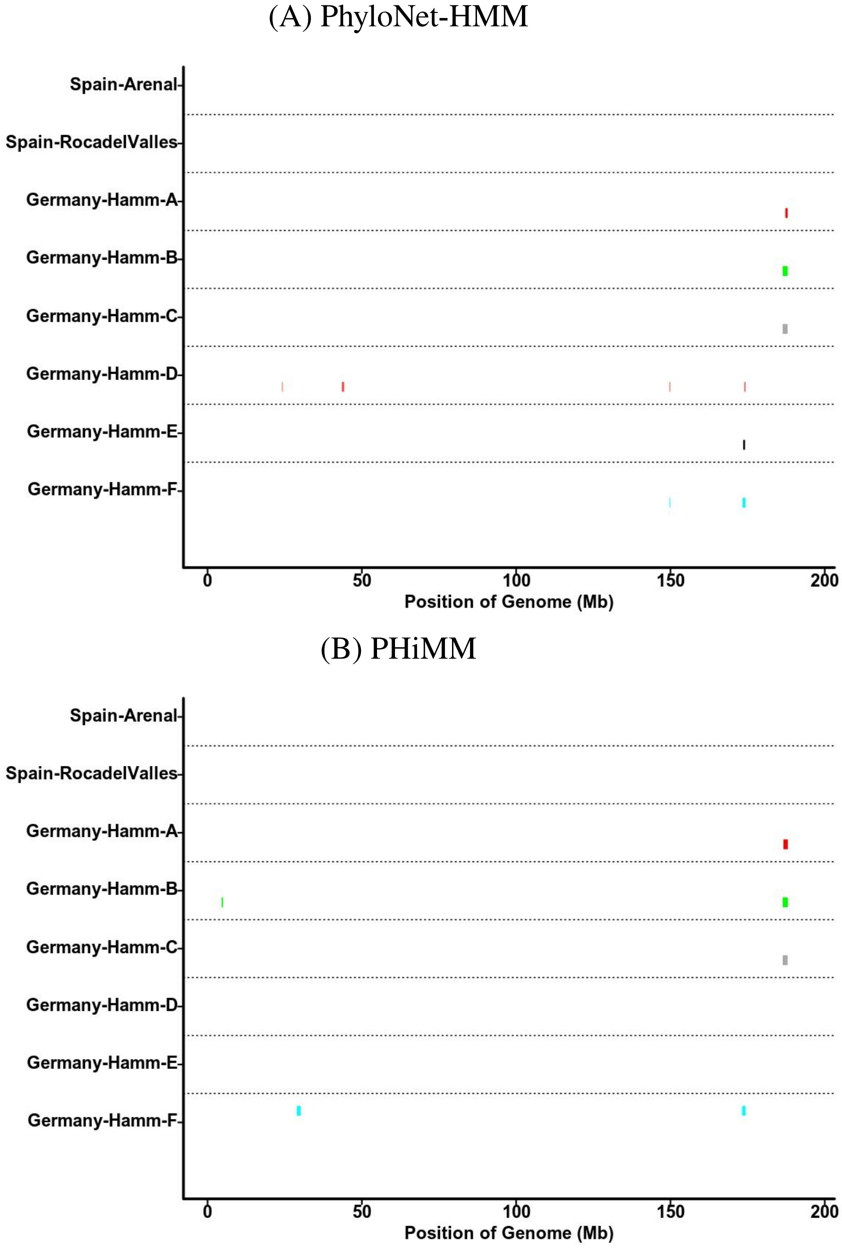
[261] Derek Setter, Sylvain Mousset, Xiaoheng Cheng, Rasmus Nielsen, Michael DeGiorgio, and Joachim Hermisson. VolcanoFinder: Genomic scans for adaptive introgression. *PLOS Genetics*, 16(6):1–44, 06 2020. doi: 10.1371/journal.pgen.1008867. URL https://doi.org/10.1371/journal.pgen.1008867.

[262] Graham Gower, Pablo Iáñez Picazo, Matteo Fumagalli, and Fernando Racimo. Detecting adaptive introgression in human evolution using convolutional neural networks. *eLife*, 10: e64669, may 2021. ISSN 2050-084X. doi: 10.7554/eLife.64669. URL https://doi.org/10.7554/eLife.64669.

[263] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

APPENDIX A

THE COMPARISON OF INTROGRESSION PATTERNS BETWEEN PHYLONET-HMM AND PHIMM ON EACH CHROMOSOME OF MOUSE DATA.



(A) PhyloNet-HMM

(B) PHiMM

Figure A.1 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 1.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].

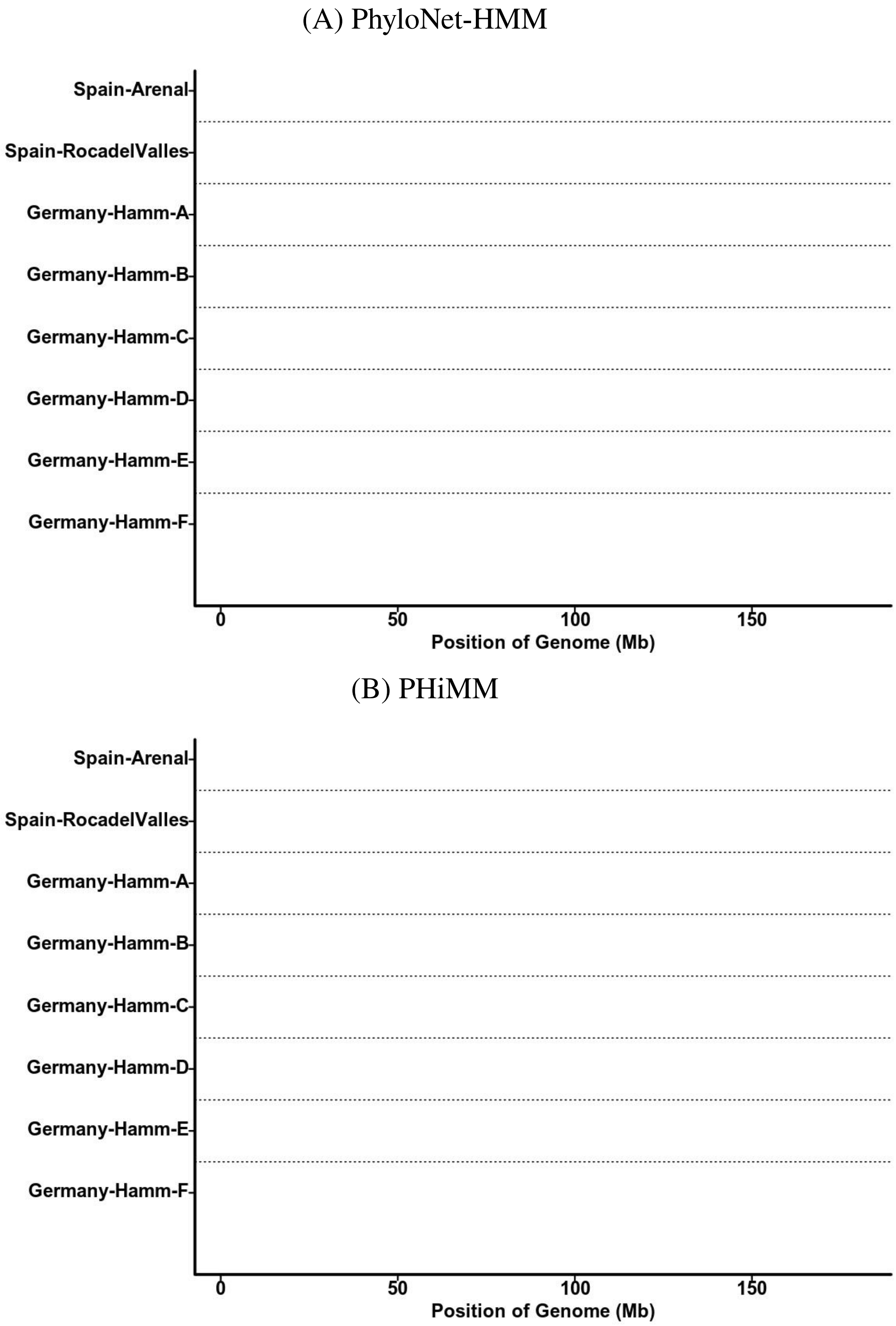


Figure A.2 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 2.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].

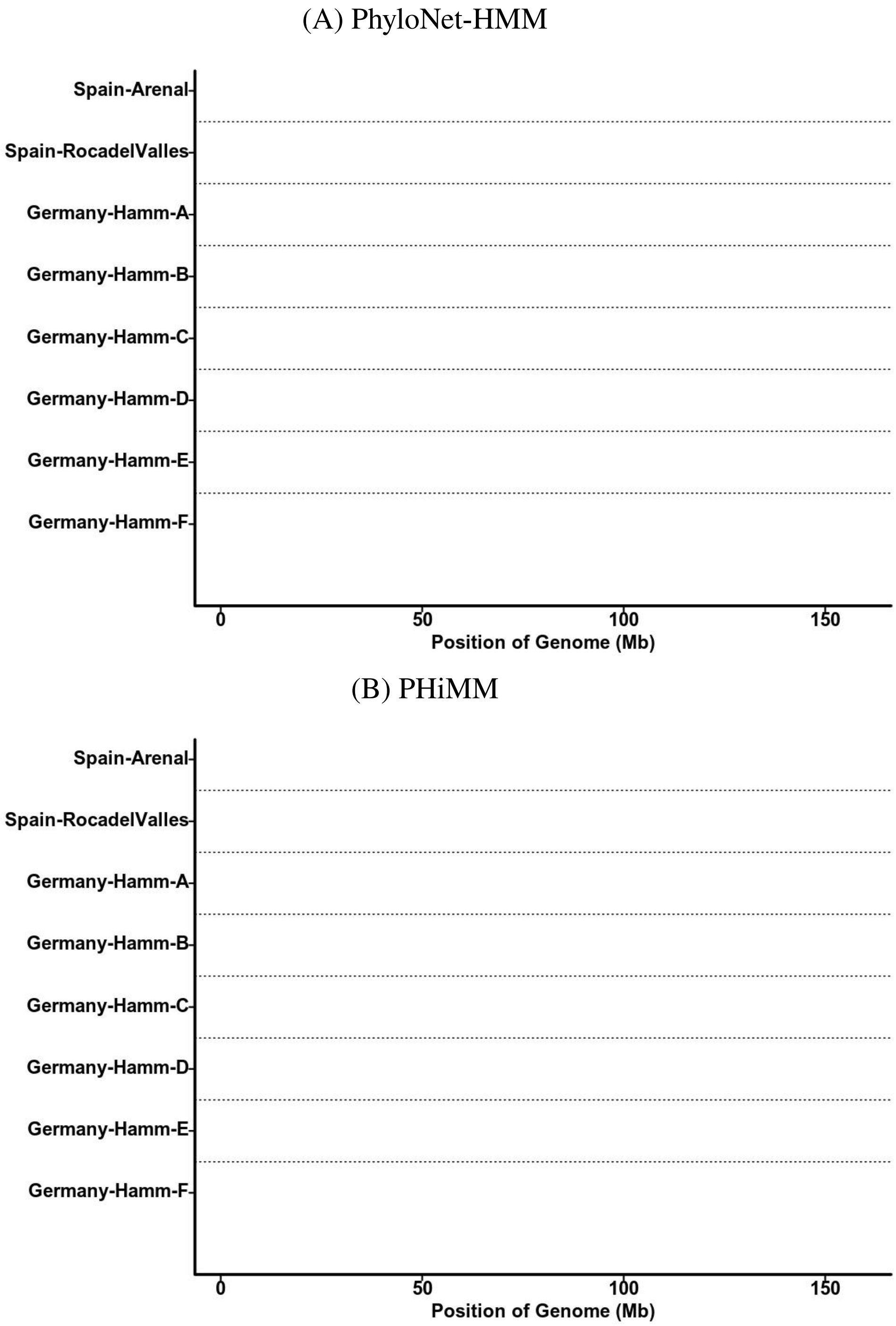
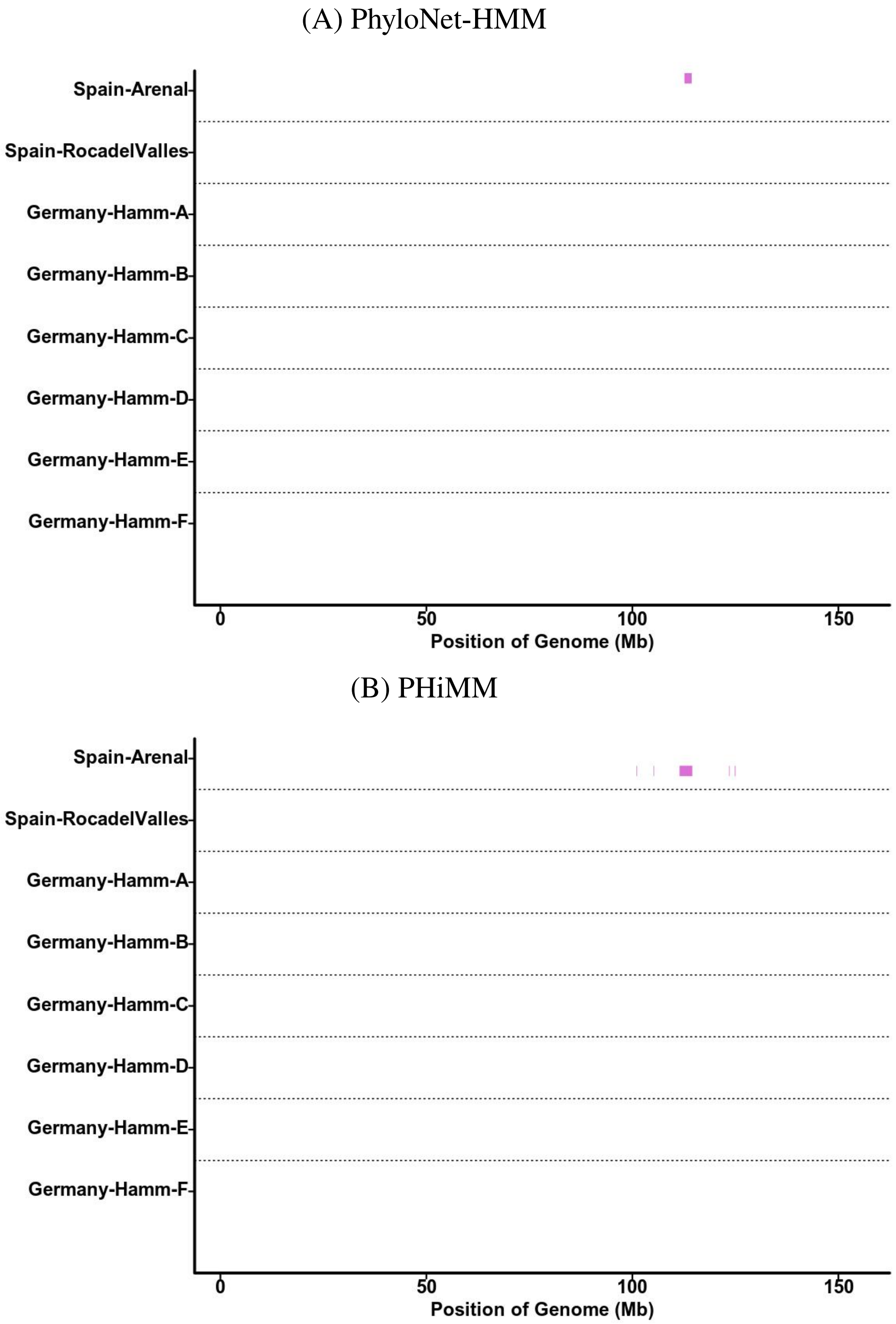


