University of South Carolina
Scholar Commons

Theses and Dissertations

2016

A Hierarchical Framework for Phylogenetic and Ancestral Genome Reconstruction on Whole Genome Data

Lingxi Zhou
University of South Carolina

Follow this and additional works at: https://scholarcommons.sc.edu/etd

 Part of the Computer Engineering Commons, and the Computer Sciences Commons

Recommended Citation

Zhou, L.(2016). *A Hierarchical Framework for Phylogenetic and Ancestral Genome Reconstruction on Whole Genome Data.* (Doctoral dissertation). Retrieved from https://scholarcommons.sc.edu/etd/3827

This Open Access Dissertation is brought to you by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact digres@mailbox.sc.edu.

A Hierarchical Framework for Phylogenetic and Ancestral Genome Reconstruction on Whole Genome Data

by

Lingxi Zhou

Bachelor of Science
Jilin University 2011

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosphy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2016

Accepted by:

Jijun Tang, Major Professor

John Rose, Committee Member

Jianjun Hu, Committee Member

Homayoun Valafar, Committee Member

Bob Friedman, Committee Member

Lacy Ford, Senior Vice Provost and Dean of Graduate Studies

© Copyright by Lingxi Zhou, 2016
All Rights Reserved.

Acknowledgments

First and foremost I wish to thank my advisor Dr. Jijun Tang. It has been my privilege to have him as my PhD advisor. His great mind and personal charisma filled me with admiration. He granted me the freedom to discover my real passion in research, and also provided continuous support, guidance and encouragement along the way. Thank you, Jijun! I sincerely wish to thank Dr. John Rose,Dr. Jianjun Hu, Dr. Homayoun Valafar and Dr. Bob Friedman for their willingnesses to serve on my dissertation committee and their valuable comments! Finally, I would like to acknowledge my family and friends for their continued love and support through- out all my endeavors!

Abstract

Gene order gets evolved under events such as rearrangements, duplications, and losses, which can change both the order and content along the genome, through the long history of genome evolution. Recently, the accumulation of genomic sequences provides researchers with the chance to handle long-standing problems about the phylogenies, or evolutionary histories, of sets of species, and ancestral genomic content and orders. Over the past few years, such problems have been proven so interesting that a large number of algorithms have been proposed in the attempt to resolve them, following different standards. The work presented in this dissertation focuses on algorithms and models for whole-genome evolution and their applications in phylogeny and ancestor inference from gene order. We developed a flexible ancestor reconstruction method (FARM) within the framework of maximum likelihood and weighted maximum matching. We designed binary code based framework to reconstruct evolutionary history for whole genome gene orders. We developed algorithms to estimate/predict missing adjacencies in ancestral reconstruction procedure to restore gene order from species, when leaf genomes are far from each other. We developed a pipeline involving maximum likelihood, weighted maximum matching and variable length binary encoding for estimation of ancestral gene content, to reconstruct ancestral genomes under the various evolutionary model, including genome rearrangements, additions, losses and duplications, with high accuracy and low time consumption. Phylogenetic analyses of whole-genome data have been limited to small collections of genomes and low-resolution data, or data without massive duplications. We designed a maximum-likelihood approach to phylogeny analysis (VLWD) based

on variable length binary encoding, under maximum likelihood model, to reconstruct phylogenies from whole genome data, scaling up in accuracy and make it capable of reconstructing phylogeny from whole genome data, like triploids and tetraploids. Maximum likelihood based approaches have been applied to ancestral reconstruction but remain primitive for whole-genome data. We developed a hierarchical framework for ancestral reconstruction, using variable length binary encoding in content estimation, then adjacencies fixing and missing adjacencies predicting in adjacencies collection and finally, weighted maximum matching in gene order assembly. Therefore it extensively improves the performance of ancestral gene order reconstruction. We designed a series of experiments to validate these methods and compared the results with the most recent and comparable methods. According to the results, they are proven to be fast and accurate.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Contributions

Gene-order data have been extensively recognized and successfully conducted in the biological research over the last few decades. Although nucleotide sequences and amino acid sequences still dominate in phylogenetic research problems, gene order data aligned from the permutation of genes along chromosomes are likely having the potential to return more convincing and meaningful results. Since operations on genes are much harder to occur than point mutations at the nucleotide level, gene ordering allows researchers to trace further back in time than nucleotide sequences. A set of evolutionary events based on rearrangements of genes and modifications of gene contents has been biologically identified and mathematically modeled [5]. Deep mathematical and algorithmic methods to copy with gene order permutations have been developed to solve various biological problems. However, the performances are still far from satisfaction, and with the emerging of whole genome and high-resolution data, it is clear that novel approaches and algorithms are greatly in need to improve the performance of current solutions of these problems.

1.1 Literature Review

In 1936, Dobzhansky and Sturtevant, for the first time, proposed to use the degree of disorder between the ordering of genes in two genomes as a measurement of an evolutionary distance between species. They described a scenario of inversion events, to explain chromosomal differences among 17 groups of flies [57, 18]. But what on earth allows us to utilize the order of genes to carry out all kinds of studies in compar-

ative genomics? The key is that genes themselves are less subject to mutations and are therefore rarely cut by rearrangement [49]. Therefore by viewing a chromosome as a permutation of genes (or conservative blocks) in the order and several chromosomes are then placed as a genome. The organizing of geneorder data enables the reconstructing of evolutionary events far back in time [51, 7].

Watterson, Later in 1982, presented the very initial and formative definition of the chromosome inversion problem [66] and they were intended to come up with a distance measurement between two organisms, in order to reconstructing a phylogenetic tree.

So how to compute the minimum amount of inversion events (defined as the edit distance) to transform one genome into the other? Until nearly a decade later in 1995, Hannenhalli and Pevzner [29] provided the first polynomial algorithm for the chromosome inversion problem, in which their finding has greatly advanced the development of gene order research. The next significant progress in distance measurement between two genomes is the introduction of double-cut-and-join (DCJ) distance [70, 5]. Although DCJ is not directly and biologically observable or provable through gene order study, the DCJ distance is then extensively favored since it can emulate a variety of other events, while greatly simplifies the computational model.

Mostly, researchers who work on gene order data, focus on copying with two different yet related problems: the phylogenetic reconstruction problem and the ancestral reconstruction problem. Both together are widely known as Big Phylogeny Problem. The phylogenetic reconstruction problem aims to reconstruct the phylogeny in terms of a binary tree from a set of genomes of extant species, while the ancestral reconstruction problem searches for the most plausible gene order of an ancestral genome. An internal node in a phylogeny tree represent a ancestral genome.

A number of methods have been proposed for phylogenetic reconstruction problem from gene order and they can be roughly classified into parsimony-based and distance-based according to the standards they follow. Saitou [50] presented the first

distance-based method called `Neighbor-joining` aimed for treating DNA sequences. `Neighbor-joining` was soon adopted for solving the phylogeny problem using gene-order data since all distance-based methods are based on statistical clustering from a distance matrix computed between each pair of genomes, In 2002, Desper [17] presented a faster and more accurate algorithm for phylogeny reconstruction called `FastMe` based on the minimum-evolution principle and the nearest neighbor interchanges (NNIs). Since the edit distance often severely underestimates the true number of events, some forms of corrections are needed. Empirical derived estimation (`EDE`) [43] estimates the true number of inversions in which the minimum number of inversions is initially computed between two genomes and an empirical correction is applied based on a statistical model to compute the true inversion distance. Later Lin developed `TIBA` [37] which provides a more accurate estimation mechanisim for the true pairwise distances.

On the other hand, there are a wide collection of parsimony-based method for gene order based phylogenetic reconstruction problem. Most of these parsimony-based methods use direct optimization techniques. In particular, `BPAnalysis` [51] was the first program written by Blanchette and Sankoff in 1998, to reconstruct phylogenies based on the breakpoint parsimony of gene orders. Moret and Tang [43, 44] then in 2002 presented `GRAPPA` which greatly improves the results and the efficiency of `BPAnalysis` through replacing the breakpoint median solver with an inversion median solver. Around the same time, Bourque and Pevzner proposed the `MGR` [8] which instead of using the breakpoint distance, addressed the issue of handling multichromosomal genomes.

Another type of parsimony-based methods relies on the encoding techniques of gene order data, which transforms permutations into sequences and then uses existing analysis tools for sequential data to reconstruct a gene order phylogeny. In particular, Cosner proposed the first method of this kind application called Maximum Parsimony

on Binary Encodings (MPBE) [14, 15] which produces one character for each gene adjacency present in the data. Wang [65] later gave the second method called MPME (M stands for multistates) in which each signed gene has exactly one character. In all evaluations, both MPBE and MPME were easily surpassed by direct optimization approaches.

To date, however, probabilistic methods for solving the gene order phylogenetic reconstruction problem are introduced by a single effort from Larget [34], in which a Bayesian approach showed a evidence of success on a couple of fairly close data sets; this approach, however, failed to converge on a harder data set later analyzed by Tang [61].

Although gene duplications and losses have long been studied by molecular biologists, their integration with rearrangements in a unified model has seen relatively little work to date by bioinformists or computational scientists. In particular, Tang [61] introduced a way of determining the gene content when solving the median problem in GRAPPA. Later Zhang [74] presented a new distance measurement for genomes with gene inversions and losses which complies with triangle inequality standard. He showed it that his method is remarkably more accurate than its predecessors, while handling gene duplication is still out-of-reach. For distance methods, El-Mabrouk [20] first introduced an exact algorithm for the computation of edit distances with inversions and losses. More recently, Yancopoulos [71] suggested a way to compute edit distances under indels, duplications, and DCJ operations, and Swenson [58] developed an algorithm to approximate the true evolutionary distance under indels, duplications, and inversions for single chromosomal genomes, showing good results under simulation study. In 2011, Hu [30] introduced the first successful attempt to use ML reconstruction based on whole-genome data; later, Lin [36] developed a faster and more accurate yet simpler method MLWD in which they introduced a biased transition model and a simplified gene-encoding scheme.

For the ancestral reconstruction problem, a handful of methods have been developed using different methods and techniques. Traditional parsimony methods such as GRAPPA and MGR are capable to compute the phylogeny and ancestral gene orders at the same time, but are computational NP-hard problem. In order to boost the accuracy and scalability at the same time, many works were published in the last few years. MGRA relies on the notation of the multiple breakpoint graphs and is a more recent derivative of MGR developed by Alekseyev [1] in 2009. GASTS, later developed by Xu [69], is based on a fast and accurate heuristic for the inversion median solver which is developed by [48]. It can scale up linearly instead of exponentially with the size of the genomes involved. The Single-Cut-or-Join (SCJ) operation [23, 6] was proposed as a new rearrangement distance between multi-chromosomal genomes, leading to a fast median solver and Fitch-style algorithm for ancestral genome reconstruction.

A new framework InferCars has been established in 2006 by Ma [40] and attracted a lot of attention in the last a couple of years. Unlike previous methods which explicitly focus on a set of predefined evolutionary events, this framework focus on gene adjacencies and the goal is usually to determine how likely an adjacency can be observed in an ancestor. Later he presented a probabilistic version InferCarsPro [39] by incorporating a modified Jukes-Cantor model. Gagnon introduced a new concept of "Gapped Adjacency" and proposed a method called GapAdj [25] in 2012. GapAdj is considered flexible since it can handle data set with unequal gene-content. By mixing the framework of event-based (GRAPPA) and adjacency-based (InferCarsPro) methods, Zhang [75] proposed a method which inherits the high performance of direct optimization and reduces its difficulty by fixing a portion of adjacencies before the exact optimization step.

All the works presented in this dissertation has been accomplished with close collaboration with Dr.Jijun Tang. Only the works that we have taken the lead are presented, including Variable Length Binary Encoding (`VLBE`) and its successor Maximum Likelihood on Whole-genome Data (`VLWD`) for the phylogeny problem, Maximum Likelihood based method using Weighted maximum matching for Ancestral Genomics (`FARM`) and its extension for the ancestral genome reconstruction.

On Phylogenetic Reconstruction

In chapter 5, we described a series of maximum-likelihood approaches to phylogenetic analysis from whole genome data. Following the previous framework, `VLBE` enables `VLWD` to run significantly better, even the whole genome data set with a dozen of thousands genes can be analyzed within hours.

Our methods possess the following advantages:

(i) Our methods utilize the maximum-likelihood analyzing tools which allow them to run significantly better and faster than their parsimonious predecessors; even the whole genome data set with a dozen of thousands genes can be analyzed within hours. (ii) Our methods are very accurate and outperform the other competitors in almost all cases according to our simulation experiments. (iii) A remarkable advantage of our methods is their independence over evolutionary events, indicating that they can handle any existing event in an unified way.

Related publications are listed below.

1. Zhou, Lingxi, et al. "Phylogeny Reconstruction from Whole-Genome Data Using Variable Length Binary Encoding." Bioinformatics Research and Applications: 12th International Symposium, ISBRA 2016, Minsk, Belarus, June 5-8, 2016, Proceedings. Vol. 9683. Springer, 2016.

2. Zhou, Lingxi, et al. "Phylogeny Reconstruction Using Variable Length Binary Encoding." BMC Bioinformatics submitted 2016.

On Ancestral Genome Reconstruction

In chapter 3 and chapter 4, we described two methods for ancestral genome reconstruction `FARM` and its extension `FARM+`. `FARM` and `FARM+` fall into typical adjacency-based probabilistic approaches which try to answer how likely an adjacency to be observed in an ancestor. And in `FARM+`, we introduce an missing adjacencies prediction mechanism.

First, our methods are fast and is able to scale up to handle whole genome data polynomially. This is achieved by treating each adjacency in the leaf genomes as a unique and independent (binary) character. So we only need to compute a small portion of all possible adjacencies and also cut the number of states for an adjacency character to 2. Second, we adopted our biased transition model into the marginal reconstruction [72] to calculate the posterior probability of an adjacency in an ancestor. This model has been proved in `MLWD` to be very useful in phylogeny reconstruction. Third, `FARM` is able to handle gene losses, insertions and duplications, other than rearrangement. through a novel probabilistic approach for inferring ancestral genome contents using the variable length binary encoding. The underlying idea is straightforward: by treating each occurance of a gene as a bit, we can compute the probability of observing this occurance in an ancestral genome. Fourth, `FARM` implemented a more sophisticated way to assemble gene adjacencies into a valid gene order permutation. It replaces the greedy strategy and TSP solver with an exact solution by solving weighted maximum matching problem. This strategy not only massively increases the performance of method, but also significantly mitigates the issue of bad assembly of gene adjacencies.

Related publications are listed below.

1. Zhou, Lingxi, et al. "Ancestral reconstruction under weighted maximum matching." Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on. IEEE, 2015.

2. Zhou, Lingxi, et al. "Ancestral reconstruction with duplications using binary encoding and probabilistic model." Bioinformatics and Computational Biology (BICoB), 2015.

Chapter 2

Background

2.1 Introduction to Gene Order

It is the adequacy of DNA replication that accounts for the diversity among living organisms. A nucleotide sequence may evolve at the level of nucleotides in different types, and it is well known that nucleotide sequences may also evolve by modifying their orderings at a larger scale. Pioneerd by Dobzhansky and Sturtevant in 1936 [57], they, for the first time, proposed to use the degree of disorder between the permutation of genes in two genomes as a measurement of an evolutionary distance between organisms. They depicted a scenario of inversion to explain chromosomal difference between 17 groups of flies [18]. As more operations are later discovered and generalized, such large-scale evolutionary operations are often called genome rearrangements, including inversion, transposition and etc.; The other type of operations that change the gene content of the genome are typically insertion, deletion and duplication. The segments of a genome that all these operations act on are often biologically genes. Therefore from the view that a genome is the collection of genes in the order of which they are placed along one or more chromosomes, Because genes are less subject to point or bit mutations and are rarely cut by rearrangement [49], gene-order data enables the reconstruction of evolutionary history with events far back in time [51, 7].

2.2 Gene Order Format and Evolutionary events

Given a set of n genes labeled as $G = \{1, 2, \cdots, n\}$, a genome can be represented by an set of chromosomes of these genes. A chromosome can be linear or circular in which its end meets head. Each gene is presented with an orientation, which is either marked as i or -i. A gene can also be represented by two ends of it, like i being $(i^t \rightarrow i^h)$. Two genes i and j are *adjacent*, if i is immediately followed by j, or, equivalently, $-j$ is immediately followed by $-i$, and they therefore form an adjacency, denoted as (i^h, j^t). If a gene k is located at either end of a linear chromosome, we define k as being adjacent to an extremity e to mark the beginning or ending of a chromosome, noted as (e, k^h) or (k^t, e), called telomeres.

Genomic rearrangement events can change the ordering of genes on a chromosome, or exchange and combine content across chromosomes. For example, let G be the genome with a single linear chromosome,

$$G = \{(1, 2, \cdots, i-1, i, \cdots, j, j+1, \cdots, k, k+1, \cdots, n)\}.$$

An *inversion*, or reversal, reverses a segment of genes on a chromosome. An inversion between indices i and j $(i \leq j)$, transforms G to a new genome with linear ordering

$$G' = \{(1, 2, \cdots, i-1, -j, -(j-1), \cdots, -i, j+1, \cdots, n)\}.$$

A *transposition* on genome G acts in this way, three indices i, j, k, with $i \leq j$ and $k \notin [i, j]$, picking up the interval $i, i+1, \cdots, j$ and inserting it immediately after k. Thus genome G is replaced by (assume $k > j$)

$$G' = \{(1, \cdots, i-1, j+1, \cdots, k, i, i+1, \cdots, j, k+1, \cdots, n)\}$$

There are other events that are common as well. *Translocation* breaks at two chromosomes and reattaches a part to another chromosome. A *fusion* joins two chromosomes, while *fission* splits one chromosome into two. A *deletion* deletes a single or

a segment of genes from a genome, while its opposite operation, an *insertion*, introduces a gene or a segment of genes that have not been presented into a chromosome at a time.

As to the multiple copy of genes, a *whole genome duplication* (*WGD*) accounts for the operation on an ancestral node, by which genome G is transformed into

$$G^{'} = \{(1, 2, \cdots, i-1, i, \cdots, j, j+1, \cdots, k, k+1, \cdots, n),$$

$$(1^{'}, 2^{'}, \cdots, i-1^{'}, i^{'}, \cdots, j^{'}, j+1^{'}, \cdots, k^{'}, k+1^{'}, \cdots, n^{'})\}.$$

A segment genome duplication operates on one gene or a piece of genes instead of entire genome. For instance, the genome G is transformed into

$$G^{'} = \{(1, 2, \cdots, i, ..., j, j+1, ..., k, i^{'}, ...j^{'}, k+1, ..., n-1, n)\},$$

for $1 \leq i \leq j \leq k \leq n$.

Given two genomes G_1 and G_2, we define the *edit distance* $d(G_1, G_2)$ as the minimum number of events required to transform one into the other. The *inversion distance* between two genomes measures the minimum number of inversions needed to transform one genome into another. Hannenhalli and Pevzner [29] developed a mathematical and computational framework for signed gene-orders and provided a polynomial-time algorithm to compute inversion distance between two signed gene-orders; Bader et al. [3] later showed that this edit distance can be computed in linear time.

Yancopoulos et al. [70] proposed a universal double-cut-and-join (DCJ) operation that accounts for common events such as inversions, translocations, fissions and fusions, which resulted in a new genomic distance that can be computed in linear time. Although there is no direct biological evidence for DCJ operations, these operations are very attractive because they provide a simpler and unifying model for genome rearrangement.

