1-1-2013

Phylogeny and Ancestral Genome Reconstruction from Gene Order Using Maximum Likelihood and Binary Encoding

Fei Hu
University of South Carolina - Columbia

Phylogeny and Ancestral Genome Reconstruction from Gene Order using Maximum Likelihood and Binary Encoding

by

Fei Hu

Bachelor of Science
Huazhong University of Science and Technology, 2008

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosphy in

Computer Science

College of Engineering and Computing

University of South Carolina

2013

Accepted by:

Jijun Tang, Major Professor

Max Alekseyev, Committee Member

Jianjun Hu, Committee Member

John Rose, Committee Member

László Székely, External Examiner

Lacy Ford, Vice Provost and Dean of Graduate Studies

Acknowledgments

First and foremost I wish to thank my advisor Dr. Jijun Tang. He encouraged and guided me into the world of computational biology and grants me the freedom to explore it. He could always come up with enlightening ideas and inspire me to move forward. Without his insights and continuous supports, I would never be able to make any academic accomplishment. I sincerely wish to thank Dr. László Székely, Dr. Max Alekseyev, Dr. Jianjun Hu and Dr. John Rose for their willingnesses to serve on my dissertation committee and their valuable comments.

Abstract

Over the long history of genome evolution, genes get rearranged under events such as rearrangements, losses, insertions and duplications, which in all change the ordering and content along the genome. Recent progress in genome-scale sequencing renews the challenges in the reconstructions of phylogeny and ancestral genomes with gene-order data. Such problems have been proved so interesting that a large number of algorithms have been developed rigorously over the past few years in attempts to tackle these problems following various principles. However, difficulties and limitations in performance and scalability largely prevent us from analyzing emerging modern whole-genome data, our study presented in this dissertation focuses on developing appropriate evolutionary models and robust algorithms for solving the phylogenetic and ancestral inference problems using gene-order data under the whole-genome evolution, along with their applications.

To reconstruct phylogenies from gene-order data, we developed a collection of closely-related methods following the principle of likelihood maximization. To the best of our knowledge, it was the first successful attempt to apply maximum likelihood optimization technique into the analysis of gene-order phylogenetic problem. Later we proposed `MLWD` (in collaboration with Lin and Moret) in which we described an effective transition model to account for the transitions between presence and absence states of an gene adjacency. Besides genome rearrangements, other evolutionary events modify gene contents such as gene duplications and gene insertion/deletion (indels) can be naturally processed as well. We present our results from extensive testing on simulated data showing that our approach returns very accurate results

very quickly.

With a known phylogeny, a subsequent problem is to reconstruct the gene-order of ancestral genomes from their living descendants. To solve this problem, we adopted an adjacency-based probabilistic framework, and developed a method called `PMAG`. `PMAG` decomposes gene orderings into a set of gene adjacencies and then infers the probability of observing each adjacency in the ancestral genome. We conducted extensive simulation experiments and compared `PMAG` with `InferCarsPro`, `GASTS`, `GapAdj` and `SCJ`. According to the results, `PMAG` demonstrated great performance in terms of the true positive rate of gene adjacency. `PMAG` also achieved comparable running time to the other methods, even when the traveling sales man problem (TSP) were exactly solved.

Although `PMAG` can give good performance, it is strongly restricted from analyzing datasets underwent only rearrangements. To infer ancestral genomes under a more general model of evolution with an arbitrary rate of indels , we proposed an enhanced method `PMAG+` based on `PMAG`. `PMAG+` includes a novel approach to infer ancestral gene contents and a detail description to reduce the adjacency assembly problem to an instance of TSP. We designed a series of experiments to validate `PMAG+` and compared the results with the most recent and comparable method `GapAdj`. According to the results, ancestral gene contents predicted by `PMAG+` coincided highly with the actual contents with error rates less than 1%. Under various degrees of indels, `PMAG+` consistently achieved more accurate prediction of ancestral gene orders and at the same time, produced contigs very close to the actual chromosomes.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the last few decades, gene-order data have been widely recognized and successfully used in the biological research. Although sequential data of nucleotides and amino-acids still dominate and have been thoroughly studied, gene-order data generated from the permutation of genes along a genome, has the potential to return more meaningful and convincing results. Since operations on genes are much more difficult to happen than point mutations at nucleotide level, gene ordering allows researchers to trace farther 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 formalized. Deep mathematical and algorithmic methods handling gene-order permutations to solve various biological problems have been developed, however the results are still far from satisfaction. By the emerging of whole-genome and high-resolution data, it is clear that new methods and algorithms are greatly needed to improve on the current solutions of these problems.

1.1 Literature Review

Pioneerd by Dobzhansky and Sturtevant in 1936, 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 [54, 15]. But what on earth allows us to utilize the order of genes to carry out all kinds of

studies in comparative genomics? The key is that genes themselves are less subject to mutations and are therefore rarely cut by rearrangement [44]. Therefore by viewing a genome as a permutation of genes (conservative blocks) in the order in which they are placed along one or more chromosomes, gene-order data enables the reconstruction of evolutionary events far back in time [46, 5].

Later in 1982, Watterson presented the very first and formal description of chromosome inversion problem [63], in which they wished to calculate a measure of distance between two species for the purpose of constructing a phylogenetic tree. Then how to calculate the minimum number of inversion events (defined as the edit distance) necessary to transform one genome into the other? Until nearly a decade later in 1995, Hannenhalli and Pevzner [24] provided the first polynomial solution for the chromosome inversion problem and their finding has greatly promoted the development of gene-order study. The next significant advance in distance metrics between two genomes is the introduction of double-cut-and-join (DCJ) distance [66]. Although DCJ is not directly observable or provable through biological study, the DCJ distance is favored since it can emulate a variety of other events which greatly simplifies the computational model.

Researchers working on gene-order data mainly focus on tackling two different yet related problems : the phylogeny problem and the ancestor inference problem. The phylogeny problem aims to reconstruct the phylogeny in terms of a binary tree from a set of gene-orders of extant species, while the ancestor inference problem searches for the most plausible gene-order of an ancestral genome. An ancestral genome is represented by an internal node in a phylogeny tree.

Methods for phylogenetic reconstruction from gene-order data can be roughly classified into distance-based and parsimony-based according to the criterion they follow. Saitou [45] introduced the first distance-based methods called `Neighbor-joining` intended for treating DNA sequences. As all distance-based methods are based on

statistical clustering from a distance matrix computed between each pair of genomes, thus `Neighbor-joining` was soon adopted for solving the phylogeny problem using gene-order data. In 2002, Desper [14] proposed 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, hence some forms of corrections are needed. Empirical derived estimation (`EDE`) [37] 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 estimate the true inversion distance. Later Lin developed `TIBA` [31] which provides a more accurate estimate of the true pairwise distances.

On the other hand, there are a wide selection of parsimony-based method for gene-order phylogeny problem. Most of these parsimony-based methods use direct optimization technique. In particular, `BPAnalysis` [46] was written by Blanchette and Sankoff in 1998, which is the first program to reconstruct phylogenies based on the breakpoint parsimony of gene orders. Moret and Tang [39, 38] in 2002 released `GRAPPA` which greatly improves on the results and on the efficiency of `BPAnalysis` by replacing the breakpoint median solver with an inversion median solver. Around the same time, Bourque and Pevzner published the `MGR` [6] which also abandoned the breakpoint distance and addressed the issue of handling multichromosomal genomes.

Another type of parsimony-based methods relies on the encoding techniques on gene-order data which transforms permutations into sequences and then uses existing analysis tools for sequence data to compute for a gene-order phylogeny. In particular, Cosner proposed the first method of this kind called Maximum Parsimony on Binary Encodings (`MPBE`) [12, 13] which produces one character for each gene adjacency present in the data. Later Wang [62] gave the second method called `MPME` (M stands for multistates) in which each signed gene has exactly one character. In all evalu-

ations, both MPBE and MPME methods were easily surpassed by direct optimization approaches.

Yet to date, probabilistic methods for solving the gene-order phylogeny problem are represented by a single effort from Larget [29], in which a Bayesian approach showed evidence of success on a couple of fairly easy datasets; the same approach, however, failed to converge on a harder dataset analyzed by Tang [58].

While 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. In particular, Tang [58] proposed a way of determining the gene content when solving for the median in GRAPPA. Zhang showed his method is remarkably more accurate than its predecessor, however handling gene duplication is still out-of-reach. For distance methods, El-Mabrouk [16] first presented an exact algorithm for the computation of edit distances for inversions and losses. More recently, Yancopoulos [67] suggested a way to compute edit distances under indels, duplications, and DCJ operations. Swenson [55] gave an algorithm to approximate the true evolutionary distance under indels, duplications, and inversions for single chromosomal genomes and showed good results under simulation study.

For the ancestor inference 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 ancestor at the same time, but they are NP-hard. In order to boost the accuracy and scalability at the same time, many methods were published in the last few years. MGRA relies on the notion of the multiple breakpoint graphs is a more recent derivative of MGR developed by Alekseyev [1]. GASTS, developed by Xu [65], is based on a fast and accurate heuristic for the inversion median solver that they developed [42] which scales up linearly instead of exponentially with the size of the genomes involved. The Single-Cut-or-Join (SCJ) operation [18, 4] was proposed as a new rearrangement distance

between multi-chromosomal genomes, leading to a fast median solver and Fitch-style algorithm for ancestor inference.

A new framework has been established and attracted a lot of attention in the last a couple of years. Unlike previous methods that explicitly focus on a set of predefined evolutionary events, these methods work on gene adjacencies and the goal is usually to determine whether or not an adjacency can be observed in an ancestor. The pioneer method `InferCars` was developed by Ma [35] and later he presented a probabilistic version `InferCarsPro` [33] by incorporating a modified Jukes-Cantor model. Gagnon introduced a new concept of "Gapped Adjacency" and proposed a method called `GapAdj` [20]. `GapAdj` is considered flexible since it can handle dataset with unequal gene-content. By mixing the framework of event-based (`GRAPPA`) and adjacency-based (`InferCarsPro`) methods, Zhang [70] developed a method which inherits the high performance of direct optimization and reduces its difficulty by fixing a portion of adjacencies before exact optimization.

1.2 Academic Contributions

All the work 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 Maximum Likelihood on Binary Encoding (`MLBE`) and its successor Maximum Likelihood on Whole-genome Data (`MLWD`) for the phylogeny problem, Probabilistic Method of Ancestral Genomics (`PMAG`) and its extension `PMAG+` for the ancestor reconstruction.

The first successfully attempt of applying maximum-likelihood into gene-order analysis

In chapter 3, we described a series of maximum-likelihood approaches to phylogenetic analysis from whole-genome data. `MLBE` and its two variants (`MLBE2` and `MLME`) proposed in [25] was the first attempt to apply maximum-likelihood criterion into the analysis of gene-order phylogeny. Although gene-encoding based methods `MPBE` and `MPME`, have been available for over a decade, 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 dataset 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 uniform manner. Later in close collaboration with Lin and Moret, following the previous framework, we developed a faster and more accurate yet simpler method `MLWD` in which we introduced a biased transition model and a simplified gene-encoding scheme.

Related publications are listed below.

1. Fei Hu, Nan Gao, Meng Zhang and Jijun Tang, "Maximum Likelihood Phylogenetic Reconstruction Using Gene Order Encodings", The 8th Annual IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'11), 117-122.

2. Yu Lin, Fei Hu, Jijun Tang and Bernard Moret, "Maximum Likelihood Phylogenetic Reconstruction from High-Resolution Whole-Genome Data and a Tree of 68 Eukaryotes." Pacific Symposium on Biocomputing 18:285-296(2013)

Robust probabilistic methods for ancestral genome reconstruction

In chapter 4 and chapter 5, we described a method for ancestral genome reconstruction `PMAG` and its extension `PMAG+`. `PMAG` series fall into typical adjacency-based probabilistic approaches which try to answer how likely an adjacency to be observed in an ancestor. We carefully investigated currently available methods and overcame several major issues associated with them.

First, our methods are fast and scale up to handle whole-genome data. This is achieved by treating each adjacency in the leaf genomes as a unique and independent 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 [68] to calculate the posterior probability of an adjacency in an ancestor. This model has been proved in `MLWD` to be very useful in phylogeny inference. Third, `PMAG+` is able to handle gene losses and insertions though a novel probabilistic approach for inferring ancestral genome contents. The underlying idea is straightforward: by treating a gene as a character, we can compute the probability of observing this gene in an ancestor genome. Fourth, `PMAG+` implemented a more sophisticated way to assemble gene adjacencies into a valid gene-order permutation. It replaces the greedy assembly with an exact solution by solving an instance of symmetric TSP. This strategy not only increases the performance of method, but also significantly mitigates the issue of bad assembly of gene adjacencies.

Related publications are listed below.

1. Fei Hu, Lingxi Zhou and Jijun Tang. "Reconstructing Ancestral Genomic Orders Using Binary Encoding and Probabilistic Models." Bioinformatics Research and Applications. Springer Berlin Heidelberg, 2013. 17-27.

2. Fei Hu, and Jijun Tang. "Probabilistic Reconstruction of Ancestral Genomes with Gene Insertions and Deletions." Asia Pacific Bioinformatics Conference 2014 (APBC'14), Accepted

Chapter 2

Background

2.1 Gene Order Format and Genome Rearrangements

Given a set of n genes $\{g_1, g_2, \cdots, g_n\}$, a genome can be represented by an *ordering* of these genes. To indicate the strandedness of genes, each gene is assigned with an orientation that is either positive, written g_i, or negative, written $-g_i$. Two genes i and j are said to be *adjacent* in genome G if i is immediately followed by j, or, equivalently, $-j$ is immediately followed by $-i$. A *breakpoint* of two genomes is defined as an adjacency appears in one but not in the other.

Let G be the multi-chromosomal genome with signed ordering $\{a_1, a_2, \cdots, a_n\}$, $\{b_1, b_2, \cdots, b_m\}, \cdots$ ($\{\cdots\}$indicates a chromosome). An *inversion* (also called *reversal*) between indices i and j ($i \leq j$) of chromosome a, produces a chromosome a' with linear ordering

$$a_1, a_2, \cdots, a_{i-1}, -a_j, -a_{j-1}, \cdots, -a_i, a_{j+1}, \cdots, a_n.$$

A *transposition* on a chromosome a acts on three indices i, j, k, with $i \leq j$ and $k \notin [i, j]$, picking up the interval $a_i, \cdots, a_j$ and inserting it immediately after a_k. Thus the chromosome a of the genome is replaced by (assume $k > j$):

$$a_1, \cdots, a_{i-1}, a_{j+1}, \cdots, a_k, a_i, a_{i+1}, \cdots, a_j, a_{k+1}, \cdots, a_n.$$

A *translocation* on genome G acts on two indices i, j belonging to different chromosomes, picking up the interval $a_i, \cdots, a_n$ and $b_j, \cdots, b_m$ and then changing their

places. Thus the chromosomes a, b of genome G become

$$\{a_1, a_2, \cdots, a_{i-1}, b_j, \cdots, b_m\}, \{b_1, b_2, \cdots, b_{j-1}, a_i, \cdots, a_n\}$$

A *transversion* is a transposition followed by an inversion of the transposed subsequence; it is also called an *inverted transposition*. A *fusion* operations join two linear chromosomes into one linear chromosome and the *fission* is the reverse of *fusion* such that a *fission* operates on a single chromosome and make the separation at a chosen point.