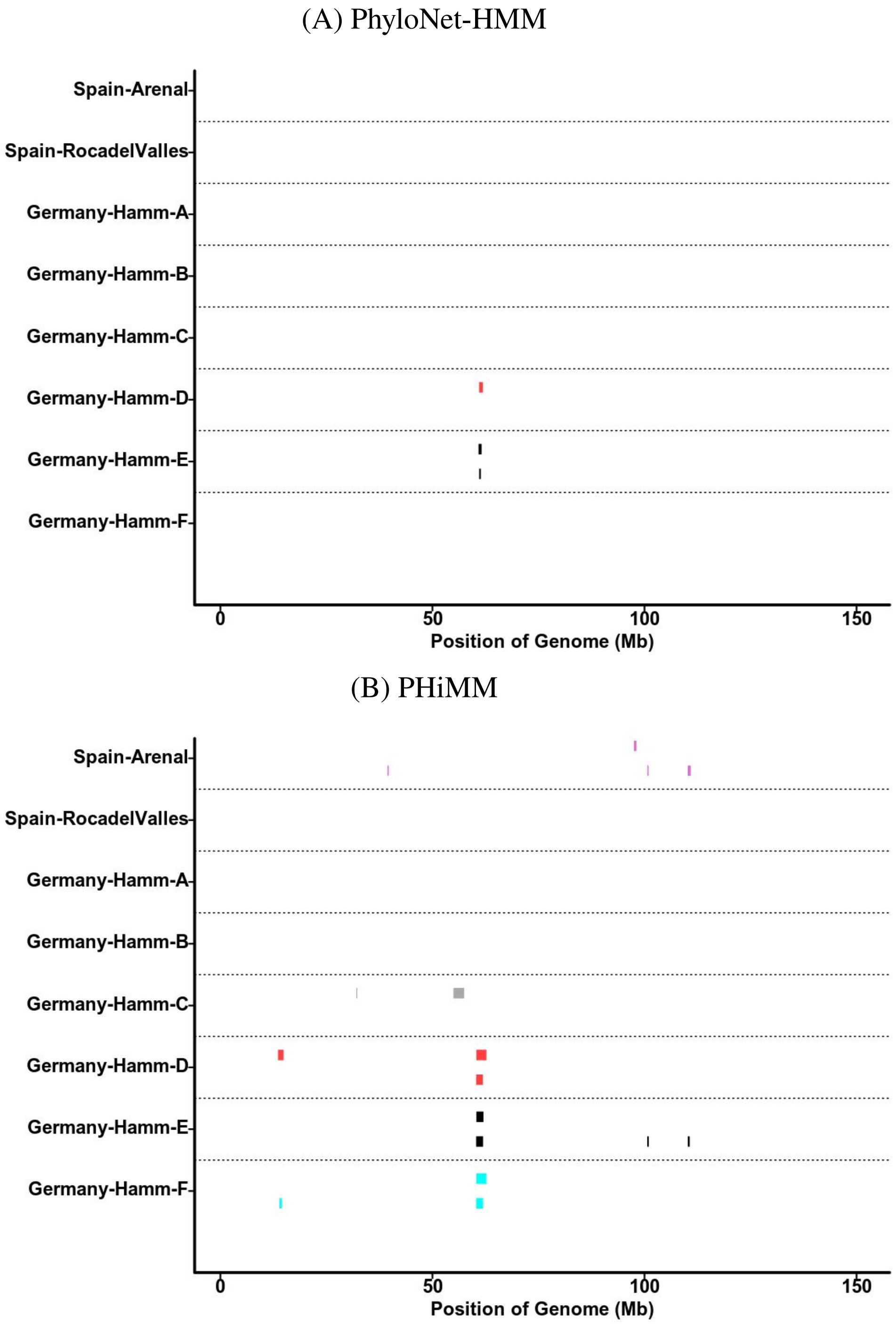
Figure A.3 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 3.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

(B) PHiMM

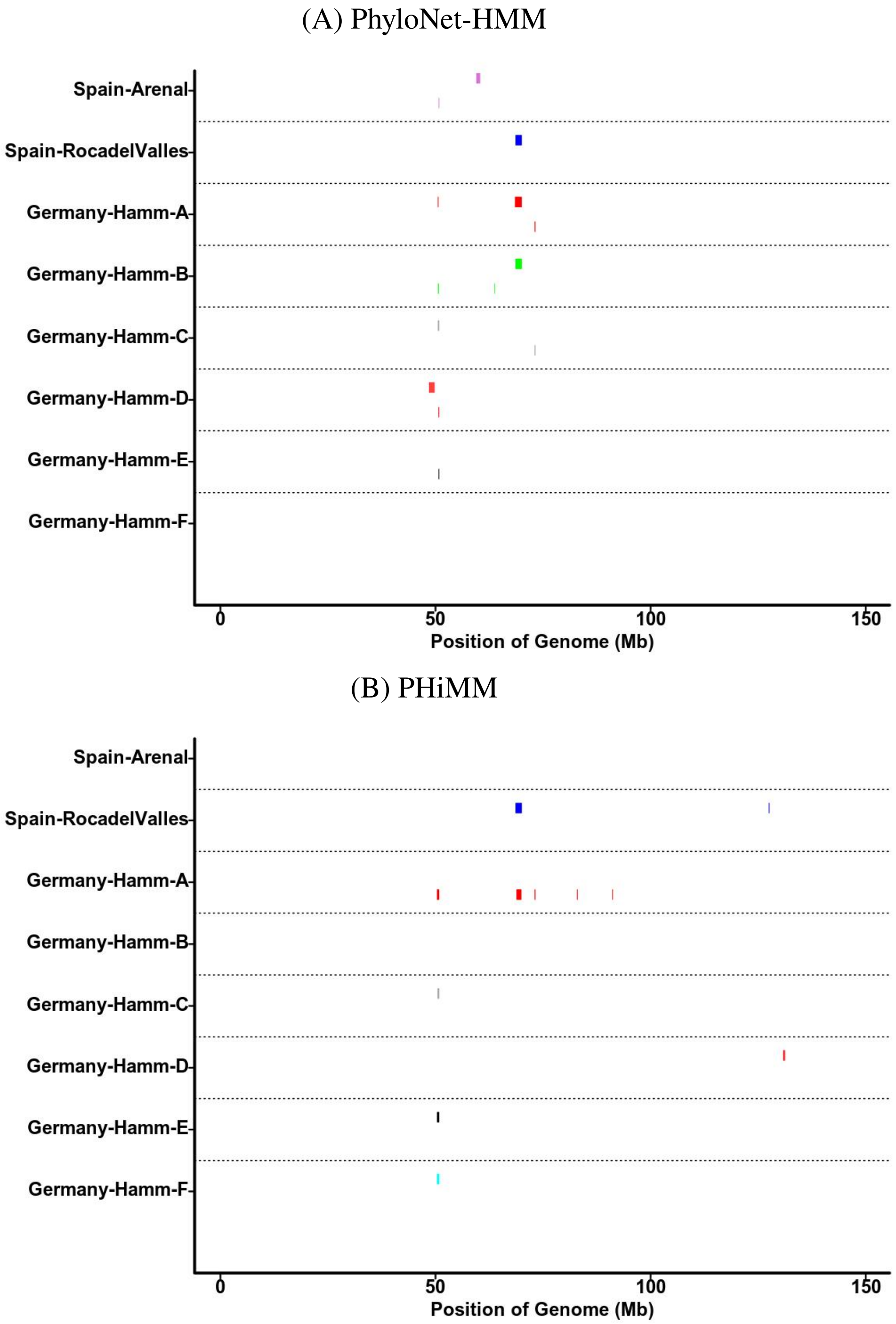
Figure A.4 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 4.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

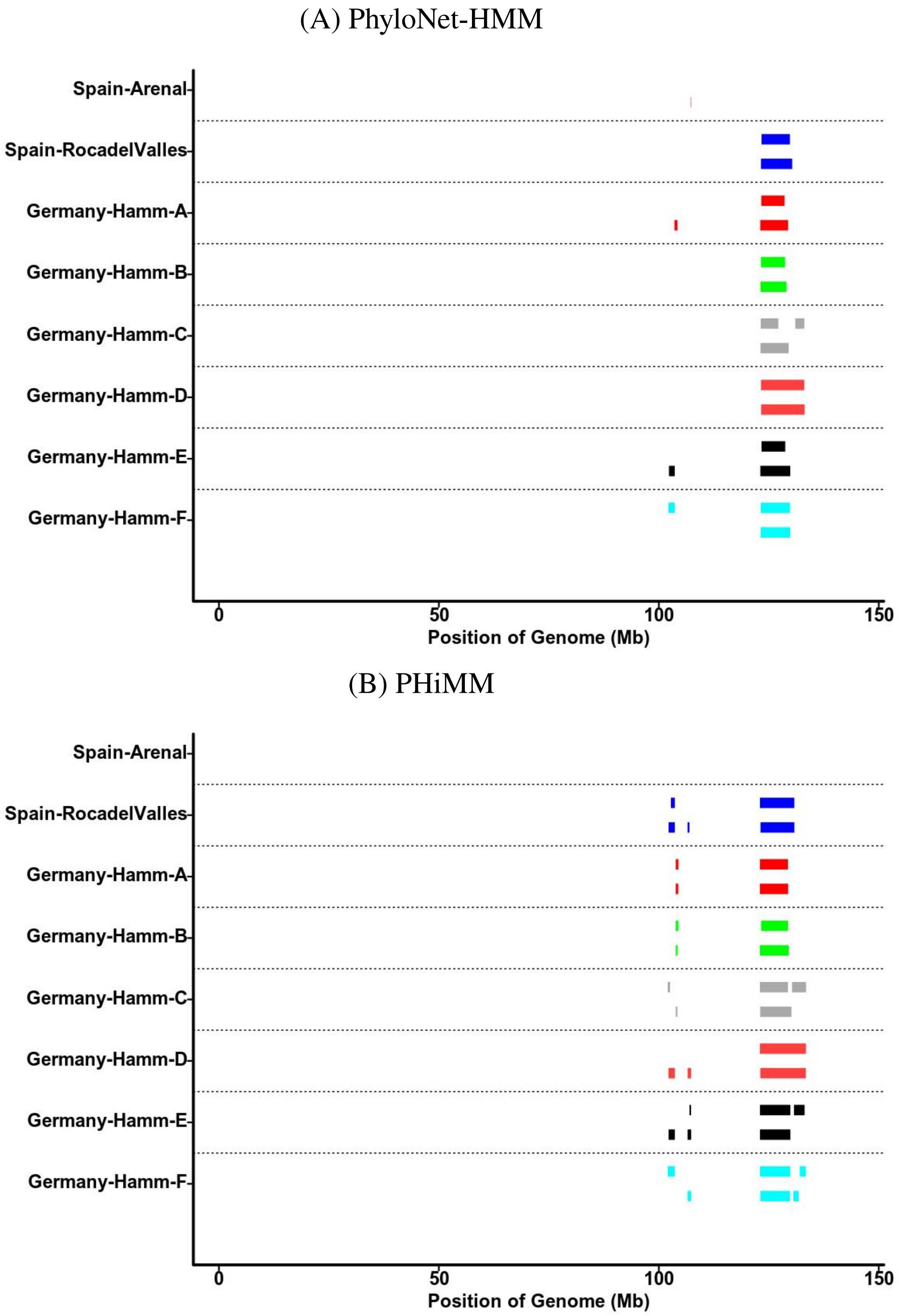
(B) PHiMM

Figure A.5 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 5.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].