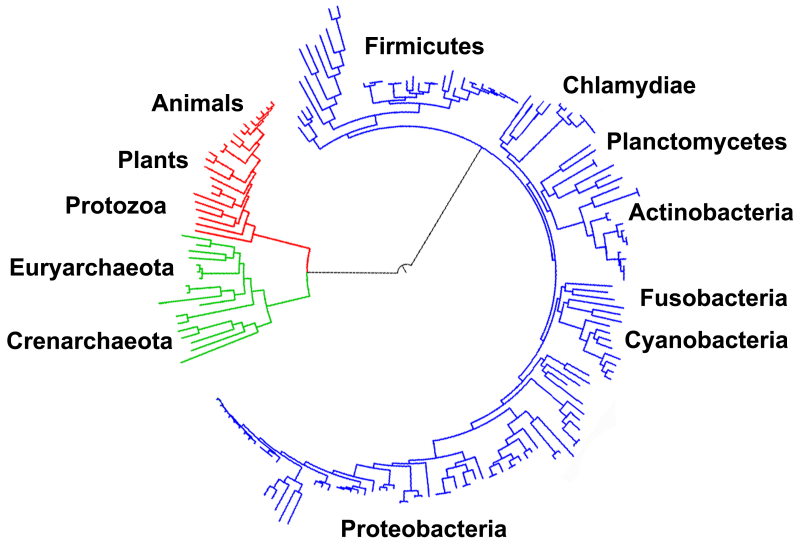


Figure 2.1 A highly resolved Tree Of Life, based on completely sequenced genomes, from $https://commons.wikimedia.org/wiki/File:Tree\_of\_life\_int.svg$

2.3 Phylogenetic Reconstruction with Gene Order Data

A phylogeny is a term that represents the reconstructed evolutionary history of a set of organisms in the form of a binary tree (rooted or un-rooted), in which the given set of organisms are descendants placed at the leaves, and internal nodes stand for extinct ancestors connected by the edges. Figure 1 shows a highly resolved Tree Of Life, based on completely sequenced genomes.

Many types of data can be used to reconstruct phylogenetic history from geographic and ecological, through the morphological and metabolic to the molecular data [60]. By the rapid accumulation of molecular data and also due to its merit of exact and easy accessibility, sequence-based data of a few genes long has become the predominant source for phylogenetic analysis. But it suffers from some prominent issues, especially, the well-known gene tree vs. species tree problem [46, 41]. Gene order data, a relatively novel and promising data type, studies the whole-genome at the same time from a higher-level perspective and hence naturally avoids the gene tree vs. species tree problem. At the meanwhile, there are great mathematical challenges

encountered in detecting and handling the genome-scale changes, not to mention to employ existing techniques directly for sequences data. In the recent years, the phylogenetic reconstruction from gene-order data has drawn a lot of attention from both computer scientists and biologists. Researchers have developed many methods [35] in coping with this problem.

Methods for phylogenetic reconstruction of gene-order data can be roughly classified into three groups according to the criterion they follow.

- Maximum likelihood based methods: MLBE [30], MLWD [36], VLWD [77].
- Distance-based methods: Neighbor-join [50], FastME [17] and TIBA [37]
- Parsimony-based methods: BPanalysis [7], GRAPPA [43], MGR [8], SCJ [22], MPBE [15] and MPME [43].

Neighbor-joining and FastME use a bottom-up clustering method for the creation of phylogenetic trees. Distance-based methods are sometimes favored due to their excellent scalability with the number and size of genomes as well as an acceptable performance they can achieve. Their performances largely depend on how the distance measurement is defined and how well such lengths are congruent with the real distance. Although Hannenhalli and Pevzner [29] provided the first polynomial algorithm for computing the minimum number of inversion between two genomes, the actual evolutionary distance, is always severely underestimated. To approach the real number of evolutionary operations, Lin et al. proposed TIBA that relies on a simple structural characterization of a genome pair under the DCJ model and significantly improves the accuracy of distance methods.

Parsimony methods are built upon the fundamental assumption that the real phylogeny along with a set of ancestors must minimize the total number of evolutionary operations required to generate the descendants from a common root node. Every tree traversed is scored by summing the edit distance between the two nodes of each

edge. In the context of gene-order data, the first heuristic BPAnalysis was proposed by Blanchette et al. based on the breakpoint distance. BPAnalysis enumerates all $(2n-5)!!$ trees and uses an iterative heuristic to label the internal nodes with signed gene orders. To improve the speed and performance of BPAnalysis, Moret et al. later reimplemented BPAnalysis and developed GRAPPA. GRAPPA not only successfully augmented the BPAnalysis with more sophisticated search strategies and high-performance algorithmic engineering, but also showed excellent extensibility to accommodate newly-defined evolutionary distance. However, parsimony methods following direct optimization depend on solving numerous instances of the median problem. In particular, the median problem is defined as given three genomes, search the fourth genome that minimizes the sum of the distances between it and the other three. But for most evolutionary distance, solving for the exact median genome is still NPhard [11, 9, 63]. Therefore, direct optimization methods are rather accurate but also extremely time-consuming. One exception is the breakpoint like Single-cut-or-join (SCJ) which has a polynomial time solution for the median problem, but in overall, an exact, branch-and-bound search for the phylogeny with SCJ is still NP-hard.

Later, other parsimony methods that transform gene-order data into sequence-like string have also been proposed. For example, MPBE (Maximum Parsimony on Binary Encoding) converts adjacency pairs from the signed permutation into strings of binary characters. These strings are further converted into nucleotide sequences and analyzed using common sequence parsimony software (e.g. PAUP* 4.0 [59]) to return a phylogeny. Wang et al. later proposed a new set of encodings schemes called MPME (Maximum Parsimony on Multistate Encoding) to improve the accuracy. These encoding-based parsimony methods can achieve slightly better accuracy compared to the neighbor-joining method, yet they are still computationally very expensive.

Maximum-likelihood based methods that also transform gene-order data into sequence-like string have been proposed in the past a few years. MLBE [36] by Hu encodes gene order information into a binary sequence based adjacency list. Then the sequence is converted into a sequence of amino acids by some strategy. Finally the amino-acid-like sequence is fed into a maximum likelihood approach, like RAxML to reconstruction the phylogenetic history. Later, Lin [36] proposed a new scheme (MLWD) to encode whole genome data into binary sequences and also designed an independent transition model for state change, leaving out the amino acids enco ding apart. This approach has been proven to be of high accuracy, due to applying its state transition model. Since this method avoids the process of transforming binary encodings into artificial biological sequence and directly use ML reconstruction program RAxML to build a tree from these sequences, it also significantly reduces its running time to a very lower level.

2.4 Ancestral Gene Order Reconstruction

The success of phylogenetic reconstruction has demonstrated the power of revealing the evolutionary relation of a group of organisms by computational means. As phylogeny often takes the form of the rooted binary tree, each internal node of the tree can be naturally regarded as the common ancestor of the living organisms descended from it. The predication of ancestral gene orders of these internal nodes has been further investigated by both computer scientists and biologists, and a numb e of methods have been developed to attack this problem.

Depending on whether or not the phylogeny tree is given, ancestral genome reconstruction problem can b e classified into the small phylogeny problem (SPP) and the big phylogeny problem (BPP). The SPP defines when the phylogenetic tree is given, and the goal is only to reconstruct the ancestral genomes, while the BPP searches the most appropriate tree along with a set of ancestral genomes with an optimized score,

usually. In this study, we are interested in tackling the small phylogeny problem. Most of the current metho ds solving SPP adopt either adjacency-based approach in which rearrangements are only implicitly considered or rearrangement-based approach that involves computing numerous instances of median problems. In particular, adjacency based metho ds mainly fo cus on the analysis of indep endent gene adjacencies, try to calculate or estimate a score for each gene adjacency to b e present in an ancestor. A graph in which genes and adjacencies are vertices and edges is often constructed, and gene adjacencies are rejoined into contiguous ancestral regions (CARs) by optimizing the total score.

From another point of view, some methods employ a parsimonious framework and suggest to use least number of changes to explain observed data; while the rest estimates the parameters and use probabilities or likelihood to score the gene adjacencies. Table 2.1 summaries the difference between a number of methods for solving SPP given gene-order data.

Table 2.1 Summary of current methods for solving small phylogeny problem (SPP) from gene-order data.

	Parsimonious	Probabilistic
$Adjacency - based$	InferCARs [40] GapAdj [25]	InferCARsPro [39] PMAG's [21, 31] FARM [76]
$Rearrangement - based$	GRAPPA [43], MGR [8, 1] GASTS [69] SCJ [47]	N/A

In the context of rearrangement-based parsimonious methods, the median problem can be formalized as follows: given a set of m genomes with permutations $\{x_i\}_{1 \leq i \leq m}$ and a distance measurement d, find another permutation x_t such that the median score defined as $\sum_{i=1}^{m} d(x_i, x_t)$ is minimized. GRAPPA and MGR (as well as their recently successors) are similar methods that implemented a set of median solvers for phylogeny and ancestral gene order reconstruction. However solving even the sim-

plest case of median problem when m equals to three is NP-hard for most distance measurements [12, 9, 63]. Specifically, GRAPPA, given a tree topology, iteratively assigns median genomes to ancestral nodes in the tree until converged. Then the set of gene order assignments that minimizes the tree score are reported as the resulting ancestral genomes. Since the scoring procedure of GRAPPA involves solving numerous instances of median problems, a fast median solver is playing a crucial crucial rule in this method. Exact solutions to the problem of finding a median of three genomes can be obtained for the inversion, breakpoint and DCJ distance measurements [13, 53, 68]. Among all the median solvers, the best one is the DCJ median solver proposed by Xu and Sankoff (ASMedian [68]) based on the concept of adequate subgraph. Adequate subgraphs allow decompositions of an multiple breakpoint graph into smaller and easier graphs. Though the ASMedian solver could remarkably scale down the computational costs of median searching, it yet runs very slow when the genomes are really distant. On the other hand, GASTS and SCJ are two heuristic methods that are scaled up to handle high-resolution vertebrate genomes. GASTS is based on a fast and accurate heuristic strategy for the inversion median [48] problem searching procesure, in which only a few of the simplest decompositions of adequate graphs will be solved. It provides a fast and robust scoring approach for a fixed tree and presents very high accuracy in the simulation experiments, compared to MGR. Signle-cut-or-join (SCJ) defines a breakpoint-like operation under which the median problem and SPP can be resolved in polynomial time. It utilizes the Fitch's small parsimony algorithm to solve the SPP, in which each adjacency is viewed as a binary character of state, being either present or absent. Ultimately, all adjacencies are determined in ancestral genomes. This is the only known evolutionary operation for which the SPP has a polynomial time solution.

Adjacency-based parsimonious method was formally introduced in InferCARs by Ma in 2006 for the first time. It identifies a most-parsimonious scenario for the

changes of each individual adjacency, introduces weights to the graph edges and uses a greedy heuristic approach to search for vertex-disjoint paths in the graph. Such path is known as contiguous ancestral regions (CAR). Later Ma introduced InferCARsPro—an suceessor of the previous work in the probabilistic framework for reconstructing ancestral genomes. The kernel of InferCARsPro is to predict the posterior probability of observing an certain adjacency in the ancestral node based on an extended Jukes-Cantor model for breakpoints. However, neither of them is able to handle data set with unequal gene content and greedy heuristic often returns a large number of CARs. Besides, both methods require users to input a phylogeny with accurate branch lengths. To deal with these problems, GapAdj is developed to handle unequal gene contents and uses TSP solver to assemble gene adjacencies into genomes with a more reasonable number of CARs at a little sacrifices of accuracy. The core of GapAdj is to consider a pair of genes, separated by up to a give number of genes, as direct gene adjacency. GapAdj can also analyze data sets of unequal gene contents by first inferring the ancestral gene content through a natural procedure proposed in [27].

The adjacency-based Maximum likelihood method was first introduced in InferCARsPro by Ma in 2010 and later formally described by Hu in 2013. Given a set of genomes, along with its corresponding phylogenetic tree, *PMAG* series methods go through three Phase 1: estimating the gene content of an internal node to predict genes likely to present in this node; Phase 2: calculating the probability of each gene adjacency collected from given data set; Phase 3: formalizing and solving an assembly problem to place genes on chromosomes.

The content of a genome can be encoded into a sequence as in PMAG+ [31]. For a gene i, if it does not appear in genome G, we will mark it as 0 in the sequence representing G; otherwise, it is then encoded by its number of appearance and if the copy number is larger than 9, it uses letters from A (10) t0 V. Table 3.2(c) shows

the encoding of the two genomes in this table, where gene 3 appears twice in genome G_1, while gene 4 does not occur in G_2. This encoding scheme by itself has several limitations. First, it can only copy with data set with copy number of a gene no larger than 32. Second, the encoded output, which serves as the input for RAxML, ignores the transition model in it. Third, when there is an missing state of in the encoded sequence, RAxML fails to return a sound result. Considering them, we designed a new encoding scheme to overcome these shortcomings. The detail can be found in Chapter 3.

Given the gene order of a genome, we also can easily obtain a set of adjacencies equivalently representing each chromosome from the genome, and form a binary sequence that specifies presence or absence of all the adjacencies [36], by viewing all chromosomes in each genome as linear and applying an one-to-one encoding. A gene i can be denoted by its head i^h and tail i^t, so that there are a total of four scenarios for two consecutive genes a and b in forming an *adjacency*: $\{a^t, b^t\}$, $\{a^h, b^t\}$, $\{a^t, b^h\}$, and $\{a^h, b^h\}$. If gene c is a telomere, we have a corresponding singleton set, $\{c^t\}$ or $\{c^h\}$. A genome can then be expressed as a multiple-set of adjacencies and telomeres. We further write 1 (0) to indicate presence (absence) of an adjacency and we consider only those adjacencies and telomeres that appear at least once in the input genomes. For instance, genome $G_1 = (1, 2, -3, 3, 4)$ will be encoded with a set of adjacencies $T = \{(1^h, 2^t), (2^h, 3^h), (3^t, 3^t), (3^h, 4^t), (0, 1^t), (4^h, 0)\}$. For an encoded adjacency $t = (i^x, j^y)$ and $t' = (j^y, i^x)$, x and y $\in \{h, t\}$, t and t' are equivalent to each other. Table 3.2 shows a result of encoding two artificial genomes into binary sequences. So given a set of N genomes, *PMAG* methods apply this encoding to each chromosome producing N adjacency sets, $T_1, ..T_N$, and recording each unique adjacency into an adjacency list A. They then conduct a binary encoding for each adjacency set in terms of the unique adjacency list A, generating N binary sequences. Once obtaining the binary sequence encoding from input gene orders, *PMAG* meth-

ods use the extended probabilistic approach for sequence data, described by Yang [72], to compute the probability at each site. It applies the RAxML package, to estimate the conditional probability for each site and the evolutionary distance, t, for each branch. *PAMG** iterates through steps, as described above, to compute the probability of all adjacencies for each internal genome.

when these probabilities are obtained, all *PMAG* methods convert the problem into a assembling problem–find the ancestral adjacencies. In PMAG, ancestral adjacencies are assembled by the greedy heuristic based on the adjacency graph proposed by Ma [39]. This greedy method starts from a contig with the first gene and picks its neighbor by using the adjacency with the highest probability; it then continues adding new genes until there is no more valid connection, in which case the current contig is closed and a new one will be formed. There are two issues with this approach that motivated us to replace the greedy assembler with an new solver. First, the greedy heuristic can achieve good approximation only when the data set is closely related in which case most vertices in the graph have only one outgoing edge. Second, the greedy heuristic tends to return an excessive number of contigs as it frequently leads itself into dead end.

In PMAG+ and PMAG++, obtaining gene orders from (conflict) adjacencies can be transformed into an instance of symmetric Traveling Salesman Problem (TSP), as shown in [25] and [62]. In this case, we can transform gene ends into cities and adjacency probabilities into edge weights in the TSP graph. However, the TSP problem is NP hard, currently the best TSP solver is limited within number of 85,900 cities, so these methods have difficulties in handling large genomes (with thousands of genes), and the returned tour is not necessarily optimal, when a heuristic strategy is used to scale up its input size. To overcome these and achieve better result, we reduce these assembly problem into a Weighted Maximum Matching Problem. Detailed description of our work will be covered in Chapter 3.

Chapter 3

Ancestral reconstruction under weighted maximum matching

3.1 Motivation of *FARM*

Our new method *FARM* is designed to reconstruct ancestral genomic content and its ordering under a flexible evolutionary scenario, which includes various evolutionary events, including rearrangements, additions, losses, and duplications. Given a set of genomes, along with its corresponding phylogenetic tree, this framework goes through five phases: content estimation, adjacency collection, probability computation, adjacency selection, and gene order assembly. The rest of this section describes these steps in detail with an example input as given in Figure 3.1 and Table 3.1, in which genomes experienced events of inversion, insertion, deletion, and duplication.

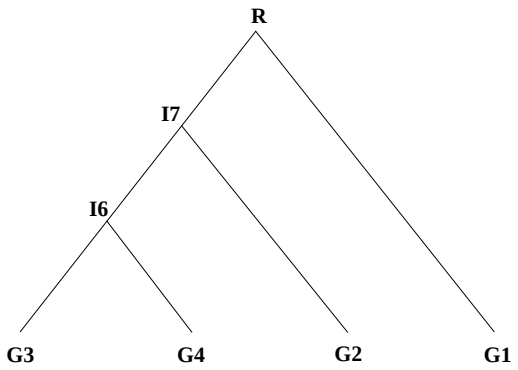


Figure 3.1 A tree of four species.

3.2 Content Estimation

Given the information of leaf species and the phylogeny topology, *FARM* first predicts all possible ancestral gene content in the target node. Unlike the method with rearrangement events only, in which every genome has exactly a complete and equal copy of genes, every internal genome here has to consider all of the gene copies observed in the leaves since a gene might either be absent or present in multiple copies.

The inference procedure views each observed gene as an independent character with multiple states. Specifically, given a data set D with N species and that a set of n distinct genes $S = \{g_1, g_2, ..., g_n\}$ are observed. For each leaf species $G_i \in D$, it has gene content $S_i = \{g_{i_1}, ..., g_{i_k}\}$ possibly with $g_{i_x} = g_{i_y}$ when $x \neq y$. It can be equivalently represented by a set of copy number, $\pi = \{\pi_{g_1}, \pi_{g_2}, ..., \pi_{g_n}\}$, in which each element g_{i_j} has a copy value, if $T_{i_j} = \{g \mid g = g_{i_j} \cap g \in S_i\}$, $\pi_{i_j} =\mid T_{i_j} \mid$; otherwise $\pi_{i_j} = 0$ for $1 \leq j \leq k$. For instance, a total of six distinct genes $\{1, 2, 3, 4, 5\}$ can be identified from four species *G1*, *G2*, *G3* and *G4* with gene orders as represented in Table 3.1, respectively. However, differing from what's applied in PMAG+ to estimate the gene content for the target node, *FARM* needs to deal with multiple copies of a gene. Considering this, we adopt a multiple state encoding (ME) scheme for genes, inspired by our multi-state encoding described in [30]. Our encoding of gene content is analogous to the encoding of gene adjacencies.

The number of copies of each gene in the ancestral nodes is going to be estimated, by expanding our alphabet from binary to multi-state. We use difference characters to represent different number of copies. First we sort all the labels of genes in ascending order and for each leaf genome, we go through every gene and put 1 if one copy gene of that gene is found or 0 otherwise. For the number of gene is larger than 9, til 35, we use characters from $A \cdots Z$ to encode it.

Given three genomes $G1, G2, G3$ and $G4$ as shown in Table 3.2(a), by applying the encoding of gene content, we come up with the sequences shown in Table 3.2(b).

