Spring 5-15-2015

Integration of Alignment and Phylogeny in the Whole-Genome Era

Hongtao Sun
Washington University in St. Louis

Recommended Citation

Sun, Hongtao, "Integration of Alignment and Phylogeny in the Whole-Genome Era" (2015). *Engineering and Applied Science Theses & Dissertations*. 93.
http://openscholarship.wustl.edu/eng_etds/93

WASHINGTON UNIVERSITY IN ST. LOUIS

School of Engineering and Applied Science

Department of Computer Science and Engineering

Dissertation Examination Committee:
Jeremy Buhler, Chair
Yixin Chen
Justin Fay
Tao Ju
Robert Pless
Gary Stormo

Integration of Alignment and Phylogeny in the Whole-Genome Era

by

Hongtao Sun

A dissertation presented to the
Graduate School of Arts and Sciences
of Washington University in
partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

May 2015
Saint Louis, Missouri

Table of Contents

List of Tables

List of Figures

Acknowledgments

First, I'd like to thank my advisor Jeremy Buhler. He is very knowledgeable and passionate about research. He always shows great insight in problems and has very good sense about nearly all topics related to my research. Besides research, he pays great attention to his students, always trying to enlighten them during their hard times and igniting their self-motivation. Without his guidance and support, it would have been impossible for me to go this far.

I am also very thankful for my wife Ziyan and my parents Gang Sun and Xiulan Chang. Without their support, I wouldn't have been able to finish the thesis.

A special thanks goes to the many graduate students and faculty within my department who have reviewed this thesis and helped support the related research.

This work was supported in part by NSF award DBI-0237902.

Hongtao Sun

Washington University in Saint Louis
May 2015

Dedicated to my advisor Jeremy Buhler for his continuous support and extreme patience and generosity, my wife Ziyan, my son Ping and my parents for their love and encouragement.

ABSTRACT OF THE DISSERTATION

Integration of Alignment and Phylogeny in the Whole-Genome Era

by

Hongtao Sun

Doctor of Philosophy in Computer Science

Washington University in St. Louis, 2015

Professor Jeremy Buhler, Chair

With the development of new sequencing techniques, whole genomes of many species have become available. This huge amount of data gives rise to new opportunities and challenges. These new sequences provide valuable information on relationships among species, e.g. genome rearrangement and conservation. One of the principal ways to investigate such information is multiple sequence alignment (MSA). Currently, there is large amount of MSA data on the internet, such as the UCSC genome database, but how to effectively use this information to solve classical and new problems is still an area lacking of exploration. In this thesis, we explored how to use this information in four problems, i.e. sequence similarity search, multiple alignment improvement, short read mapping, and genome rearrangement inference.

The first problem is sequence similarity search, i.e., given a query sequence, search its similar sequences in a database. The expansion of DNA sequencing capacity has enabled the sequencing of whole genomes from a number of related species. These genomes can be combined in a multiple alignment that provides useful information about the evolutionary history at each genomic locus. One area in which evolutionary information can productively be exploited is in aligning a new sequence to a database of existing, aligned genomes. However, existing high-throughput alignment tools are not designed to work effectively with multiple genome alignments. We introduce PhyLAT, the Phylogenetic Local Alignment Tool, to compute local alignments of a query sequence against a fixed multiple-genome alignment of closely related species. PhyLAT uses a known phylogenetic tree on the species in the multiple alignment to improve the quality of its computed alignments while also estimating the placement of the query on this tree. It combines a probabilistic approach to alignment with seeding and expansion heuristics to accelerate discovery of significant alignments. We provide evidence, using alignments of human chromosome 22 against a 5-species alignment from the UCSC Genome Browser database, that PhyLAT's alignments are more accurate than those of other commonly used programs, including BLAST, POY, MAFFT, MUSCLE, and CLUSTAL. PhyLAT also identifies more alignments in coding DNA than does pairwise alignment alone. Finally, our tool determines the evolutionary relationship of query sequences to the database more accurately than do POY, RAxML, EPA, or pplacer.

The second problem is multiple alignment quality improvement, i.e., given a multiple alignment, correct any wrong matches, i.e., matches between non-orthologous characters (bases or residues). This is important to all other data analysis based on multiple alignments. However, existing methods either compute alignments non-iteratively or use complex models which are very time-consuming and have the risk of overfitting. We developed an optimization algorithm to iteratively refine the multiple alignment quality. In each iteration, we take out one sequence from the multiple alignment, and realign it to the rest of the sequences using our phylogeny-aware alignment framework. We tested several strategies for picking sequences, i.e., picking out the most distant species from the rest species, picking out the closest species from the rest species and randomly picking out a sequence. Experiment

results showed that different picking strategies gave very similar results. In other words, our method is very insensitive to sequence picking strategy, which makes it a stable algorithm for improving alignments of any number of sequences. The results showed that our method is more accurate than existing methods, i.e. MAFFT, Clustal-O, and MAVID, on test data from three sets of species from the UCSC genome database.