(A) PhyloNet-HMM

(B) PHiMM

Figure A.6 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 6.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

(B) PHiMM

Figure A.7 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 7.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].

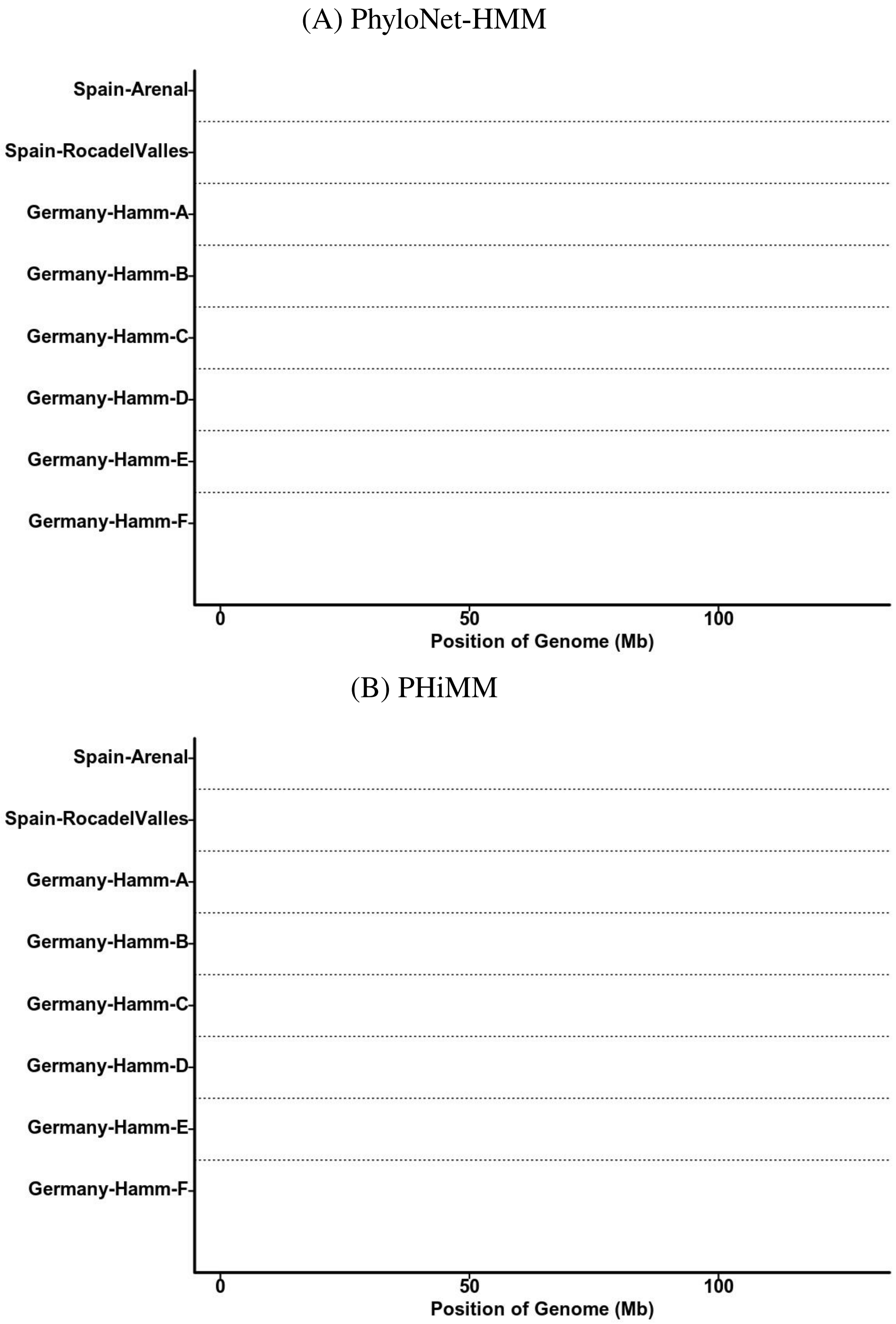
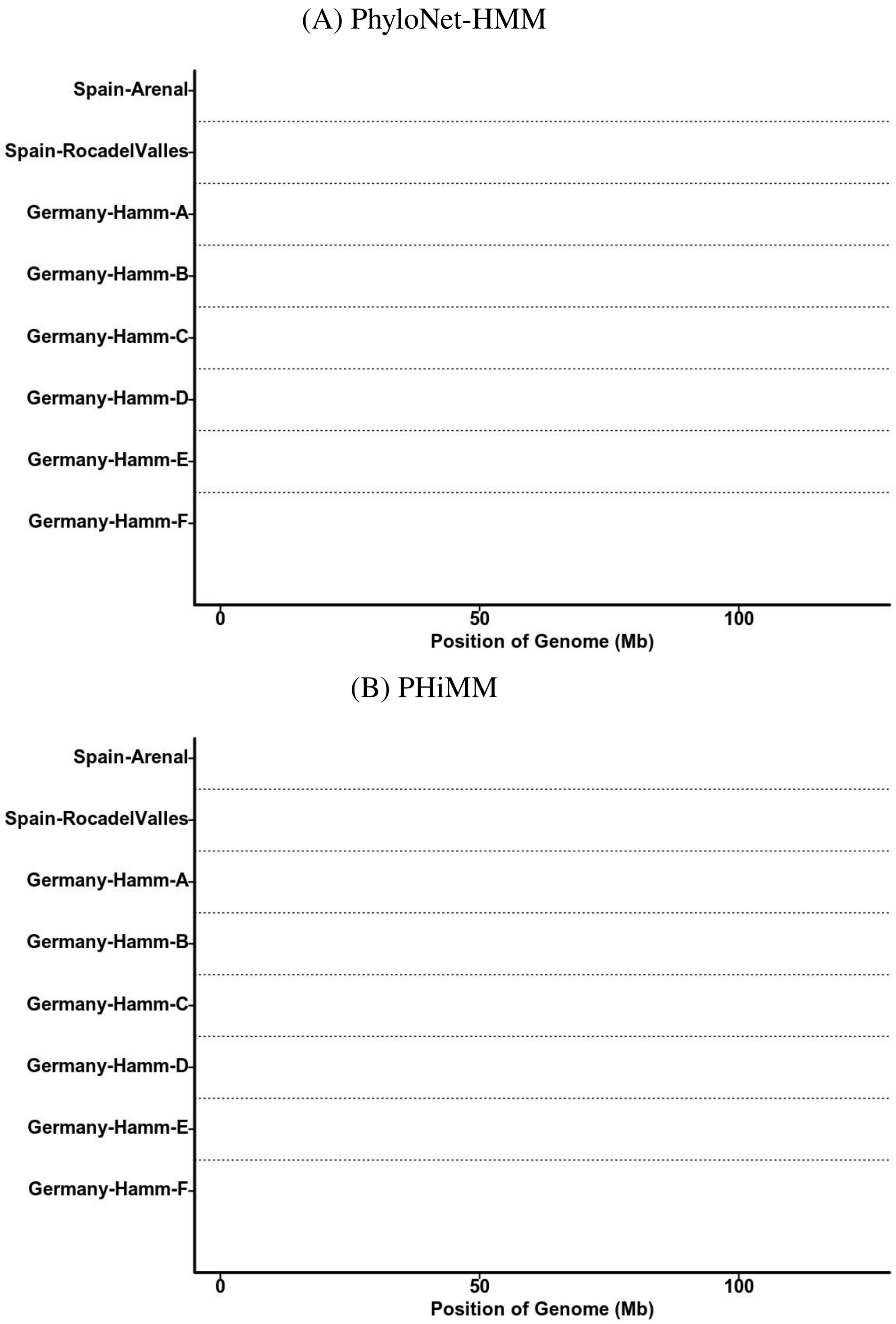
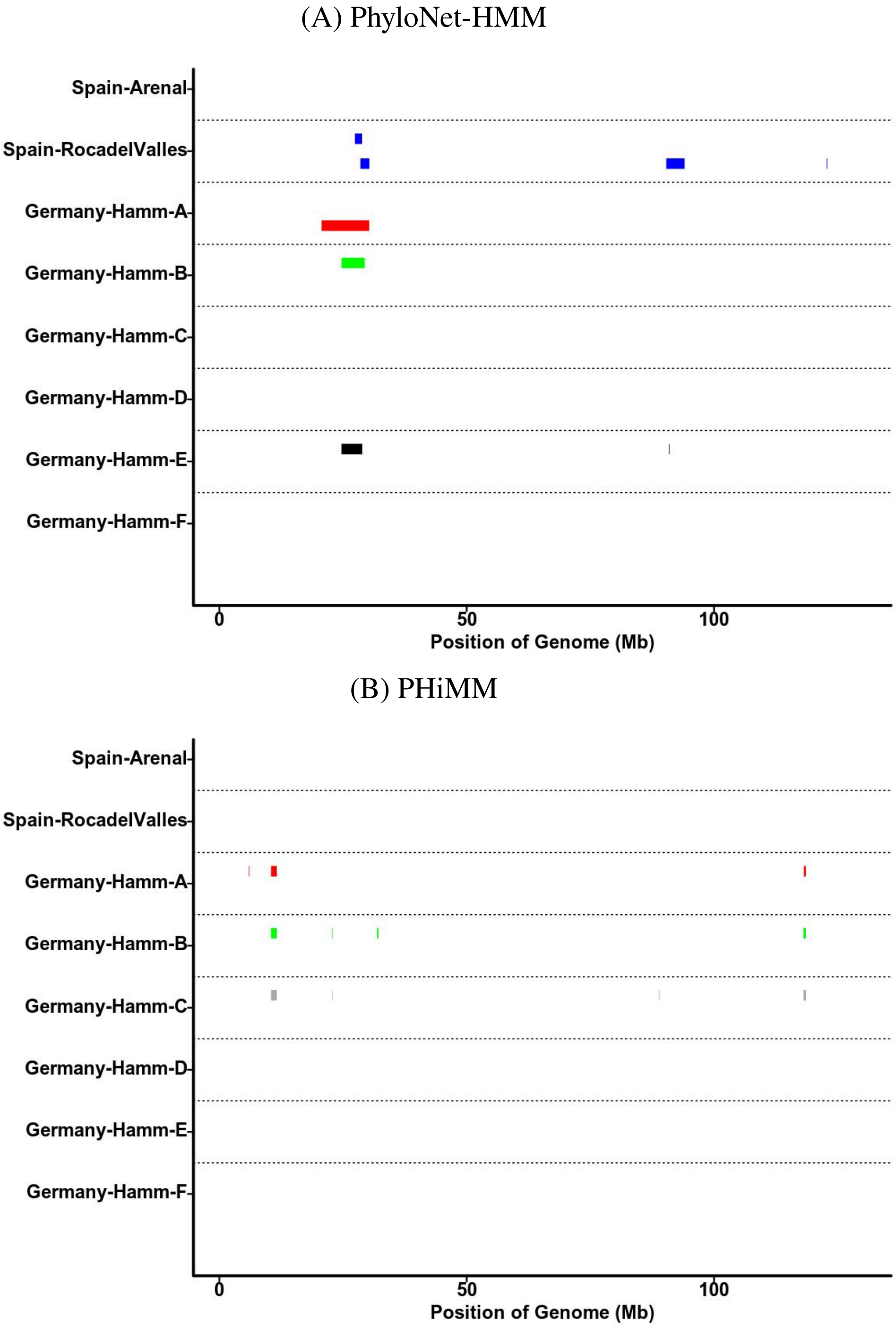


Figure A.8 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 8.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].