There are another set of operations which can alter the gene content in a genome. A *deletion* (also called *loss*) deletes a single or a segment of genes from the genome. Its reverse operation called *insertion* introduces one or a segment of genes that have not seen before into a chromosome at a time. *Whole genome duplication* (WGD) creates an additional copy of the entire genome of a species.

The universal double-cut-and-join operation that accounts for inversions, transposition and translocations which resulted in a new genomic distance that can be computed in linear time. In particular, a DCJ operation consists of cutting two connections (breakpoints) in a genome, and rejoining the resulting four unconnected ends in two new pairs. 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.

Later single-cut-or-join (`SCJ`) was proposed as a breakpoint-like rearrangement event. Basically any operation that destroys a gene adjacency or reestablishes a gene adjacency from two telomeres can be regarded as a valid `SCJ` operation. Using `SCJ` provides the first polynomial solution for ancestral genome inference, although its accuracy is worse than other competing methods as shown in the following tests.

Figure 2.1: Phylogeny reconstructed from 12 plants from the Campanulaceae family

2.2 Surveys on Methods for Gene-Order Phylogenetic Reconstruction

Phylogenies is a term that represents the reconstructed evolutionary relationship of a set of organisms in the form of a binary tree 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 2.1 shows an example phylogeny reconstructed from 12 plants from the Campanulaceae family [12].

All types of data can be used for phylogenetic reconstruction from geographic and ecological, through the morphological and metabolic to the molecular data [57]. By the rapid accumulation in molecular data and also due to its merit of exact and easy accessibility, sequence-based data of a few genes long has became the predominant source for phylogenetic analysis. But they suffer from a number of prominent problems, especially the well-known gene tree vs. species tree problem [41, 36]. Gene-order data, a relatively recent and promising data type, studies the whole-genome at once from a higher-level perspective and hence naturally resolves the gene tree vs. species tree problem. On the other hand, there are great mathematical challenges encountered in detecting and handling the genome-scale changes, not to mention to directly employ existing techniques for sequences data. In the recent years, the phylogenetic reconstruction from gene-order data has attracted a lot of attention from both computer scientist and biologist and a number of methods have been developed in coping with this problem.

`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 great scalability with the number and size of genomes as well as an acceptable performance they can achieve. Their performances largely depend on how the dis-

tance measurement is defined and how well such distances are congruent with the true distance. Although Hannenhalli and Penvzner provided the first polynomial algorithm for computing the minimum number of inversions between two genomes, the true evolutionary distance is always severely underestimated. In order to approach the true number of evolutionary operations, `TIBA` replies 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 on the fundamental assumption that the true 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, `BPAnalysis` is based on the breakpoint distance. It enumerates all $(2n-5)!!$ trees and uses an iterative heuristic to label the internal nodes with signed gene orders. To improve its speed and performance, it was reimplemented and evolved into `GRAPPA` (Genome Rearrangement Analysis under Parsimony and other Phylogenetic Algorithms). `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 often need to solve numerous instances of median problem. In particular the median problem can be formalized as follows: give 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. But for almost all evolutionary distances, solving for the exact median genome is NP-hard [9, 7, 60]. Therefore direct optimization methods are rather accurate but also extremely time-consuming. On exception is the breakpoint-like Single-cut-or-join (`SCJ`) which has a polynomial time solution for the median problem, but in overall a

branch-and-bound search for the phylogeny with SCJ is still NP-hard.

MPBE (Maximum Parsimony on Binary Encoding) transforms adjacency pairs from the signed permutation into strings of binary characters. These strings are further converted into nucleotide sequences and analyzed using ordinary sequence parsimony software (e.g. PAUP* 4.0 [56]) to obtain a phylogeny. MPME (Maximum Parsimony on Multistate Encoding) uses a new kind of encoding scheme to improve the accuracy. These encoding based parsimony methods can achieve slightly better accuracy compared to the uncorrected distance method, yet they are computationally very expensive.

2.3 Surveys on Methods for Ancestral Gene-Order Reconstruction

The success of phylogenetic reconstruction demonstrates the power of revealing the evolutionary relation of a group of organisms by computational means. As phylogeny often takes the form of 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 orders of these ancestors has been investigated in-depth and several methods have been developed for the task.

Depending on whether the phylogeny tree is given, ancestral genome reconstruction problem can be 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. In this study, we are interested in solving the small phylogeny problem. Majority of current methods 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 methods mainly focus on the analysis of independent gene adjacencies and try to calculate or estimate a

score for each gene adjacency to be 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.

But from another point of view, some of the methods employ a parsimonious framework and suggest to use least number of changes to explain observed data; while the others estimate the parameters and use probabilities or likelihoods to score the gene adjacencies. Table 2.1 summaries the differences between methods solving the 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 GapAdj	InferCARsPro
$Rearrangement - based$	GRAPPA, MGR GASTS, SCJ	N/A

In the context of rearrangement-based parsimonious methods, `GRAPPA` and `MGR` (as well as their recently enhanced versions) are two similar methods that implemented a selection of median solvers for phylogeny and ancestral gene-order inference. In detail, given a tree topology, `GRAPPA` iteratively assigns median genomes to ancestral nodes in the tree until the total tree score will not decrease. Then the set of gene-order assignments that minimizes the tree score are reported as the ancestral genomes. Since the scoring procedure of `GRAPPA` involves solving numerous instances of median problems, a fast median solver is crucial. Exact solutions to the problem of finding a median of three genomes can be obtained for the inversion, breakpoint and DCJ distances [10, 49, 64]. Among all the median solvers, the best one is the DCJ median solver `ASMedian` [64] based on the concept of adequate subgraph. Adequate subgraphs allow decompositions of a multiple breakpoint graph into smaller and easier graphs. Though the `ASMedian` solver could remarkably scale down the computational expenses

of median searching, it yet runs very slow when the genomes are distant. On the other hand, `GASTS` and `SCJ` can scale up to handle high-resolution vertebrate genomes. `GASTS` is based on a fast and accurate heuristic for the inversion median [42] in which only a few of the simplest decompositions of adequate graphs will be solved; it provides a fast and robust scoring method for a fixed tree and demonstrated very high accuracy in the simulation experiments compared to `MGR`. `SCJ` utilizes the Fitch's small parsimony algorithm to solve the SPP in which each adjacency is viewed as a binary character of state either presence or absence and 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 methods were firstly introduced in `InferCARs`. It identifies the 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 `InferCARsPro` was introduced as an extension to the previous work in the probabilistic framework. The kernel of `InferCARsPro` is to predict the posterior probability of observing an adjacency in the ancestor based on an extended Jukes-Cantor model for breakpoints. However, neither of them is able to handle dataset with unequal gene content and greedy heuristic often returns an unmatched number of CARs. Besides both methods require users to input a phylogeny with accurate branch lengths. To address these problems, `GapAdj` was 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 pairs of genes separated by up to a given number of genes as direct gene adjacencies. `GapAdj` can also analyze datasets with unequal gene contents by first inferring the ancestral gene content through a natural process [22].

Chapter 3

Phylogeny Reconstruction from Gene Order Encoding

3.1 Motivation

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 [21] and `PAUP*` [56] 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, besides parsimonious framework, sequence data can be analyzed by searching the phylogeny with maximized likelihood score as suggested by Felsenstein [19] 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 [52] have made the maximum

likelihood approach feasible for large scale analysis of molecular sequences. These improvements motivated us to utilize the technique and apply it for gene order phylogeny analysis through encodings of gene orders.

In the rest of this chapter, we will first describe the Maximum Likelihood on Binary Encoding (MLBE and MLBE2) and Maximum Likelihood on Multistate Encoding (MLME). Then Maximum Likelihood on Whole-Genome Data (MLWD) will be introduced. Finally we will show our experimental design along with evaluations of various methods.

3.2 Maximum Likelihood on Binary Encoding

Maximum Likelihood on Binary Encoding with Amino Acid Characters (MLBE)

Let G be a signed permutation of n genes. For linear genomes, genes 0 and $n+1$ are added to indicate the start and end of a genome respectively. For the pair $(i, j), 0 \leq i, j \leq n+1$, we set up a character to indicate the presence or absence of this adjacency. If i is immediately followed by j in the gene ordering, or $-j$ is immediately followed by $-i$, we then put a 1 to the sequence at the corresponding site where the character represents this pair and put a 0 otherwise. 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. Table 3.1 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.

After converting the gene orders into strings of 0 and 1, we further convert these sequences into amino acid sequences and utilize the power of those widely used ML packages developed for molecular sequences. We tested several ML packages such as `TREE-PUZZLE` [48] and `GARLI` [23] and among them, `RAxML` [52] is the best by

Table 3.1: Example of the binary encoding (0 indicates the start of a genome, 6 indicates the end of a genome)

.

$$G_1 \ : \ (4, 5, -2, -1, -3) \qquad (3.1)$$
$$G_2 \ : \ (-1, 4, 2, -3, -5) \qquad (3.2)$$
$$G_3 \ : \ (3, 2, -5, -4, -1) \qquad (3.3)$$

(a) Three signed linear genomes

	Adjacencies															
	0,4	4,5	5,-2	-2,-1	-1,-3	-3,6	0,-1	-1,4	4,2	2,-3	-3,-5	-5,6	0,3	3,2	-4,-1	-1,6
G_1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
G_2	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
G_3	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1

(b) Binary Encoding

incorporating the rapid bootstrapping [53]. In the method of `MLBE`, the following steps are used to convert 1 and 0 into amino acids:

- For a dataset, randomly pick an amino acid to code *absent state*.
- For an adjacency, code *present state* by randomly picking one from the remaining 19 amino acids, such choice will be preserved for all genomes on the site corresponding to the given adjacency.

Table 3.2 shows an example of amino acid sequences produced by `MLBE` from the binary strings of the genome presented in Table 3.1.

Table 3.2: Example of the converted sequences using MLBE, V is picked to encode absent state in all sequences.

	Adjacencies															
	H,4	4,5	5,-2	-2,-1	-1,-3	-3,T	H,-1	-1,4	4,2	2,-3	-3,-5	-5,T	H,3	3,2	-4,-1	-1,T
G_1	Q	K	S	A	N	A	V	V	V	V	V	V	V	V	V	V
G_2	V	V	V	V	V	V	W	Q	R	C	Y	Y	V	V	V	V
G_3	V	K	S	V	V	V	V	V	V	V	V	V	L	F	M	H

Maximum Likelihood on Binary Encoding with Nucleotide Characters (MLBE2)

MLBE2 uses a simple code that treats every 1 as A or T, and consequently every 0 as C or G. Like what we regulate in MLBE, at the site of a given adjacency across all genomes, we enforce the presence of nucleotides to be either A associating with C or T associating with G. Such choice is randomly assigned so that a balanced number of the two pairings is presented in the new sequences.

Table 3.3 shows the example of nucleotide coding of the binary strings of the genomes presented in Table 3.1. Again, RAxML will be used to obtain trees from these nucleotides sequences.

Table 3.3: Example of the converted sequences using MLBE2. Nucleotides are used in pairs to substitute binary characters.

	Adjacencies															
	0,4	4,5	5,-2	-2,-1	-1,-3	-3,6	0,-1	-1,4	4,2	2,-3	-3,-5	-5,6	0,3	3,2	-4,-1	-1,6
G_1	T	T	A	T	T	T	C	G	G	C	C	C	C	G	G	C
G_2	G	G	C	G	G	G	A	T	T	A	A	A	C	G	G	C
G_3	G	T	A	G	G	G	C	G	G	C	C	C	A	T	T	A

3.3 Maximum Likelihood on Multistate Encoding

Bryant [8] proposed an encoding method called Multistate Encoding. Let n be the number of genes in each genome; then each gene order is translated into a sequence with $2n$ characters. For every gene i, $1 \leq i \leq n$, site i takes the value of the gene immediately following i; site $n + i$ takes the gene immediately following gene $-i$. Table 3.4 shows an example of such encoding.

Once the states of each site are determined for all genomes, we can easily convert them into molecular sequences by randomly assigning amino acid to a given state and then use RAxML to compute the phylogeny. We call such method MLME in this study and an example of converted amino acid sequences is shown in Table 3.5. Since

Table 3.4: Examples of multistate encoding, 0 indicates the start of a genome, 6 indicates the end of a genome.

$$G_1 \ : \ (4, 5, -2, -1, -3) \tag{3.4}$$
$$G_2 \ : \ (-1, 4, 2, -3, -5) \tag{3.5}$$
$$G_3 \ : \ (3, 2, -5, -4, -1) \tag{3.6}$$

(a) Three signed linear genomes

	Genes									
	1	2	3	4	5	-1	-2	-3	-4	-5
G_1	2	-5	1	5	-2	-3	-1	6	0	-4
G_2	0	-3	-2	2	3	4	-4	-5	1	6
G_3	4	-5	2	5	-2	6	-3	0	-1	-4

(b) Multistate Encoding

`RAxML` only deals with 20 amino acids, hence no site can have more than 20 states. As a result, `MLME` is limited to handle datasets with no more than 20 genomes at this stage.

Table 3.5: Example of the converted sequences using MLME. No site can have more than 20 states.

	Genes									
	1	2	3	4	5	-1	-2	-3	-4	-5
G_1	G	D	R	L	Y	W	N	S	K	E
G_2	N	K	M	T	C	P	M	G	W	V
G_3	C	D	P	L	Y	Q	I	H	V	E

3.4 Maximum Likelihood Reconstruction from Whole-Genome Data

Previous three methods are the premature version of `MLWD`, as they rely on existing transition models designed for the sequential data with nucleotide or amino acid characters, however as a method for gene-order data, it is more desirable to develop a designated model from the characteristics of gene rearrangements. Let us give a close look at gene adjacencies. One DCJ operation randomly selects two adjacencies

(or telomeres) uniformly to break up followed by a creation of two new adjacencies. Since each genome has $n+O(1)$ adjacencies and telomeres (n is the number of genes. $O(1)$ is the number of linear chromosomes in the genome, viewed as a small constant). Thus the transition probability from 1 to 0 at some fixed index in the sequence is $\frac{2}{n+O(1)}$ 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 $\frac{2}{2n^2+O(n)}$. 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)—in effect, a version of homoplasy for adjacencies. In order to set up the $2n$ ratio, we simply add a direct assignment of the two base frequencies in the code.

Once we have the binary sequences encoding the input genomes and have computed the transition parameters, we can avoid the process of transforming binary encodings into artificial biological sequence and directly use ML reconstruction program `RAxML` to build a tree from these sequences. We call this approach Maximum Likelihood on Whole-genome Data (MLWD)

3.5 Experimental Design and Simulation Results

Experimental Design and Simulation Result on MLBE, MLBE2, MLME

We tested `MLBE`, `MLBE2`, `MLME` using simulated datasets. In our simulations, we generated model tree topologies from the uniform distribution on binary trees, each with 10, 20 and 40 leaves. We chose genomes of 200 and 1,000 genes, spanning the range from organelles to small bacteria. On each tree, we evolved signed permutations using various numbers of evolutionary rates: letting r denote the expected number

of rearrangement events (80% inversion and 20% transposition) along an edge of the true tree, we used values of $r = 20, 35, \cdots, 80$ for 200 genes and $r = 100, 175, \cdots, 400$ for 1000 genes. The actual number of events along each edge was sampled from a uniform distribution on the set $\{1, 2, \cdots, 2r\}$. For each combination of parameter settings, we ran 10 datasets and averaged the results.