Table 3.1 Gene orders of four species.

species					
G1	2	1	3	4	-5
G2	-1	-2	-4	-3	
G3	-2	-4	-3	2	1
G4	2	1	2	-4	

Since the encoded sequences have no difference with a common aligned sequences, by giving the true phylogeny, we are able to infer the gene content in the ancestors. Therefore our inference of gene content shares the same paradigm with the posterior calculation of gene adjacencies;

however the existence status of a gene is purely determined by its own probability. In particular, if the probability of seeing character , say, "1", at the site is greater than 0.5, we regard the gene as presence, otherwise it is absence. Once we finish the inference of gene content for the ancestor under inference, those adjacencies that contains absent genes are filtered out from the assemble of genome. It is worth noting that by relabeling discontinuous gene identifiers into continuous ones, we can still use the same greedy heuristic to assemble gene adjacencies.

3.3 Adjacency Collection

A genome can equivalently be encoded as a set of adjacencies, representing each chromosome from the genome. In this dissertation, we view all chromosomes in the input genome as circular and apply a one-to-one encoding on each gene. Given a gene i, we encode it into $i^h \rightarrow i^t$, or, in the case of -i, as $i^t \rightarrow i^h$. So a genome can be encoded in to a set of adjacencies. For instance, a circular chromosome (1, -2, 4, 3) will then be encoded with a set of adjacencies, $T = \{(1^t, 2^t), (2^h, 4^h), (4^t, 3^h), (3^t, 1^h)\}$. For an encoded adjacency $t = (i^x, j^y)$ and $t^{'} = (j^y, i^x)$, x and y $\in \{h, t\}$, t and $t^{'}$ are equivalent to each other.

Given a set of N genomes, we apply this encoding to each genome producing N

adjacency sets, $T_1, ..T_N$, and recording each unique adjacency into an adjacency list A. We then conduct a binary encoding the same to PMAG++ for each adjacency set in terms of the unique adjacency list A, generating N binary sequences. As shown in Table 3.5 and 3.3, we give the adjacencies set for each genome and binary sequences of them, provided the input in Table 3.1.

3.4 Probability Computation

Once we collect the binary sequence encoding from input gene orders, we use the extended probabilistic approach for sequence data, described by Yang [72], to compute the probability at each site. We apply the same software package RAxML as *PMAG* methods have used, since it can infer ancestral states of large scale sequence data. As in our example, if we want to compute the probability for each adjacency at internal node I6, we simply re-root the input tree while reserving the topology, to put the target node I7 at root position.

Table 3.2 Example of encoding gene orders into binary sequences.

$$\begin{aligned} G_1 &: (1,\ 2, -3,\ 3,\ 4) \\ G_2 &: (3, -2,\ 1) \end{aligned}$$

(a) Two signed linear genomes with inserted/deleted and duplicated genes

	1	2	3	4
G_1	1	1	2	1
G_2	1	1	1	0

(b) Sequences for gene contents

	$\{1^t\}$	$\{1^h, 2^t\}$	$\{2^h, 3^h\}$	$\{3^t, 3^t\}$	$\{3^h, 4^t\}$	$\{4^h\}$	$\{2^t, 1^t\}$	$\{1^h\}$	$\{3^t\}$
G_1	1	1	1	1	1	1	0	0	0
G_2	0	0	1	0	0	0	1	1	1

(c) Binary sequences for gene orders

Table 3.3 A multi-state encoding with three species.

species	1	2	3	4	5
G1	1	1	1	1	1
G2	1	1	1	1	0
G3	1	2	1	1	0
G4	1	2	0	1	0

Table 3.4 Estimated result for the two internal nodes.

species	1	2	3	4	5
I6	1	2	1	1	0
I7	1	1	1	1	0

3.5 Weighted Maximum Matching (WMM) in Adjacency Selection

Given a undirected graph $G(V, E)$, a matching M in G is a set of pairwise non-adjacent edges; that is, no two edges share a common vertex. A maximum matching is a matching that contains the largest possible number of edges. If edges are assigned with weights, a weighted maximum matching (WMM) algorithm is then used to find a maximum matching with minimum score. This problem can be solved in polynomial time using Edmonds' algorithm [19]. For demonstrative purpose, Figure 3.2 shows a graph with its weighted maximum matching solution highlighted with dashed lines.

To select the desired adjacencies, we build a graph based on the leaf species' adjacency set $A = \{a_1, a_2, ..., a_m\}$, content information $S_I = \{g1, g2, ..., gk\}$ estimated in Table 3.4, as well as probabilities estimated from Table 3.5. We encode each gene into multiple copies, if necessary, so that each possible occurrence is preserved in the

Table 3.5 Adjacencies of four species.

species					
G1	$(2^t, 1^h)$	$(1^t, 3^h)$	$(3^t, 4^h)$	$(4^t, 5^t)$	$(5^h, 2^h)$
G2	$(1^h, 2^t)$	$(2^h, 4^t)$	$(4^h, 3^t)$	$(3^h, 1^t)$	
G3	$(2^h, 4^t)$	$(4^h, 3^t)$	$(3^h, 2^h)$	$(2^t, 1^h)$	$(1^t, 2^t)$
G4	$(2^t, 1^h)$	$(1^t, 2^h)$	$(4^h, 2^h)$	$(2^t, 4^t)$	

Table 3.6 Binary encoding of adjacencies for genome G1, G2, G3 and G4.

$(2^t, 1^h)$	$(1^t, 3^h)$	$(3^t, 4^h)$	$(4^t, 5^t)$	$(5^h, 2^h)$	$(2^h, 4^t)$	$(3^h, 2^h)$	$(1^t, 2^t)$	$(4^h, 2^h)$	$(2^t, 4^t)$	$(1^t, 2^h)$
1	1	1	1	1	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0
1	0	1	0	1	0	1	1	0	0	0
1	0	0	0	0	0	0	0	1	1	1

target node. With gene content we have estimated in Table 3.4 right, for gene *2*, it's going to have two copy in ancestral node I6. So we extend it into *2a* and *2b*. Gene *2a* and *2b* have all adjacencies (together with their probabilities) that gene *2* has.

We keep this mapping information by M Note we have expected no gene *5* in internal node I6. We get a set of genes $S = \{1, 2a, 2b, 3, 4\}$. We build a graph G(V, E). The set of nodes V include each gene $g \in S$. If two ends v, $u \in S$ and adjacency $(u,\ v) \in A$, edge $(u,\ v)$ belongs to E. As the estimated probabilities range from 0 to 1, using them directly as edge weights may introduce undesirable impact associated with handling small float points. The most straightforward way is to linearly correlate the edge weight with its probability, however in such case, differences of weights between adjacencies are too strong and adjacencies with smaller probabilities can hardly be considered. To assign weight to each adjacency in a precise and fine-grained

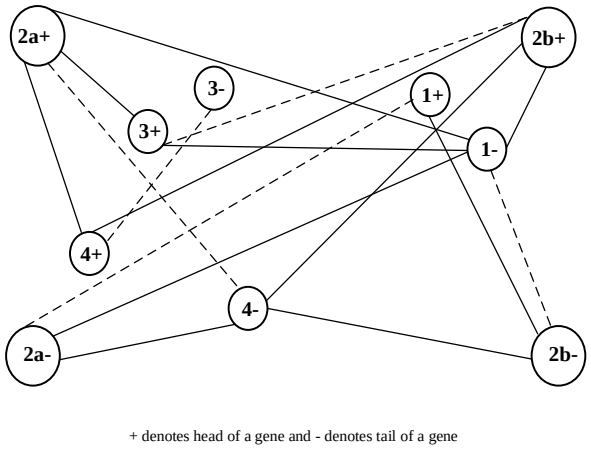


Figure 3.2 Weighted Maximum Matching Graph.

way(guarantee the return result is optimal), we curve up probabilities to sound edge weights using the following formula:

$$w_{(f,g)}(m) = log_2(10^m \times (1 - p_{(f,g)}) + t) \tag{3.1}$$

Here, $(f, g) \in A$, and $p_{(f,g)}$ is the probability of the observed adjacency *(f, g)* and is the sole parameter determining the shape of the curve. m and t are two shift parameters, ensuring $w_{(f,g)}(m)$ is within the range of $(0, log_2(10^m))$. According to our experiments, *FARM* yields good results when $m = 6$ and $t = 1$, empirically. *FARM* applies a revised N-cubed weighted matching algorithm to solve this maximum matching problem we've formalized.

It then selects a set of potential adjacencies from the given encoded markers. With this conversion, Table 3.5 is transferred into weighted edges in Graph G. Then we apply the weighted maximum matching algorithm to that graph. As shown in Figure 3.2 and we get selected adjacency set

$R = \{(1+, 2a-), (1-, 2b-), (2a+, 4-), (2b+, 3+), (4+, 3-)\}$.

Table 3.7 Probabilities of adjacencies for genome G1, G2, G3 and G4.

$(2^t, 1^h)$	$(1^t, 3^h)$	$(3^t, 4^h)$	$(4^t, 5^t)$	$(5^h, 2^h)$	$(2^h, 4^t)$	$(3^h, 2^h)$	$(1^t, 2^t)$	$(4^h, 2^h)$	$(2^t, 4^t)$	$(1^t, 2^h)$
1.000	0.957	1.000	0.957	0.957	0.999	0.999	0.999	0.999	0.976	0.976

3.6 Gene Order Assembly

When *FARM* gets the selected set of adjacencies of encoded markers, we work toward recovering the estimated gene order in two steps:

We, first, chain them up by the encoding nature that, i^h and i^t are two ends of a marker i and decode the adjacencies back to the gene-like order of encoded markers;

Second, then apply the mapping relation M to map the encoded mark back to real gene order domain and in this step, duplicated genes are recovered. As in our example, we get the gene order for node $I6 = \{(-1, -2, -4, -3, 2)\}$.

Since we add telemere markers to encode both ends of each chromosome from leaf genomes, we will easily get a chromosome by viewing the gene order between two telemere markers as one. In the TSP solution by Hu, multiple connected extremities are shrank to a single one and a segment genes between two extremities are taken as a contig. Our construction of matching topology is a little different, we add only a specical marker to encode all the extremities of each chromosome. It remains the finall assembled contig number much closer than TSP solver to real ones. However GapAdj requires extra steps and information to adjust the contig number. Instead our inference of ancestral genome is uniform and directly from the solution of WMM, minimizing the risk of introducing artifacts. This assembly mechanism, while maintaining the assembled contig number in a very accurate way, will sometimes add one or two rearrangment events to the final chromosome gene order.

3.7 Experimental Results

Experiments setup

To evaluate the performance of *FARM*, we generate a set of simulation gene order data. The simulating procedure is carried out as follows. First, we produce a birth-death tree T, which obeys the same way as [35]. Then we find the longest path between two leaf nodes, with length = K. We apply different evolutionary rates $r \in \{1, 2, 3, 4\}$ so that the tree diameters are in the range of $d \in \{1n, 2n, 3n, 4n\}$: larger diameter means a genome is more distant from its ancestor, and hence more computationally expensive this data set will be. By timing $1/K$ to tree diameter, we then get the length for a certain branch, but right now each branch on a tree has the same length. To vary the length of each branch, we apply a variation coefficient to each branch in this way: given a parameter c, for each branch we sample a number s uniformly from the interval (-c,c) and multiply the original branch length by e^s. For

the experiments in this paper, we set c with the value of 1. Thus, a branch would get its length L get by,

$$L = r \times n \times (1/K) \times e^s$$

For evolving on each branch, we use a series of evolutionary events, including inversions, fusions, fissions, translocations, indels, segment duplications and whole genome duplications. We set each event with a specific value of probability to be selected during the simulation process.

We set up comparative experiments with InferCarsPro, GASTS and PMAG++ to evaluate the performance of *FARM* under equal content model where each gene occurs exactly once in each genome and deletion, insertion and duplication are not allowed. As PMAG++ methods are still the most flexible for ancestral genome reconstruction to date for unequal content ancestral genome reconstruction, we only compare *FARM* with PMAG++ under unequal content. Within equal content testing, the genome settings for all methods are 10 genomes and 1000 genes (considering the capability of InferCarsPro), each data set with 80% inversion and 20% translocations. Within equal content, we also test on large scale. The genome setting is 40 genomes and 5000 genes. Since both InferCarsPro and GASTS cannot handle large scale data, we only compare *FARM* with PMAG++ on this data set. For the unequal content testing, the genome settings for both use 10, 20, and 40 genomes containing 2000 genes, and 10, 20, and 40 genomes containing 5000 genes. Each of these setups are generated both without WGD, 5 chromosomes per genome, and with WGD at root, 10 chromosomes per genome.

We generate 10 data sets for each setting and report the average accuracy of content and adjacency using the equation

$$E = \frac{|T \cap T'|}{|T \cup T'|} \times 100\%,$$

where T represents the amount of gene content, or gene adjacencies and telomeres

in the true ancestral genome, and T' represents the amount of gene content, or gene adjacencies and telomeres in the reconstructed genome.

We also report the average absolute difference of contigs per node using

$$\frac{\sum_{i=1}^{N} |c_i - C|}{N},$$

where C is the number of chromosomes of the true ancestor and c_i is the actual number of contigs in the reconstructed genome. In our experiment, this value is set to 5 in the test without whole genome duplication, and 10 for data set with whole genome duplication.

Small scale comparison under equal content

In this section, we pick three main competitors from both event-based and adjacency-based methods, and compare them with *FARM*. In particular, we supply InferCARsPro with multichromosomal genomic distances as its branch lengths computed by GRIMM [64].The event-based method GASTS is simply run by providing the true evolutionary tree and the input genomes.

As shown in Figure 3.3, we give the comparison on average adjacency accuracy for reconstructed genomes. Both InferCarsPro and GASTS present significantly lower accuracy than *FARM*. *FARM* runs slight better than PMAG++, and both of them conserves the same trend of performance.

For the performance on assembly accuracy, we summarized the number of contigs produced by various methods and computed the average absolute difference per node for all cases in Figure 3.4. From the figure, the event-based method GASTS and the TSP solver based method PMAG++ produced more relevant number of contigs than *FARM* does, but the difference is really small.

InferCarsPro performs the worst among all the methods and as the evolutionary rate gets larger, the result is getting worse. For the time consumption, InferCarsPro

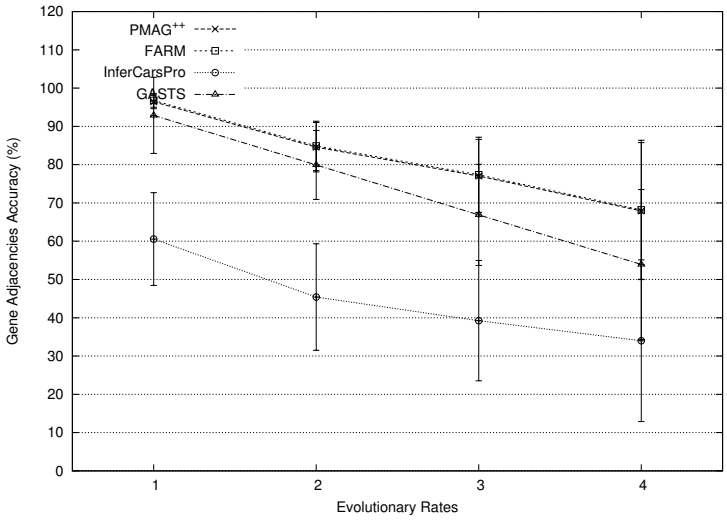


Figure 3.3 Accuracy of adjacency on data with 80% inversions, 20% translocations. the x-axis represents the evolutionary rate for each data set with 10 genomes and 1000 genes, by which the tree diameter is $\{1 \times 1000, 2 \times 1000, 3 \times 1000, 4 \times 1000\}$.

.

does the worst and takes 445 mins to finish the easiest case, which is with evolutionary rate of 1. GAST could get back a result within an hour and PMAG++ within 10 mins. *FARM* does the best and can finish every setting with 3 mins. It completes all the test cases almost at the same time level, even though the tree diameter is getting larger.

Large scale comparison under equal content

We compare *FARM* with PMAG++ to evaluate the performance under rearrangement only with large scale data set. As shown in Figure 3.6, we give the comparison on average adjacency accuracy for reconstructed genomes. *FARM* runs slight better than PMAG++ for all the cases, while both of them conserves the same trend of performance. For the performance on assembly accuracy, we summarized the number of contigs produced by both methods and compute the averages of assembly accuracy for all cases in Figure 3.7. From the Figure, we can see that *FARM* shows great assembly performance, and is significantly better than PMAG++.

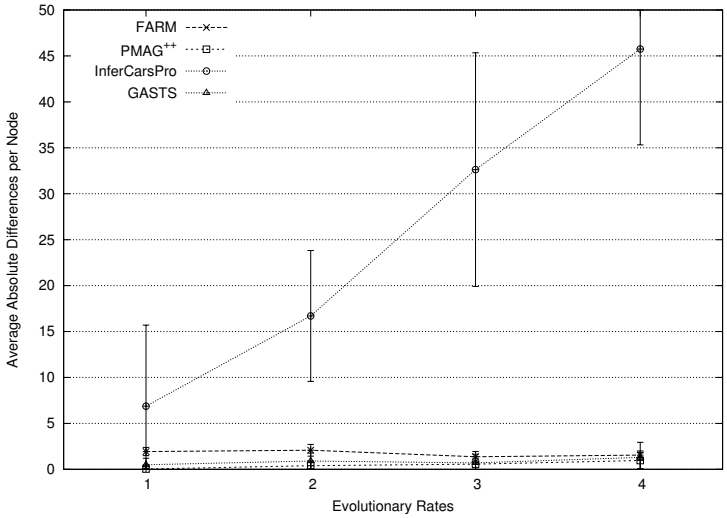


Figure 3.4 Average Absolute Difference per node for contig number with 80% inversions, 20% translocations. the x-axis represents the evolutionary rate for each data set with 10 genomes and 1000 genes, by which the tree diameter is $\{1 \times 1000, 2 \times 1000, 3 \times 1000, 4 \times 1000\}$.

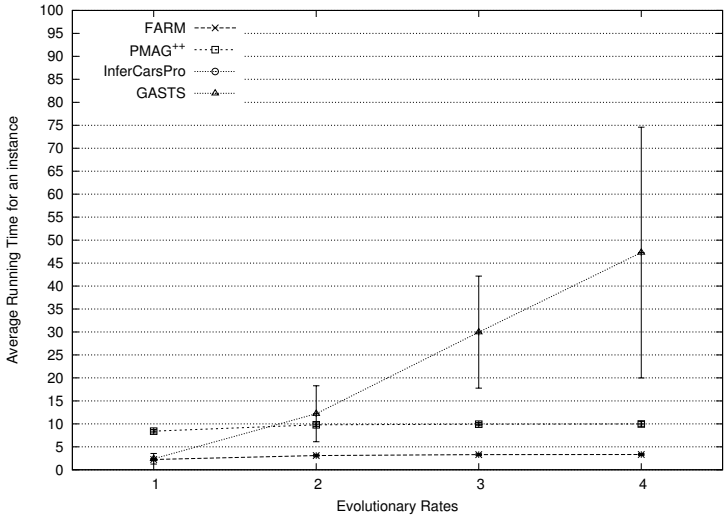


Figure 3.5 Time for running on data with 80% inversions, 20% translocations. Since *InferCarsPro* takes 445 mins when the evolutionary rate r = 1, the curve for its running time doesn't display on the figure. the x-axis represents the evolutionary rate for each data set with 10 genomes and 1000 genes, by which the tree diameter is $\{1 \times 1000, 2 \times 1000, 3 \times 1000, 4 \times 1000\}$.