(A) PhyloNet-HMM

(B) PHiMM

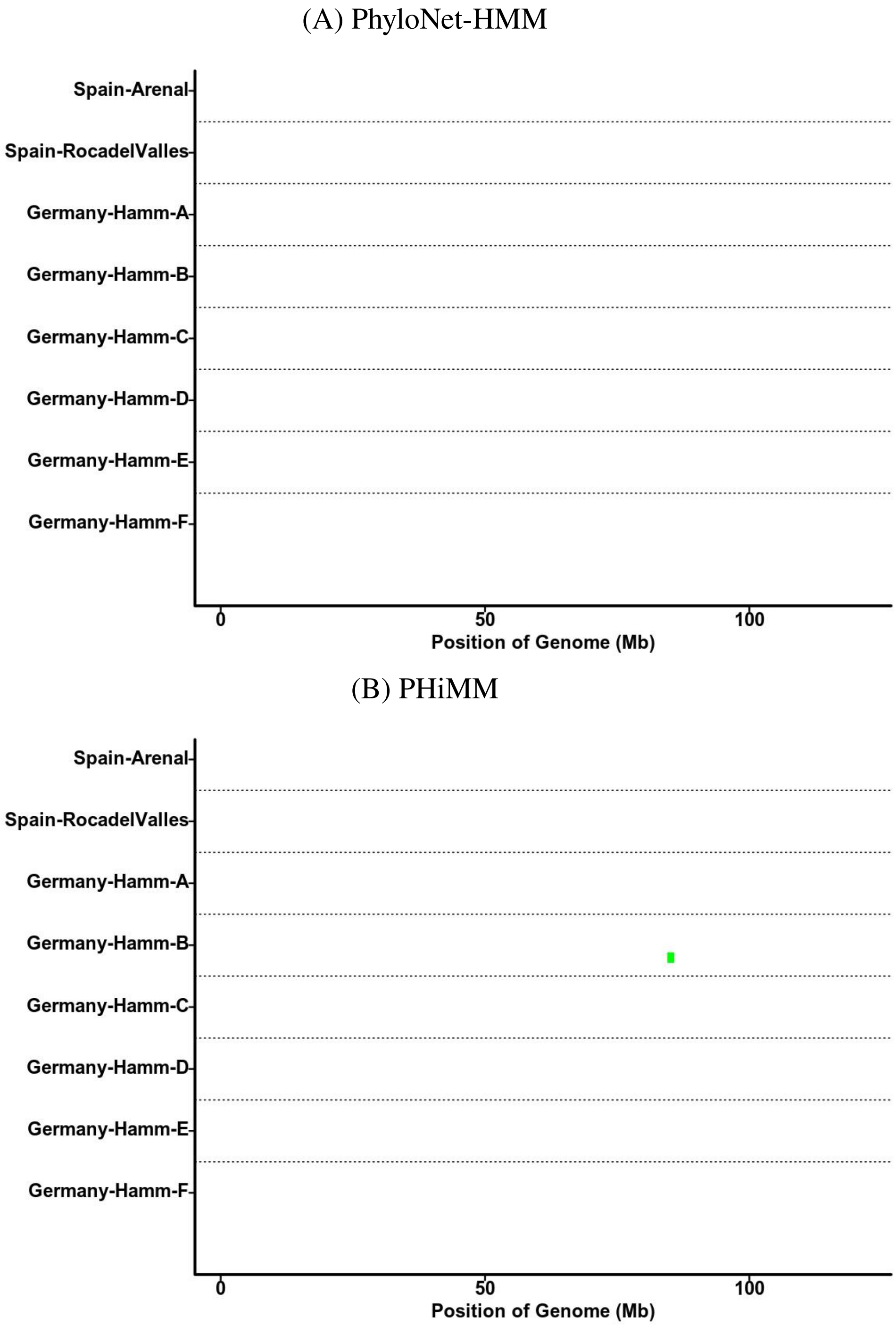
Figure A.9 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 9.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

(B) PHiMM

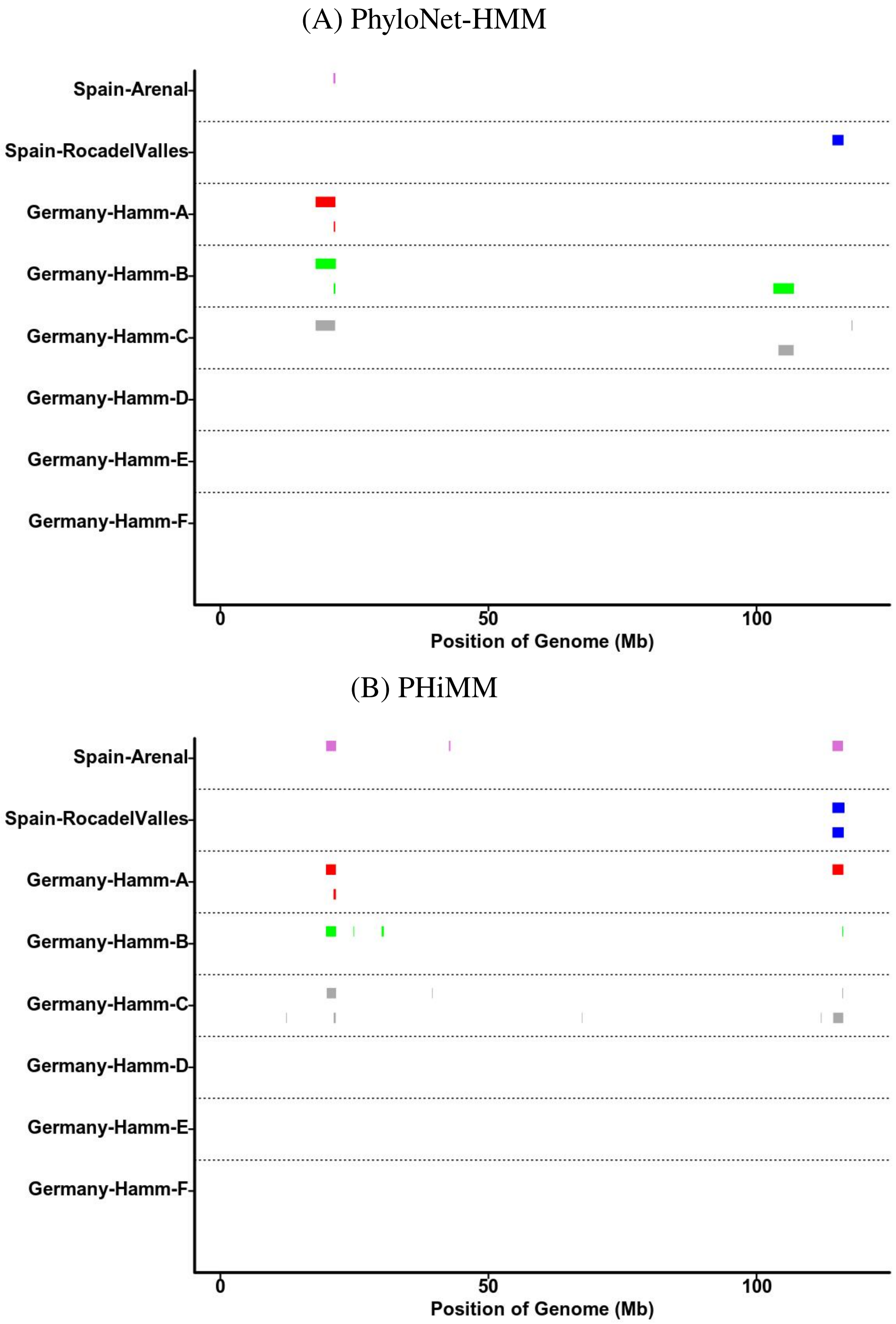
Figure A.10 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 10.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

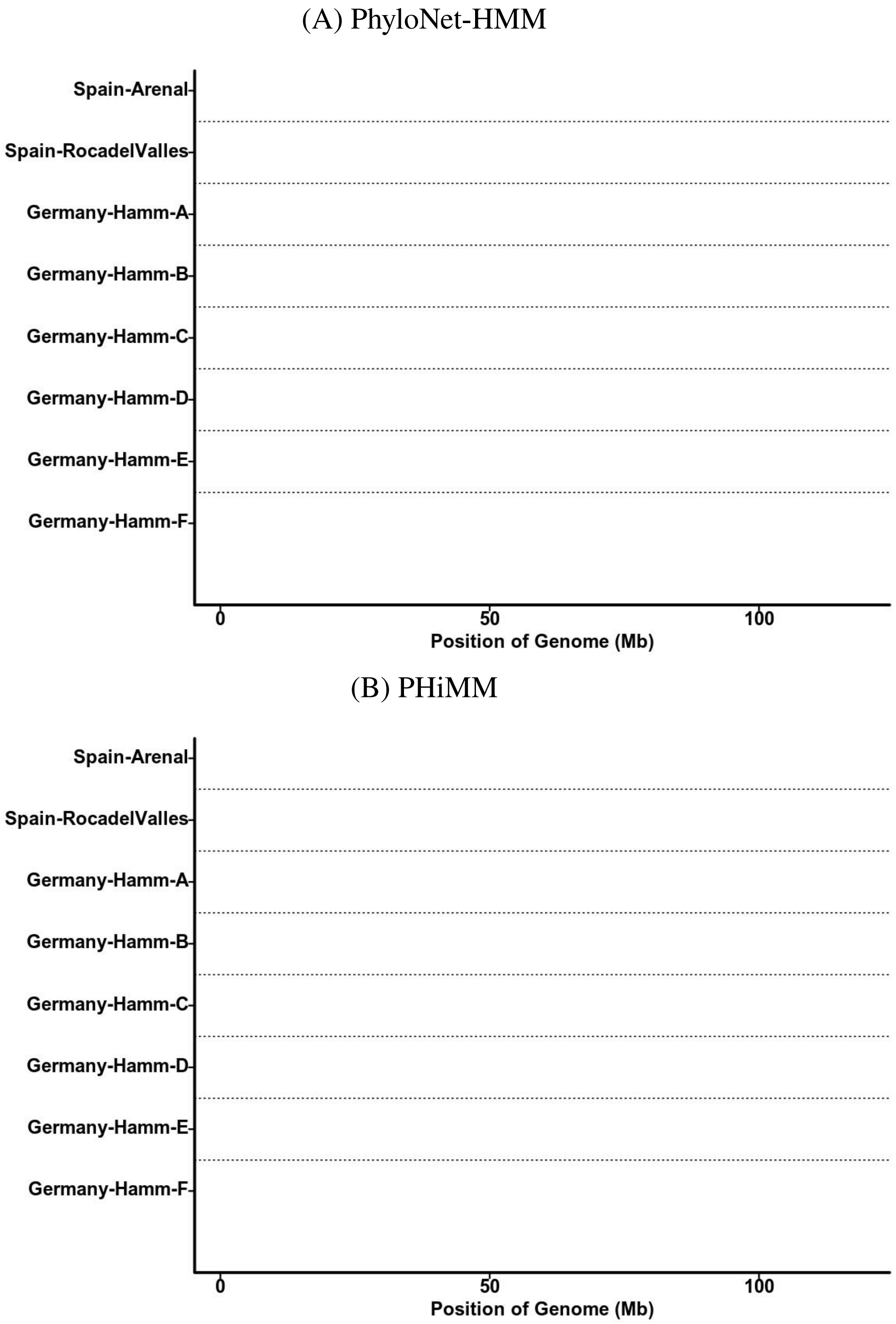
(B) PHiMM

Figure A.11 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 11.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].


(A) PhyloNet-HMM

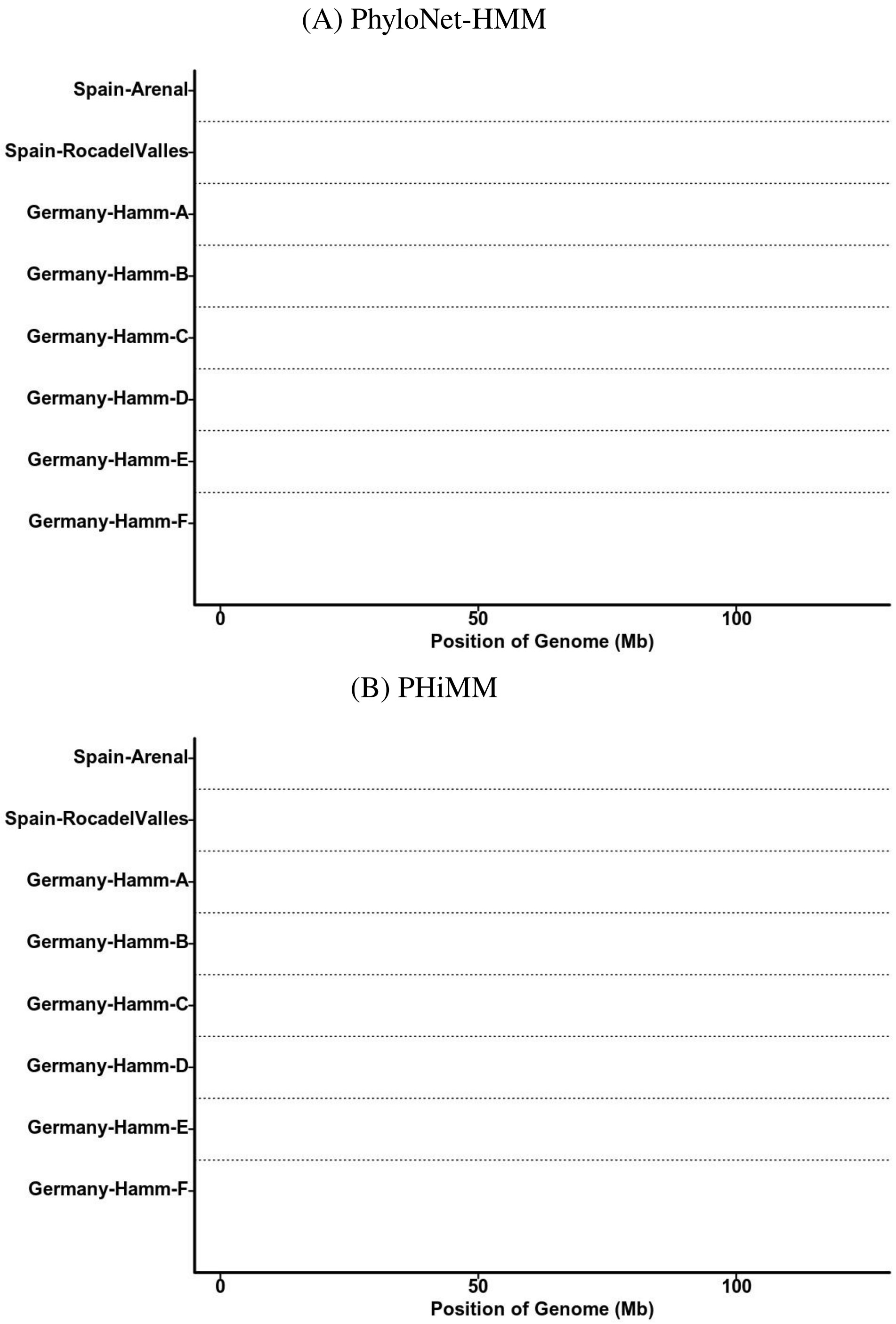
(B) PHiMM

Figure A.12 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 12.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].