We compared ML methods with other two methods: `FastME` with the true distance estimator based on DCJ distance (`CDCJ`) [31], `FastME` with Empirical Distance Estimation (`EDE`). `MPBE` was also added to the test only with the datasets of 10 genomes so that `MPBE` can accomplish the test with branch-and-bound search in an appropriate time. Since neither `GRAPPA` nor `MGR` can finish any of the above tests within days of computation, we therefore conducted a special experiment to accommodate `GRAPPA` and `MGR` by simulating datasets of 10 genomes matching mitochondrial DNA consisting of 37 genes where transpositions are dominant [50]. In particular, the `GRAPPA` was configured to use Caprara's inversion median solver [11] and enable EDE distance estimator. And `MGR` was tested given the parameter $-c$ and $-H1$ for efficiency and speed. Similarly the number of events was the value of $r = 2, 3, \cdots, 6$ and the actual events were sampled from the set of $\{1, 2, \cdots, 2r\}$. Finally we ran `RAxML` with the same setting but on binary strings as a control test to demonstrate the efficiency of our three approaches of encodings which is called RAxML-Binary in this study.

We assess topological correctness by computing the *false negatives* (FN) and *false positives* (FP) [43] rates. The *false negatives* are those edges in the true tree but not in the inferred tree. The *false positives* are those edges in the inferred tree that do not exist in the true tree. The *false negatives rate* is the number of false negatives divided by the number of internal edges. The *false positives rate* is similarly defined. The *Robinson-Foulds* (RF) rate is then defined as the average of the FN and FP rates. An RF rate of more than 5% is generally considered too high [57].

Figure 3.1 and Figure 3.2 show the topological accuracy of these methods (`MLME`

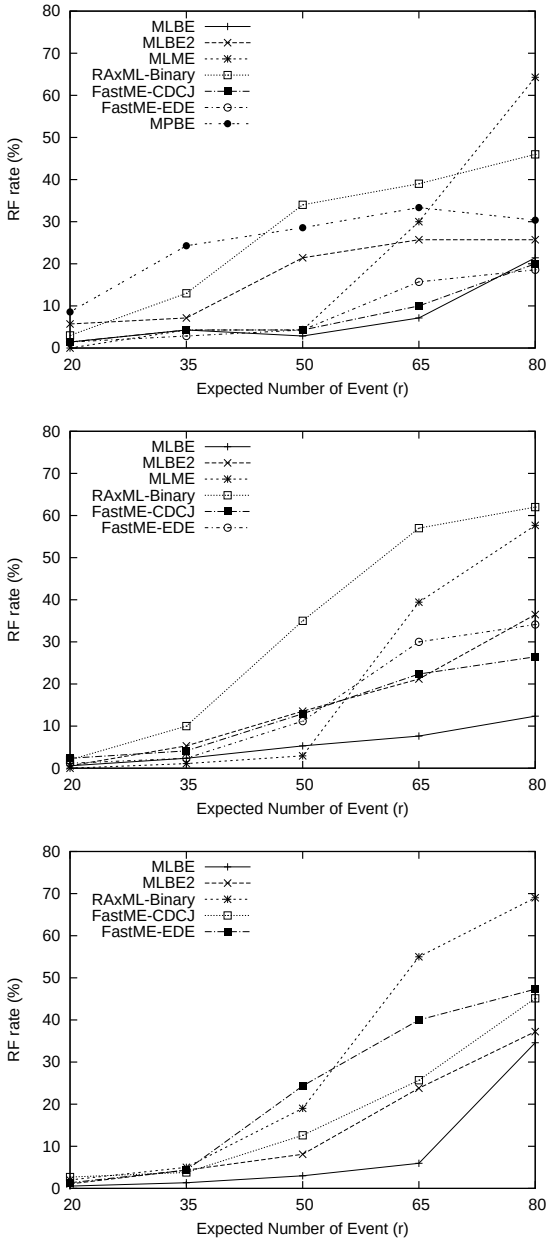


Figure 3.1: RF rates for MLBE, MLBE2, MLME, RAxML-binary, FastME-CDCJ, FastME-EDE, MPBE on 200-gene datasets (top: 10 genomes, middle: 20 genomes(MPBE was excluded), bottom: 40 genomes(MLME and MPBE were excluded)).

is not applicable for 40-genome datasets and `MPBE` is too slow to finish the brand-and-bound search for datasets containing 20 and 40 genomes). Both figures show that `MLBE` was of the most accuracy in most of the cases when genome number is 20 and 40, except for a few occasions when `MLME` becomes the best(20 genomes, 200

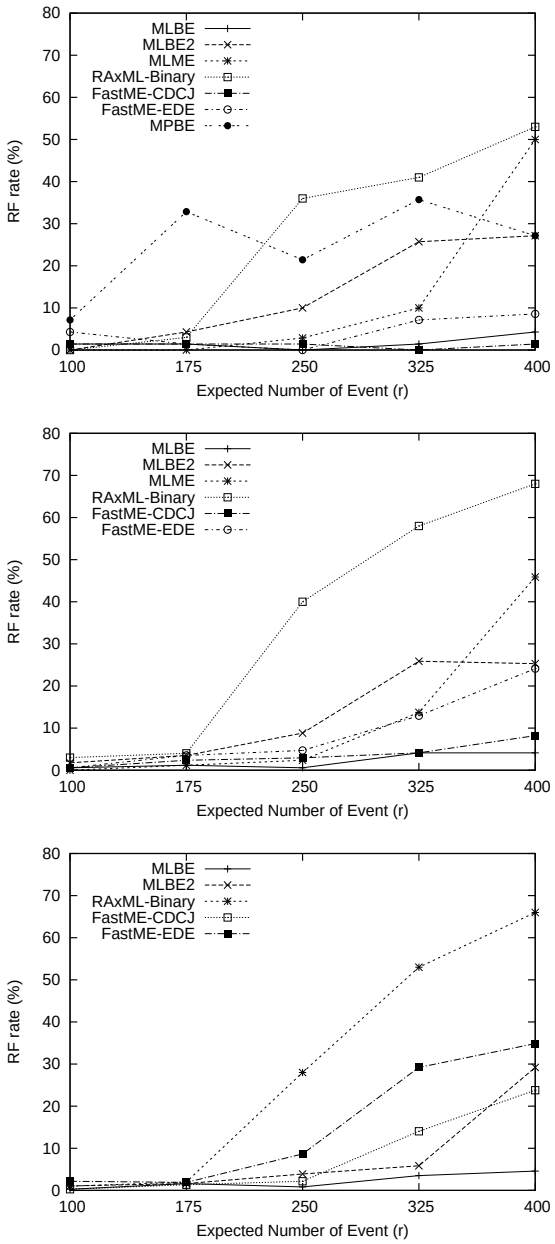


Figure 3.2: RF rates for MLBE, MLBE2, MLME, RAxML-binary, FastME-CDCJ, FastME-EDE, MPBE on 1000-gene datasets (top: 10 genomes, middle: 20 genomes(MPBE was excluded), bottom: 40 genomes(MLME and MPBE were excluded)).

genes, fewer than 50 events). As to the results for datasets of 10 genomes, `MLBE` and distance methods quite matched each other in performance and both outperformed the other methods. Figure 3.3 shows the results of simulated mitochondrial gene orderings with only transpositions applied when `GRAPPA` and `MGR` were also present in

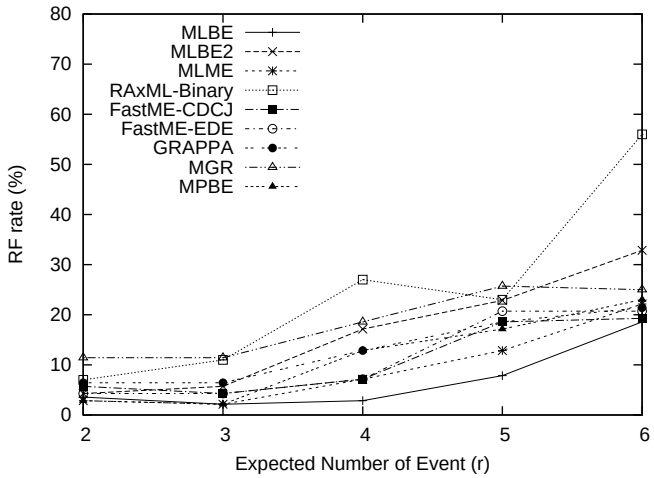


Figure 3.3: RF rates for MLBE, MLBE2, MLME, RAxML-binary, FastME-CDCJ, FastME-EDE, GRAPPA, MGR, MPBE on simulated 37-gene and 10-genome datasets where only transposition existed.

the contest. The results suggested the `MLBE` possessed the greatest performance in all conditions compared to the parsimony methods(`GRAPPA`, `MGR` and `MPBE`) and distance methods(`FastME-CDCJ` and `FastME-EDE`). In contrast `RAxML-Binary` is significantly worse in accuracy and stability in most of cases. Although `MLBE` and `MLBE2` are both based on the same principle of binary encoding, `MLBE` is more accurate by using amino acids to code the binary strings. The performance of ML methods improves with more genes, indicating that the length of the sequences has big impact on their accuracy.

`FastME` was always the fastest in our testing, while the speed of these ML methods were acceptable. Table 3.6 and 3.7 present the average time used by these methods.

These two tables show that `MLBE2` is very fast and generally requires less than one hour to compute, while `MLME` is very slow and may take up to three days to finish. `MLBE` is much slower than `MLBE2` and its speed quickly decreases with the increase of number of characters. However, it only requires fewer than 13 hours even for the most difficult datasets. Comparing to the results in Figures 3.1 and 3.2, such computation time is worthwhile and is easily offset by the increased accuracy of inferred phylogenies. Since all our tests were conducted on single processors and did

not use the parallel version of RAxML, MLBE has the potential to handle several dozens of large nuclear genomes if the full computational power of RAxML is utilized.

Table 3.6: Time usage of ML methods on 200 genes(- indicates missing data since MLME cannot be used for more than 20 genomes).

methods	time (in minutes)								
	r=20			r=65			r=80		
	N=10	N=20	N=40	N=10	N=20	N=40	N=10	N=20	N=40
MLBE	6	30	222	10	96	318	15	150	468
MLBE2	0.3	2	7	1	7	12	2	12	34
MLME	18	72	-	35	144	-	65	948	-

Table 3.7: Time usage of ML methods on 1000 genes (- indicates missing data since MLME cannot be used for more than 20 genomes).

methods	time (in minutes)								
	r=100			r=250			r=400		
	N=10	N=20	N=40	N=10	N=20	N=40	N=10	N=20	N=40
MLBE	18	132	354	24	174	414	30	240	756
MLBE2	1	4	15	2	10	30	2.8	15	55
MLME	85	324	-	110	558	-	400	4200	-

Experimental Design and Simulation Result on MLWD

We ran a series of experiments on simulated datasets in order to evaluate the performance of MLWD against a known "ground truth" under a wide variety of settings. Our simulation studies follow standard practice in phylogenetic reconstruction. 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 datasets of leaf genomes for which we know the complete evolutionary history. We then reconstruct trees for each dataset by applying different reconstruction methods and compare the results against the model tree.

Experimental Design

A model tree consists of a rooted tree topology and corresponding branch lengths. The trees are generated by a three-step process. We first generate birth-death trees with a birth rate of 0.001 and a death rate of 0, which simulates the development of a model tree under a uniform, time-homogeneous birth-death process. The branch lengths in such trees are ultrametric (the root-to-leaf paths all have the same length), so, in the second step, the branch lengths are modified as follows. We choose 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 study, we set $c = 2$). Thus, each branch length is multiplied by a possibly different random number. Finally, we rescale all branch lengths to achieve a target diameter D (the length of the longest path, defined as the sum of the edge lengths along that path) for the model tree. (Note that the unit of "length" is one expected evolutionary operation.)

Our experiments are conducted by varying three main parameters: the number of taxa , the number of genes, and the target diameter. We used two values for each of the first two parameters: 50 and 100 taxa, and $1,000$ and $5,000$ genes. For the third parameter, the diameter of the tree, we varied it from n to $4n$, where n is the number of genes. For each setting of the parameters, we generated 100 datasets; data presented below are averages over these 100 datasets.

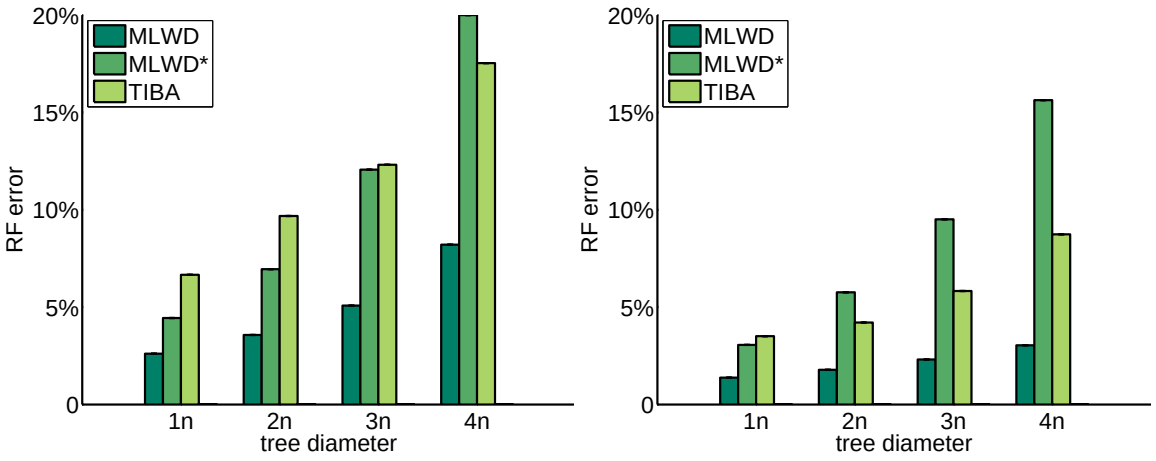
In the rearrangement-only model, all evolutionary events along the branches are DCJ operations. The next event is then chosen uniformly at random among all possible DCJ operations.

In the general model, an event can be a DCJ operation or one of a gene duplication, gene insertion, or gene loss. Thus we randomly sample three parameters for each branch: the probability of occurrence of a gene duplication, p_d, the probability of occurrence of a gene insertion, p_i and the probability of occurrence of a gene loss, p_l.