The third problem is phylogeny-aware short read mapping using multiple informant sequences. Given a set of short reads from next-generation sequencing results, mapping them back to their orthologous locations in a reference genome is called short read mapping. This is a new problem arising with the development of next-generation sequencing techniques. Existing methods cannot deal with indels in alignments, and cannot do interspecies mapping. We developed a model, PhyMap, to align a read to a multiple alignment allowing mismatches and indels. PhyMap computes local alignments of a query sequence against a fixed multiple-genome alignment of closely related species. PhyMap uses a known phylogenetic tree on the species in the multiple alignment to improve the quality of its computed alignments while also estimating the placement of the query on this tree. We showed theoretically that our model can differentiate orthologous sequences from paralogous sequences. Thus our algorithm can align short reads to their homologous positions in reference sequences. Our experiment results have proved this and showed that our model can differentiate between orthologous and paralogous alignments. Furthermore, we compared our method with other popular short read mapping tools (BWA, BOWTIE and BLAST) on simulated data, and found that our method can map more reads to their orthologous locations in their closely-related species' genomes than any one of them.

The fourth problem is genome rearrangement inference, i.e., given a set of orthologous alignments along with the genomic orders in each aligned sequence and a set of new sequences orthologous to the given alignments, determine the genomic order of the new sequences. Existing methods on genome rearrangement inference have several shortcomings. First, most of these methods rely on annotated genes. They are only applicable to genomes with annotated genes. They cannot infer on parts of genomes where there are no genes or do not have annotated genes. Second, they either only infer a set of conserved intervals without specifying their order or just infer a fixed single order of all the genes without giving alternative solutions. We gave a simple genome rearrangement model which can express inversions, translocations and inverted translocations on aligned genome segments. We also developed an MCMC algorithm to infer the order of the query segments. We proved that using any Euclidian metrics to measure distance between two sequence orders in the tree optimization goal function will lead to a degenerated solution where the inferred order will be the order of one of the leaf nodes. We also gave a graph-based formulation of the problem which can represent the probability distribution of the order of the query sequences.

This thesis is not only about the four problems. Our goal is to give an attempt to solve different problems in this field using the same underlying model. We feel that with the fast accumulation of biological data and deeper and deeper understanding people have on these data, different problems become related to each other and can be integrated under the same framework. With a unified model, different problems do not need ad-hoc solutions. The ultimate vision is that all problems can be expressed in their biologically realistic representations and solved by algorithms based on this real model. All of the four problems studied in this thesis are independent in their own areas, but are also related to each other in the sense that a better solution to one problem will give better solutions to other problems. On one hand, the problem of MSA quality improvement is a foundation of the other two problems, and on the other hand, building a high-quality MSA on new sequences requires accurate short read mapping in the first place. First, we showed that by doing alignment and short read mapping together, we can get more accurate short read mapping. Second, we showed that improving existing multiple alignment quality can be aided by incorporating phylogenetic information and a probabilistic scoring system. Last, we showed that multiple sequence alignment can be used to infer genome rearrangement events with the help of phylogenetic information. Our work does not only explored the solutions to the four problems, but also provides a new viewpoint that with the development of new techniques and availability of new data, many existing and new problems can be viewed in the same framework and solved using the same model.

Chapter 1

Introduction

With the development of new sequencing techniques, whole genomes of many species have become available. This huge amount of data gives rise to new opportunities and challenges. These new sequences provide valuable information on relationships among species, e.g. genome rearrangement and conservation. One of the principal ways to investigate such information is multiple sequence alignment (MSA). Currently, there is large amount of MSA data on the internet, such as the UCSC genome database [130], but this data has not consistently been applied to solve classical and new problems in biosequence analysis. In this thesis, we explore how to use this information in four problems, i.e. sequence similarity search, multiple alignment improvement, short read mapping, and genome rearrangement inference.

Before diving into details of problems and solutions, several core concepts need to be introduced.

Sequence Alignment: a sequence alignment is a method of comparing similar biological sequences. Usually an alignment of N sequences is represented as a matrix with N rows, each row containing the characters in a sequence plus some gaps to make similar characters be in the same column. An example is shown in Figure 1.1.

Finding accurate alignments is usually the first step to many bioinformatic problems. High-score alignment means more similarities between the aligned sequences, which is often a good

```
sequence1 CACCTAAGTACT
sequence2 CACGTAA--ACT
sequence3 CTCCTAAGTACA
sequence4 CACCCAAGTACT
```

Figure 1.1: An example of multiple alignment of four sequences. Gaps are inserted into the second sequence to make similar characters aligned in the same columns.

indication that the aligned sequences may share the same biological functions or have the same ancestral sequence.