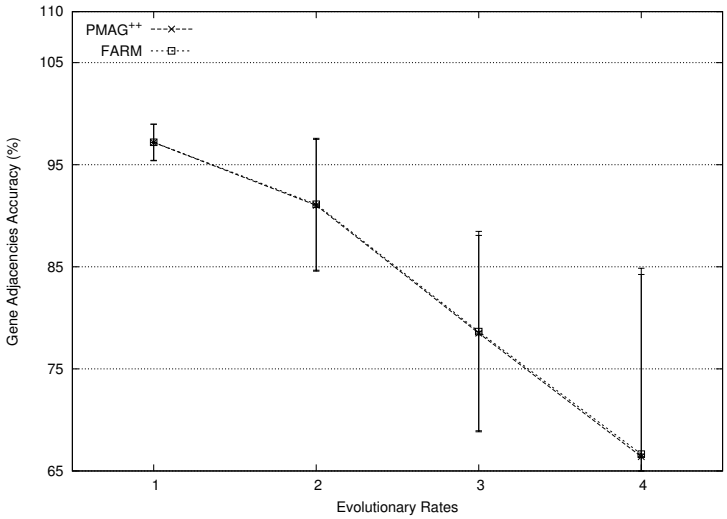


Figure 3.6 Accuracy of adjacency on data with 80% inversions, 10% translocations, 5% fissions and 5% fusions. the x-axis represents the evolutionary rate for each data set with 40 genomes and 5000 genes, by which the tree diameter is $\{1 \times 5000, 2 \times 5000, 3 \times 5000, 4 \times 5000\}$.

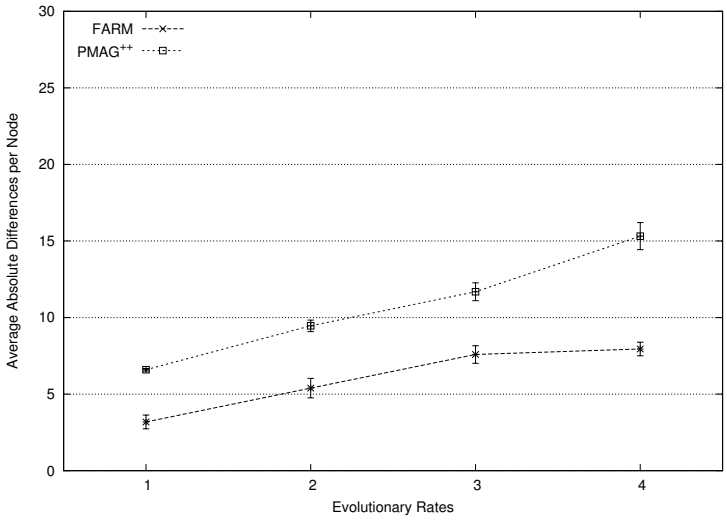


Figure 3.7 Average Absolute Difference per node for contig number with 80% inversions, 10% translocations, 5% fissions and 5% fusions. the x-axis represents the evolutionary rate for each data set with 40 genomes and 5000 genes, by which the tree diameter is $\{1 \times 5000, 2 \times 5000, 3 \times 5000, 4 \times 5000\}$.

Comparison under unequal content

As we have mentioned, *FARM* and PMAG++ both aim to formulate the conditional probabilities of gene adjacencies, however due to applying TSP solver to handle assembling, it is much more computationally demanding than *FARM*. In this section, we compare the performance of *FARM* to PMAG++ on data set without whole genome duplication and with whole genome duplication, together with other evolutionary events.

To compare on data set without whole genome duplication, we set the evolutionary setting as described is Figure 4.6. In our experiments we see that *FARM* always outperforms PMAG++ for every data setting on adjacency accuracy as shown in Figure 3.8. It confirms that *VLBE* does performs better than multiple state encoding in the phase of content estimation of PMAG++. *FARM* can achieve a minimum average accuracy of above 70% in our testing cases. The improvement on adjacency accuracy is much more significant than PMAG++, when the tree diameter r, is getting larger. As for the performance on contig assembly, both of them have comparable performance, as we can see from Figure 3.9. *FARM* can approximately reflect the actual number of chromosomes in the true genomes as PMAG++ does.

To compare on data set with whole genome duplication, we set the evolutionary setting as described in Figure 4.7. *FARM* continues to have a stable performance on ancestral genomes assembling, when compared with the performance on the data set without WGD. As shown in Figure Figure 3.10, in the most difficult case ($50k \times 10$ and $r = 4$), *FARM* presents an improvement of more than 10 percent in adjacency accuracy. Although the performance on contig assembling is slightly lower, when compared with PMAG++, it is still competitive to each counterpart as shown in Figure 3.11.

All tests are conducted on a workstation of 2.4Ghz, 8 core CPU and 4 GB RAM. In Figure 3.13 and Figure 3.12, we summarize the running time of each method in

each test case. Figure 3.12 and 3.13 also indicate another significant achievement in this work is that *FARM* generally runs 3-5 times faster than PMAG++. PMAG++ is more computationally demanding than *FARM*, for which PMAG++ is limited to copy with small tree diameter data sets, while larger tree diameter shows little impact on the running time of *FARM*.

3.8 Conclusion

In this study, we implement a Flexible Ancestral Reconstruction Method embedded with maximum likelihood and a weighted maximum matching algorithm. The achievement in this work is we apply the weighed maximum matching to the ancestral reconstruction problem, which can be computed in polynomial time. That allows *FARM* to be a flexible framework for the ancestor inference problem, which can be extended into real gene order data. We set up comparison experiments with InferCarsPro, GASTS, and PMAG++ separately with various genomic settings and evolutionary rates, under both equal and unequal content model. According to the results, we can see that *FARM* can not only outperform other methods under both

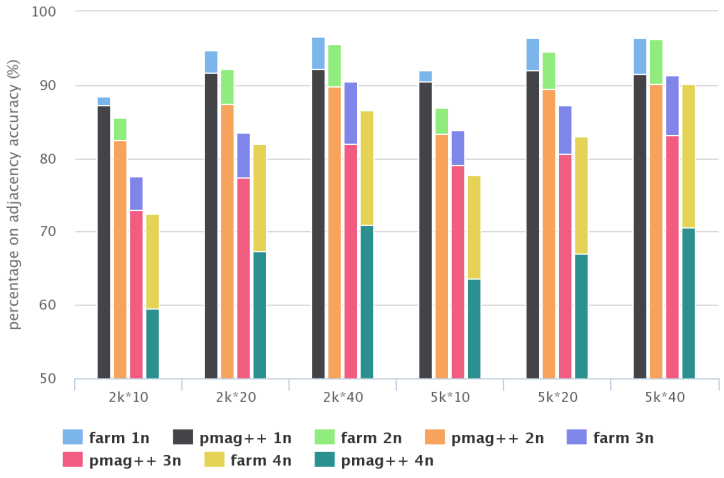


Figure 3.8 Accuracy of adjacency on data with 60% inversions, 5% fissions, 5% fusions, 10% translocations, 5% insertions, 5% deletions, 10% duplications. $n \times N$ means the datasets have n genes and N genomes.

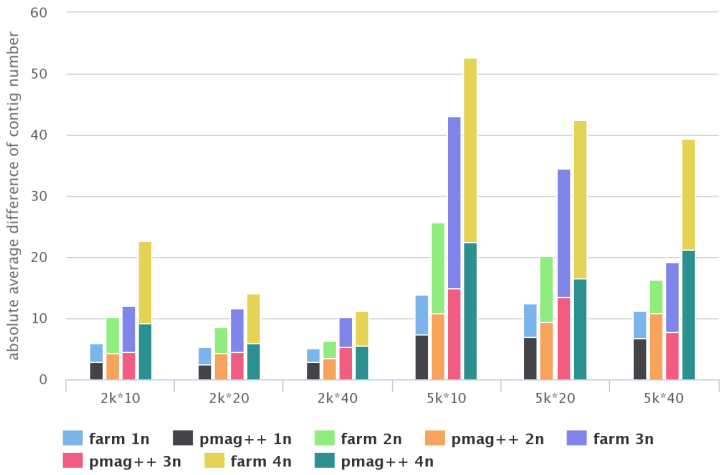


Figure 3.9 Absolute average difference of contig number on data with 60% inversions, 5% fissions, 5% fusions, 10% translocations, 5% insertions, 5% deletions, 10% duplications. $n \times N$ means the datasets have n genes and N genomes.

equal content and unequal content model, in term of accuracy, but also achieves a significant reduction in running time. This is because the weighted maximum matching problem can be solved in polynomial time, while the TSP solvers embedded in PMAG++ is an NP-hard problem. So *FARM* is fast and also flexible across a wide

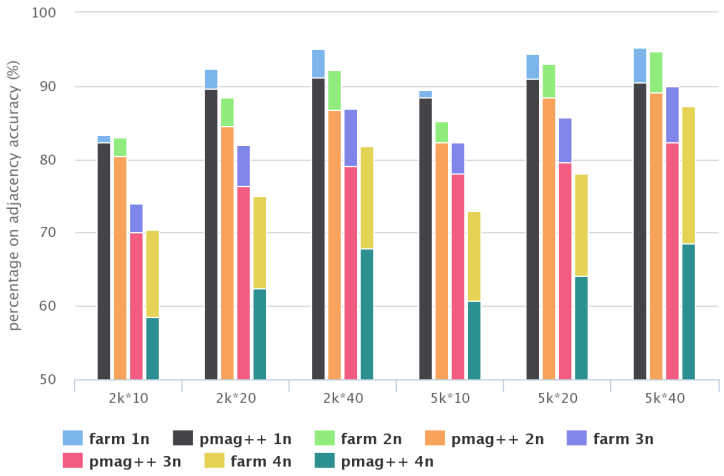


Figure 3.10 Accuracy of adjacency on data with 60% inversions, 5% fissions, 5% fusions, 10% translocations, 5% insertions, 5% deletions, 10% duplications, and one whole genome duplication on the root node. $n \times N$ means the datasets have n genes and N genomes.

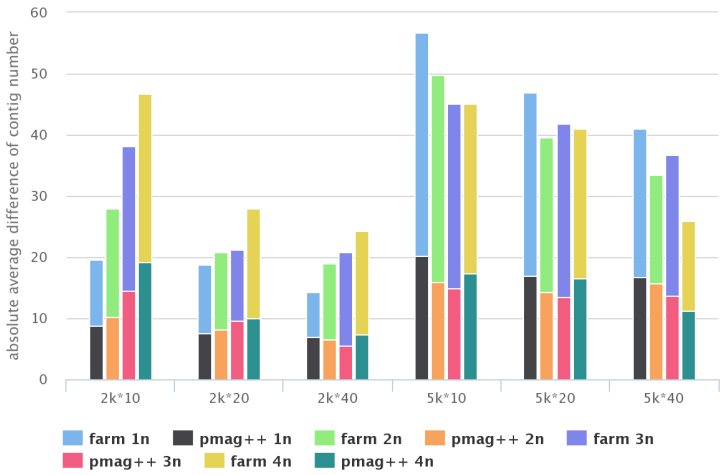


Figure 3.11 Absolute average difference of contig number on data with 60% inversions, 5% fissions, 5% fusions, 10% translocations, 5% insertions, 5% deletions, 10% duplications, and one whole genome duplication on the root node. $n \times N$ means the datasets have n genes and N genomes.

range of configurations and can be further applied into ancestral reconstruction on real biological gene order data.

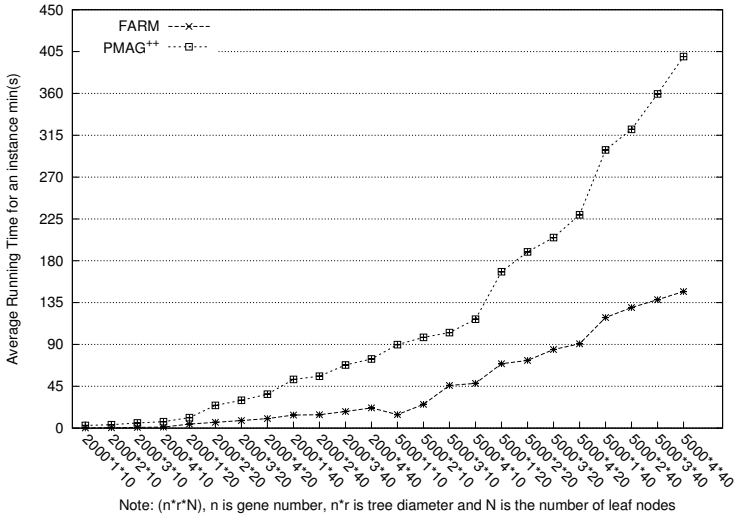


Figure 3.12 Running time of *FARM* over *PMAG+* and *FARM* over *PMAG++* (in minute). The data sets (without whole genome duplication)are represented as $n \times N \times d$, indicating they have n genes, N genomes and the tree diameters are $n \times d$.

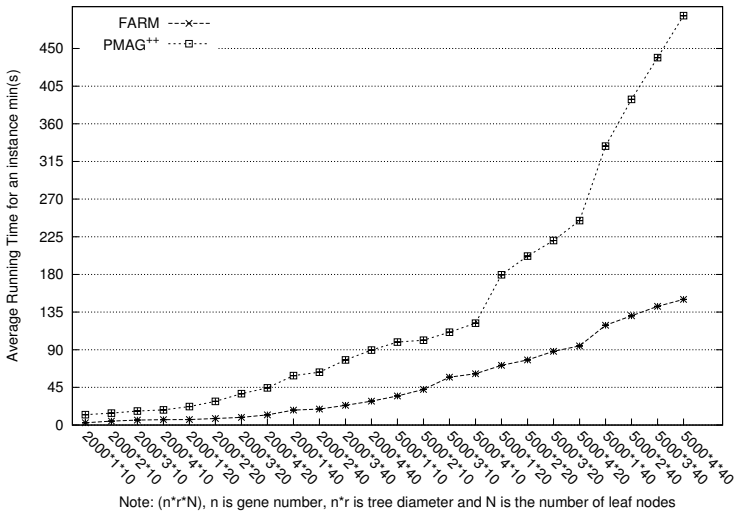


Figure 3.13 Running time of *FARM* over *PMAG+* and *FARM* over *PMAG++* (in minute). The data sets (with whole genome duplication) are represented as $n \times N \times d$, indicating they have n genes, N genomes and the tree diameters are $n \times d$.

Chapter 4

Ancestral Reconstruction with Adjacency Enhancement

4.1 Motivation

As described in the last chapter 3, after the calculation of probabilities for observing each gene adjacency in an ancestor, the final task is to assemble gene adjacencies into valid gene orderings. Since multiple options are available from a gene to another, an efficient algorithm is much needed for the assembly. In the past, by modeling the problem into an instance of TSP problem, an exact solution can be successfully found. In GRAPPA, TSP solvers are implemented for solving the breakpoint median problem. Later Tang [62] proved that the problem of searching the longest path in a graph by visiting each gene's head and tail exactly once is indeed a TSP problem; however the edge weights can be either 0 or 1 which is oversimplified and indistinguishable. Recent method GapAdj [25] developed a better scoring mechanism to score the gene adjacencies and reduced the problem to TSP problem. Before that, Ma [40] proposed a greedy heuristic to stepwise add heaviest edges (highest probabilities) into the path until no edge can be added and then detect and break cycles by removing the edge with the smallest weight. This heuristic procedure has been implemented in InferCars [40].

In PMAG [21], Hu chose to adopt the greedy heuristic to assemble the gene adjacencies based on the facts that heuristic is efficient and produced acceptable results according to our simulation study. However, there are mainly two reasons

to find a substitution for existing strategies. For the first, the greedy heuristic can only achieve good approximation, when the dataset is closely related in which case most nodes in the graph have only one outgoing edge. For the second, as a sign of bad assembly, greedy heuristic tends to return an excessive number of contiguous ancestral regions (CARs) that is partly due to missing adjacencies. In PMAG+ [31], Hu applied a TSP solver strategy to assemble gene adjacencies in to gene orders. However, still, the performance of TSP relies heavily on an appropriate construction of graph and assignment of edge weights that fit our problem.

As coverd in Chapter 3, to solve an ancestral genome reconstruction problem by adjacency-based methods heavily depends on the leaf genomes, since they are the raw materals (containing adjacencies we need) for later gene order assembly for an internal genome. Ideally, we want the leaf material containing all the adjacencies for ancestor genome. However, when it's in the case where genomes of given leaf nodes are evolved from distant tree topology that the true adjacencies presented in the ancetral nodes are enormously different from the adjacencies from leaf nodes, it prevents existing adjacency-based methods from achieving a good enough result. As we mentioned in Chapter 3, InferCarsPro, PMAG series methods and FARM are trying their best to score each observed adjaceny with a unique value, e.g, InferCarsPro directly uses probablity as adjacency weight, so that, by each choosing strategy, they could select a set of adjacencies, which could optimize the ancestral genome to as close as possible to the true one. However, there are usually conflicts existing to achieve a optimal one in a general model. An example of explanation for this will be gvien later. 2011, Zhang [75] proposed a framework to improve the ancestral genome reconstruction through fixing adjacencies estimated from a maximum likelihood method [39]. In his work, he uses ASMedian to absorb adjacencies from PMAG into ASMedian. Zhang's method produces more accurate ancestral genomes than the maximum likelihood method while the computation time is far less than that of pure median method.

On the other hand, other than being difficult to score adjacencies, there are a significantly large amount of adjacencies missing from leaf genomes. As we can see from the statistics from Table 4.1, the leaf nodes can only provide 76.4% of the materials that could contribute (percentage) in the reconstruction process. In other word, there is 23.6% the materails missing from the procedure. Both enable *FARM* to reconstruct ancestral genomes with significant improvement.

Table 4.1 Adjacency missing rate with under genome setting with 1000 genes and 60 genomes, of 40% inversion, 5% fission, 5% fusion, 10% translocation, 10% insertion, 10% deletion and 20% duplication.

PMAG methods				
Tree Diameter	1×1000	2×1000	3×1000	4×1000
Loss percentage	3.2%	7.3%	14.6%	23.6%

With these two observed, we are inspired to either exclude a set of adjacencies from being ambiguity or absorb a set of adjacencies that are very likely to present in the ancestral genome. Besides, we apply the variable length binary encoding in the step of gene content estimation to preserve multiple copies of content and adjacencies. Through these stategies, we could, to a large extent, improve the correctness in ancestral reconstruction.

Before we get into our algorithms, we first give some definitions that we'll use later.

Definition for functions

- $A - structure$

  In a tree, we define a node with its two children as $A - structure$. If its two children are leaf nodes, we say it is a leaf $A - structure$.

- $Adj(g)$

  Given a node (in tree representation) or a genome (in gene data representation), $Adj(g)$ represents all the adjacencies it contains.

- $Left(g)$

  Given a node g, $Left(g)$ represents its left child.

- $Right(g)$

  Given a node g, $Right(g)$ represents its right child.

- $Parent(g)$

  Given a node g, $Parent(g)$ represents its parent.

- $Uncle(g)$

  Given a node g, $Uncle(g)$ represents the other child of $Parent(Parent(g))$, if there is one.

- $AdjIntersect(g_1, ..., g_n)$

  Given several nodes or genomes, $AdjIntersect(g_1, ..., g_n)$ represents the set of adjacencies presenting in all nodes or genomes.

- $UnionIntersect(g_1, ..., g_n)$

  Given several nodes or genomes, $UnionIntersect(g_1, ..., g_n)$ represents the set of all adjacencies presenting in these nodes or genomes.

4.2 Variable Length Binary Encoding (VLBE) in Content Estimation

Thanks to the applying of Variable Length Binary Encoding in FARM as shown in Chapter 4, the evolutionary history inference can be improved with the assisting of Varialbe Length Binary Encoding as we have tested out in last chapter. Given the information of leaf species and the phylogeny topology, *FARM* first predicts all possible ancestral gene content in the target node. Unlike the method with rearrangement events only, in which every genome has exactly an whole and equal copy of genes,

every internal genome here has to consider all of the gene copies observed in the leaves since a gene might either be absent or present in multiple copies.