(The probability of occurrence of a DCJ operation is then just $p_r = 1 - p_d - p_i - p_l$.) The next evolutionary event is chosen randomly from the four categories according to these parameters. For gene duplication, we uniformly select a position to start duplicating a short segment of chromosomal material and place the new copy to a new position within the genome. We set $L_{\max}$ as the maximum number of genes in the duplicated segment and assume that the number of genes in that segment is a uniform random number between 1 and $L_{\max}$. In our simulations, we used $L_{\max} = 5$. For gene insertion, we tested two different possible scenarios, one for genomes of prokaryotic type and the other for genomes of eukaryotic type. For the former, we uniformly select one position and insert a new gene; for the latter, we uniformly select one existing gene and mutate it into a new gene. Finally, for gene loss, we uniformly select one gene and delete it.

Results for simulations under rearrangements

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

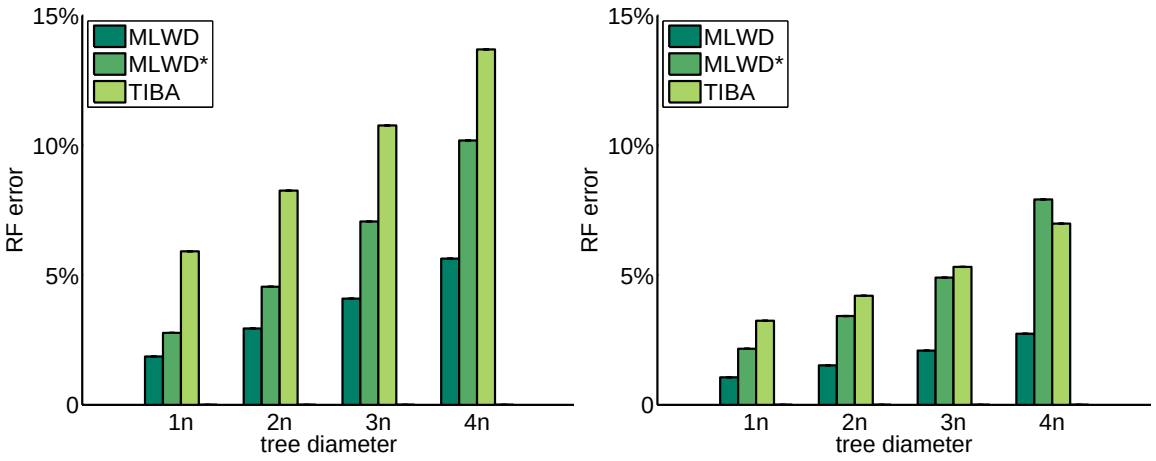


(a) 1,000 genes *(b) 5,000 genes*

Figure 3.4: RF error rates for different approaches for trees with 50 species, with genomes of $1,000$ and $5,000$ genes and tree diameters from one to four times the number of genes, under the rearrangement model.

settings. Our experiments are conducted by varying three main parameters: the number of taxa , the number of genes, and the target diameter. We used two values for each of the first two parameters: 50 and 100 taxa, and 1,000 and 5,000 genes. For the third parameter, the diameter of the tree, we varied it from n to $4n$, where n is the number of genes. For each setting of the parameters, we generated 100 datasets; data presented below are averages over these 100 datasets.

We compared the accuracy of three different approaches, `MLWD`, `MLWD*` and `TIBA`. `MLWD` (Maximum Likelihood on Whole-genome Data) is our new approach; `MLWD*` follows the same procedure as `MLWD`, but does not use our computation of transition probabilities—instead, it allows `RAxML` to estimate and set them; finally, `TIBA` is a fast distance-based tool to reconstruct phylogenies from rearrangement data [32], which combines a pairwise distance estimator [31] and the `FastME` [14] distance-based reconstruction method. We did not compare with the approaches of `MLBE` or `MPBE`, because they are too slow for these test cases. Figures 3.4 and 3.5 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 [43] error rates, the standard measure of error



(a) 1,000 genes *(b) 5,000 genes*

Figure 3.5: RF error rates for different approaches for trees with 100 species, with genomes of 1,000 and 5,000 genes and tree diameters from one to four times the number of genes, under the rearrangement model.

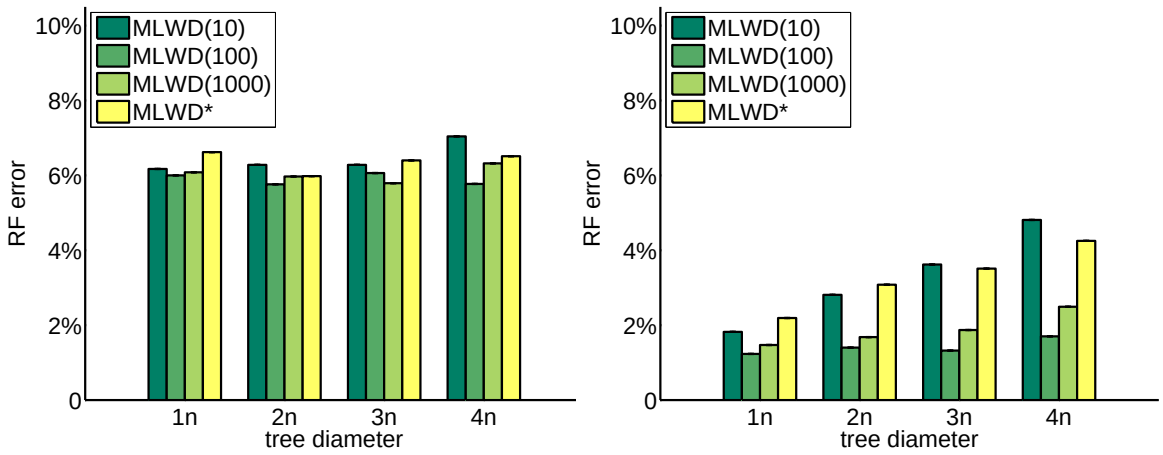
for phylogenetic trees—the RF rate expresses the percentage of edges in error, either because they are missing or because they are wrong.

These simulations show that our `MLWD` approach can reconstruct much more accurate phylogenies from rearrangement data than the distance-based approach `TIBA`, in line with experience in sequence-based reconstruction. `MLWD` also outperforms `MLWD`*, underlining the importance of estimating and setting the transition parameters before applying the sequence-based ML method.

Results for simulations under the general model

Here we generated more complex datasets than for the previous set of experiments. For example, among our simulated eukaryotic genomes, the largest genome has more than 20,000 genes, and the biggest gene family in a single genome has 42 members.

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 4.2. Thus we set different bias values and compare them under simulation results. If the transition probability of any gene or adjacency from 0 to 1 in `MLWD` is set to be m times less than that in the opposite direction, we name it `MLWD(`m`)` ($m = 10, 100, 1000$). Figure 3.6 summarizes the RF error rates. Whereas the best ratio in the rearrangement model was $2n$ (as derived in Section 4.2), the best ratio under the general model is much smaller. This difference can be attributed to the relatively modest change in gene content compared to the change in adjacencies: since we encode presence or absence of a gene, but not the number of copies of the gene, not only rearrangements, but also many duplication and loss events will not alter the encoded gene content.



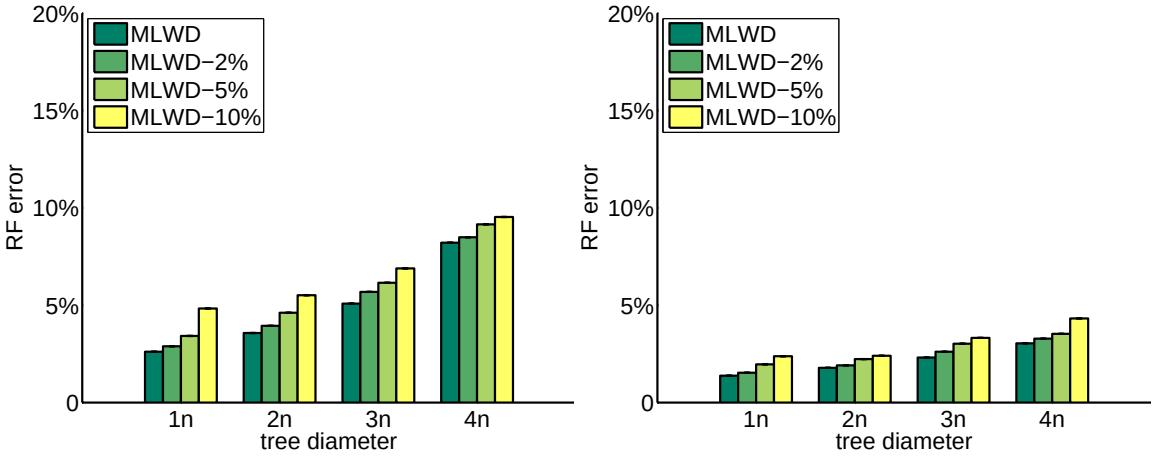
(a) 1,000 genes *(b) 5,000 genes*

Figure 3.6: RF error rates for different approaches for trees with 50 species, with initial genomes of size $1,000$ and $5,000$ and tree diameters from one to four times the number of genes in the initial genome, under the general model of evolution.

Results for simulated poor assemblies

High-throughput sequencing has made it possible to sequence many genomes, but the finishing steps—producing a good assembly from the sequence data—are time-consuming and may require much additional laboratory work. Thus many sequenced genomes remain broken into a number of contigs, thereby inducing a loss of adjacencies in the source data. In addition, some assemblies may have errors, thereby producing spurious adjacencies while losing others. We designed experiments to test the robustness of our approach in handling genomes with such assembly defects. We introduce artificial breakages in the leaf genomes by "losing" adjacencies, which correspondingly breaks chromosomes into multiple contigs. For example, `MLWD-`$x\%$ represents the cases of losing $x\%$ of adjacencies, that is, $x\%$ of the adjacencies are selected uniformly at random and discarded for each genome.

Figure 3.7 shows RF error rates for `MLWD` on different quality of genome assemblies under the rearrangement model. Our approach is relatively insensitive to the quality of assembly, especially when the tree diameter is large, that is, when it includes highly



(a) 1,000 genes *(b) 5,000 genes*

Figure 3.7: RF error rates for MLWD on different qualities of genome assemblies, for trees with 50 species, with genomes of size $1,000$ and $5,000$. with tree diameters from one to four times the number of genes, under the rearrangement model.

diverged taxa. Note that this finding was to be expected in view of the good results of our approach using an encoding that, as observed earlier, does not uniquely identify the ordering of the genes along the chromosomes.

3.6 Conclusion

In this chapter, we presented a set of three maximum likelihood methods (`MLBE`, `MLBE2` and `MLME`) as well as a simpler yet more powerful methods (`MLWD`) for gene order phylogenetic reconstructions. Our tests on simulated datasets show that all these methods are very accurate and scale well to accommodate large genomes. With a biased transition probability, `MLWD` needs not to further encode binary sequences and hence runs greatly faster than `MLBE`, making it a perfect tool for handling dozens of whole-genome data within an acceptable amount of time.

Chapter 4

Reconstruct Ancestors under Genome Rearrangement

4.1 Motivation

Overview

Evolutionary biologists have had a tradition in reconstructing genomes of extinct ancestral species. Mutations in a genomic sequence are made up not only at the level of base-pair changes but also by rearrangement operations on chromosomal structures such as inversions, transpositions, fissions and fusions [27]. Over the past few years, ancestral gene-order inference has brought profound predictions of protein functional shift and positive selection [40].

Current methods such as `GRAPPA` and `InferCARsPro` are capable to handle modern whole-genome data due to their intrinsic high complexity. And the performance between `InferCarsPro` and `GRAPPA` has been comprehensively investigated through simulation experiment. In general, the probabilistic method are less computational expensive by abandoning the NP-hard procedure and its performance in terms of the number of correct adjacencies recovered is largely comparable to the classic parsimony methods.

On the other hand, several heuristic parsimonious methodes are developed to mitigate the complexity. `SCJ` reduces the running time to polynomial, but fails to achieve comparable performance and `GASTS` employs a heuristic median solver to score a tree. In this study, we propose a new probabilistic method called `PMAG` to reconstruct

ancestral genomic orders given a phylogeny. We conducted extensive experiments to evaluate the performance of `PMAG` with other existing methods. According to the results, `PMAG` can outperform all the other methods under study and still run at least hundreds of times faster than `GRAPPA` and `InferCARsPro`. Although the time expense of `SCJ` was negligible, it recovered much less correct adjacencies than any other method even in the easy test.

Reconstructing the ancestral gene order in probabilistic frameworks

The probabilistic approach `InferCARsPro` proposed by Ma is based on Bayes' theorem such that every possible predecessor and successor of a signed gene i denoted as X_i in the ancestral genome x, given D_x representing the observed data, can be expressed as

$$P(X_i \ in \ x|D_x) = \frac{P(D_x|X_i \ in \ x)P(X_i \ in \ x)}{\sum_{j=1}^{q} P(D_x|X_j \ in \ x)P(X_j \ in \ x)} = \frac{P(D_x|X_i \ in \ x)}{\sum_{j=1}^{q} P(D_x|X_j \ in \ x)}$$

where priors are assumed equal and the likelihood $P(D_x|X_i \ in \ x)$ can be calculated recursively in a post-order traversal fashion summed over q possible configurations. Its transition matrix is defined as an extension of the Jukes-Cantor model such that probability of transition from any character to any different character is always equal.

Let $s_x(\cdot)$ denote the successor of a gene and $p_x(\cdot)$ denote the predecessor of a gene, an adjacency pair $A_x(i, j)$ can be viewed as $s_x(i) = j$ and $p_x(j) = i$ simultaneously. After finishing the calculation of conditional probabilities for every successor and predecessor relationships, the conditional probability of an adjacency $A_x(i, j)$ in genome x can approximated as

$$P(A_x(i, j)|D_x) = P(p_x(j) = i|D_x) \times P(s_x(i) = j|D_x)$$

Finally a fast greedy algorithm is adopted to connect adjacencies into contiguous ancestral regions. Although `InferCARsPro` showed good results and speedup over

parsimonious methods, it is still too slow and inaccurate when dealing with even small number of distant genomes.

We investigated the following intrinsic characteristics of `InferCARsPro` that account for its difficulties in handling complex datasets, which in turn motivated us to propose our new method.

- `InferCARsPro` uses a neutral model accounting for all changes of adjacencies, however biased model for phylogeny reconstruction has been successfully applied for genome rearrangement scenarios [30].
- The total number of states for each gene is exactly equal to $2 \times n - 2$ where n is the number of genes. Thus computing the likelihood score on such excessive number of states clearly incurs huge computational burden.
- The conditional probability of an adjacency is approximated from the predecessor and successor relations. Although such approximation is intuitive, it is more desirable to directly calculate the conditional probability of an adjacency.
- `InferCARsPro` tends to produce an excessive number of chromosomes. To achieve better assembly, `GapAdj` scarifies considerable accuracy.
- `InferCARsPro` requires branch lengths of a given phylogeny as part of its inputs, but it is not always handy to obtain in practice.

4.2 Algorithm Detail

Given the topology of a model tree and a collection of gene orders at the leaves, our approach first encodes the gene orders into binary sequences and estimates the parameters in the transition model for adjacency changes. Ancestral nodes in the model tree are inferred independently and in each inference, we reroot the model tree to have the target ancestor as the root of a new tree. Then we utilize a probabilistic

inference tool to compute the conditional probabilities of all the adjacencies encoded in the binary sequence of the target ancestor. At last we use a greedy algorithm as used in Ma's work to connect the adjacencies into contiguous regions. We call our new approach *Probabilistic Method of Ancestral Genomics (PMAG)*.