(A) PhyloNet-HMM

(B) PHiMM

Figure A.13 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 13.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

(B) PHiMM

Figure A.14 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 14.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].

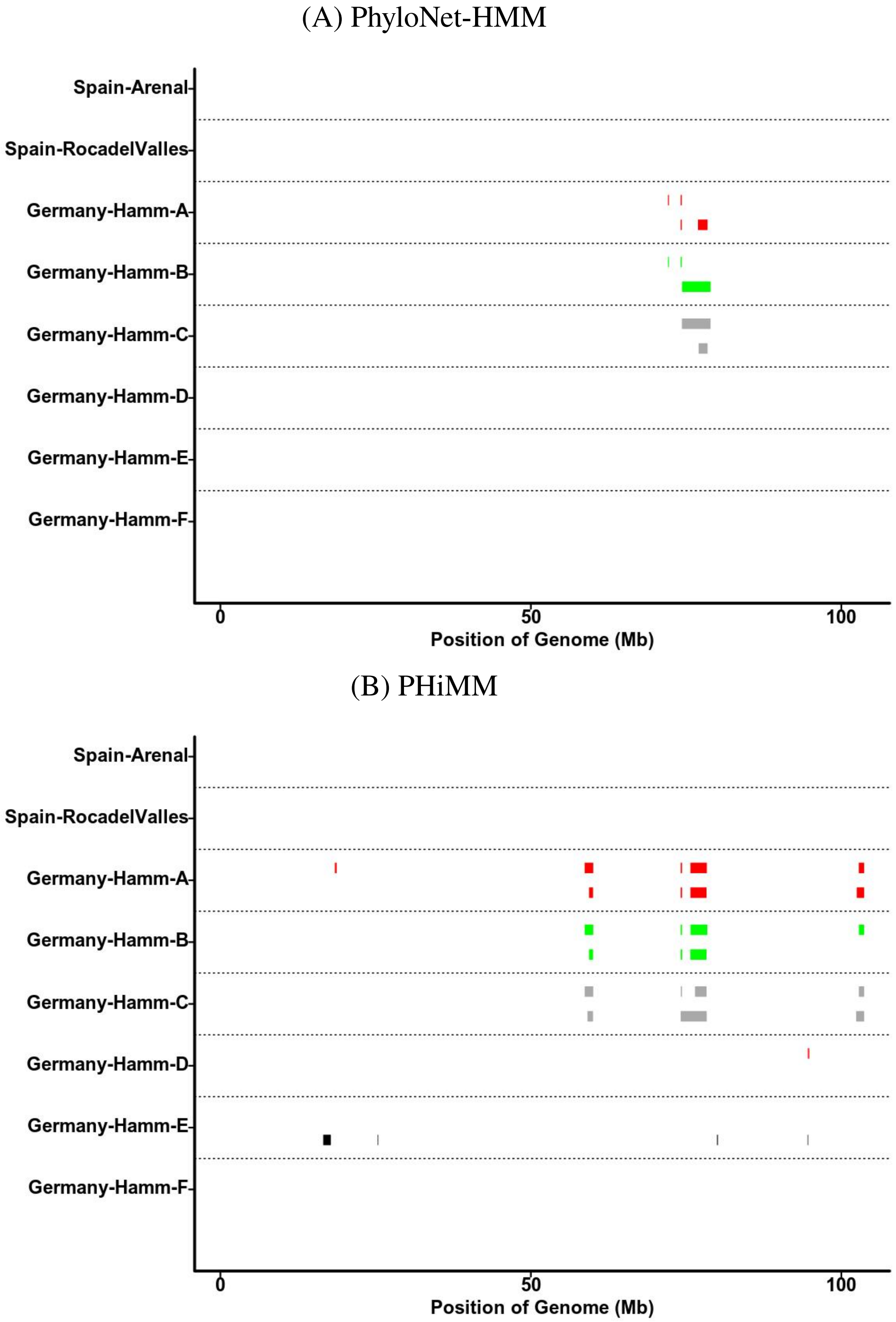
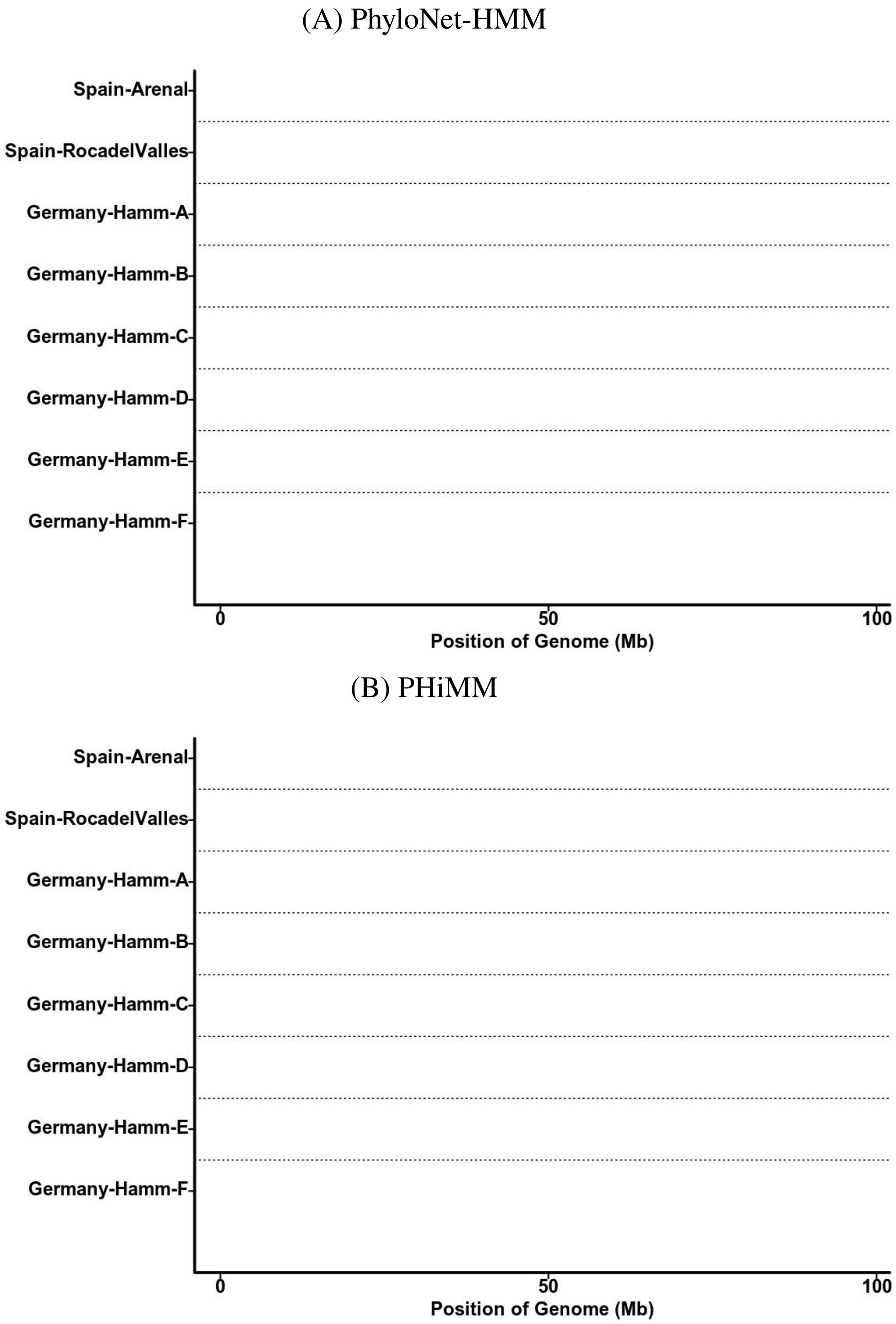


Figure A.15 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 15.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

(B) PHiMM

Figure A.16 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 16.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].

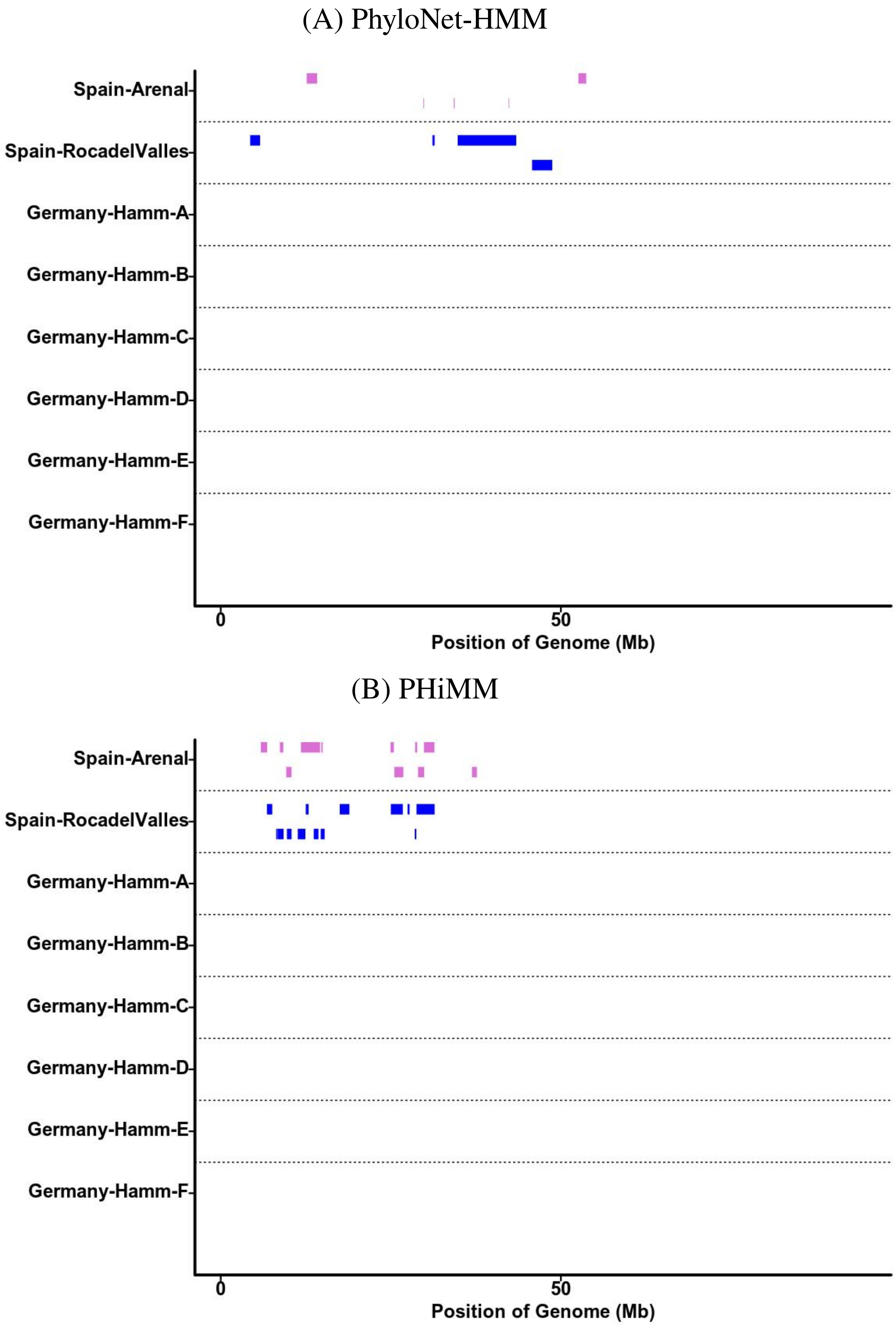
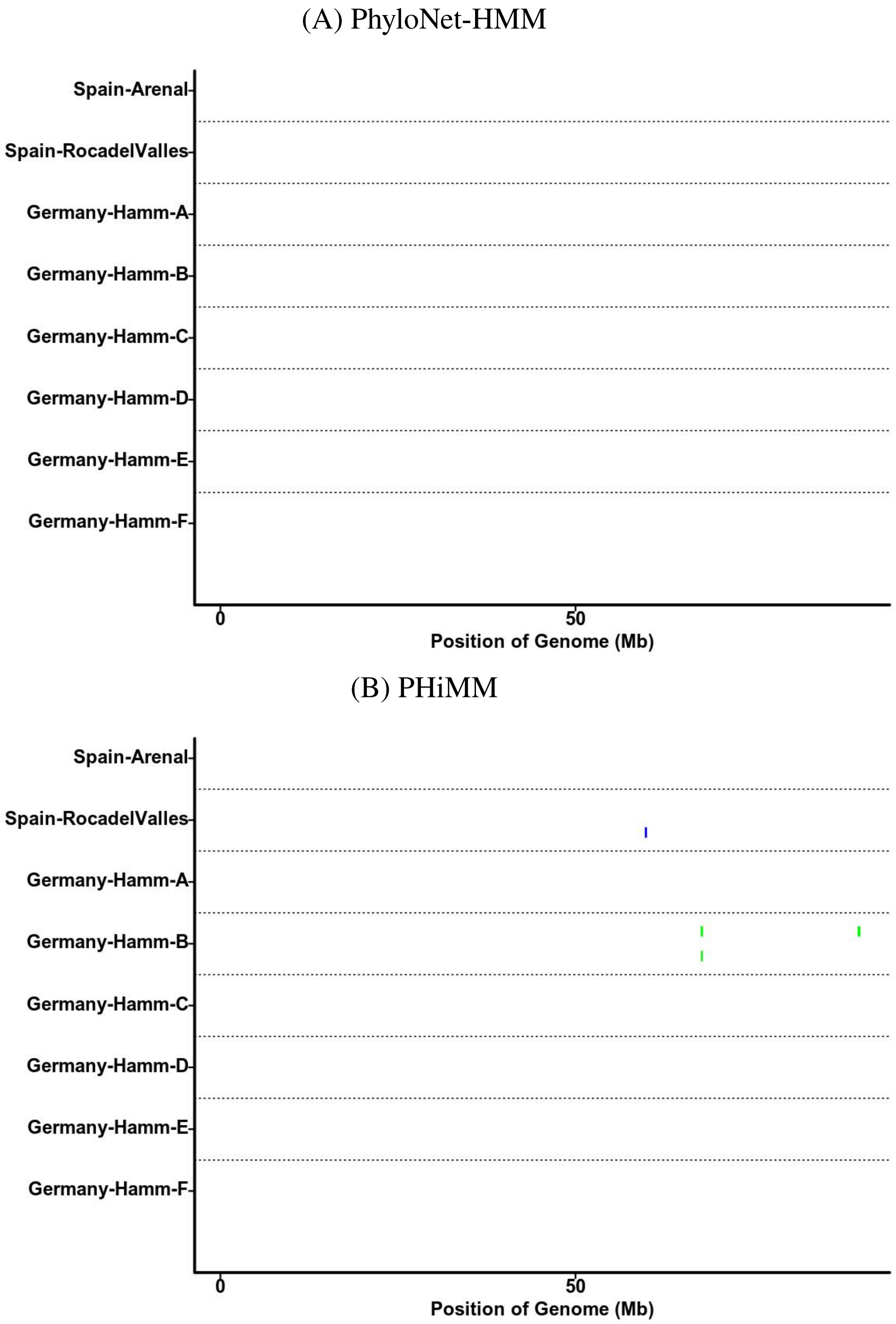