The inference procedure views each observed gene as an independent character with multiple states. Specifically, given a data set D with N species and that a set of n distinct genes $S = \{g_1, g_2, ..., g_n\}$ are observed. For each leaf species $G_i \in D$, it has gene content $S_i = \{g_{i_1}, ..., g_{i_k}\}$ possibly with $g_{i_x} = g_{i_y}$ when $x \neq y$. It can be equivalently represented by a set of copy number, $\pi = \{\pi_{g_1}, \pi_{g_2}, ..., \pi_{g_n}\}$, in which each element g_{i_j} has a copy value, if $T_{i_j} = \{g \mid g = g_{i_j} \cap g \in S_i\}$, $\pi_{i_j} = \mid T_{i_j} \mid$; otherwise $\pi_{i_j} = 0$ for $1 \leq j \leq k$. For instance, a total of six distinct genes $\{1, 2, 3, 4, 5\}$ can be identified from four species *G1*, *G2*, *G3* and *G4* with gene orders as represented in Table 3.1, respectively. However, differing from what's applied in PMAG+ to estimate the gene content for the target node, *FARM* needs to deal with multiple copies of a gene. Considering this, we adopt a variable-length binary encoding (VLBE) scheme for genes, and the advantages that VLBE has over the *multiple-state* encoding that PAMG++ adopted are, 1), VLBE has no limitation in copy number of a gene, by which the multiple-state encoding is limited within 32 states; 2), the VLBE more accurately describes the transferring cost from one state to the other. In this encoding scheme, the cost for a transition is proportional to the gap between two transferring states. However, the transiting cost in multiple-state encoding is following a neutral transition model of protein. 3), this encoding scheme by itself avoids the issue caused by missing states (There will be no missing states in the coded sequences), because the RAxML could not handle sequence with missing states.

The *VLBE* goes in this way: (1) screen through each genome from input data set and capture the maximum state T and maximum gene marker M. (2) then for each genome, we allocate an M chunks of blocks, each block with T cells. For each chunk at i, it stores a copy number information for gene i that encode each gene $g_i = i$ into a binary sequence s_i of length T, using the number of 1s to indicate occurrences of

that gene and place these 1s in right first order. The rest is filled with 0's. (3) we append each s_g at the right end of s_{g-1}. As shown in table 3.4, The gene content of genome G_1, G_2, G_3 and G_4 are encoded into four binary sequences correspondingly.

4.3 Improve Ancestral Reconstructin by Fixing Adjacencies

As we can see in Figure 4.1, in internal node $I1$, we are observing a set of adjacencies

$$Adj(I1) = \{(1^h, 2^t), (2^h, 5^h), (5^t, 4^h), (4^t, 3^h), (3^t, 1^t)\}.$$

If we were provided with these adjacencies of high probabilities, we'll have edges, mapping to these adjacences, with low weight in the matching graph, in which we are trying to extract a set of adjacencies for assembling ancestral genome $I1$. However, it is difficult to locate them correctly. Let's go a little bit further. What evolved from $I1$ are $Adj(left(I1)) = \{(1^h, 2^t), (2^h, 5^h), (5^t, 3^t), (3^h, 4^t), (4^h, 1^t)\}$ and $Adj(right(I1))$ $= \{(1^h, 4^t), (4^h, 5^t), (5^h, 2^h), (2^t, 3^h), (3^t, 1^t)\}$.

So $AdjIntersect(\ Adj(left(I1)),\ Adj(right(I1))) = \{(1^h, 2^t),\ (2^h, 5^h)\}$, and intuitively, adjacencies in this set are going to be assigned with high probability in $I1$'s reconstruction raw materials. But still beyond enough. So we put our attention on set $AdjUnion(Adj(left(I1)),\ Adj(right(I1)))\ - AdjIntersect(\ Adj(left(I1)),$ $Adj(right(I1)))$.

Let's look at the upper structure, the subtree with $Parent(I1)$ as its root, here, $Parent(I1) = Root$. In the right child of $Root$, we see $Adj(right(Root)) = \{(2^t, 1^h),$ $(1^t, 3^t), (3^h, 4^t), (4^h, 5^t), (5^h, 2^h)\}$. We find that adjacencies $\{(3^t, 1^t), (3^h, 4^t), (4^h, 5^t)\}$ from genome $G3$, can be found either in genome $G1$ or $G2$. Most likely, thse adjacencies are coming from the common ancestor – $Root$, and we can applies this observing in ancestral reconstruction procedure, by fixing them with relatively high probability as well. So here we propose a Adjacency Fixing algorithm to improve the performance of ancestral reconstruction under Maximum likelihood and weighted

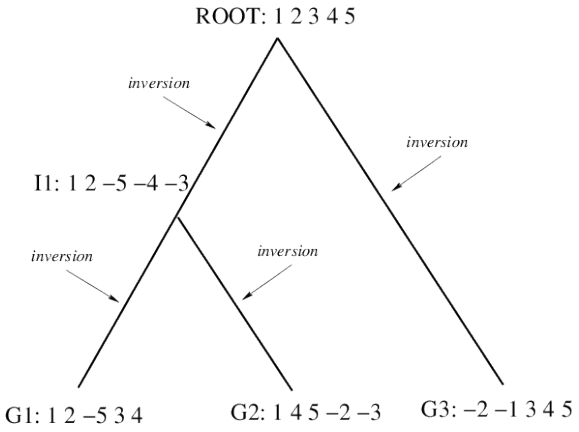


Figure 4.1 A phylogenetic topology of three genomes, showing step by step how new adjacencies are evolved.

maximum matching framework. Since extracting adjacencies from matching graph using weighted maximum matching algorithm, to some extent, let's say, is to avoid bad decision in choosing, through $AdjFix$ algorithm much more "correct" or "expected" adjacencies are signally enhenced to be chosen during the course of applying WMM.

Algorithm $AdjFix$ 1. Find an $A - structure$ in tree topology;

2. Extract the intersect of adjacencies in two children;

3. Find the uncle of root of $A - structure$;

4. Find the union of $Adj(left(root(A)))$ and $Adj(Uncle(root(A)))$, and the union of $Adj(right(root(A)))$ and $Adj(Uncle(root(A)))$, add them in to the raw materials in $root(A)$.

5. Iterate up in the tree until every internal node is fixed.

4.4 Ancestral Reconstruction from FARM

We use Figure 4.2 to demonstrates an example where the gene adjacency $(1, 2)$ in the ancestor $I1$ is missing in all its descendants—$G1$ and $G2$. Non-observed adjacencies is assigned with an extremely large number in the Weighted Maximum Matching graph in order to guarantee bypassing of these edges. However as we mentioned, it

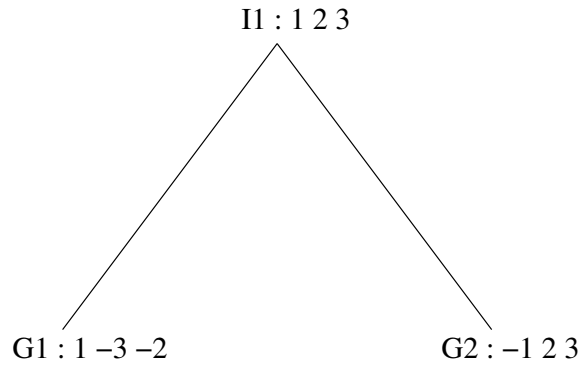


Figure 4.2 Demonstration the loss of gene adjacencies in descendant genomes.

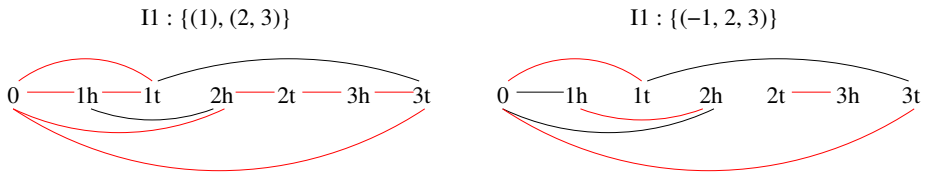


Figure 4.3 Ancestral node inferred by the greedy heuristic (left) and WMM (right).

is possible that a number of gene adjacencies show up in an ancestral genome but are not found in any leaf genome. In case of using heuristic, missing edges lead to additional CARs since the greedy heuristic is not able to foresee the breakage until it is met. In contrast, WMM procedure can always find a complete set of edges at the cost of introducing false positives. Using figure 4.2 as a toy dataset, figure 4.3 shows the WMM graph constructed from the leaf genomes as well as the inferred ancestors from both approaches. We used red color to indicate the edges included in solution. Table 4.2 summaries the results in which the last column "dist" measures the DCJ distances between true and inferred ancestor. Results implies that WMM is weaker against the impact of missing adjacencies as it must always return a single complete path. It is also noticeable that although greedy heuristic can recover one more correct adjacency, their DCJ distance remains the same.

Since adding one missing adjacencies can at least reduce two false positives. If we can retrieve some of the missing adjacencies back into the WMM graph with adequate weights, the result of WMM will be enhanced. Based on the observation that none

Table 4.2 Comparison of inferred ancestors against true ancestor.

	True Adjs	FP	FN	Dist
Greedy	(0,1) (2,3) (3,0)	(1,0) (0,2)	(1,2)	1
WMM	(2,3) (3,0)	(0,-1) (-1,2)	(1,2)	1

of current adjacency-based methods is capable of retrieving missing adjacencies while rearrangement-based methods such as GASTS can find a large portion of missing adjacencies back in their solution, we conducted a test which shows GASTS is able to retrieve around 60% of missing adjacencies when genomes are not distant (20 out of 29.5 is found at $1n$ diameter). According to this finding, we propose to use a mixture framework of both type of methods to enhance the performance of FARM. This framework relies GASTS to initialize all internal nodes and then uses a randomized method inspired from Zhang's work [75] to add missing adjacency into TSP graph. In particular, the framework follows these steps:

1. Run GASTS and FARM separately and compute the collection of adjacencies which are in GASTS but not in FARM. The collection must contain all missing adjacencies GASTS can retrieve.

2. Randomly select a certain percentage of adjacencies from the collection and add the adjacencies to the WMM graph. Their edge weight is set to the average weight of edges in the solution of WMM. Selected adjacencies contain both correct and incorrect adjacencies, however correct adjacencies are more likely to stay in the new solution. Thus we keep track of the appearances of selected adjacencies in the new WMM solution.

3. Repeat step 2 a certain times and sort the frequencies of appearances for all adjacencies in descending order.

4. Starting from the adjacency (g, f) with the current largest number of appearances, we remove it from the list and add it to the graph with minimum weight

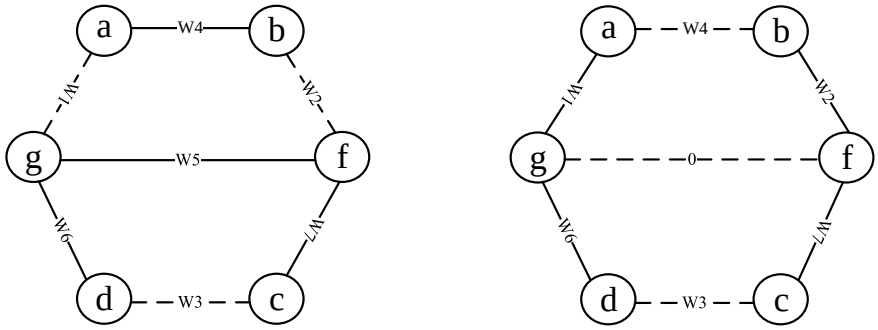


Figure 4.4 The scenarios before and after adding the (g, f) with 0 edge weight.

of 0. Suppose current WMM score is S and (g, a), (b, f), (d, c), (a, b), (g, f), (g, d), and (c, f) are the seven edges connecting with a, b, c, d, g and f of costs at $w1$, $w2$, $w3$, $w4$, $w5$, $w6$ and $w7$ respectively as shown in figure 4.4.

5. In the new WMM solution with score S', by connecting (g, f), we assume two edges (g, a) and (b, f) are removed. Therefore if $S - S' \geq w1 + w2 - w4$ which indicates adding such adjacency at least does not increase the previous WMM score, we then trust it as a missing adjacency.

6. If an adjacency is trusted, we update the current best WMM score and the WMM graph, then repeat the step 4. If the list is empty or $S - S' < w1 + w2 - w4$, we stop the whole process and return the current WMM solution as our final result.

The rationale behind this procedure is that adding a missing adjacency not only releases two detouring edges ($(0, 1t)$ and $(1h, 2h)$ as in figure 4.3) but also allow the released genes to connect to their correct genes (0 and $1h$ are released and can now join into the correct adjacency $(0, 1h)$). Thus in overall, the gain in WMM score by adding a missing adjacency is expected to be no less than just rescuing two detouring edges.

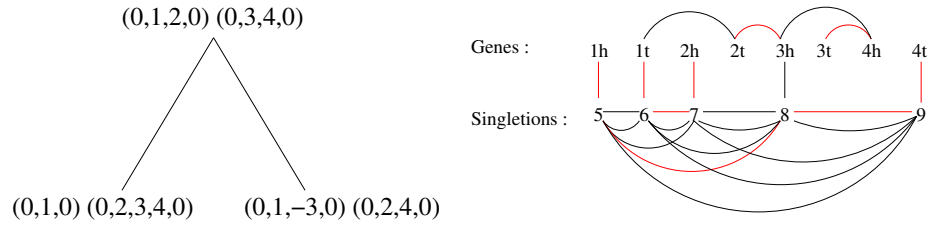


Figure 4.5 Demonstration of mapping telomeres of chromosomes into unique singletons.

Another aspect of proposed work is to amend the WMM graph to handle multi-chromosomal genomes, since current graph can only return a single CAR containing all the genes; on the other hand, greedy heuristic always generates a large number of CARs. Investigation of this issue has been make in GapAdj, as its name implies, it uses the concept of gapped adjacencies to find the relationship between CARs and combine them into longer ones. Experiments shows that the number of CARs GapAdj produced are highly correlated with the actual amount of chromosomes. PMAG using greedy heuristic also suffers from this issue and the amount of CARs is too large. Therefore we propose to modify the formation of WMM graph to allow more accurate inference of CARs. We expect our inference of CARs be even closer to the true amount than GapAdj can achieve. Our proposed approach relies on mapping different telomeres in the leaf genomes into unique singletons. As gene is represented by two vertices in the graph—its head and tail, singletons stands for telomeres which are distinguished by the gene it connects to. Singleton is useful because it keeps track of the two extremities of a chromosome. For example, label "0" is used to represent telomeres of the genomes in the figure 4.5 and we identified five distinct telomere adjacencies which are $(0, 1)$,$(1, 0)$,$(0, 2)$,$(4, 0)$ and $(-3, 0)$. By mapping telomeres into distinct singleton labels, we can insert these new vertices and construct a new graph as shown in the figure 4.5. We call the collection of vertices representing actual heads or tails of genes as gene community, while singleton community consists of all distinct telomeres. With in a singleton community, each pair of vertices is connected with

minimum weight 0. Singleton community and gene community are connected only through the five telomere adjacencies. It is obvious that any solution of WMM must traverse the telomere adjacencies n times, where n must be even. Then the number of chromosomes is $\frac{n}{2}$. Unused singletons will join with other singletons and contiguous singletons will finally be collapsed in the solution. It is also noticeable that singleton with low probabilities should be excluded in the first place, since a large potion of telomere adjacencies are indeed false positives and hence introduce too many such edges complicate our problem. Finally we screen the solution of WMM and cut the path at two or more contiguous singletons to form a chromosome.

4.5 Experimental Results

Test VLBE under unequal content

As we have mentioned, *FARM* and PMAG++ both aim to formulate the conditional probabilities of gene adjacencies, however due to applying TSP solver to handle assembling, it is much more computationally demanding than *FARM*. In this section, we compare the performance of *FARM* to PMAG++ on data set without whole genome duplication and with whole genome duplication, together with other evolutionary events.

To compare on data set without whole genome duplication, we set the evolutionary setting as described is Figure 4.6. In our experiments we see that *FARM* always outperforms PMAG++ for every data setting on both content accuracy and adjacency accuracy as shown in Figure 4.6 and 3.8. It confirms that *VLBE* does performs better than multiple state encoding in the phase of content estimation of PMAG++. *FARM* can achieve a minimum average accuracy of above 70% in our testing cases. The improvement on adjacency accuracy is much more significant than PMAG++, when the tree diameter r, is getting larger. As for the performance on contig assembly,

both of them have comparable performance, as we can see from Figure 3.9. *FARM* can approximately reflect the actual number of chromosomes in the true genomes as PMAG++ does.

To compare on data set with whole genome duplication, we set the evolutionary setting as described in Figure 4.7. *FARM* continues to have a stable performance on ancestral genomes assembling, when compared with the performance on the data set without WGD. As shown in Figure 4.7, *FARM* shows slight improvement on content estimation; and from Figure 3.10, in the most difficult case ($50k \times 10$ and $r = 4$), *FARM* presents an improvement of more than 10 percent in adjacency accuracy. Although the performance on contig assembling is slightly lower, when compared with PMAG++, it is still competitive to each counterpart as shown in Figure 3.11.

All tests are conducted on a workstation of 2.4Ghz, 8 core CPU and 4 GB RAM. In Figure 3.13 and Figure 3.12, we summarize the running time of each method in each test case. Figure 3.12 and 3.13 also indicate another significant achievement in this work is that *FARM* generally runs 3-5 times faster than PMAG++. PMAG++ is more computationally demanding than *FARM*, for which PMAG++ is limited to copy with small tree diameter data sets, while larger tree diameter shows little impact on the running time of *FARM*.

Comparison on 12 Drosophila species

It is very difficult to evaluate the accuracy of our methods using real biological data as we do not know true ancestral gene orders. Nonetheless, we test *FARM* and PAMG++ on 12 fully sequenced drosophila species. Since the ground truth for ancestral gene orders of these 12 drosophila species is unknown, we evaluate the result in this way: First, we calculate the DCJ distance by UniMoG [4] on each branch and compare it between *FARM* and PMAG++. Second, we average out the sum of all these real lengths returned from each method. The tree topology of these 12

Drosophila genomes is given in Figure 4.8 and branch lengths are presented in Table 4.6. The overall sum of branch lengths from *FARM* is 6001 and that of PMAG++ is 6099, which means *FARM* can obtain a better phylogenetic score than that of PMAG++. On average, reconstructed ancestral genomes from *FARM* reduces 4.45 DCJ events per branch. By these, it confirms that by using the weighted maximum matching and variable length binary encoding, *FARM* reconstructs internal genomes with fewer events to explain the evolutionary history. Figure 4.8 shows the details of how *FARM* outperforms PMAG++. For example, branch $(A5, A7)$ and $(A7, A8)$ are two branches that bring significant difference between these two results. In addition, PMAG++ requires more than 40 minutes to reconstruct ancestors, while *FARM* finishes within a minute.

4.6 Conclusion

In summary, we introduced our ground works in chapter 2 and chapter 3 on gene order phylogeny and ancestral genome inference. Our proposed work focuses on the extension of FARM to address the following problems:

- Extend FARM to handle gene insertion, deletion and duplication.

  Deduce gene content of ancestral genomes.

  Assemble gene adjacencies into genomes when genes have duplications.

- Reduce our problem to an instance of WMM to replace the greedy heuristic.

  Convert probabilities into edges weights.

  Retrieve missing adjacencies from rearrangement-based methods.

  Produce appropriate number of CARs.