Encoding gene orders into binary sequences

A gene order can be expressed as a sequence of adjacency information that specifies presence or absence of all the adjacencies [25, 30]. Denote the head of a gene i by i^h and its tail by i^t. We refer $+i$ as an indication of direction from head to tail $(i^h \rightarrow i^t)$ and otherwise $-i$ as $(i^t \rightarrow i^h)$. 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 at the first or last place of a linear chromosome, then we have a corresponding singleton set, $\{c^t\}$ or $\{c^h\}$, called a *telomere*. A genome can then be expressed as a multiset of adjacencies and telomeres. For instance, a linear chromosome consists of four genes, (+1,+2,-3,-4,) can be represented by the multiset of adjacencies and telomeres $\{\{1^h\}, \{1^t, 2^h\}, \{2^t, 3^t\}, \{3^h, 4^t\}, \{4^h\}\}$. 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. Table 4.1 shows an example of encoding two artificial genomes $G_1 : (1, 2, -3)$ and $G_2 : (3, -2, 1)$ into binary sequences.

Table 4.1: Example of encoding gene orders into binary sequences

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

Given a dataset D with m species and each of n genes, let k indicate the total number of linear chromosomes in D, then there are up to $\binom{2n+2}{2}$ distinct adjacencies and telomeres. However in reality if the length of the binary sequences extracted

from D is l, then l is typically far smaller. In fact, in the extreme case when genomes in D share no adjacency and telomere, l equals at most to $n \times m + k$, and since m and k are commonly much smaller than n, thus the length of the binary sequences for a dataset is usually linear rather than quadratic to the number of genes.

Estimating transition parameters

Since we are handling binary sequences with two characters, we use a general time-reversible framework to simulate the transitions from presence (1) to absence (0) and vice versa. Thus the rate matrix is

$$Q = \{q_{ij}\} = \begin{bmatrix} \cdot & a \\ a & \cdot \end{bmatrix} \begin{bmatrix} \pi_0 & 0 \\ 0 & \pi_1 \end{bmatrix}$$

The matrix involves 3 parameters: the relative rate a, and two frequencies π_0 and π_1.

Several models have been proposed to probabilistically characterize the changes of gene adjacencies by common types of rearrangement operations such as inversion, transposition as well as DCJ [47]. In this study, we use the model that has been successfully applied for phylogeny reconstruction in the context of genome rearrangement as suggested in [30]. In particular, every DCJ operation breaks two random adjacencies uniformly chosen from the gene-order string and subsequently creates two new ones. Since each genome contains $n + O(1)$ adjacencies and telomeres where n is the gene number and $O(1)$ equals to the number of linear chromosomes in the genome, thus the probability that an adjacency changes from presence (1) to absence (0) in the sequence is $\frac{2}{n+O(1)}$ under one operation. Since there are up to $\binom{2n+2}{2}$ possible adjacencies and telomeres, the probability for an adjacency changing from absence (0) to presence (1) is $\frac{2}{2n^2+O(n)}$. Therefore we come to the conclusion that the transition from 1 to 0 is roughly $2n$ times more likely than that from 0 to 1.

Inferring the probabilities of ancestral adjacencies for the root node

In principle, our probabilistic inference is categorized as marginal reconstruction which assigns characters to a single ancestral genome at a time. Once we have the tree topology and binary sequences encoding the input gene orders, we use the extended probabilistic approach for sequence data described by Yang [68] to infer the ancestral gene orders at the root node. In the binary sequences, each site represents an adjacency with character either 0 (absence) or 1 (presence) and for each site we seek to calculate the conditional probability of observing that adjacency. As the true branch lengths are not available, we take advantage of the widely-used maximum-likelihood estimation from the binary sequences at the leaves to estimate the branch length.

Suppose x is the root of a model tree, then the conditional probability that node x has the character s_x at the site, given D_x representing the observed data at the site in all leaves of the subtree rooted at x, is

$$P(s_x|D_x) = \frac{P(s_x)P(D_x|s_x)}{P(D_x)} = \frac{\pi_{s_x} L_x(s_x)}{\sum_{s_x} \pi_{s_x} L_x(s_x)}$$

where π_{s_x} is the character frequency for s_x. The conditional probability in the form of $L_x(s_x)$ is defined as the probability of observing the leaves that belong to the subtree rooted at x, given that the character at node x is s_x. It can be calculated recursively in a post-order traversal fashion suggested by Felsenstein [19] as:

$$L_x(s_x) = \begin{cases} 1 & \text{if } x \text{ is a leaf with character} = s_x \text{ at the site} \\ 0 & \text{if } x \text{ is a leaf with character} \neq s_x \text{ at the site} \\ \left[\sum_{s_f} p_{s_x s_f}(t_f) L_f(s_f)\right] \times \left[\sum_{s_g} p_{s_x s_g}(t_g) L_g(s_g)\right] & \text{otherwise} \end{cases}$$

where f and g are the two direct descendants of x. $p_{ij}(t)$ defines the transition probability that character i changes to j after an evolutionary distance t. Following

the deduction of transition probability in [19], our transition-probability matrix can be written as

$$p_{ij}(t) = \pi_j + e^{-t}(\delta_{ij} - \pi_j)$$

Here the δ_{ij} is 1 if $i = j$, otherwise δ_{ij} is 0. In order to set up the $2n$ ratio, we simply set the rate a to 1 and add a direct assignment of the two frequencies in the code. For instance, if the character frequencies are $\pi_0 = 0.1$ and $\pi_1 = 0.9$, then the rate of 0 to 1 transitions is 10 times as high as the rate of transitions in the other direction under the same evolutionary distance.

`RAxML` [52, 51] is one of the most widely used program for sequence-data analysis which implements the method for ancestral sequence inference developed by Yang [68]. In this study, we modified `RAxML` to infer the conditional probabilities of gene adjacencies at all sites.

Assembling gene adjacencies into chromosomes

Once we obtain the conditional probability of every adjacency for the target ancestor x, we can construct an adjacency graph for x in which each gene i corresponds to its head and tail, i^h and i^t, and each adjacency is connected by an edge with weight equals to the conditional probability of seeing that adjacency in x. A telomere is viewed as an edge between the gene and a cap, and each cap is expressed by a unique vertex in the graph, representing the edge of a chromosome. Finally for the adjacencies absent in the binary encoding, their edges are given the infinite weight so they will be excluded from further consideration. The problem of searching the longest path in such a fully connected graph by visiting each gene's head and tail exactly once is indeed an instance of symmetric TSP as shown initially in Tang and Wang's study [59]. In our method, we adopt this approach and utilize one of the most popular TSP solver `Concorde` [2] to find the optimal path with maximal probability.

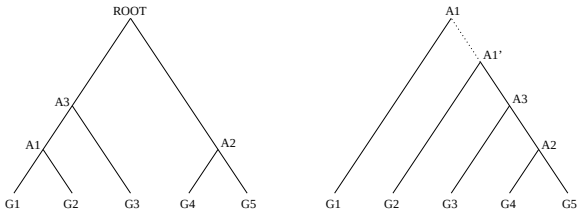


Figure 4.1: Rerooting the phylogeny tree from the original root to the ancestral node under inference which is $A1$ in this case. Auxiliary node A' is added to preserve its binary structure.

In the solution path, multiple contiguous caps are shrank into a single one and a gene segment between two caps is taken as a chromosome.

Rerooting the tree topology

To infer the genomic order of a non-root ancestral node x, if x is taken as the root of the tree such that only the leaves in the subtree of x are considered into the recursive calculation of likelihood, potentially many good adjacencies in the outgroup of the subtree will be neglected and result in a loss of information. To minimize the influence, we incorporate the technique of rerooting so that original tree is rearranged and the target node x becomes the root of a new tree. As a standard procedure, rerooting has already found use for ancestral genome reconstruction [33]. We use figure 4.1 to demonstrate the rerooting procedure for genome $A1$ by adding an auxiliary node $A1'$. The branch length between $A1$ and $A1'$ (dashed edge) is always 0.

4.3 A Quantitative Example

We have introduced the algorithm of `PMAG` for ancestral genome reconstruction from equal gene content. In this section, we give a detailed example about how to computer the posterior probability for a character state. Here we use a tree of three leaves to compute the most likely state of the character in the root, as shown in figure 4.2. For convenience, we adopt the F80 model and set the branches $t_1 = t_2 = t_3 = 0.1$ and

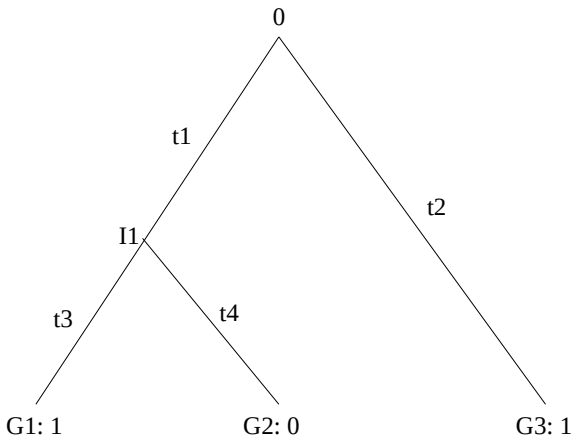


Figure 4.2: A phylogenetic topology of three genomes. The 0 or 1 following the leaf label indicate a gene adjacency is in absence or presence state.

$t_4 = 0.3$. The transition probability matrix of F80 model can be calculated by the following equations:

$$P_{ij}(t) = \begin{cases} \pi_i + (1 - \pi_i) \times e^{-\beta t} & \text{if } i = j \\ \pi_j \times (1 - e^{-\beta t}) & \text{if } i \neq j \end{cases} \tag{4.1}$$

$$\beta = \frac{1}{(1 - \pi_A^2 - \pi_C^2 - \pi_G^2 - \pi_T^2)} \tag{4.2}$$

By substituting the two branch lengths and base compositions π of $(0.33, 0.67)$ for 0 and 1, two matrices of transition probability with order 0 and 1 are:

$$P(0.1) = \begin{pmatrix} 0.865 & 0.135 \\ 0.066 & 0.934 \end{pmatrix} \tag{4.3}$$

$$P(0.2) = \begin{pmatrix} 0.757 & 0.243 \\ 0.120 & 0.880 \end{pmatrix} \tag{4.4}$$

Likelihood score can be computed by a recursive function suggested by Felsenstein [19]:

$$L_x(s_x) = \begin{cases} 1 & \text{if } x \text{ is a leaf with character} = s_x \text{ at the site} \\ 0 & \text{if } x \text{ is a leaf with character} \neq s_x \text{ at the site} \\ \left[\sum_{s_f} p_{s_x s_f}(t_f) L_f(s_f)\right] \times \left[\sum_{s_g} p_{s_x s_g}(t_g) L_g(s_g)\right] & \text{otherwise} \end{cases} \tag{4.5}$$

If we consider internal node $I1$ with two direct descendants $G1$ and $G2$, the first entry of the likelihood vector for $I1$ is $L_{I1}(0) = p_{01}(0.1) \times p_{11}(0.2) = 0.135 \times 0.880 = 0.119$.

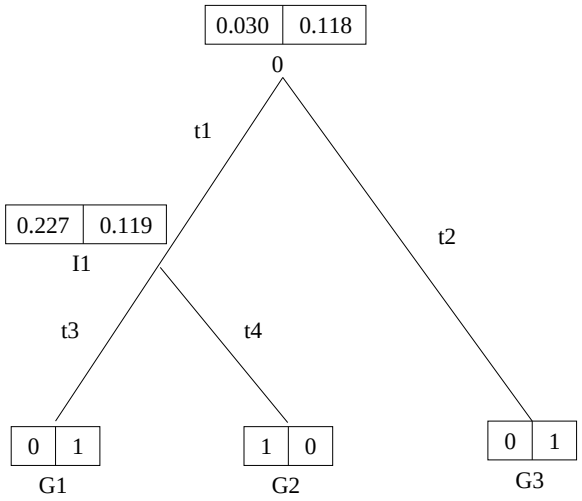


Figure 4.3: Likelihood score for each character (0 on the left) at each internal node is calculated based on the empirical based probabilities and F80 transition model.

This is the likelihood of observing 0 and 1 at leaves $G1$ and $G2$, given that node $I1$ has 0. Samewise $L_{I1}(1), L_0(0), L_0(1)$ can be computed from bottom-up. Figure 4.3 shows the likelihood scores we have computed for every internal node. Then the posterior probabilities of the character state s_x at root is computed based on the Bayes' theorem:

$$P(s_x|D_x) = \frac{P(s_x)P(D_x|s_x)}{P(D_x)} = \frac{\pi_{s_x} L_x(s_x)}{\sum_{s_x} \pi_{s_x} L_x(s_x)} \tag{4.6}$$

At root, the denominator is the probability of data at the site which can be calculated by $\pi_0 L_0(0) + \pi_1 L_0(1) = 0.33 \times 0.030 + 0.67 \times 0.118 = 0.08896$. Therefore the posterior probability at root are 0.11 and 0.89 for character states 0 and 1 respectively. So 1 is the most probable state at the root with posterior probability 0.89. In another word, the probability of observing such adjacency at the root is 0.89.

4.4 Experimental Results

Experimental design

Since actual ancestors are rarely known for sure, it is difficulty to evaluate ancestral reconstruction methods with real datasets. In order to carry out a complete evalua-

tion over a group of methods under a wide range of configurations, we conducted a collection of simulation experiments following the standard steps of such tests that have been extensively adopted in genome rearrangement studies [26, 30].

In particular, a group of tree topologies were first generated with edge lengths representing the expected number of evolutionary operations. An initial gene order was assigned at the root so it can evolve down to the leaves following the tree topology mimicking the natural process of evolution, by carrying out a number of predefined evolutionary events. In this way, we obtained the complete evolutionary history of the model tree and the whole set of genomes it has.

Normally we utilized the simulator proposed by Lin *et al.* [32] to produce birth-death tree topologies. Since `SCJ` has its own simulator for SPP, we therefore used that simulator for a fair comparison in the tests involving `SCJ`. With a model tree, we were able to produce genomes of any size and difficulty by simply adjusting four main parameters: the number of genomes m, the number of chromosome c, the number of genes n, and the tree diameter d (equivalently branch length l in `SCJ`'s simulator). To closely mimic the rearrangement scenarios in bacterial genomes with multi-chromosomes, we generated datasets with 10 genomes, each with 500 genes and 5 chromosomes. Along each branch, we performed 80% random inversions and 20% random translocations to account for intra- and inter-chromosomal rearrangements respectively.