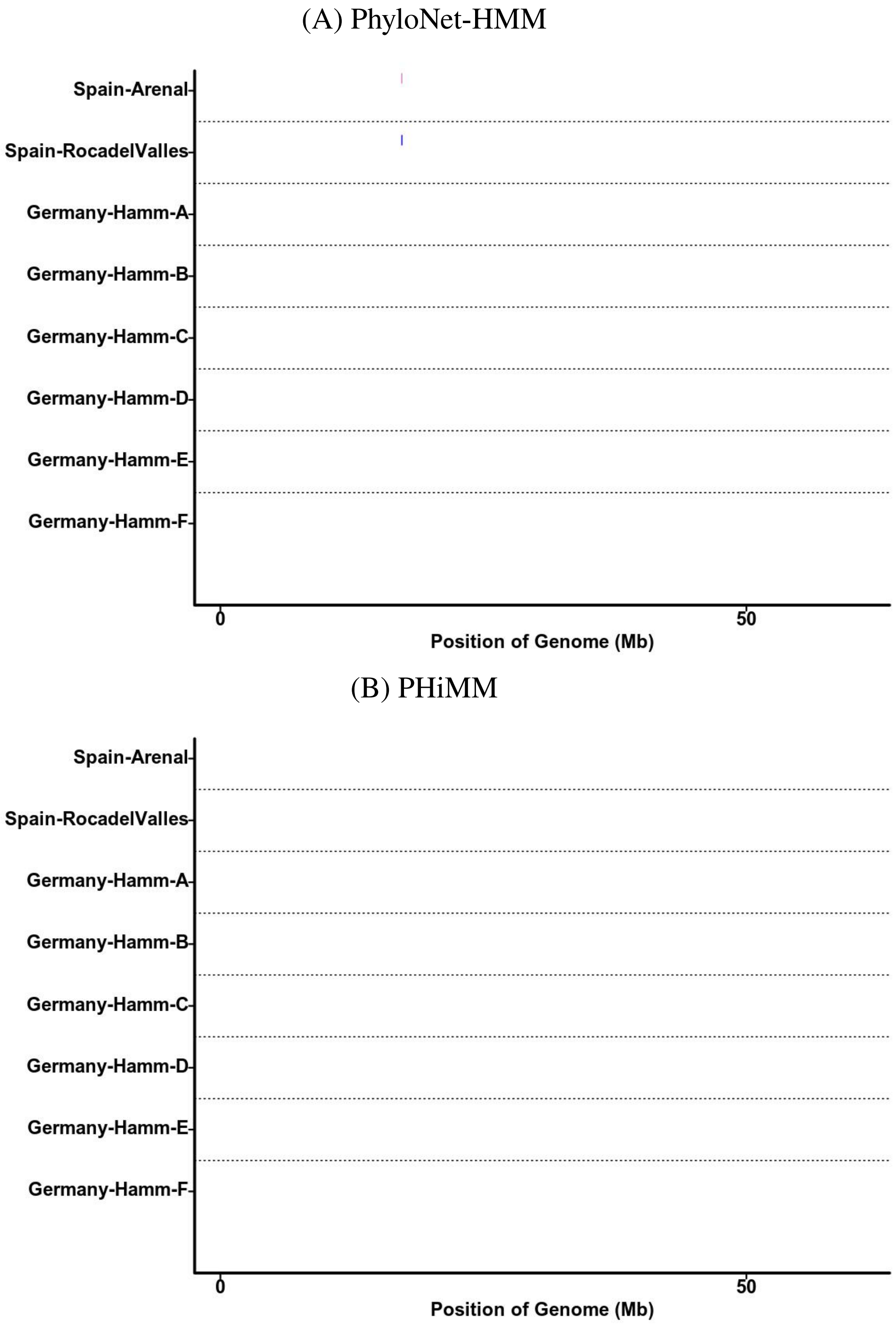
Figure A.17 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 17.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

(B) PHiMM

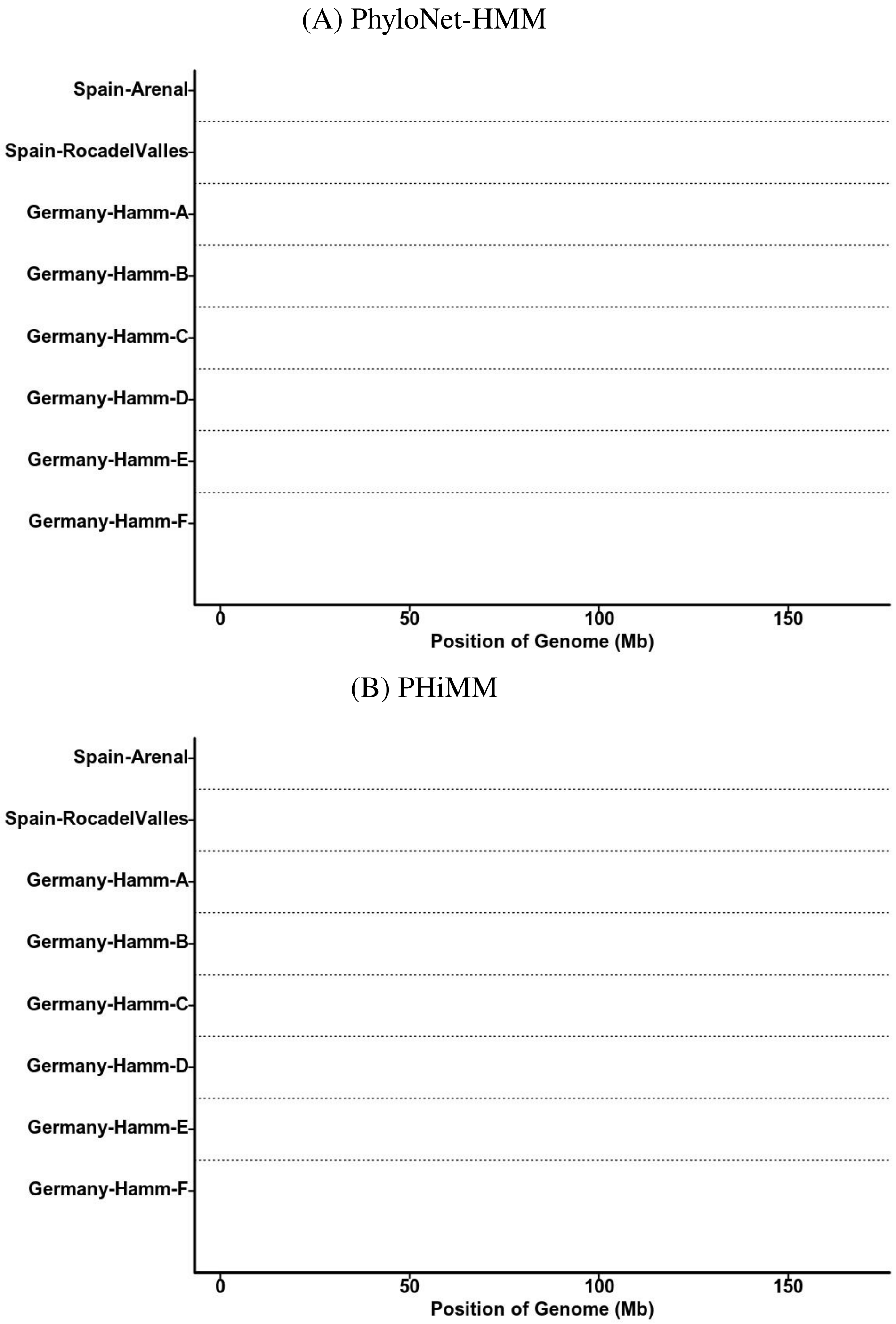
Figure A.18 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 18.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

(B) PHiMM

Figure A.19 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome 19.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].



(A) PhyloNet-HMM

(B) PHiMM

Figure A.20 **The comparison of introgression patterns between (A) PhyloNet-HMM and (B) PHiMM on chromosome X.** Introgressed tracts along the genome are shown in megabases (Mb). Results are reported for the introgression from donor *M. spretus* samples to the recipient *M. m. domesticus* samples (two Spanish mice and six German mice). Each *M. m. domesticus* sample is shown with two haploid genomes (the above bar is haplotype 1, while the below represents haplotype 2). This figure comes from Wuyun et al. [122].

APPENDIX B

THE INTROGRESSION PATTERNS OF DACS ON DARWIN'S FINCH DATA.

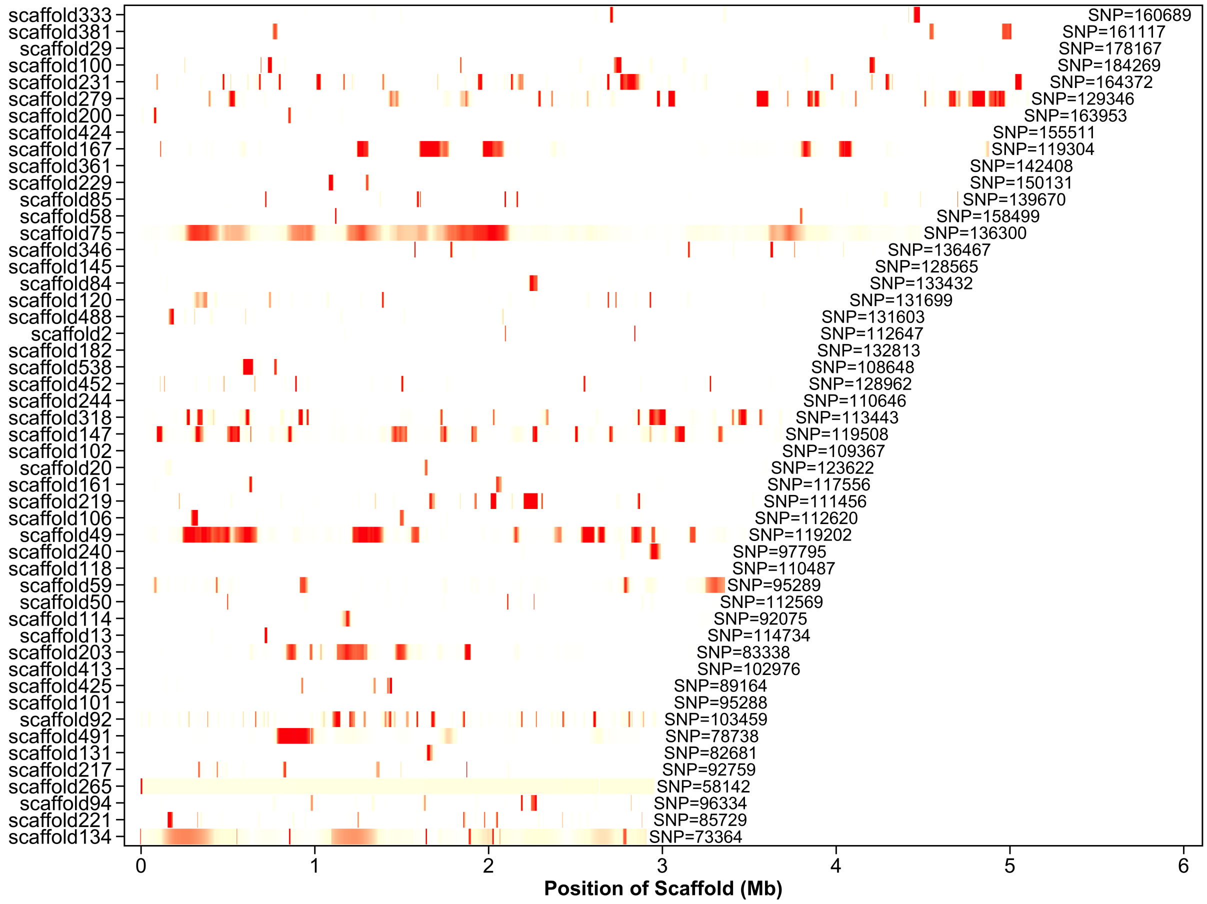


Figure B.1 **The introgression patterns of DACS on top 51-100 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in megabases (Mb).