We identified most difficulties and proposed our solution accordingly for each item. In the evaluation phase, we will conduct extensive experiment and validate the per-

Table 4.3 DCJ distance of each branch on the tree of 12 Drosophila genomes. A1 to A11 are ancestral genomes.

branch	$(A1, Dsec)$	$(A1, Dsim)$	$(A1, A2)$	$(A2, Dmel)$	$(A2, A3)$	$(A3, Dyak)$
FARM	33	112	36	53	55	98
PMAG++	33	112	28	65	67	90
branch	$(A3, A4)$	$(A4, Dere)$	$(A4, A5)$	$(A5, Dana)$	$(A5, A7)$	NA
FARM	29	255	298	453	216	NA
PMAG++	35	249	294	464	249	NA
branch	$(A6, Dpse)$	$(A6, Dper)$	$(A6, A7)$	$(A7, A8)$	$(A8, Dwil)$	$(A8, A9)$
FARM	299	94	489	238	1302	977
PMAG++	295	98	474	282	1308	986
branch	$(A9, Dgri)$	$(A9, A10)$	$(A10, Dviri)$	$(A10, A11)$	$(A11, Dmoj)$	NA
FARM	76	50	379	19	440	NA
PMAG++	78	64	368	46	413	NA

formance of our new implementations to the best of current competitors. Finally, as all current adjacency-based methods evaluated their results by counting the number of correct adjacencies, we propose to use DCJ distance between the inferred genome and its according true ancestor as a direct measurement, after all our goal is to infer ancestral genome, not ancestral adjacencies and gene adjacencies also can not reflect structural variance between genomes.

In this study, we extended a Flexible Ancestral Reconstruction Method embedded with variable length binary encoding. The achievement is, we use a variable binary encoding scheme to estimate gene content, with which we improve the estimation of ancestral gene content. We set up comparison experiments with PMAG++ with various genomic settings and evolutionary rates, unequal content model (Since PMAG++ is the only method can handle unequal content). We also compare the performance of *FARM* with PMAG++ using genomes of 12 fully sequenced drosophila species. According to the results, we can see that *FARM* can not only outperform other methods under both equal content and unequal content model, in term of accuracy, but also achieves a significant reduction in running time.

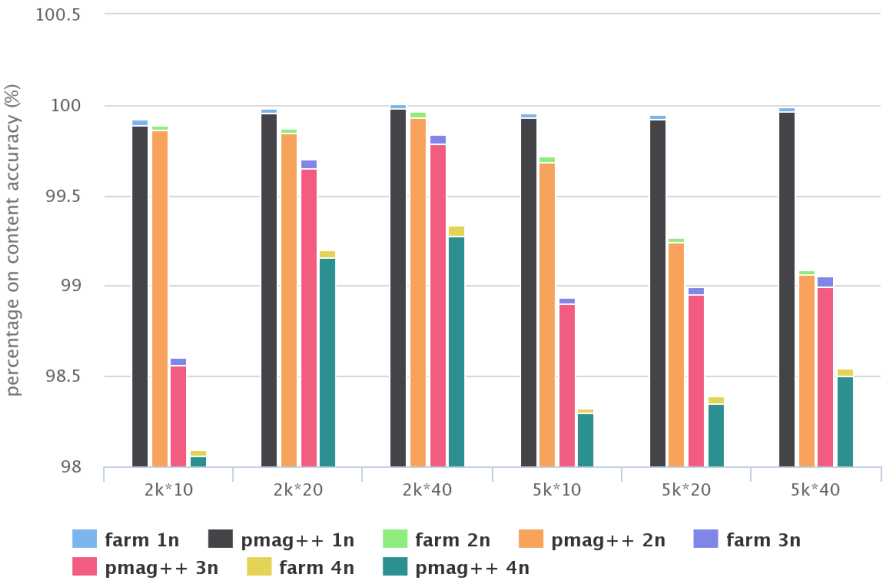


Figure 4.6 Accuracy of content on data with 60% inversions, 5% fissions, 5% fusions, 10% translocations, 5% insertions, 5% deletions, 10% duplications. $n \times N$ means the datasets have n genes and N genomes.

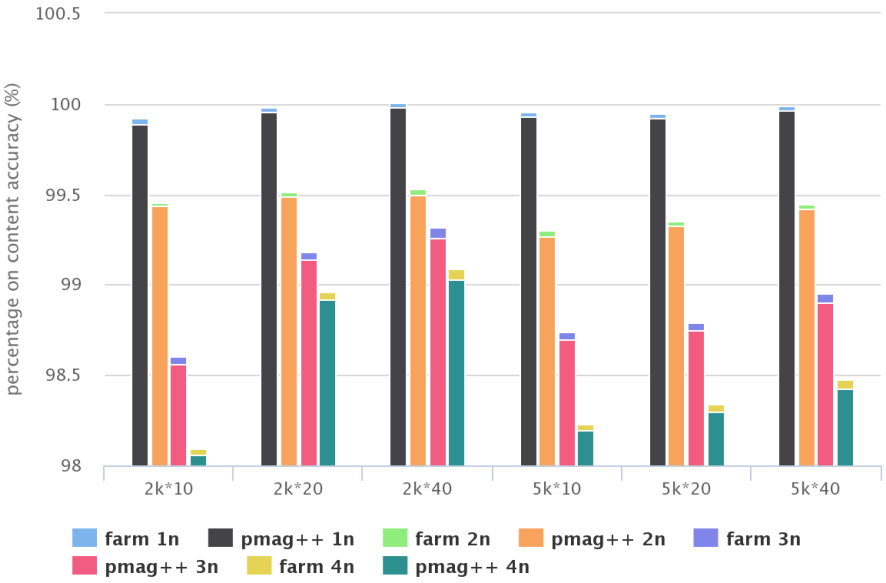


Figure 4.7 Accuracy of content on data with 60% inversions, 5% fissions, 5% fusions, 10% translocations, 5% insertions, 5% deletions, 10% duplications, and one whole genome duplication on the root node. $n \times N$ means the datasets have n genes and N genomes.

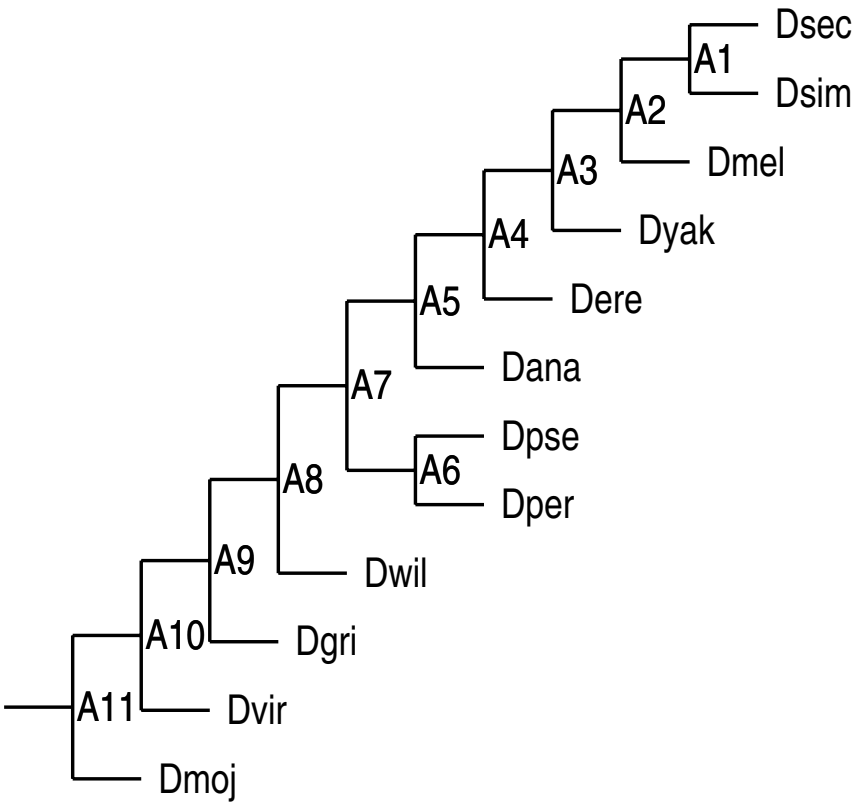


Figure 4.8 The tree topology of 12 drosophila genomes.

Chapter 5

Phylogeny Reconstruction from Whole Genome Data Using Variable Length Binary Encoding

In this chapter, we designed a flexible frame work for Phylogeny Reconstruction, based on maximum likelihood. First, we are going to explore, under maximum likelihood scheme, how encoding scheme from whole genome data to sequence can assist on phylogeny reconstruction and therefore design a method to reconstruct phylogeny with high accuracy, roubusticity and scalability. Finally, we give the evaluation design at the end of each part.

5.1 Motivation

Phylogenetic analysis is one of the main tools of evolutionary biology. Most of it to date has been carried out using sequence data (or, more rarely, morphological data)[55, 60, 54, 32]. Nowadays, sequence data can be collected in large amounts at very low cost and, at least in the case of coding genes, is relatively well understood, but it needs accurate determination of orthologies and gives us only local information – and different parts of the genome may evolve at different rates or according to different models. Events that affect the structure of an entire genome may hold the key to building a coherent picture of the past history of contemporary organisms. Such events occur at a much larger scale than sequence mutations – entire blocks of a genome may be permuted (rearrangements), duplicated, or lost. As whole genomes are sequenced at increasing rates, using whole-genome data for phylogenetic analy-

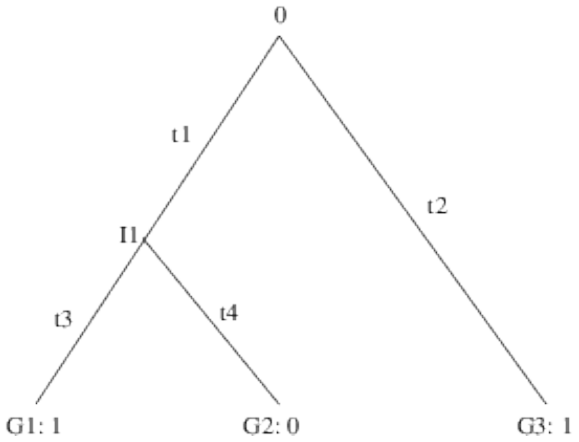


Figure 5.1 A phylogenetic topology of three genomes. The 0 or 1 following the leaf label represent absence or presence of a gene adjacency.

ses is attracting increasing interest, especially as researchers uncover links between large-scale genomic events (rearrangements, duplications leading to increased copy numbers) and various diseases (such as cancer) or health conditions (such as autism). However, using whole-genome data in phylogenetic reconstruction has been proved far more challenging than using sequence data and numerous problems plague existing methods: oversimplified models, poor accuracy, poor scaling, lack of robustness, lack of statistical assessment, etc.

Determining the phylogeny between a group of organisms plays an essential role in our understanding of evolution. A wide selection of methods have been developed for a specific biological data type, which are commonly aligned sequences of nucleotides or amino acids. As nowadays more and more genomes are completely sequenced, gene order of whole-genomes as a relatively new type of data attracts a lot of attention in recent years. As we mentioned, MPBE and MPME are the first two methods that reconcile the sequence data and gene-order data such that gene orders can be encoded into aligned sequences without loss of information. Therefore we can use parsimony softwares such as TNT [26] and PAUP* [59] developed for molecular sequences to conduct gene order phylogeny searching. Although MPBE and MPME failed to compete with direct-optimization approaches such as GRAPPA,

they show great speedup and pave the way for future improvements. From another aspect, beyond parsimonious framework, sequence data can be analyzed by searching the phylogeny with maximized likelihood score as suggested by Felsenstein [24] in 1981. Such probabilistic approach is attractive since it is accurate and statistically well-founded; even with very short sequence, it tends to outperform other methods. Recent algorithm developments and the introduction of high-performance computation tools such as RAxML [55] have made the maximum likelihood approach feasible for large scale analysis of molecular sequences.

Current approaches in the area of phylogenetic analysis are limited to very small collections of closely related genomes using low-resolution data (typically a few hundred syntenic blocks); moreover, these approaches typically do not include duplications and loss events. It was not until 2011, however, that the first successful attempt to use ML reconstruction based on whole genome data was published [30]; results from this study on bacterial genomes were promising, but somewhat diffcult to explain, while the method appeared too time-consuming to handle eukaryotic genomes. later 2012, Yu [36] describes a maximum likelihood (ML) approach for phylogenetic analysis that takes into account genome rearrangements as well as duplications, insertions, and losses. This approach can handle high-resolution genomes (with 40,000 or more markers) and can be used in the same analysis for genomes with very different numbers of markers. However, since the embeded encoding scheme in it igores the copy information of both adjacency and content, its performance fades out when genomes experienced a large number of duplications or whole genome duplications.

As we've discovered in last chapter. Variable Length Binary Encoding works with a better performance on ancestral content estimation than Binary Encoding or Multiple-State Encoding in ancestral genome reconstruction. This improvement indicates that VLBE reserves more information than the simple Binary Encoding or MLME method. Maximum-likelihood (ML) approaches seek the tree and related

model parameters that maximize the probability of producing the given set of leaf genomes. Theoretically, such approaches are much more computationally expensive than both distance-based and parsimonybased approaches, but their accuracy has long been a major attraction in sequence-based phylogenetic analysis. Moreover, in the last few years, packages such as RAxML [56] have largely overcome computational limitations and allowed reconstructions of large trees (with thousands of taxa) and the use of long sequences (to a hundred thousand characters). These improvements motivate us to utilize the technique and apply it for gene order phylogeny analysis through encoding gene orders. Because of using RAxML package, our approach is able to scale up to large trees reconstruction.

In the rest of this section, we will first describe three variations of Variable Length Binary Encoding, transition model design, phylogeny reconstruction with VLWDx and experiment design and analysis on VLWDx. Finally we will show our experimental design along with evaluations of various methods.

5.2 Variable Length Binary Encoding

In this section, we first describe several versions of Variable Length Binary Encoding schemes (VLBE) and then introduce Variable Length Binary Encoding based Phylogeny Reconstruction with Maximum Likelihood on Whole-Genome Data with VLBE (VLWDx). All of the methods are founded on the binary encoding of gene orderings. By encoding, we want to produce a sequence like string while reserving as completely as possible about the gene order information, and by incorporateing a dedicated transition model deduced from adjacencies changes, VLWDx aims at achieving more robust and scalable phylogenetic reconstruction performance, and keeping running-time at a reasonable low level.

Before getting into the encoding detail, let's first take a look at the way to interpret genomes. Given a gene g, we denote the tail of it by g^t and its head by g^h. We write

$+g$ to indicate an orientation from tail to head (g^t, g^h), $-g$ otherwise (g^h, g^t). Two consecutive genes a and b can be connected by an adjacency with one of the following four types: (a^t, b^h), (a^h, b^h), (a^t, b^t), and (a^h, b^t). If gene c lies at one end of a linear chromosome, then we have a corresponding singleton set for it, c^t or c^h, called a telomere; otherwise, they are all adjacencies, if it's a circular genome. A genome can then be represented as a multiset of adjacencies and telomeres (if there's any). For example, a simple genome composed of one linear chromosome $(+a, +b, -c, +a, +b, -d, +a)$, and one circular one, $(+e, -f)$, can be represented by the multiset of adjacencies and telomeres $S = \{(a^t), (a^h, b^t), (b^h, c^h), (c^t, a^t), (a^h, b^t), (b^h, d^h), (d^t, a^t), (a^h), (e^h, f^h), (e^t, f^t)\}$. In the presence of duplicated genes, there is no one-to-one correspondence between genomes and multiset of genes, adjacencies, and telomeres. For example, the genome composed of the linear chromosome $(+a, +b, -d, +a, +b, -c, +a)$ and the circular one $(+e, -f)$, would have the same multiset of adjacencies and telomeres as our toy example. For data limited to rearrangements (i.e. for genomes with identical gene content), we encode only the adjacency information. For a possible adjacency or telomere, we apply $VLBE_1$ to encode its presence or absence detail in a genome. We consider only those adjacencies and telomeres that exist in at least one of the input genomes. If the total number of distinct genes among the input genomes is n, then the total number of distinct adjacencies and telomeres is $2n^2 + 2$, but the number of adjacencies and telomeres that appear in at least one input genome is typically far smaller – in fact, it is usually linear in n rather than quadratic. For the general model, which includes gene duplications, insertions, and losses in addition to rearrangements, we extend the encoding of adjacencies by also encoding the gene content. For each gene, we apply $VLBE_2$ or $VLBE_3$ to indicate the presence or absence state of this gene in a genome. In next tree subsections, we give three algorithms, and with genomes given in Table 5.2(1), we give the encoding results for each algorithm. Figure 5.2 gives a graphic representation for genomes in Table 5.2(1).

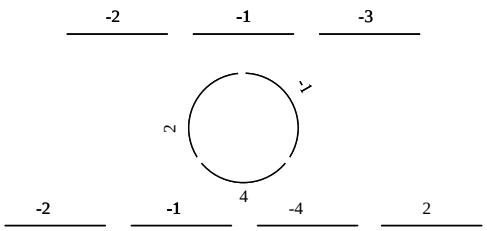


Figure 5.2 An example of a set of three genomes.

Variable Length Binary Encoding 1 ($VLBE_1$)

Let G be a signed permutation of n genes. For linear genomes, gene 0 is added to indicate the start and end of a genome generally. For a pair $(i, j), 0 \leq i, j \leq n$, we apply $VLBE_1$ to encode each adjacency. A detailed algorithm is provided as following.

- Given a data set D of n genomes, screen over it, collect all unique adjacencies, and get a list A of m adjacencies; for each adjacency $a \in A$, count the maximum state number of a, denote as $MaxAdj(a)$.

- For each genome $D_i \in D$, $1 \leq i \leq n$, iterate each adjacency $a \in A$, get the copy number of $a \in D_i$, denote by $AdjNum(i, a)$.

- For each adjacency $a = A[j]$ if present in genome D_i, we encode it into a binary sequence in this way:

  Place $AdjNum(i, a)$ 1's to indicate its copy number; append $MaxAdj(a)$ - $AdjNum(i, a)$ 0's to its left; otherwise place $MaxAdj(a)$ 0s to indicate its absence; get sequence $seq(i, a)$, or $seq(i, A[j])$.

  Append $seq(i, A[j])$ to $seq(i, A[j-1])$, for each adjacency $a = A[j]$, $1 \leq j \leq m$; get a binary sequence for genome D_i denoted by S_i.

Table 5.1 Example of the binary encoding throught $VLBE_1$ (0 indicates the start of a genome, 6 indicates the end of a genome).

$$G_1 : (-2, -1, -3) \quad (5.1)$$
$$G_2 : (-1, 4, 2) \quad (5.2)$$
$$G_3 : (-2, -1, -4, 2) \quad (5.3)$$

(a) Three signed linear genomes

	Adjacencies								
A	0,-2	-2,-1	-1,-3	-3,0	0,-1	-1,4	4,2	-1,-4	-4,2
S_1	01	1	1	1	0	0	0	0	0
S_2	00	0	0	0	1	1	1	0	0
S_3	11	1	0	0	0	1	0	1	1

(b) Binary Encoding

- Encode each genome D_i into S_i, We get a set of binary sequences, denoted as S.

Table 5.2 gives an example of such encoding. Most gene pairs are not shown in this table because they do not appear in any of these genomes. Although there are up to $\binom{2n+2}{2}$ possible adjacencies, we can further reduce the length of these sequences by removing those characters at which every genome has the same state.

After converting the gene orders into strings of 0 and 1, we tested several ML packages such as TREE-PUZZLE [52], GARLI [28] and etc. Among them, RAxML [55] is the best by incorporating the rapid bootstrapping [56].

Variable Length Binary Encoding 2 ($VLBE_2$)

$VLBE_1$ is designed to encode a gene order into a binary sequence reserving as much information as possible, concerning the gene order detail. However, we might also want to keep the content information as well. So we propose $VLBE_2$ to encode a gene order into binary sequence. The algorithm is given as follows.