Predicted ancestral genomes produced from a method were evaluated by three measurements. We first calculated the `adjacency accuracy` C computed as the total number of correctly inferred adjacencies (i.e. those also appear in the true ancestral genomes) divided by the total number of adjacencies in both true genome and predicted genome. In particular, if S represents the set of gene adjacencies in the real genome and S' the predicted genome.

$$C = \frac{|S \cap S'|}{|S \cup S'|} \times 100\%$$

Second, we calculated `distance accuracy` D defined as the DCJ distance between a predicted ancestor and its corresponding true genome. Apparently for genome rearrangement study, `distance accuracy` is more appropriate as it not only considers the adjacency changes, but also takes differences in genome structures into account. Finally, to assess the assembly capabilities, we computed `assembly accuracy` A as the absolute differences of the number of chromosomes between a predicted ancestor and its corresponding truth. For each dataset, the average of each measurement across all ancestors were computed and for each tree diameter, we produced 10 datasets and reported their average, as well as their standard deviation.

Assessing the impact of the biased transition model and reroot procedure

The underlying structure of `PMAG` is the probabilistic framework of inferring the conditional probability of observing a gene adjacency in the target genome using the Bayes' theorem. Moreover we enhance the general framework with a transition model and a reroot procedure. In this section, we show how the transition model and the reroot procedure can respectively influence the performance of `PMAG` by comparing `PMAG` to its three variants created as follows:

- `Naive` : The naive version of `PMAG` with neutral model of adjacency changes and fixed tree topology for all ancestral nodes.
- `Naive+Model` : `Naive` method cooperating with the biased transition model.
- `Naive+Reroot` : `Naive` method cooperating with the reroot procedure.

Figure 4.4 summarizes the comparison among the four methods under tree diameters from $0.5n$ (easy case) to $3n$ (very difficult). In general, higher tree diameter effectively increased the difficulties and hence reduced the portion of correct adjacencies

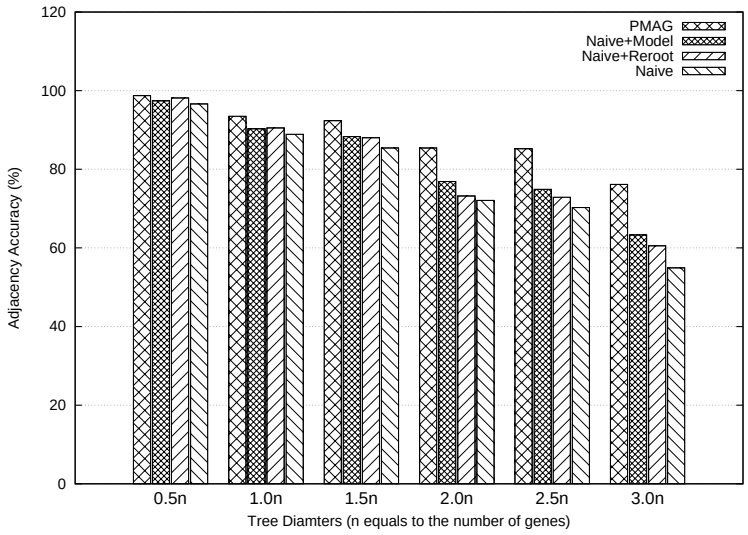


Figure 4.4: Comparison of adjacency accuracy between `PMAG` and its three premature versions. Datasets is produced with 10 genomes, each with 5 chromosomes and a total of 500 genes. X-axis represents the tree diameters from 0.5 to 3 times the number of genes.

all methods can recover. Unsurprisingly the `Naive` method is the least accurate in all cases and both `Naive+Model` and `Naive+Reroot` can independently enhance the accuracy of `Naive` method. By incorporating both mechanisms, `PMAG` not only inherits both improvements but obtains additional improvements as well, suggesting the transition model and reroot procedure are useful and indispensable for our method.

Evaluation of PMAG against other methods

In this section, we picked three main competitors from both event-based and adjacency-based methods, and compared them with `PMAG` using the same datasets. In particular we supplied `InferCARsPro` with multi-chromosomal genomic distances as its branch lengths computed by `GRIMM` [61]. And in `GapAdj`, the cutoff value and maximal iterations were set to 0.6 and 25 as suggested by the authors. The event-based method `GASTS` was simply ran by providing the tree and dataset.

Figure 4.5 shows the comparison of `adjacency accuracies`. When the tree diameters were small ($0.5n$ and $1n$), all methods were able to consistently produce

highly accurate ancestral genomes ($> 90\%$) and the differences among methods were not significant. In particular, `GASTS` was the most accurate method, while the performances of `PMAG` and `InferCARsPro` were similar and both were better than `GapAdj`. As the tree diameters went larger, `GASTS` quickly became unreliable which is coincided with the experimental results reported in the study of `GASTS`. In all tests, `PMAG` showed great robustness against disturbance and achieved the highest accuracy when the tree diameter is greater than $1n$.

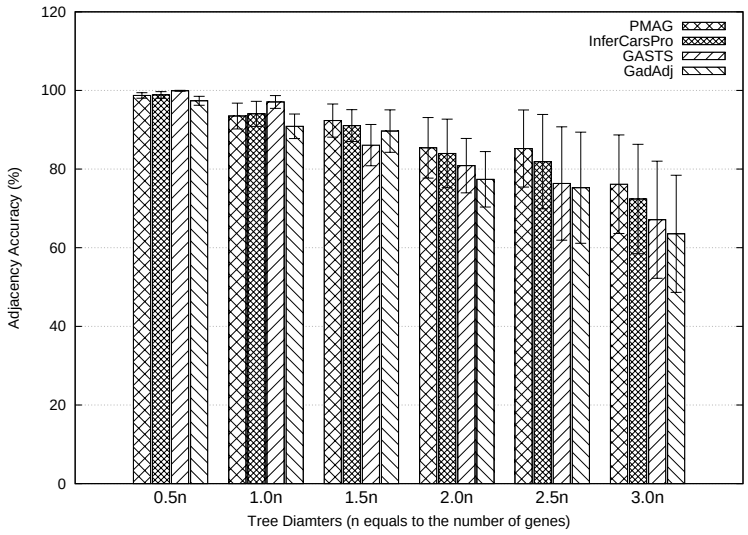


Figure 4.5: Comparison of adjacency accuracy between `PMAG`, `InferCARsPro`, `GASTS` and `GapAdj`. Use the same datasets as used in figure 4.4. Standard deviations are given at the top of bars. X-axis represents the tree diameters from 0.5 to 3 times the number of genes.

Figure 4.6 shows the comparison of `distance accuracies`. Generally from the figure, `distance accuracies` are highly correlated with `adjacency accuracies` except for a couple of cases. Interestingly, at $1.0n$ diameter, `PMAG` showed less `adjacency accuracy` than `InferCARsPro`, but when measuring `distance accuracy`, `PMAG` achieved better results, showing its ability in preserving good genome structures.

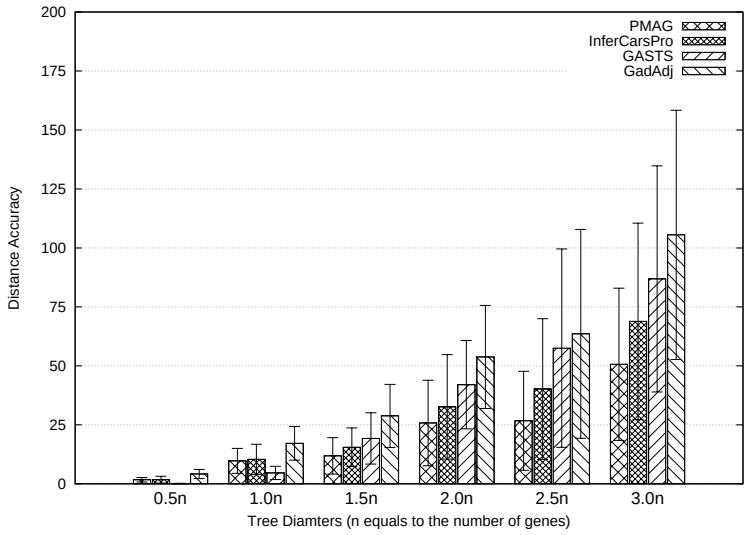


Figure 4.6: Comparison of distance accuracy between `PMAG`, `InferCARsPro`, `GASTS` and `GapAdj`. Use the same datasets as used in figure 4.4. Standard deviations are given at the top of bars. X-axis represents the tree diameters from 0.5 to 3 times the number of genes.

Evaluation of PMAG against SCJ

Simulator embedded in the `SCJ` program was used and the measurement of difficulty became the branch length l, denoting the expected number of evolutionary events along an edge of the tree which is sampled from an uniform distribution on the set $\{1, 2, 3...d\}$, where d equals to $l \times n$ and n is the number of genes. As before, those events were consisted by 80% of inversions and 20% translocations. Since `SCJ` and `PMAG` are both fast enough, we therefore generated a set of larger dataset containing 32 genomes, each with 5 chromosomes and a total of 2,000 genes.

Figures 4.7 and 4.8 demonstrates the `adjacency accuracy` and the `distance accuracy` of `PMAG` and `SCJ` respectively. These figures clearly demanstrate that `PMAG` can significantly outperform `SCJ` in all test cases.

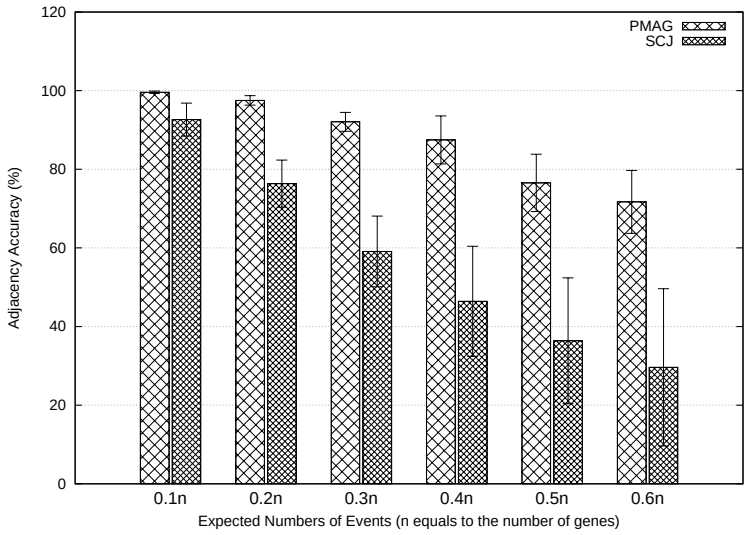


Figure 4.7: Comparison of adjacency accuracy between `PMAG` and `SCJ`. Datasets were produced by the simulator provided in SCJ program that contain 32 genomes, each with 5 chromosomes and a total of 2,000 genes. X-axis represents the expected number of events from 0.1 to 0.6 times the number of genes.

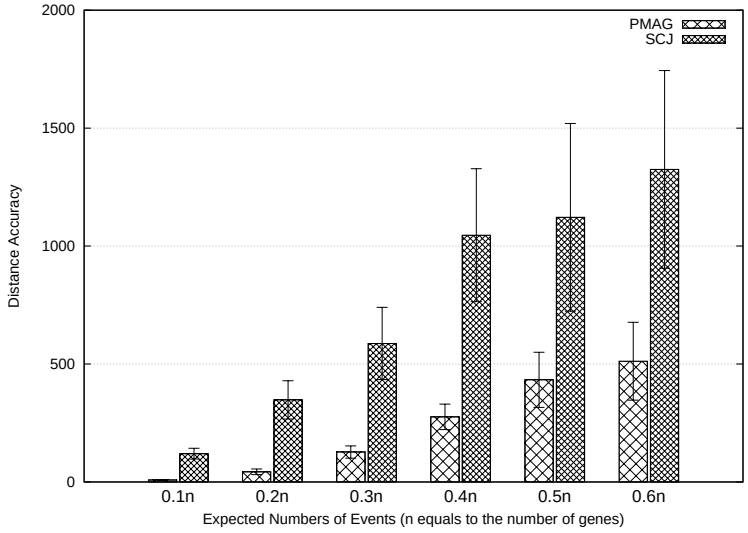


Figure 4.8: Comparison of distance accuracy between `PMAG` and `SCJ`. Use the same datasets as used in figure 4.7. Standard deviations are given at the top of bars. X-axis represents the expected number of events from 0.1 to 0.6 times the number of genes.

Comparison of performances on assembly

The final step of adjacency-based methods often involves assembly of adjacencies into contiguous segments. These segments can be viewed as chromosomes or more

precisely contigs. Previous methods `InferCARsPro` employing a greedy algorithm for assembly often ends up with an excessive number of contigs. Later the assembly accuracy was improved by `GapAdj` using the concept of gapped adjacencies with a sacrifice of adjacency accuracy.

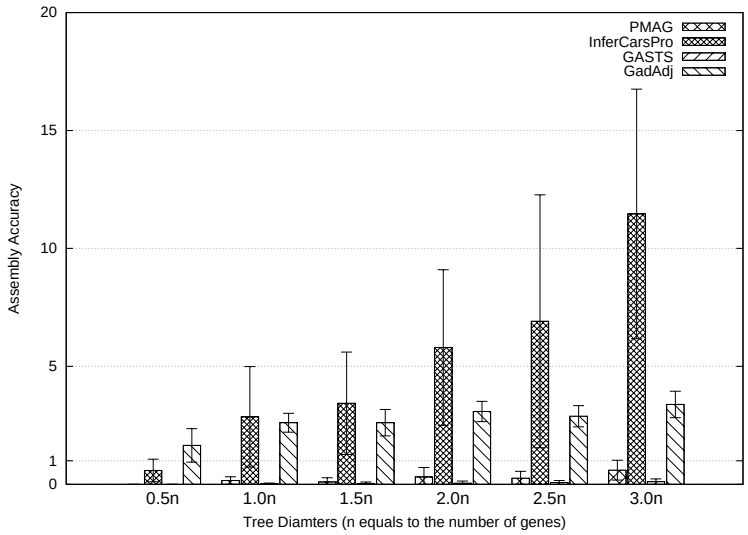


Figure 4.9: Comparison of assembly accuracy between `PMAG`, `InferCARsPro`, `GASTS` and `GapAdj`. Assembly accuracies were summarized from the test results as shown in figure 4.5. X-axis represents the tree diameters from 0.5 to 3 times the number of genes.

We summarized the number of contigs produced by each methods and computed the average of assembly accuracy in each case, as demonstrated in Figures 4.9 and 4.10. From the figures, the event-based method `GASTS` without the need for assembly produced the most relevant number of contigs in all cases. Among the adjacency-based methods, `PMAG` showed much better assembly performance and in fact its performance was very close to `GASTS`. Compared with `SCJ` (figure 4.10), `PMAG` yielded very accurate amount of contigs (unobservable from the figure); however since `SCJ` is overly conservative, it missed a large portion of true adjacencies and produced a massive amount of contigs.