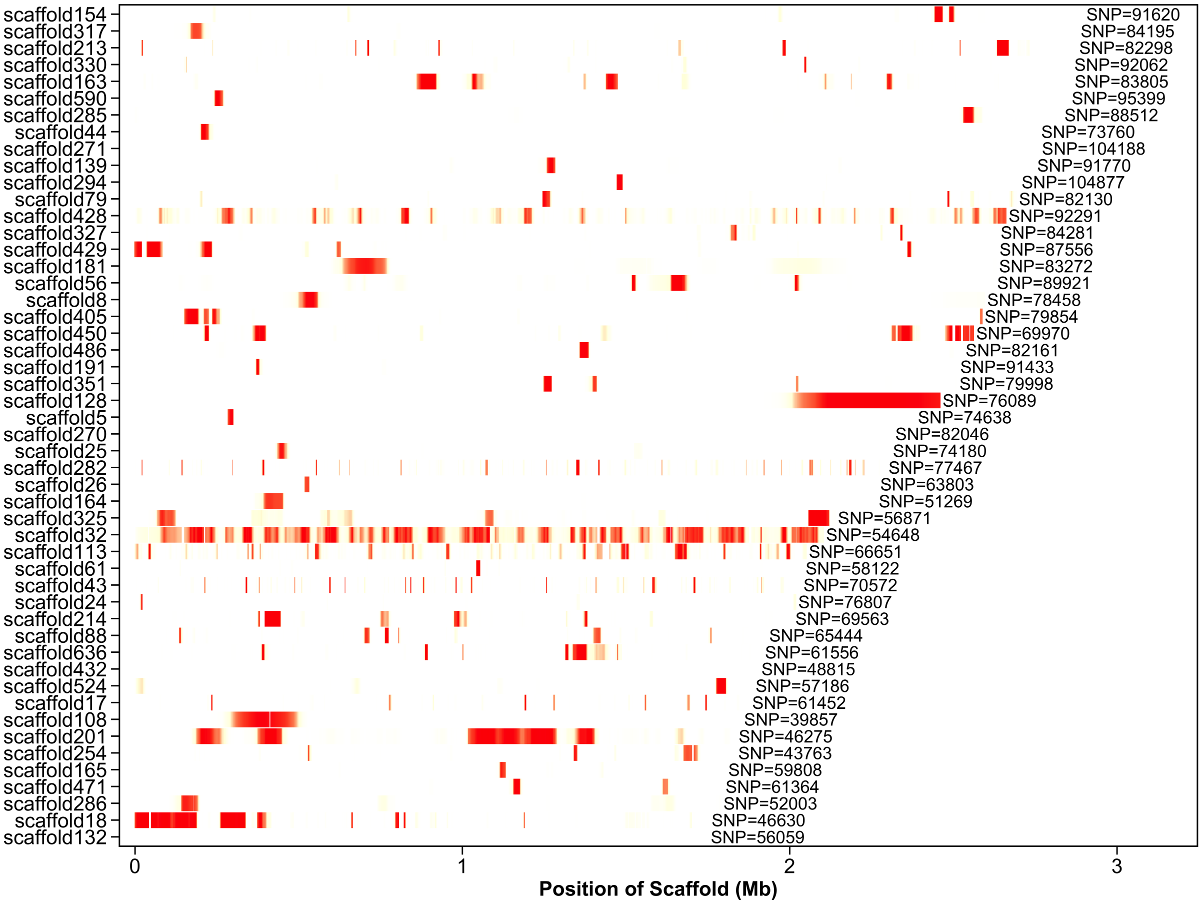


Figure B.2 **The introgression patterns of DACS on top 101-150 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in megabases (Mb).

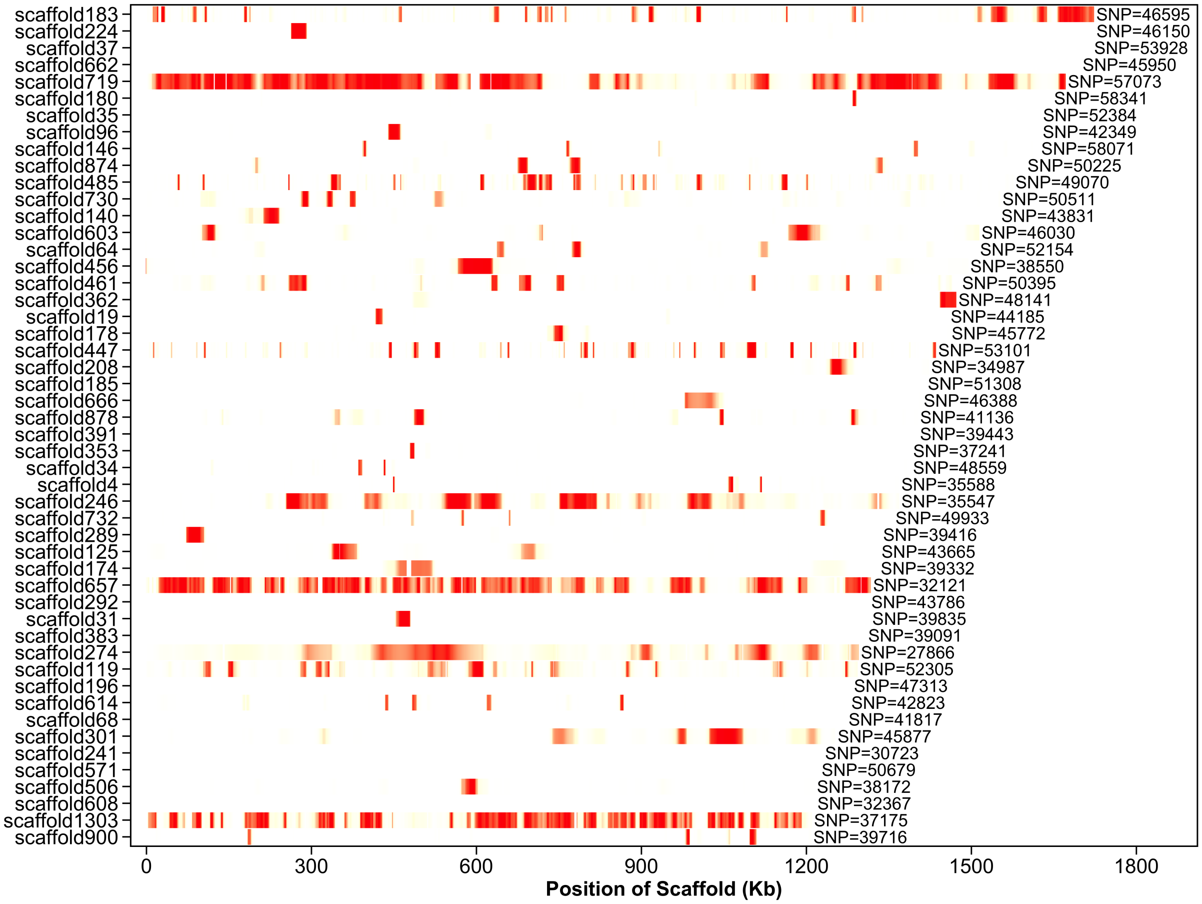


Figure B.3 **The introgression patterns of DACS on top 151-200 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in kilobases (Kb).

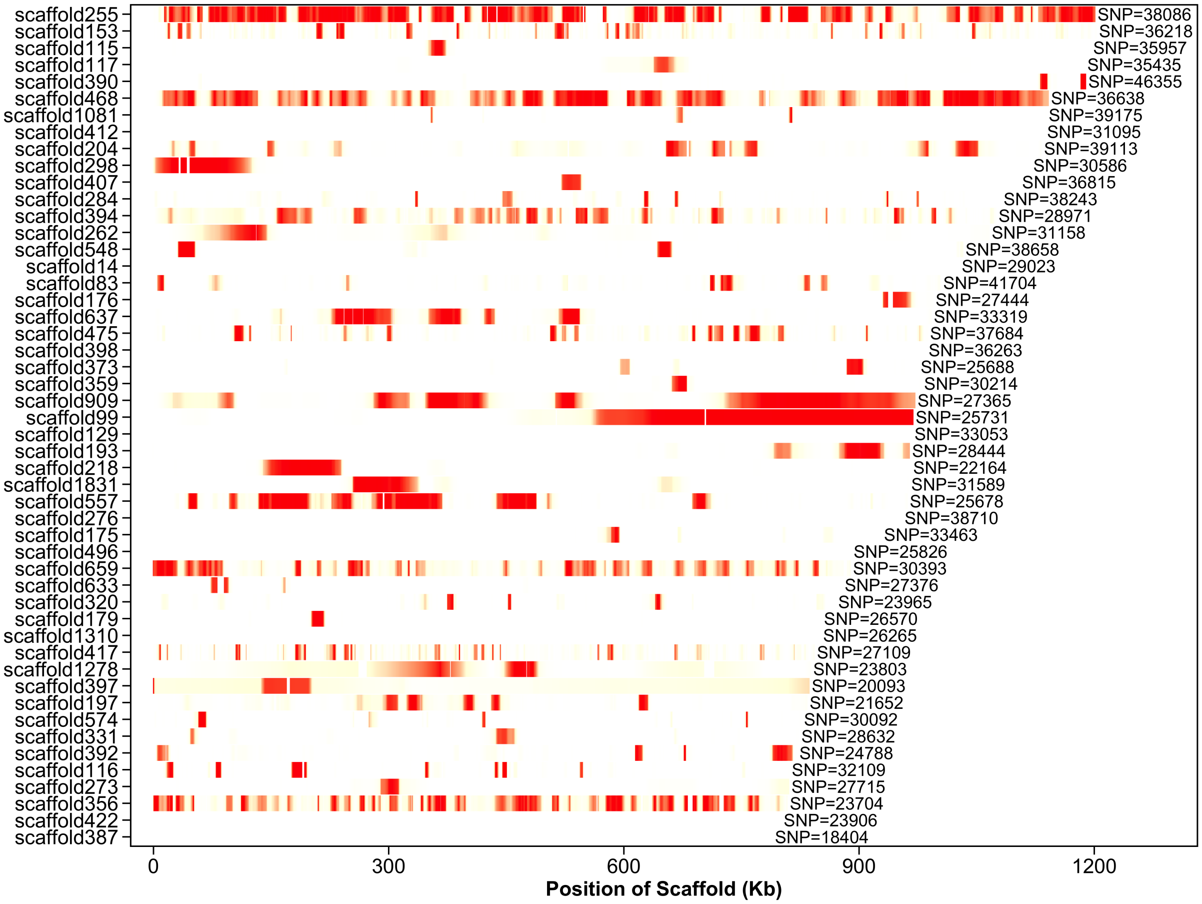


Figure B.4 **The introgression patterns of DACS on top 201-250 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in kilobases (Kb).