- Given a data set D of n genomes, screen over it, collect all unique adjacencies, and record them into a list A of m adjacencies;

  For each adjacency $a \in A$, count the maximum state number of a, denote as $MaxAdj(a)$;

  Collect all unique content, get a list C, with maximum gene denoted as $MaxGene$.

- For each genome $D_i \in D$, $1 \leq i \leq n$, iterate every adjacency $a \in A$, get the copy number of $a \in D_i$, denote by $AdjNum(i, a)$.

- For each adjacency $a = A[j]$, we encode it into a binary sequence in this way:

  Place AdjNum(i, a) 1's to indicate its copy number; append $MaxAdj(a)$ - $AdjNum(i, a)$ 0's to its left, if present in genome D_i; otherwise place $MaxAdj(a)$ 0s to indicate its absence; result a sequence $seqAdj(i, a)$, or $seqAdj(i, A[j])$;

  Append $seqAdj(i, A[j])$ to $seqAdj(i, A[j-1])$, for each adjacency $a = A[j]$, $1 \leq j \leq m$; get a binary sequence for all adjacencies from A_i, get $SAdj(i)$;

  For each content $g = C[t] \in C$, append 1 to $seqCont(i, C_{t-1})$, if g presents in D_i. denote $SCont(i)$.

- To encode each genome D_i into S_i, We combine $SAdj(i)$ and $SCont(i)$ togather and get a set of binary sequences, denoted as S.

Table 5.2 Example of the binary sequences using $VLBE_2$ (0 indicates the start of a genome, 6 indicates the end of a genome).

	Adjacencies									Content			
	0,-2	-2,-1	-1,-3	-3,0	0,-1	-1,4	4,2	-1,-4	-4,2	1	2	3	4
G_1	01	1	1	1	0	0	0	0	0	1	1	1	0
G_2	00	0	0	0	1	1	1	0	0	1	1	0	1
G_3	11	1	0	0	0	1	0	1	1	1	1	0	1

Table 5.2 shows an example of binary sequences produced by $VLBE_2$ from whole genome data presented in Table 5.2(a). After converting the gene orders into strings of 0 and 1, We'll use RAxML package to reconstruct the phylogeny.

Variable Length Binary Encoding 3 ($VLBE_3$)

$VLBE_2$ is designed to encode a gene order into a binary sequence with as much information as possible, concerning the gene order detail. We further want to know how variable length binary encoding on content could make difference on phylogeny reconstruction. So we propose $VLBE_3$ to encode a gene order into binary sequence. The detail is given as follows:

- Given a data set D of n genomes, screen over it, collect all unique adjacencies, and record them into a list A of m adjacencies;

  For each adjacency $a \in A$, count the maximum state number of a, denote as $MaxAdj(a)$;

  Collect all unique content, get a list C, with maximum gene denoted as $MaxGene$; for each gene $g \in C$, count the maximum copy number of g, denote as $MaxCont(g)$.

- For each genome $D_i \in D$, $1 \leq i \leq n$, iterate every adjacency $a \in A$, get the copy number of $a \in D_i$, denote by $AdjNum(i, a)$; iterate every adjacency $g \in C$, get the copy number of $g \in D_i$, denote by $ContNum(i, a)$.

- For each adjacency $a = A[j]$, we encode it into a binary sequence in this way:

  Place $AdjNum(i, a)$ 1's to indicate its copy number and append $MaxAdj(a)$ - $AdjNum(i, a)$ 0's to its left, if a presents in genome D_i; otherwise place $MaxAdj(a)$ 0s to indicate its absence; get a sequence $seqAdj(i, a)$, or $seqAdj(i, A[j])$.

Append $seqAdj(i, A[j])$ to $seqAdj(i, A[j-1])$ for each $a = A[j]$, $1 \leq j \leq m$; get a binary sequence for all adjacencies from D_i, get $SAdj(i)$;

- For each content $g = C[t]$, we encode it into a binary sequence in this way:

  Place $ContNum(i, g)$ 1's to indicate its copy number; append $MaxCont(g)$ - $ContNum(i, g)$ 0's to its left; otherwise place $MaxCont(g)$ 0s to indicate its absence, if g presents in genome D_i; get a sequence $seqCont(i, g)$, or $seqAdj(i, C[t])$.

  Append seqAdj(i, $C[t]$) to seqAdj(i, $C[t-1]$) for each $g = C[t]$, $1 \leq t \leq MaxGene$; get a binary sequence for all content from D_i, get $SCont(i)$;

- To encode each genome D_i into S_i, We combine SAdj(i) and $SCont(i)$ togather and get a set of binary sequences, denote as S.

Table 5.3 Example of binary sequences using $VLBE_3$ (0 indicates the start of a genome, 6 indicates the end of a genome).

	Adjacencies									Content			
	0,-2	-2,-1	-1,-3	-3,0	0,-1	-1,4	4,2	-1,-4	-4,2	1	2	3	4
G_1	01	1	1	1	0	0	0	0	0	1	01	1	0
G_2	00	0	0	0	1	1	1	0	0	1	01	0	1
G_3	11	1	0	0	0	1	0	1	1	1	11	0	1

Table 5.2 shows the example of the binary strings of the genomes presented in Table 5.2(a). Again, RAxML will be used to obtain trees from these binary sequences. However, the transition model is still in need to design, which will be covered in next subsection.

5.3 Building Transition Model

As mentioned above, $VLBE_1$, $VLBE_2$ and $VLBE_3$ aim at transforming gene order information to sequence-like string without losing important genomic information, after encoding. Since fliping a state, 1 to 0 or 0 to 1, is dependent on the transition

model within the encoding scheme, we have to design a transition model for each of the encoding scheme. As in MLWD[36], Lin gives a transion model explanation for the encoding scheme.

It is more desirable to develop a designated model from the characteristics of gene rearrangements and the composition feature of genes for a method for gene-order data under maximum likelihood method. Since our encodings are binary sequences, the parameters of the model in all of them are simply the transition probability from presence (1) to absence (0) and that from absence (0) to presence (1). So we set off from the composition of the encoding and analyze how 0 is flipped to 1 or vice versa.

Let us first take a look at adjacencies. Every DCJ operation will select two adjacencies (or telomeres) uniformly at random, and (if adjacencies) break them to create two new adjacencies. Each genome has $n + O(1)$ adjacencies and telomeres (O(1) is the number of linear chromosomes in the genome, viewed as a constant). Thus the transition probability from 1 to 0 at some fixed index in the sequence is $\frac{2}{2n+O(1)}$ under one DCJ operation. Since there are up to $\binom{2n+2}{2}$ possible adjacencies and telomeres, the transition probability from 0 to 1 is $\frac{2}{n^2+O(n)}$. Thus the transition from 0 to 1 is roughly $2n$ times less likely than that from 1 to 0. Despite the restrictive assumption that all DCJ operations are equally likely, this result is in line with general opinion about the probability of eventually breaking an ancestral adjacency (high) vs. that of creating a particular adjacency along several lineages (low)-a version of homoplasy for adjacencies.

For content encoding, as for $VLBE_2$ and $VLBE_3$, we also have transitions for gene content. Once again, the probability of losing a copy of gene independently along several lineages is high, whereas the probability of gaining the same gene independently along several lineages (the standard homoplasy) is low. However, there is no simple uniformity assumption that would enable us to derive a formula for the respective probabilities-there have been attempts to reconstruct phylogenies based

on gene content only[54, 32, 73], but they were based on a different approach-so we experimented with various values of the ratio between the probability of a transition from 1 to 0 and that of a transition from 0 to 1. Each site in our binary sequence isn't simply representing the present or absent of a single adjacency or a single certain gene. Actually, it only represents a copy of gene or adjacency. we want to bring the transion model to either a more general way or several detailed ways to accommodate various kinds of Whole-Genome Gene order data, taking the adjacency sequence length and content sequence length into consideration for mixed encoding scheme ($VLBE_2$ and $VLBE_3$).

5.4 Estimating The Phylogeny

Once we have encoded input genomes into binary sequences and have computed the transition parameters, we use the ML reconstruction program RAxML (version 7.2.8 was used to produce the results given in this experiment) to build a tree from these sequences. Because RAxML uses a time-reversible model, it estimates the transition parameters directly from the input sequences by computing the base frequencies. In order to set up the 2n ratio, we simply add a direct assignment of the two base frequencies in the code. Athough this VLBE will generate a sequence no shorter than that from other encoding mehtods metioned above(up to 2-3 times longer in our expriments), it bring no disastrous load to the computation limitation of RAxML, due to it's excellent improvement on parallel coding.

5.5 Experimental Results

Experiments Design

We ran a series of experiments on simulated data sets in order to evaluate the performance of our approach against a known "ground truth" under a wide variety of

settings. We then ran our reconstruction algorithm on a data set of *18* genomes, of yeasts, a data set of 6 genomes of plants and a data set of 11 genomes of mammalians, obtained from *the Eukaryotic Gene Order Browser (eGOB) database.23*

Our simulation studies follow standard practice in phylogenetic reconstruction.*24 citation* We generate model trees under various parameter settings, then use each model tree to evolve an artificial root genome from the root down to the leaves, by performing randomly chosen evolutionary events on the current genome, finally obtaining data sets of leaf genomes for which we know the complete evolutionary history. We then reconstruct trees for each data set by applying different reconstruction methods and compare the results against the model tree.

The simulation process is carried out as follows. First, we produce a birth-death tree T, which obeys the same way as [35]. Then we find the longest path between two leaf nodes, with length = K. We apply different evolutionary rates $r \in \{1, 2, 3, 4\}$ so that the tree diameters are in the range of $d \in \{1n, 2n, 3n, 4n\}$: larger diameter means a genome is more distant from its ancestor, and hence more computationally expensive this data set will be. By timing $1/K$ to tree diameter, we then get the length for a certain branch and we apply a variation coefficient to each branch in this way to vary the length of each branch: given a parameter c, for each branch we sample a number s uniformly from the interval $(-c, c)$ and multiply the branch length by e^s. For the experiments in this chapter, we set c with the value of 1. Thus, a branch would get its length L get by,

$$L = r \times n \times (1/K) \times e^s$$

For evolving on each branch, we use a set of evolutionary events, including inversions, fusions, fissions, translocations, indels, segment duplications and whole genome duplications. We assign each event with a specific value of probability to be selected during the simulation process.

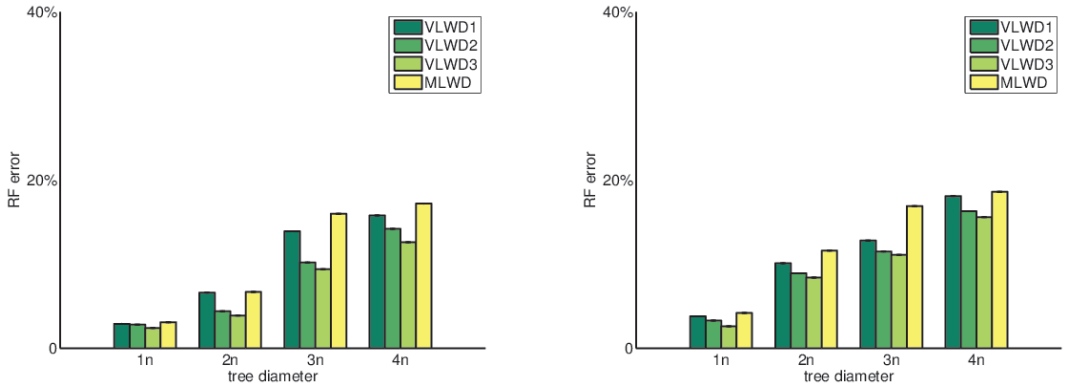
We compared the accuracy of three different approaches, $VLWD_1$, $VLWD_2$, $VLWD_3$ and MLWD. $VLWD_x$ (Variable Length Encoding Whole Genome Data, of which the subscripts represent different encoding schemes covered above) is our new approach; MLWD (Maximum Likelihood on Whole-genome Data) is a maximum likelihood based tool to reconstruct phylogeny on whole genome data, which applies the custom transition probabilities estimation and maximum likelihood estimation tool RAxML. We did not compare with the approaches of Lin, or those of Hu et al. 19 or those of Cosner et al.,27 because MLWD outperforms the first one [*citation*], and both second and third are too slow and also because the second is also limited by their character encodings to a maximum of 20 taxa.

Simulation under General Model without Duplications

We simulate two settings of data to test our proposed method, and run both our methods and MLWD. In this test, our method uses for encoding and the transition parameter uses the $2n$ ratio. our method outperforms MLWD in every data setting and the improvement is even more significant when the tree diameter gets larger for $VLWD_x$. This result is in line with the assumption (variable length binary encoding can reserve more genome information) we made earlier and encourages us to dig further in phylogenetic reconstruction through binary encoding. Figures 5.3 (a) and 5.3 (b) show error rates for different approaches; the x axis indicates the error rates and the y axis indicates the tree diameter. Error rates are RF error rates[28] the standard measure of error for phylogenetic trees. the RF rate expresses the percentage of edges in error, either because they are missing or because they are wrong.

These representative simulations show that our VLWD approach can reconstruct much more accurate phylogenies from genome data experienced various evolutionary events, than the previous binary encoding-based approach MLWD, in line with experience in sequence-based reconstruction. $VLWD_3$ also outperforms $VLWD_1$

and $VLWD_2$, underlining the importance of fullest encoding the genome order information into sequence and the importance of estimating and setting the transition parameters before applying the sequence-based ML method.



(a) 1, 000 genes *(b) 1, 000 genes*

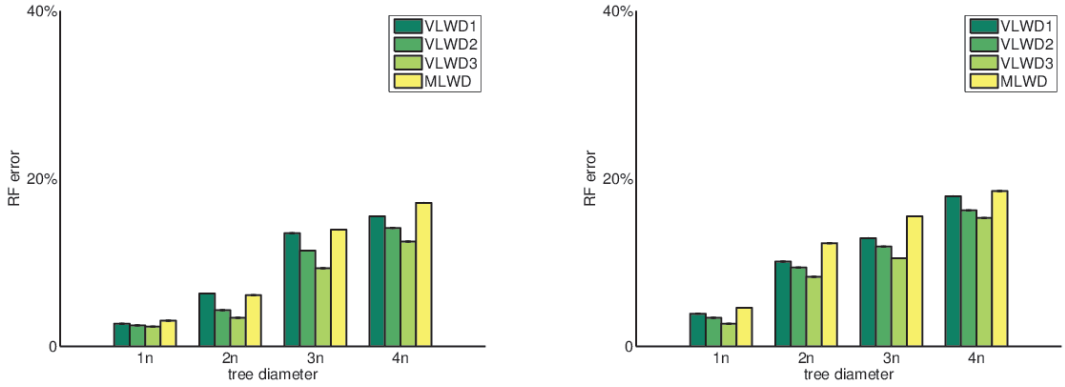
Figure 5.3 RF error rates for different appraoches for trees with 60 species, with genomes of $1,000$ genes and tree diameters from 1 to 4 time the number of genes, under the evolutionary events without duplications.

Simulation under General Model with Duplications

Here we generated more complex data sets than for the previous set of experiments. For example, among our simulated eukaryotic genomes, the largest genome has more than 4,000 genes, and the biggest gene family in a single genome has 20 members. We simulate two settings of data to test our proposed method, and run both our methods and MLWD. Through this test, different encoding methods will contribute to different performance of phylogeny reconstruction.

In our approach, the encoded sequence of each genome combines both the adjacency and gene content information, which makes it difficult to compute optimal transition probabilities, as discussed in Section 3. Thus we set a empirical value [35] under simulation results. If the transition probability of any gene or adjacency from 0 to 1 in $VLWD_x$ is set to be m times less than that in the opposite direction, we set all $VLWD_x$ (m = 1000).

Figure 5.5 (a) and 5.5 (b) summarizes the RF error rates. Whereas all $VLWD$ methods again outperform MLWD, and $VLWD_3$ can always maintain the best performance. Generally, $VLWD_x$ can reconstruct more accurate phylogeny than MLWD. Among $VLWDs$, $VLWD_3$ achieve the best result. Comparing between Figure 5.5 and 5.5, we can find that MLWD returns similar result for data set without and with whole genome duplication. Both the differences can be attributed to the encoding scheme of $VLWD_3$, which reserves the fullest genome information than others – since we encode the number of copies of the gene, many duplication and loss events will alter the encoded gene content. Whereas MLWD could only encode the presence or absence for both adjacency and content.

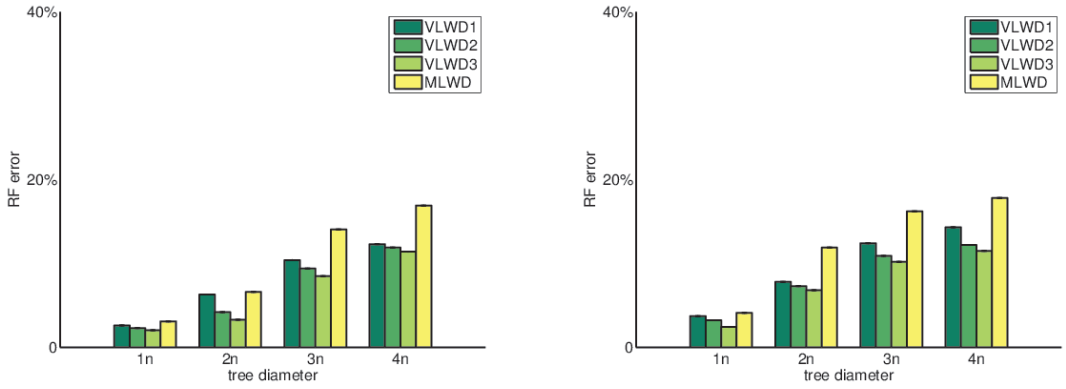


(a) 1, 000 genes *(b) 1, 000 genes*

Figure 5.4 RF error rates for different appraoches for trees with 60 species, with genomes of $1,000$ genes and tree diameters from 1 to 4 time the number of genes, under the evolutionary events with $free$ (segment) duplications.

VLBE phylogeny for real mammal genomes

In the previous results of this approach, we tested our VLBE approach on simulated data set and achieved very good performance for reconstructing the phylogeny history for the simulated genome data. Moreover, the VLBE approach can also be applied to reconstruct the phylogeny for real genome data. In this section, we obtain the whole genome data of eleven mammal species from online database Ensemble [16]. We first



(a) 1, 000 genes *(b) 1, 000 genes*

Figure 5.5 RF error rates for different appraoches for trees with 60 species, with genomes of $1,000$ genes and tree diameters from 1 to 4 time the number of genes, under the evolutionary events with both segment and whole genome duplications.

encode all of the genes into gene orders by using the same gene order to represent all of the homologous genes across different mammal genomes. If some gene has more than one copies in the same genome, we still use same gene order to represent all of the copies of this gene. Subsequently, we input the gene order content and adjacencies into the VLBE approach to reconstruct the phylogenetic relationship for these eleven mammal species 5.5. It only takes less than ten minutes for the VLBE to output the final solution. We compare the VLBE phylogeny with the NCBI taxonomy, As Figure 5.5 showing, our VLBE approach correctly assign the Macaca mulatta and Macaca fascicularis into the Macaca genus and assign the Pan troglodytes and Gorilla gorilla into the Homininae genus. The Rattus norvegicus and Mus musculus are also been correctly assigned into the subfamily Murinae. The Ovis aries and Bos taurus are also been correctly assigned to the Bovidae family. We also compare this $VLWD_3$ phylogeny with the previous gene order based mammal phylogeny study of Luo *et al.* [38]. There are eight mammal species shared by these two phylogenies, and all of the shared branches for these eight species agree with each other. Moreover, two lowest bootstrap scores (68, 71) on the middle two branches in the tree of Figure 5.5 reflect the current controversial opinions in placing primates closer to rodents or

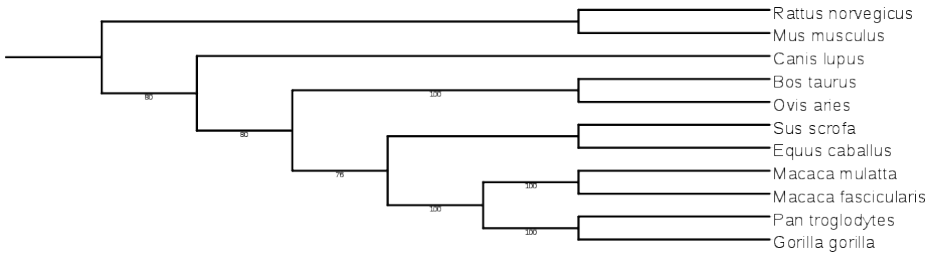


Figure 5.6 Phylogeny reconstructed by VLWD for eleven mammal genomes, with bootstrap values shown on branches.