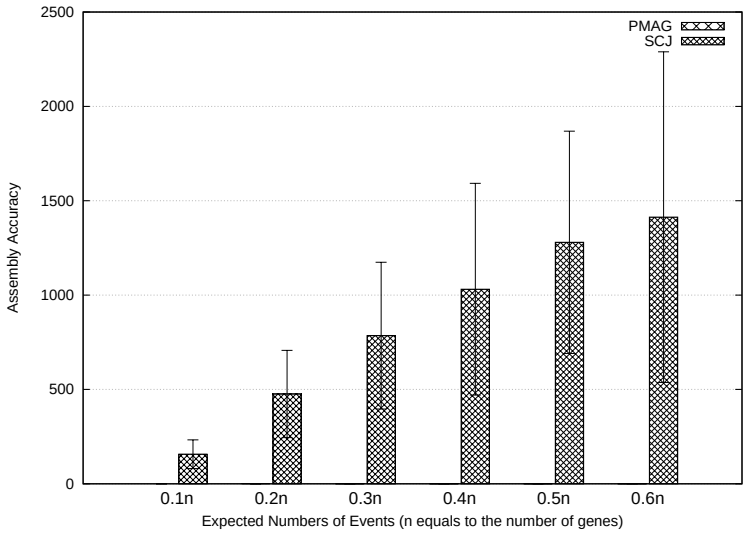


Figure 4.10: Comparison of assembly accuracy between `PMAG` and `SCJ`. Assembly accuracies were summarized from the test results as shown in figure 4.7. X-axis represents the expected number of events from 0.1 to 0.6 times the number of genes.

Time efficiency

All tests were conducted on a workstation with 2.4Ghz CPUs and 4 GB RAM. We summarized the time consumption of various methods in Table 4.2.

Table 4.2: Comparison of average time cost between four methods in seconds (n equals to the number of genes)

Tree Diameter	PMAG	InferCARsPro	GapAdj	GASTS	SCJ
0.1n	21	660	5	42	1
0.2n	39	1250	6	54	1
0.3n	29	1480	12	52	1
0,4n	36	2460	18	60	1
0.5n	36	2760	22	82	1
0.6n	39	3480	45	120	1

From the table, `SCJ` was unsurprisingly the fastest of all, while `PMAG` with an exact solution of TSP achieved running time within an acceptable level. Unlike other methods, the difficulties of datasets put minor effect on the running time of `PMAG`, profiting from the binary encoding which allows us to take only a small portion of

adjacencies into computation.

4.5 Conclusion

In this study, we introduced the adjacency-based method `PMAG` in the probabilistic framework for ancestral gene-order inference. `PMAG` determines the state of each adjacency in the binary encoding to be either present or absent in an ancestral genome according to its conditional probability. Ancestral genomes are finally retrieved by connecting individual adjacencies into continuous regions by using an exact TSP solver. Experimental results reveal that `PMAG` can not only accurately infer ancestral genomes, and also does a good job in assembling adjacencies into valid genomes. The running speed of `PMAG` is fast and also stable spanning a wide range of simulating settings.

Chapter 5

Reconstruct Ancestors under a Flexible Model

5.1 Motivation

In chapter 4, we described the method `PMAG` [17] which overcame several major issues found in `InferCarsPro`. Though simulation study, `PMAG` was faster and more accurate than the other competitors in most of the scenarios.

However `PMAG` is unable to handle datasets with unequal gene contents. From modern perspective of view, only gene loss, gene insertion and gene duplication can modify the gene content of a genome. The difficulties of handling these events in the framework of `PMAG` are obvious: in the presence of gene indels and duplication, without knowing the gene content of the ancestor genome under inference, it would be impossible to construct an appropriate adjacency graph, as the number of nodes we should place in the graph is not determined. However if we produce ancestral genomes from the adjacency graph constructed from all possible genes as the way in `PMAG`, the inferred genomes will always include the entire set of genes which in this case is simply wrong.

On the other hand, in the past few years, several new studies and methods were published especially to process datasets with unequal gene contents [34, 3, 20]. Among them, the most recent method `GapAdj` [20] showed good result. In particular `GapAdj` utilizes a natural process [22] to infer ancestral gene contents and uses such content information to construct an adequate adjacency graph for gene assembly.

Therefore we extended our previous method `PMAG` and developed `PMAG`$^+$ in order to

efficiently handle datasets underwent a large scale of rearrangements, as well as gene deletions and insertions (indels) of a single or a segments of genes. Our experimental results with `GapAdj` on simulated datasets suggest that `PMAG`$^+$ can efficiently and accurately predict both ancestral gene contents and ancestral gene orders.

5.2 Algorithm Details

Given a phylogeny, `PMAG`$^+$ computes the gene content and ordering of ancestral (internal) nodes one at a time. Prior to the inference of a target ancestral node, we reroot the given phylogeny tree to the node such that it becomes the root of the new tree. The underlying rationale is that the calculation of probabilities follows a bottom-up manner and only the species in the subtree of the target node are considered, therefore rerooting can prevent loss of information.

After rerooting, `PMAG`$^+$ proceeds the following three steps: 1) inferring the gene content of target node to determine which genes should appear; 2) computing the probabilities of gene adjacencies; 3) forming and solving a TSP problem to place genes on chromosomes. The following subsections describe these steps in detail.

Inference of Ancestral Gene Contents

The very first step of ancestral reconstruction often involves explicitly estimating gene content in ancestral nodes, using content information from leaves. A number of approaches have been developed and most of them are similar in spirit to the Fitch-Hartigan parsimony algorithm [58, 22, 28]. Figure 5.1 demonstrates a phylogeny tree with three leaf genomes involving insertion and deletion operations long the branches down from the root.

For pure rearrangements, every gene observed in leaf species should also be present in all ancestors; however in the presence of gene indels, such correspondence does not hold anymore and a gene can be either present or absent in an ancestor. Therefore

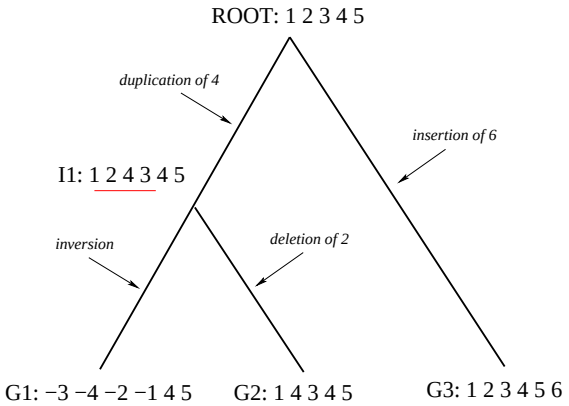


Figure 5.1: A phylogenetic tree with all genomes labeled. Its evolutionary history involves inversion, insertion and deletion.

our inference of ancestral contents relies on viewing genes as independent characters (with binary states); we can then determine the state for every gene in the ancestor. The first step involves encoding the gene contents of leaf species into binary sequences. In particular, suppose a dataset G with N species is given and a set of n distinct genes $S = \{g_1, g_2, ..., g_n\}$ is identified from G. For each leaf species G_i, its gene content $S_i = \{g_{i_1}, ..., g_{i_k}\}$ with $k \leq n$ can be equivalently represented by a sequence $\pi_i = \{\pi_{i_1}, \pi_{i_2}, ..., \pi_{i_n}\}$ in which each element has two states; if $g_j \in S_i$, $\pi_{i_j} = 1$, otherwise $\pi_{i_j} = 0$ for all j $(1 \leq j \leq n)$. For instance (table 5.1), a total of five distinct genes $\{a, b, c, d, e\}$ can be identified from two toy species $G1$ and $G2$ with gene orders $(+a, -c, +d)$ and $(+b, +a, -e)$ respectively. Note that for methods limited to equal gene content, gene a is the only informative gene as it appears in both species.

Table 5.1: Example of binary encoding on gene content.

	a	b	c	d	e
G_1	1	0	1	1	0
G_2	1	1	0	0	1

Many methods are available to infer ancestral states from binary characters, including `RAxML` [52] for maximum likelihood and `PAUP*` [56]. In this study, we chose `RAxML` (version 7.2.8 was used to produce the results given in this study) to conduct

the inference of states. Once the probabilities of presence state, $P = \{p_1, p_2, ..., p_n\}$, for the root node are computed, the gene i belongs to the gene content of root S_{root} if $p_i \geq 0.5$, otherwise, gene i is not in S_{root}. Following this paradigm, gene contents for all ancestral nodes can be separately inferred from leaf species. Our simulation shows that this approach can estimate gene contents with less than 1% error even for very difficult datasets.

Inference the Probabilities of Ancestral Gene Adjacencies

In [17], we have presented an adjacency-based method in probabilistic framework called `PMAG` to calculate the probability of observing an adjacency in the target ancestral node. The method proceeds in the following three main steps.

Step 1 Each species in the dataset is screened to identify all unique gene adjacencies and telomeres. By viewing each adjacency and telomere as an independent character with binary states—presence or absence, gene orders of species can be rigorously encoded into aligned sequences of binary characters.

Step 2 The phylogeny tree is rerooted to the target ancestral node in order to take all leaf species into consideration. At the same time, the $2n$ ratio for base compositions is setup such that the rate of presence to absence transitions is roughly $2n$ times as high as the rate of transitions in the other direction under the same evolutionary distance, where n is equal to the number of genes. Such model has been successfully used for phylogeny reconstruction [30].

Step 3 The probabilities of characters states for all gene adjacencies and telomeres at the root node are computed . The marginal ancestral reconstruction approach suggested by Yang [68] for molecular data was adopted and extended to compute for t

PMAG+ reuses the three steps as described to calculate probabilities for adjacencies and telomeres. Once these probabilities are obtained, it then uses the following step to connect gene adjacencies and telomeres into contigs, from which the ancestral gene ordering can be identified.

Assembling Ancestral Adjacencies into Ancestral Gene Orders

The last step is to assemble gene adjacencies and telomere into a valid gene order, with respect to the gene content inferred from the first step. In general, higher probability of presence state implies an adjacency or telomere should be more likely to be included in the ancestor; however the decision on choosing an adjacency or telomere cannot be solely made upon its own probability as each gene can only be selected once. In InferCarsPro, ancestral adjacencies are assembled by the greedy heuristic based on the adjacency graph. 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 exact solver. First, the greedy heuristic can achieve good approximation only when the dataset 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 ends.

Obtaining gene orders from (conflict) adjacencies can be transformed into an instance of symmetric Traveling Salesman Problem (TSP), as shown in [20, 59]. In this case, we can transform genes into cities and adjacency probabilities into edge weights in the TSP graph. In particular, suppose for the target ancestral node I, we have identified a set of m adjacencies $A = \{a_1, a_2, ..., a_m\}$ and n telomeres $T =$

$\{t_1, t_2, ..., t_n\}$ from leaf species. If the gene content of I has been inferred as $S_I = \{g_1, g_2, ..., g_k\}$ and the probabilities $P = \{p_{a_1}, ..., p_{a_m}, p_{t_1}, ..., p_{t_n}\}$ for each adjacency and telomere are known, we can create the TSP graph G as follows:

1. Each gene $g \in S_I$ is represented by two vertices—its head and tail, denoted as g^h and g^t respectively. Every extremity in the telomere $t \in T$ is represented by a unique vertex e_i, where $1 \leq i \leq n$. In this way, the total number of vertices in the graph is equal to $2 \times m + n$.

2. Edges between all pairs of head and tail of the same gene (g^h, g^t) are added with $-\inf$ to guarantee this connection is present in the solution. Edges are also established with $-\inf$ for all pairs of extremities (e_i, e_j) where $i \neq j$ and $1 \leq i, j \leq n$.

3. For every adjacency $(f, g) \in A$, the corresponding edge is added to G connecting f^t and g^h. Similarly for other combination of orientations $(-f, g)$, $(f, -g)$ and $(-f, -g)$, we can add (f^h, g^h), (f^t, g^t) and (f^h, g^t) respectively.

4. For every telomere $(e_i, g) \in T$, we add an edge to G between e_i and g^h. In case of (g, e_i), an edge between g^t and e_i are added.

5. For the rest of the edges in G, we set the edge weights to inf to exclude them from the solution.

As the inferred probabilities range from 0 to 1, using them directly as edge weights may introduce undesirable impact associated with handling small float points. It is critical for TSP to have a more precise and fine-grained set of edge weights to assure the quality of its solution. 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 a little bit smaller probabilities can hardly be considered. Therefore we decided to use the following equation to curve

the probabilities into edge weights:

$$w_{(f,g)}(m) \quad = \quad \log_2(10^m \times (1 - p_{(f,g)})) \tag{5.1}$$

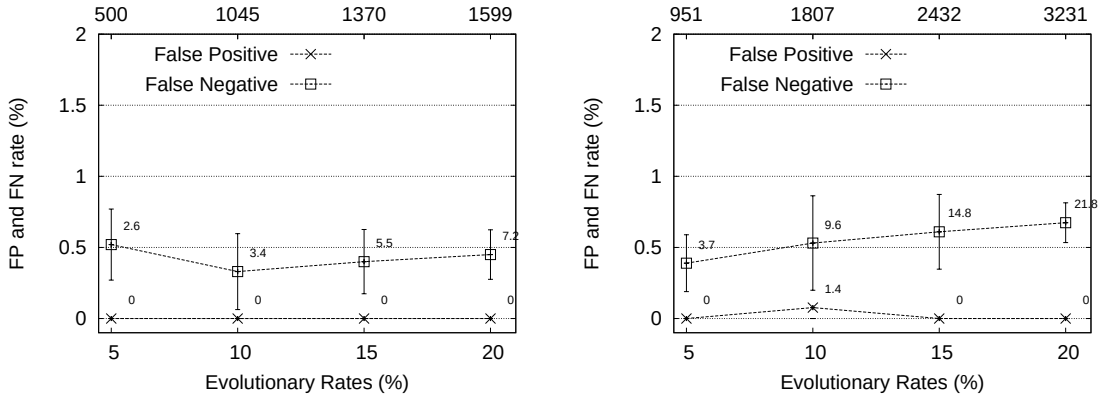
where $(f, g) \in \{A \cup T\}$ and $p_{(f,g)}$ is the probabilities of observing (f, g). m is the sole parameter determining the shape of the curve and according to our experiments, TSP yields good results when $m = 6$.

We then utilize the power of one of the most used TSP solver `Concorde` [2] to find the optimal path which traverses every vertex once with the minimum total score. In the solution path, multiple contiguous extremities are shrank to a single one and a gene segment between two extremities is taken as a contig. Our construction of TSP topology is in spirit similar to `GapAdj`, however `GapAdj` requires additional procedures and parameters to adjust the contig number. Instead our inference of ancestral genome is uniform and directly from the solution of TSP, minimizing the risk of introducing artifacts.

5.3 Experimental Results

Experimental Design

To evaluate the performance of `PMAG`$^+$, we ran a series of experiments on simulated datasets under a wide variety of settings. We generated model topologies from the uniformly distributed binary trees, each with s species. An initial gene order of n distinct genes and m chromosomes was assigned at the root so it can evolve down to the leaves following the tree topology mimicking the natural process of evolution, by carrying out a set of predefined evolutionary events. We used different evolutionary rates r with 50% relative fluctuation, thus the actual number of events per edge is in the interval $\lfloor \frac{r \times n}{2}, r \times n \rfloor$. Several evolutionary events were considered—inversions, translocations and indels and each kind of event was assigned a probability to be selected during the simulation process. In this study, we only present results with



(a) 5% Gene Insertion and Deletion

(b) 10% Gene Insertion and Deletion

Figure 5.2: *FP* and *FN* rates (divided by the numbers on upper x-axis) with standard deviations under various evolutionary rates and indel rates. Labels on upper x-axis represent the total number of genes that are inserted or deleted over all internal nodes due to indel operations. Numbers above points indicate the actual amount of errors in average.