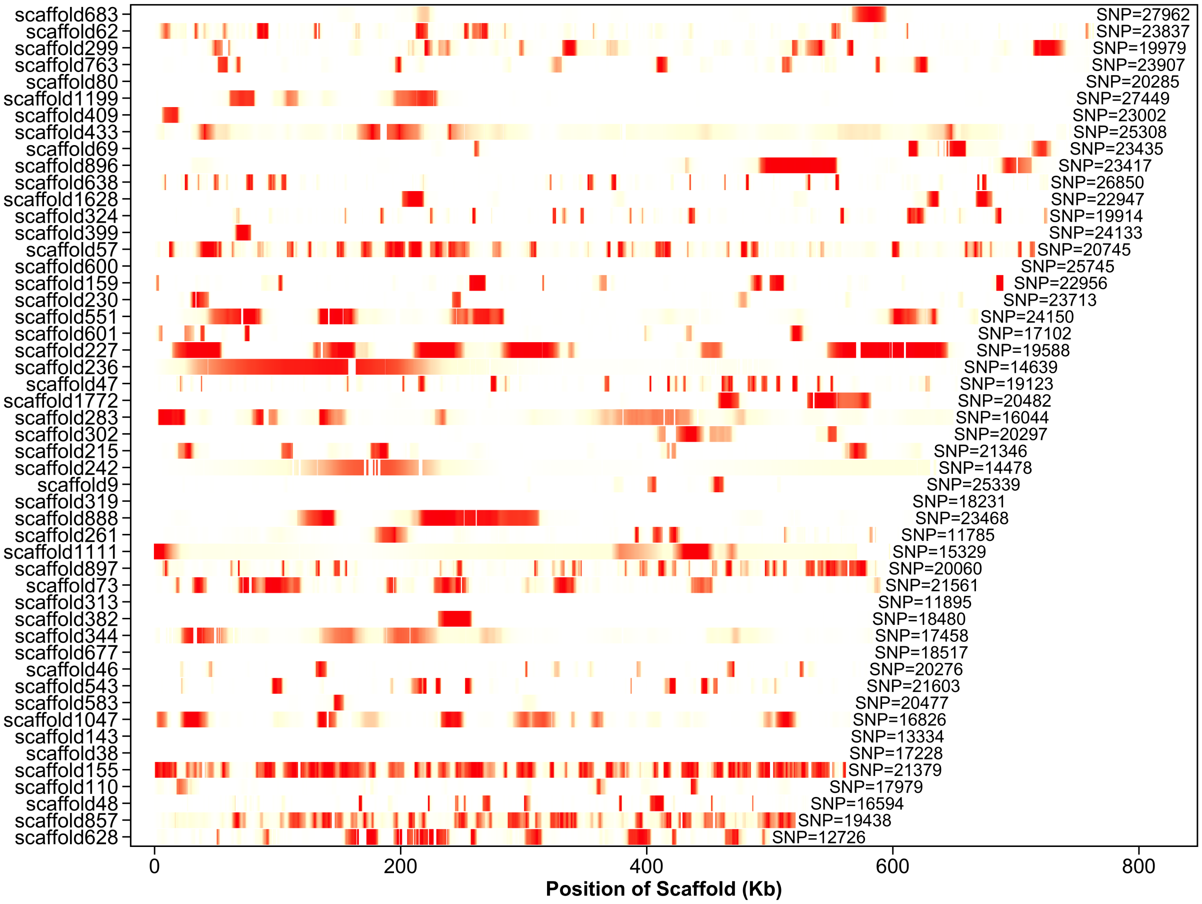


Figure B.5 **The introgression patterns of DACS on top 251-300 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in kilobases (Kb).

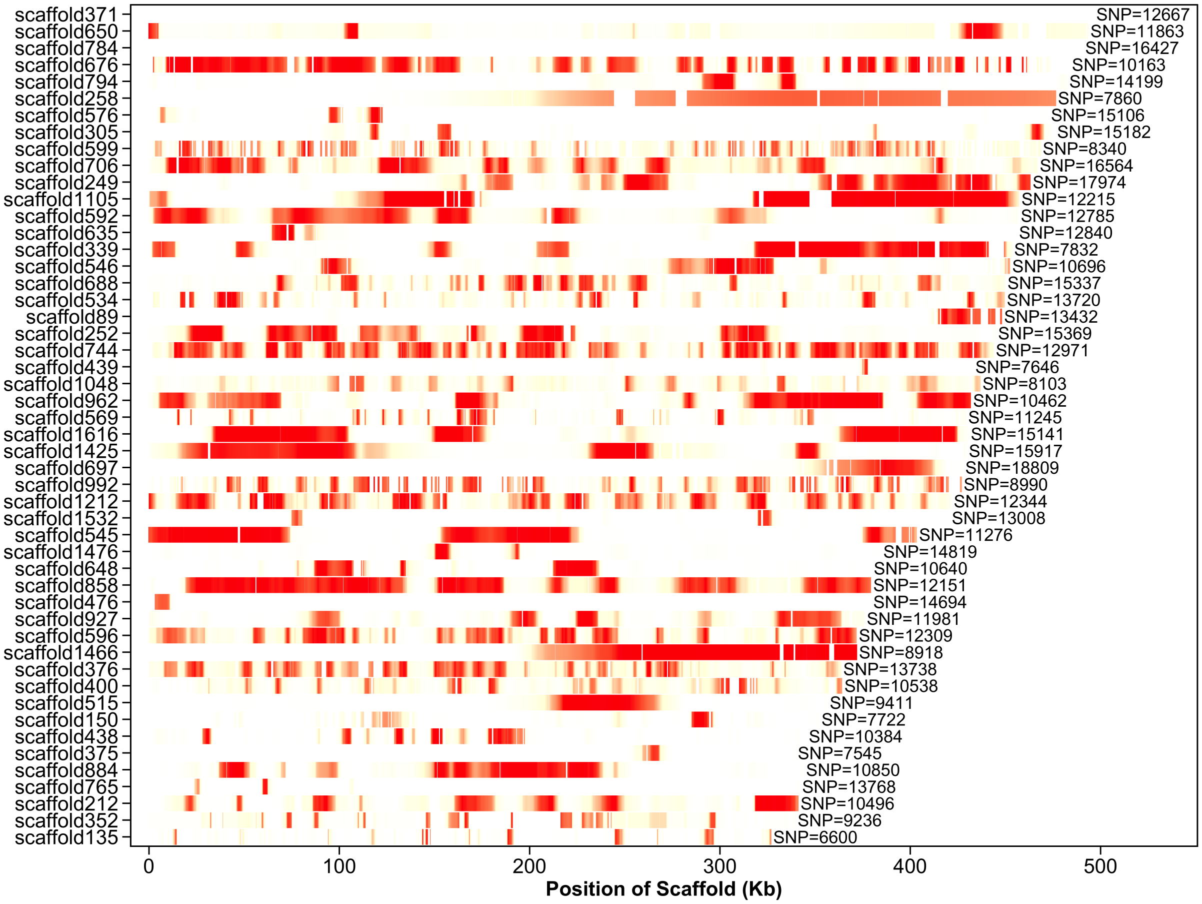


Figure B.6 **The introgression patterns of DACS on top 301-350 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in kilobases (Kb).

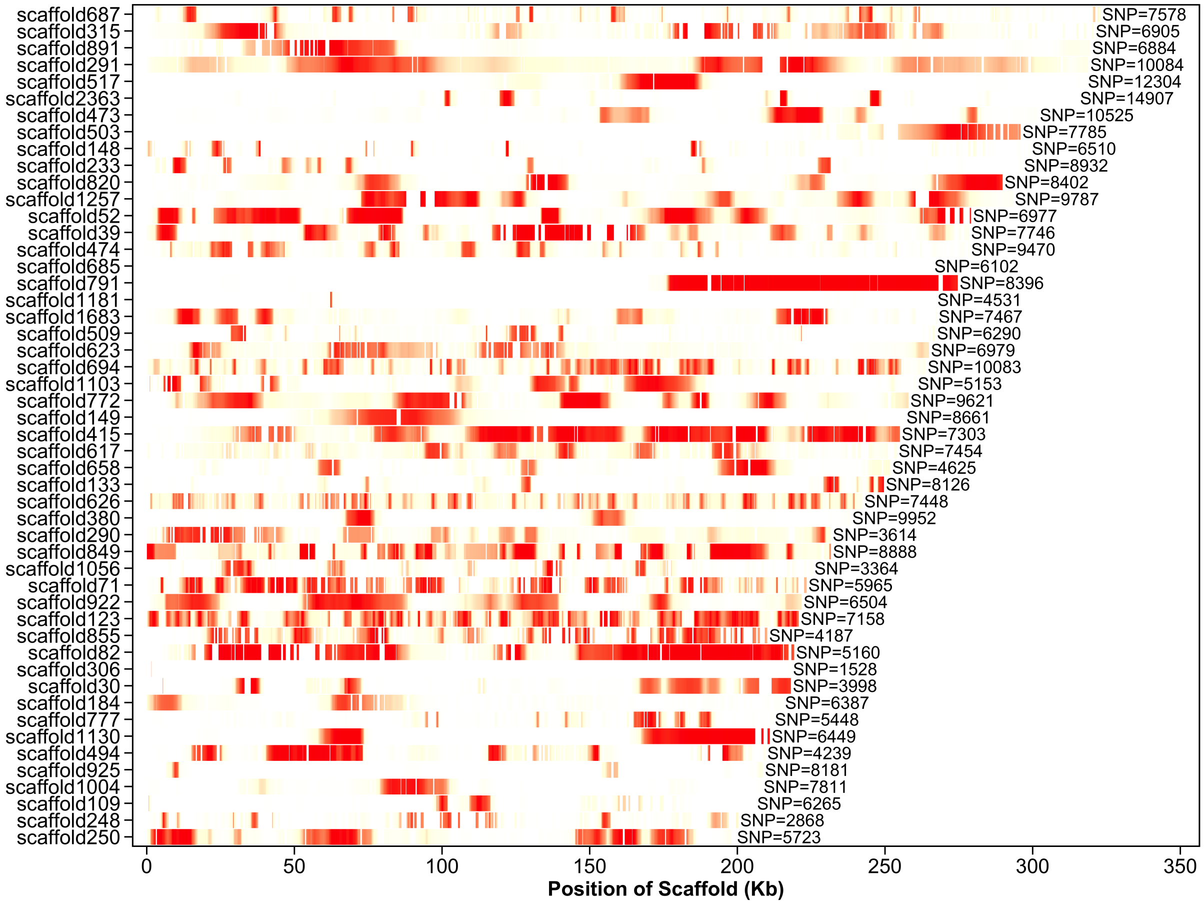


Figure B.7 **The introgression patterns of DACS on top 351-400 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in kilobases (Kb).

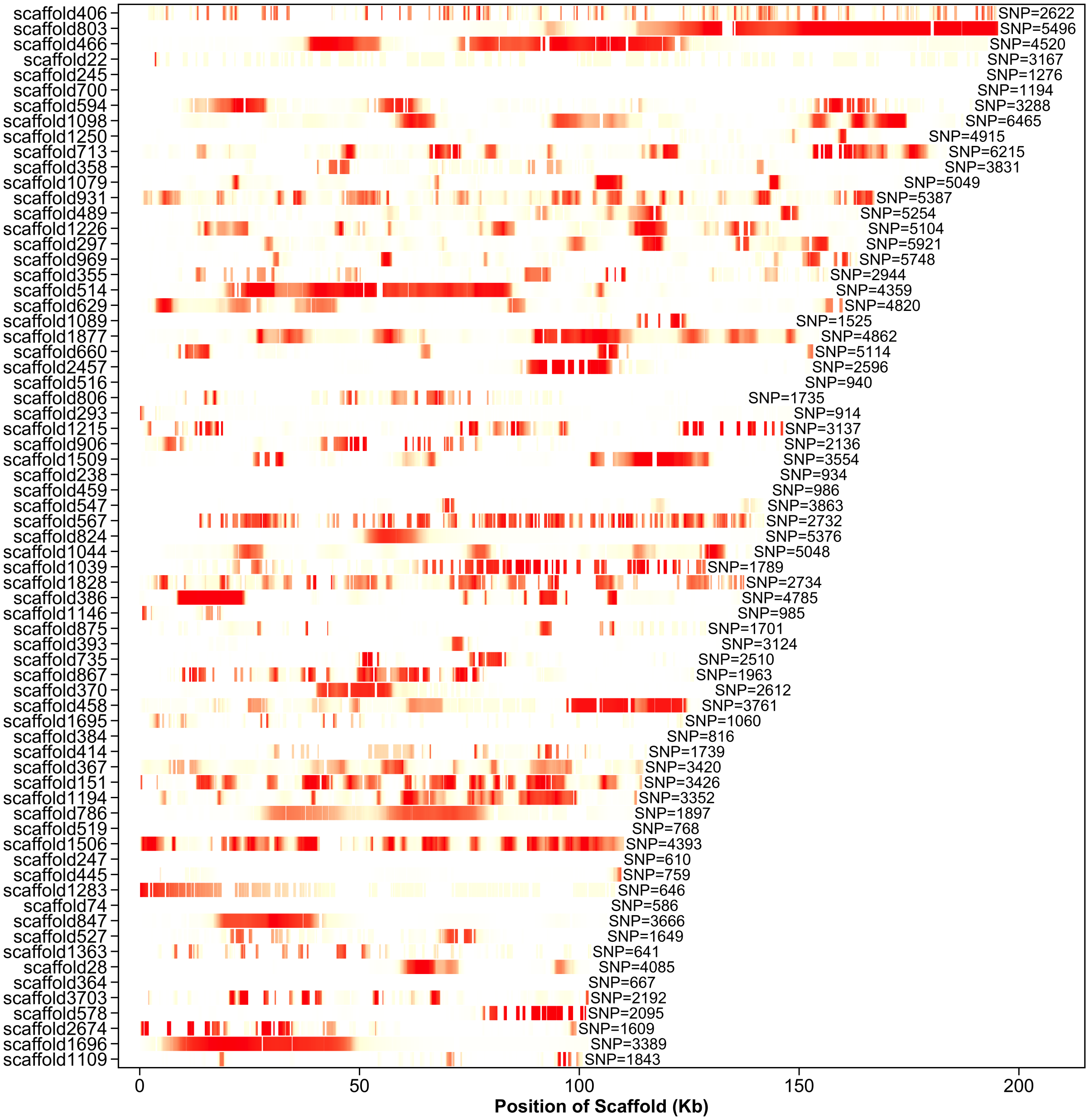


Figure B.8 **The introgression patterns of DACS on top 401-469 largest scaffolds of Darwin's finch data.** Introgressed tracts along the genome are shown in kilobases (Kb).