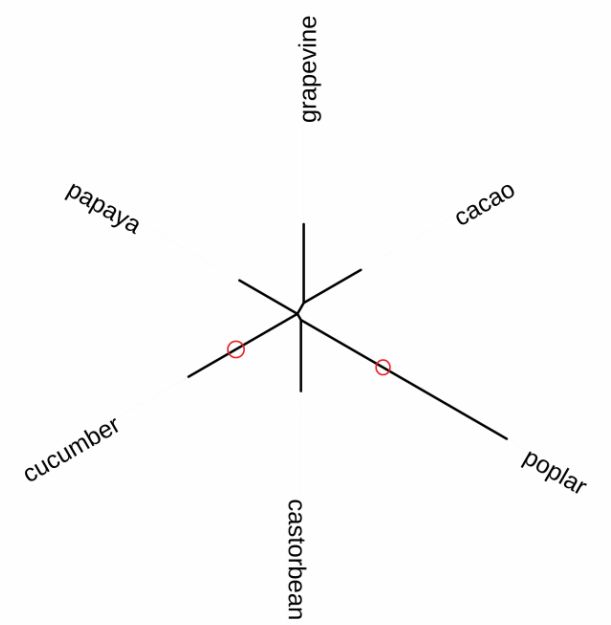


Figure 5.7 Phylogeny reconstructed by VLWD for six plant genomes, with branch lengths proportional to genomic distances.

carnivores [42, 45, 2, 33, 67, 10].

5.6 Conclusion

practice to date has continued to use pre-processed (manually) sequences of moderate length using nucleotide-, aminoacid-, or codon-level models, regardless of many attractive reasons for using whole-genome data in phylogenetic reconstruction. Mainly, it is the lack of suitable/robust tools that has prevented more extensive use of whole-

genome data and previous tools all suffered from serious problems in combined reasons of limited data types, poor accuracy and scalability. The approach we presented is trying to overcome all of these difficulties: it uses a fairly general model of genomic evolution (rearrangements plus duplications, whole genome duplication, insertions, and losses of genomic regions), is very accurate, scales as well as sequence-based approaches, is quite robust against typical assembly errors and omissions of genes, and supports standard bootstrapping methods. Our analysis of a 11-taxon collection of mammalians genomes, 6-taxon collection of plant genomes and 18-taxon collection of yeast genomes, could not have been conducted, regardless of computational resources, with any distance-based tools without accepting severe compromises in the data (e.g., equalizing gene content) or the quality of the analysis. Also we design a new encoding scheme to reserve fullest genome information in the course of phylogeny reconstruction using maximum likelihood method. Our analysis also helps make the case for phylogenetic reconstruction based on whole-genome data for either haploid or polyploid species. Indeedly, much work remains to be done. In particular, using different transition probabilities for adjacencies and for content, by running a compartmentalized analysis, should prove beneficial on large data sets.

Bibliography

[1] Max Alekseyev and Pavel Pevzner, *Breakpoint graphs and ancestral genome reconstructions*, Genome research **19** (2009), no. 5, 943–957.

[2] Heather Amrine-Madsen, Klaus-Peter Koepfli, Robert K Wayne, and Mark S Springer, *A new phylogenetic marker, apolipoprotein b, provides compelling evidence for eutherian relationships*, Molecular phylogenetics and evolution **28** (2003), no. 2, 225–240.

[3] David Bader, Bernard Moret, and Mi Yan, *A linear-time algorithm for computing inversion distance between signed permutations with an experimental study*, Journal of Computational Biology **8** (2001), no. 5, 483–491.

[4] A. Bergeron, J. Mixtacki, and J. Stoye, *Chapter 10: The inversion distance problem.*, 2005.

[5] Anne Bergeron, Julia Mixtacki, and Jens Stoye, *A unifying view of genome rearrangements*, Algorithms in Bioinformatics, Springer, 2006, pp. 163–173.

[6] Priscila Biller, Pedro Feijão, and João Meidanis, *Rearrangement-based phylogeny using the single-cut-or-join operation*, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **10** (2013), no. 1, 122–134.

[7] Mathieu Blanchette, Guillaume Bourque, David Sankoff, et al., *Breakpoint phylogenies*, Genome Informatics **1997** (1997), 25–34.

[8] Guillaume Bourque and Pavel Pevzner, *Genome-scale evolution: reconstructing gene orders in the ancestral species*, Genome Research **12** (2002), no. 1, 26–36.

[9] David Bryant, *The complexity of the breakpoint median problem*, Centre de recherches mathematiques (1998).

[10] Gina Cannarozzi, Adrian Schneider, and Gaston Gonnet, *A phylogenomic study of human, dog, and mouse*, PLoS Comput Biol **3** (2007), no. 1, e2.

[11] Alberto Caprara, *Formulations and hardness of multiple sorting by reversals*, In Proc. 3rd International Conf. on Comput. Mol. Biol., 1999, pp. 84–93.

[12] ———, *Formulations and hardness of multiple sorting by reversals*, Proceedings of the third annual international conference on Computational molecular biology, ACM, 1999, pp. 84–93.

[13] ———, *On the practical solution of the reversal median problem*, Algorithms in Bioinformatics (2001), 238–251.

[14] Mary Cosner, Robert Jansen, Bernard Moret, Linda Raubeson, Li-San Wang, Tandy Warnow, and Stacia Wyman, *An empirical comparison of phylogenetic methods on chloroplast gene order data in campanulaceae*, (2000).

[15] Mary Cosner, Robert Jansen, Bernard Moret, Linda Raubeson, Li-San Wang, Tandy Warnow, Stacia Wyman, et al., *A new fast heuristic for computing the breakpoint phylogeny and experimental phylogenetic analyses of real and synthetic data*, Proc. 8th International Conf. on Intelligent Systems for Mol. Biol. ISMB, 2000, pp. 104–115.

[16] Fiona Cunningham, M Ridwan Amode, Daniel Barrell, Kathryn Beal, Konstantinos Billis, Simon Brent, Denise Carvalho-Silva, Peter Clapham, Guy Coates, Stephen Fitzgerald, et al., *Ensembl 2015*, Nucleic acids research **43** (2015), no. D1, D662–D669.

[17] Richard Desper and Olivier Gascuel, *Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle*, Journal of computational biology **9** (2002), no. 5, 687–705.

[18] TH Dobzhansky and AH Sturtevant, *Inversions in the chromosomes of drosophila pseudoobscura*, Genetics **23** (1938), no. 1, 28.

[19] Jack Edmonds, *Paths, trees, and flowers*, Canadian Journal of mathematics **17** (1965), no. 3, 449–467.

[20] Nadia El-Mabrouk, *Genome rearrangement by reversals and insertions/deletions of contiguous segments*, Combinatorial Pattern Matching, Springer, 2000, pp. 222–234.

[21] Hu Fei, Lingxi Zhou, and Tang Jijun, *Reconstructing ancestral genomic orders using binary encoding and probabilistic models*, Bioinformatics Research and Applications (2013).

[22] Pedro Feijao and Joao Meidanis, *Scj: a variant of breakpoint distance for which sorting, genome median and genome halving problems are easy*, Algorithms in Bioinformatics (2009), 85–96.

[23] ———, *Scj: a breakpoint-like distance that simplifies several rearrangement problems*, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **8** (2011), no. 5, 1318–1329.

[24] Joseph Felsenstein, *Evolutionary trees from dna sequences: a maximum likelihood approach*, Journal of molecular evolution **17** (1981), no. 6, 368–376.

[25] Yves Gagnon, Mathieu Blanchette, and Nadia El-Mabrouk, *A flexible ancestral genome reconstruction method based on gapped adjacencies*, BMC bioinformatics **13** (2012), no. Suppl 19, S4.

[26] Pablo Goloboff, James Farris, and Kevin Nixon, *Tnt, a free program for phylogenetic analysis*, Cladistics **24** (2008), no. 5, 774–786.

[27] Jonathan L Gordon, Kevin P Byrne, and Kenneth H Wolfe, *Additions, losses, and rearrangements on the evolutionary route from a reconstructed ancestor to the modern saccharomyces cerevisiae genome*, PLoS Genetics **5** (2009), no. 5, e1000485.

[28] Robin Gutell and Robert Jansen, *Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion*, (2006).

[29] Sridhar Hannenhalli and Pavel Pevzner, *Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals*, Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, ACM, 1995, pp. 178–189.

[30] Fei Hu, Nan Gao, Meng Zhang, and Jijun Tang, *Maximum likelihood phylogenetic reconstruction using gene order encodings*, Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2011 IEEE Symposium on, IEEE, 2011, pp. 1–6.

[31] Fei Hu, Jun Zhou, Lingxi Zhou, and Jijun Tang, *Probabilistic reconstruction of ancestral gene orders with insertions and deletions*, Computational Biology and Bioinformatics, IEEE/ACM Transactions on **11** (2014), no. 4, 667–672.

[32] Daniel H Huson and Mike Steel, *Phylogenetic trees based on gene content*, Bioinformatics **20** (2004), no. 13, 2044–2049.

[33] Gavin A Huttley, Matthew J Wakefield, and Simon Easteal, *Rates of genome evolution and branching order from whole genome analysis*, Molecular biology and evolution **24** (2007), no. 8, 1722–1730.

[34] Bret Larget, Donald L Simon, and Joseph B Kadane, *Bayesian phylogenetic inference from animal mitochondrial genome arrangements*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **64** (2002), no. 4, 681–693.

[35] Yu Lin, Fei Hu, Jijun Tang, and B Moret, *Maximum likelihood phylogenetic reconstruction from high-resolution whole-genome data and a tree of 68 eukaryotes*, Pacific Symposium on Biocomputing, World Scientific, 2013, pp. 357–366.

[36] Yu Lin, Fei Hu, Jijun Tang, and Bernard ME Moret, *Maximum likelihood phylogenetic reconstruction from high-resolution whole-genome data and a tree of 68 eukaryotes*, Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 2012, pp. 285–296.

[37] Yu Lin and Bernard Moret, *Estimating true evolutionary distances under the dcj model*, Bioinformatics **24** (2008), no. 13, i114–i122.

[38] Haiwei Luo, William Arndt, Yiwei Zhang, Guanqun Shi, Max A Alekseyev, Jijun Tang, Austin L Hughes, and Robert Friedman, *Phylogenetic analysis of genome rearrangements among five mammalian orders*, Molecular phylogenetics and evolution **65** (2012), no. 3, 871–882.

[39] Jian Ma, *A probabilistic framework for inferring ancestral genomic orders*, Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on, IEEE, 2010, pp. 179–184.

[40] Jian Ma, Louxin Zhang, Bernard Suh, Brian Raney, Richard Burhans, James Kent, Mathieu Blanchette, David Haussler, and Webb Miller, *Reconstructing contiguous regions of an ancestral genome*, Genome Research **16** (2006), no. 12, 1557–1565.

[41] Wayne Maddison, *Gene trees in species trees*, Systematic biology **46** (1997), no. 3, 523–536.

[42] Ole Madsen, Mark Scally, Christophe J Douady, Diana J Kao, Ronald W DeBry, Ronald Adkins, Heather M Amrine, Michael J Stanhope, Wilfried W de Jong,

and Mark S Springer, *Parallel adaptive radiations in two major clades of placental mammals*, Nature **409** (2001), no. 6820, 610–614.

[43] Bernard Moret, Li-San Wang, Tandy Warnow, and Stacia Wyman, *New approaches for reconstructing phylogenies from gene order data*, Bioinformatics **17** (2001), no. suppl 1, S165–S173.

[44] Bernard ME Moret, Adam C Siepel, Jijun Tang, and Tao Liu, *Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data*, Algorithms in Bioinformatics, Springer, 2002, pp. 521–536.

[45] William J Murphy, Eduardo Eizirik, Warren E Johnson, Ya Ping Zhang, Oliver A Ryder, and Stephen J O'Brien, *Molecular phylogenetics and the origins of placental mammals*, Nature **409** (2001), no. 6820, 614–618.

[46] Roderic Page and Michael Charleston, *From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem*, Molecular phylogenetics and evolution **7** (1997), no. 2, 231–240.

[47] Biller Priscila, Feijao Pedro, and Meidanis Joao, *Rearrangement-based phylogeny using the single-cut-or-join operation*, IEEE/ACM Transactions on Computational Biology and Bioinformatics **99** (2012), no. PrePrints, 1.

[48] Vaibhav Rajan, Andrew W Xu, Yu Lin, Krister M Swenson, and Bernard ME Moret, *Heuristics for the inversion median problem*, BMC bioinformatics **11** (2010), no. Suppl 1, S30.

[49] Antonis Rokas and Peter Holland, *Rare genomic changes as a tool for phylogenetics*, Trends in Ecology & Evolution **15** (2000), no. 11, 454–459.

[50] Naruya Saitou and Masatoshi Nei, *The neighbor-joining method: a new method for reconstructing phylogenetic trees.*, Molecular biology and evolution **4** (1987), no. 4, 406–425.

[51] David Sankoff and Mathieu Blanchette, *Multiple genome rearrangement and breakpoint phylogeny*, Journal of Computational Biology **5** (1998), no. 3, 555–570.

[52] Heiko Schmidt, Korbinian Strimmer, Martin Vingron, and Arndt Haeseler, *Tree-puzzle: maximum likelihood phylogenetic analysis using quartets and parallel computing*, Bioinformatics **18** (2002), no. 3, 502–504.

[53] Adam Siepel and Bernard Moret, *Finding an optimal inversion median: experimental results*, Algorithms in Bioinformatics (2001), 189–203.

[54] Berend Snel, Peer Bork, and Martijn A Huynen, *Genome phylogeny based on gene content*, Nature genetics **21** (1999), no. 1, 108–110.

[55] Alexandros Stamatakis, *Raxml-vi-hpc: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models*, Bioinformatics **22** (2006), no. 21, 2688–2690.

[56] Alexandros Stamatakis, Paul Hoover, and Jacques Rougemont, *A rapid bootstrap algorithm for the raxml web servers*, Systematic biology **57** (2008), no. 5, 758–771.

[57] AH Sturtevant and TH Dobzhansky, *Inversions in the third chromosome of wild races of drosophila pseudoobscura, and their use in the study of the history of the species*, Proceedings of the National Academy of Sciences of the United States of America **22** (1936), no. 7, 448.

[58] Krister M Swenson, Mark Marron, Joel V Earnest-DeYoung, and Bernard ME Moret, *Approximating the true evolutionary distance between two genomes*, Journal of Experimental Algorithmics (JEA) **12** (2008), 3–5.

[59] David Swofford, *Phylogenetic analysis using parsimony (* and other methods). version 4*, Sunderland, MA: Sinauer Associates (2002).

[60] David Swofford, Gary Olsen, and Peter Waddell, *Phylogenetic inference, dm hillis, c*, Moritz, BK Mable, Editors, Molecular Systematics (1996), 407–514.

[61] Jijun Tang, Bernard ME Moret, LiYing Cui, and Claude W Depamphilis, *Phylogenetic reconstruction from arbitrary gene-order data*, Bioinformatics and Bioengineering, 2004. BIBE 2004. Proceedings. Fourth IEEE Symposium on, IEEE, 2004, pp. 592–599.

[62] Jijun Tang and Li-San Wang, *Improving genome rearrangement phylogeny using sequence-style parsimony*, Bioinformatics and Bioengineering, 2005. BIBE 2005. Fifth IEEE Symposium on, IEEE, 2005, pp. 137–144.

[63] Eric Tannier, Chunfang Zheng, and David Sankoff, *Multichromosomal genome median and halving problems*, Algorithms in Bioinformatics (2008), 1–13.

[64] Glenn Tesler, *Efficient algorithms for multichromosomal genome rearrangements*, Journal of Computer and System Sciences **65** (2002), no. 3, 587–609.

[65] Li-San Wang, Robert Jansen, Bernard Moret, Linda Raubeson, Tandy Warnow, et al., *Fast phylogenetic methods for the analysis of genome rearrangement data: an empirical study.*, Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 2002, p. 524.

[66] GA Watterson, Warren J Ewens, Thomas Eric Hall, and A Morgan, *The chromosome inversion problem*, Journal of Theoretical Biology **99** (1982), no. 1, 1–7.

[67] Derek E Wildman, Monica Uddin, Juan C Opazo, Guozhen Liu, Vincent Lefort, Stephane Guindon, Olivier Gascuel, Lawrence I Grossman, Roberto Romero, and Morris Goodman, *Genomics, biogeography, and the diversification of placental mammals*, Proceedings of the National Academy of Sciences **104** (2007), no. 36, 14395–14400.

[68] Andrew Xu and David Sankoff, *Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem*, Algorithms in Bioinformatics (2008), 25–37.

[69] Andrew Wei Xu and Bernard ME Moret, *Gasts: Parsimony scoring under rearrangements*, Algorithms in Bioinformatics, Springer, 2011, pp. 351–363.

[70] Sophia Yancopoulos, Oliver Attie, and Richard Friedberg, *Efficient sorting of genomic permutations by translocation, inversion and block interchange*, Bioinformatics **21** (2005), no. 16, 3340–3346.

[71] Sophia Yancopoulos and Richard Friedberg, *Sorting genomes with insertions, deletions and duplications by dcj*, Comparative Genomics, Springer, 2008, pp. 170–183.

[72] Ziheng Yang, Sudhir Kumar, and Masatoshi Nei, *A new method of inference of ancestral nucleotide and amino acid sequences.*, Genetics **141** (1995), no. 4, 1641–1650.

[73] Hongmei Zhang, Yang Zhong, Bailin Hao, and Xun Gu, *A simple method for phylogenomic inference using the information of gene content of genomes*, Gene **441** (2009), no. 1, 163–168.

[74] Yiwei Zhang, Fei Hu, and Jijun Tang, *Phylogenetic reconstruction with gene rearrangements and gene losses*, Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on, IEEE, 2010, pp. 35–38.

[75] _____, *A mixture framework for inferring ancestral gene orders*, BMC genomics **13** (2012), no. Suppl 1, S7.

[76] Lingxi Zhou, William Hoskins, Jieyi Zhao, and Jijun Tang, *Ancestral reconstruction under weighted maximum matching*, Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on, IEEE, 2015, pp. 1448–1455.

[77] Lingxi Zhou, Yu Lin, Bing Feng, Jieyi Zhao, and Jijun Tang, *Phylogeny reconstruction from whole-genome data using variable length binary encoding*, Bioinformatics Research and Applications: 12th International Symposium, ISBRA 2016, Minsk, Belarus, June 5-8, 2016, Proceedings, vol. 9683, Springer, 2016, p. 345.