20 genomes, each with 1000 genes and 5 chromosomes, to closely mimic bacterial genomes. The evolutionary rates r were set from 50 to 200 events, the later representing highly disturbed datasets. For each combination of evolutionary events, we simulated 10 datasets and reported averages and standard deviations.

Our predicted ancestral genomes are evaluated by the ratio of correct adjacencies and telomeres recovered. In specific, we used the following equation to compute the error rate of reconstruction.

$$E = (1 - \frac{|D \cap D'|}{|D \cup D'|}) \times 100\%$$

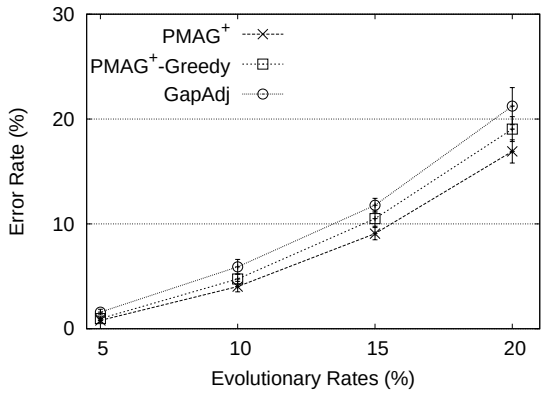
where D represents the set of gene adjacencies and telomeres in the real genome and D' the predicted genomes. We further refer an element that is contained in inferred set S' but not in true set S as a `false positive` (FP) and `false negative` (FN) is defined similarly, by swapping S and S'.

Assessing the Accuracy of Ancestral Gene Contents

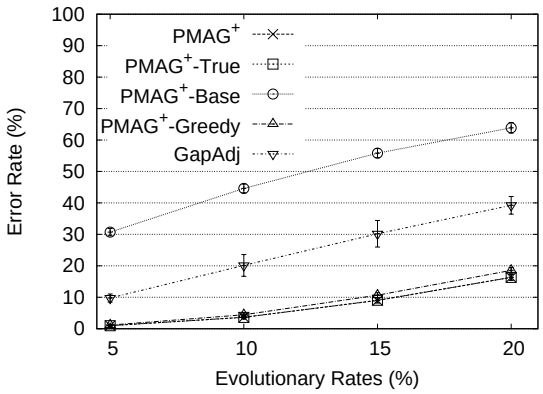
We first ran simulations to test PMAG$^+$ on the inference of ancestral gene contents. Our gene orders, derived from its direct ancestor through a number of events, underwent random indels and inversions (two boundaries of each inversion are uniformly distributed). Two different probabilities (5% and 10%) of occurrences for indels were used. We compared our inferred gene content with its corresponding true content and counted the number of *FPs* and *FNs*. For each dataset, we summed the number of *FPs* and *FNs* in all internal nodes and divided it by the total number of genes in all ancestral nodes that are missing or inserted. Figure 5.2 shows our results. From this figure, the *FP* rates are always extremely low (only one dataset produced *FPs*), indicating that our inference can prevent introducing erroneous gene content and the inferred contents are reliable. *FN* rates increase slightly when more indel operations were performed, but even in the worst case the error rate stays below 1%. At the same time, we ran GapAdj without specifying any WGD node and set the cutoff value and maximal iterations to 0.6 and 25 as suggested. According to the results, GapAdj failed to infer a large portion of inserted genes, making the *FPs* rates in all cases higher than 60%.

Assessing the Accuracy of Ancestral Gene Orders

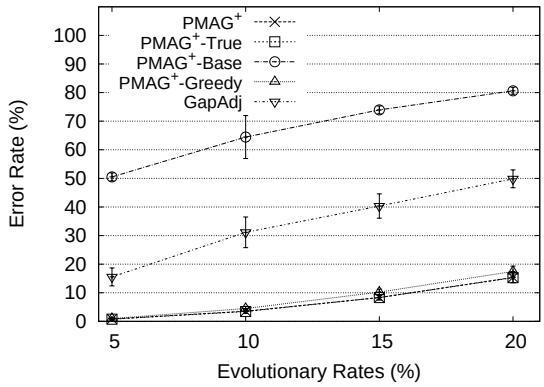
We conducted several tests to evaluate the accuracy of PMAG$^+$ under different degrees of indels. Our first test is to compare PMAG$^+$ with current standard approach that reduces the dataset into equal content by eliminating genes that are not present in every genome, which forms the baseline method (named PMAG$^+$-Base). Our second test is to give PMAG$^+$ the "ground true" content (named PMAG$^+$-True) to eliminate all impacts from gene contents. To compare the greedy heuristic to the TSP solution, we switched back to the greedy heuristic and redid the tests (named PMAG$^+$-Greedy). Finally the results of GapAdj (which is the most recent method to our knowledge)



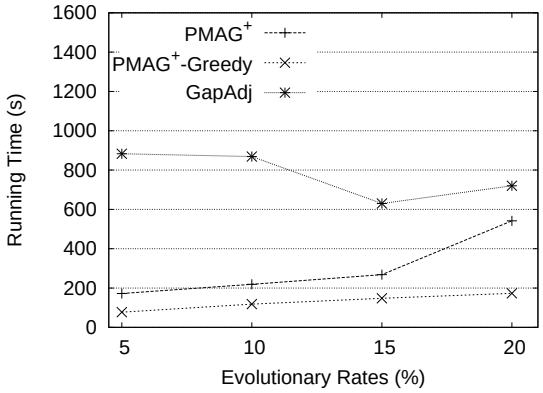
(a) 90% Inv and 10% Tsl



(b) 5% Ins and Del, 80% Inv and 10% Tsl



(c) 10% Ins and Del, 70% Inv and 10% Tsl



(d) Running time of tests in (a)

Figure 5.3: (a), (b) and (c) summarize the error rates under various evolutionary rates and combinations of evolutionary events (Ins for insertion, Del for deletion, Inv for inversion and Tsl for translocation). (d) shows the running time for methods in (a). Error bars indicate the standard deviations

were reported. For general purpose, we also compared `PMAG+` with `GapAdj` using datasets without indel operations.

Evaluation of designed experiments in terms of error rates is shown in figure 5.3. From the figure, the error rates for both `PMAG+` and `PMAG+-True` are the lowest in all cases and the difference between the two approaches is almost indistinguishable, indicating that errors introduced by a very limited amount of false contents are not significant.

As expected, `PMAG+-Base` recovered the least amount of adjacencies due to the

loss of contents. GapAdj, due to its failure in gene content inference, achieved much higher error rates in the presence of indels. Even in the test of equal gene content, PMAG$^+$ can still outperform GapAdj with around 5% higher accuracy.

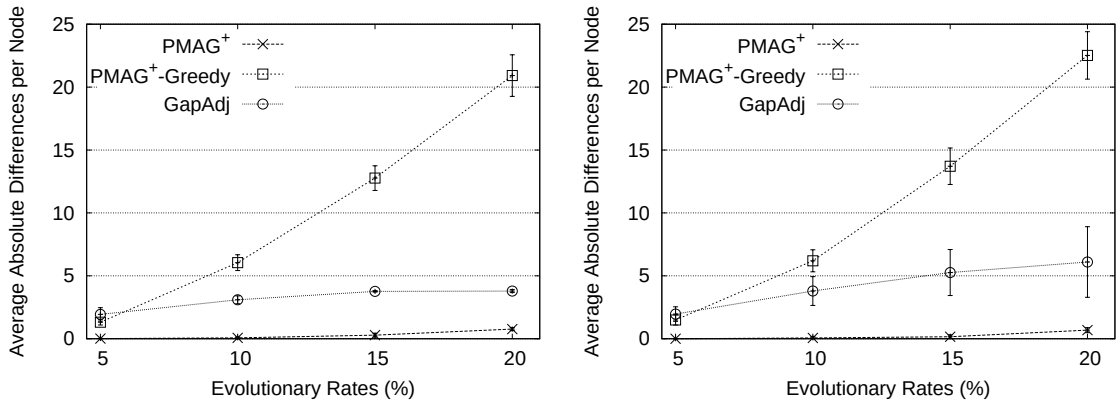
PMAG$^+$-Greedy came very close to PMAG$^+$, however in all test, PMAG$^+$ can always return more accurate reconstruction than PMAG$^+$-Greedy, suggesting the usefulness of our TSP assembler.

Using different degrees of indels has little impact on the performances of PMAG$^+$. From the perspective of adjacency evolution, an inversion operation always breaks two extant adjacencies and creates two new adjacencies, the disturbances on adjacencies introduced by an indel operation are essentially much similar to an inversion. In particular, a deletion breaks two adjacencies and creates a new one, while a insertion breaks one adjacency and introduces two new adjacencies. Therefore, as long as ancestral gene contents can be accurately predicted, PMAG$^+$ returns comparable results with all combinations of evolutionary events.

The last figure summaries the running time of all methods. From the figure, PMAG$^+$-Greedy benefits from the greedy heuristic is indeed slightly faster than PMAG$^+$, while GapAdj which solves the TSP problem heuristically took a longer time to finish than PMAG$^+$ using an exact solver.

Assessing the Number of Inferred Contigs

PMAG$^+$ by treating telomeres as a special type of adjacencies, simultaneously finds the best set of adjacencies and telomeres in one step. As translocation operations account for inter-chromosomal rearrangements which can be equivalently viewed as a fission followed by a fusion, thus all ancestors should also have the same amount of chromosomes to the root node, which is 5 in our test cases. For each dataset with N ancestors, the number of contigs c_i $(1 \leq i \leq N)$ in each ancestor was counted and the average absolute differences per ancestral node $\frac{\sum_{i=1}^{N} |c_i - 5|}{N}$ was computed to



(a) 0% Gene Insertion and Deletion (b) 10% Gene Insertion and Deletion

Figure 5.4: The average of absolute differences per ancestral node produced by various methods. Error bars indicate the standard deviations

assess the accuracy of chromosomal assembly. Figure 5.4 summaries our findings. As predicted, the amount of contigs produced by `PMAG+-Greedy` was totally irrelevant to the true number of chromosomes, while `GapAdj` can indeed reduced a large portion of redundant contigs. In comparison, the number of contigs returned by `PMAG+` can precisely reflect the actual number of chromosomes in the true genomes.

5.4 Conclusions

In this chapter, we described an extension to our previous adjacency-based method called `PMAG+`. `PMAG+` can infer the ancestral gene orders under a more general model of gene evolution, including intra-chromosomal and inter-chromosomal rearrangements as well as gene insertions and deletions. As real ancestors are unknown, we tested our method through a series of simulation studies. According to the results, `PMAG+` can accurately deduce the ancestral gene contents with error rates less than 1%. In the subsequent inference of ancestral gene orders, `PMAG+` can outperform existing methods we tested. Also by adopting a TSP solution for adjacency assembly, `PMAG+` not only overcame the issue on producing excessive contigs, but also achieved better performance than using the greedy assembler.

Chapter 6

Summary

This work investigates on two classic problems using gene order data—the phylogeny problem and the ancestral inference problem. We provided each problem with an efficient solution. The successes of these two methods are in two parts. First, our approaches use the binary encoding to simplify a complex gene-order permutation under various evolutionary events into a sequence of independent gene adjacencies. Second, our approaches use a biased transition model to account for genome rearrangements. The model was derived from standard DCJ operations and has been proved to be critical for both methods.

We conducted extensive simulation studies to evaluate each method under a wide range of settings. Our results were also compared with all the other available methods under the same profile. Statistical reveal that both methods are very accurate and flexible enough to process most types of evolutionary events. At the meanwhile, both methods demonstrated great scalability to handle extremely large dataset in an acceptable mount of time.

Bibliography

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

[2] David Applegate, ROBERT Bixby, Vasek Chvatal, and William Cook, *Concorde tsp solver*, URL http://www. tsp. gatech. edu/concorde (2006).

[3] Sèverine Bérard, Coralie Gallien, Bastien Boussau, Gergely J Szöllősi, Vincent Daubin, and Eric Tannier, *Evolution of gene neighborhoods within reconciled phylogenies*, Bioinformatics **28** (2012), no. 18, i382–i388.

[4] 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.

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

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

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

[8] ———, *A lower bound for the breakpoint phylogeny problem*, Combinatorial Pattern Matching, Springer, 2000, pp. 235–247.

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

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

[11] ———, *The reversal median problem*, INFORMS Journal on Computing **15** (2003), no. 1, 93–113.

[12] 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).

[13] 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.

[14] 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.

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

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

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

[18] Pedro Feijao and Joao Meidanis, *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.

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

[20] 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.

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

[22] 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.

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

[24] 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.

[25] 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.

[26] Jahn Katharina, Zheng Chunfang, Kovac Jakub, and Sankoff David, *A consolidation algorithm for genomes fractionated after higher order polyploidization*, BMC Bioinformatics (2012), 13(Suppl 19): S8.

[27] James Kent, Robert Baertsch, Angie Hinrichs, Webb Miller, and David Haussler, *Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes*, Proceedings of the National Academy of Sciences **100** (2003), no. 20, 11484–11489.

[28] Victor Kunin and Christos A Ouzounis, *Genetrace–reconstruction of gene content of ancestral species*, Bioinformatics **19** (2003), no. 11, 1412–1416.

[29] 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.

[30] 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.

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

[32] Yu Lin, Vaibhav Rajan, and Bernard Moret, *Fast and accurate phylogenetic reconstruction from high-resolution whole-genome data and a novel robustness estimator*, Journal of Computational Biology **18** (2011), no. 9, 1131–1139.

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

[34] Jian Ma, Aakrosh Ratan, Brian J Raney, Bernard B Suh, Webb Miller, and David Haussler, *The infinite sites model of genome evolution*, Proceedings of the National Academy of Sciences **105** (2008), no. 38, 14254–14261.

[35] 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.

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

[37] 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.

[38] 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.

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

[40] K Müller, T Borsch, L Legendre, S Porembski, I Theisen, and W Barthlott, *Evolution of carnivory in lentibulariaceae and the lamiales*, Plant Biology **6** (2008), no. 4, 477–490.

[41] 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.

[42] 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.

[43] DF Robinson and Leslie Foulds, *Comparison of phylogenetic trees*, Mathematical Biosciences **53** (1981), no. 1, 131–147.

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

[45] 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.

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

[47] _____, *Probability models for genome rearrangement and linear invariants for phylogenetic inference*, Proceedings of the third annual international conference on Computational molecular biology, ACM, 1999, pp. 302–309.

[48] 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.

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

[50] Michael Sorenson and Robert Fleischer, *Multiple independent transpositions of mitochondrial dna control region sequences to the nucleus*, Proceedings of the National Academy of Sciences **93** (1996), no. 26, 15239–15243.

[51] Alexandros Stamatakis, *New standard raxml version with marginal ancestral state computationas*, https://github.com/stamatak/standard-RAxML.

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

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

[54] 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.

[55] 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.

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

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

[58] 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.

[59] 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.

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

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

[62] 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.

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

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

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

[66] 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.

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

[68] 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.

[69] 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.

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