Phylogeny: a phylogeny is a tree structure, representing the evolutionary history of several species. Each leaf node represents an extant species. An inner node represents a common ancestor of the species in its subtree. Each branch in the tree may be associated with a length, representing the evolutionary time from the parental species to the child species. The longer the branch is, the more possible changes there may be between the parental sequence and the child sequence. Thus the branch length can also be proportional to the amount of character changes from the parental sequence to its child sequence. An example is shown in Figure 1.2.

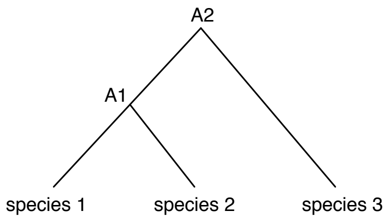


Figure 1.2: An example of phylogeny. There are three extant species, i.e. species 1, species 2 and species3. A1 is the latest common ancestor of species 1 and species 2. A2 is the latest common ancestor of all three species.

Next-generation Sequencing: next-generation sequencing is a group of technologies for sequencing biological sequences, i.e., DNA and RNA sequences. A sequence is sampled at a large number of locations. Each sample is called a read, with a length ranging from 21 to 400 bases [56, 129], depending on what kind of technology is used. A read usually contains errors and sometimes missing bases. Each location in the original sequence is sampled 40 to 500 times. Compared with previous sequencing techniques, next-generation sequencing is low-cost and much faster. An example is human genome sequencing. It cost 3 billion dollars and 13 years at the time of the Human Genome Project [139]. Now it only costs $1000 and a single run of a sequencing machine.

A common factor of the four problems is that they are intrinsically related to multiple alignment and phylogeny. However, there are very few researches which try to use a unified biological framework for these different kinds of tasks. We feel it is the right way to view these problems in such a framework that each problem can leverage the biologically meaningful models and solutions to other problems in the framework. The core of the framework should be multiple alignment and the concepts of homology and rearrangement. These are the most important aspects of modeling the real biological process of sequences evolution. Our study on sequence similarity search, PhyLAT (*Phy*logeny-aware *L*ocal *A*lignment *T*ool), showed that using a multiple alignment as reference sequence and a phylogeny-based probabilistic scoring system can improve the alignment accuracy and the accuracy of tree placement of the query species. This implies that by combining multiple alignment and phylogeny, we can do better on both problems. The next step is naturally to apply this combined framework to other related problems, which are the rest three problems we study in this thesis. So we will first introduce our study on sequence similarity search, then the rest three problems. In the multiple alignment improvement problem, we use our PhyLAT framework as the basis for aligning each sequence back to the rest of the multiple alignment. In the short read mapping problem, we use PhyLAT to align each read to its possible matching positions in a multiple alignment database. In genome rearrangement inference problem, we use PhyLAT as the first step to infer orthology mapping from query sequence segments to reference sequence blocks in a database.

1.1 Overview

In this section, we give an overview of the four problems (including our previous research on sequence similarity search) we studied, and briefly discuss our methods and results.

1.1.1 Sequence Similarity Search

The first problem we studied is sequence similarity search. Sequence similarity search is the basis for nearly all downstream researches in bioinformatics. In principle, using a reference multiple alignment as database, rather than any one of its component genomes, to align a query sequence should result in a more accurate alignment, since the aligner can use the pattern of conservation at each position to more accurately determine which query base corresponds to which multiple alignment column. Moreover, given a phylogenetic tree relating the species in the reference, an aligner should be able to use standard probabilistic models of evolution to compare the likelihoods of possible alignments, rather than resorting to an arbitrary scoring system. In fact, alignment could even infer the evolutionary relationship of the query to the reference, placing it on the tree of the reference's species.

In practice, however, most widely used alignment tools either cannot use reference multiple alignments or cannot do so in a phylogenetically aware way [6, 6, 12, 44, 45, 47, 70, 150, 159, 162]. So the first problem we study is practical implementation of high-throughput pairwise alignment between a query sequence and database of reference multiple alignments with phylogenetic information.

We developed PhyLAT (the Phylogenetic Local Alignment Tool), a tool for rapidly aligning a query DNA sequence to a database of multi-genome reference alignments. PhyLAT combines BLAST-style seeding and extension heuristics with a EM-like, phylogenetically aware backend alignment algorithm. We score alignments to references containing gaps using a model that is simplified enough for efficient implementation but disallows alignment hypotheses that are demonstrably impossible alignment given the pattern of gaps in the reference. We show that PhyLAT produces results in protein-coding regions of mammalian genomes that are better supported by external evidence than the results of pairwise alignment, and that our tool can accurately infer the evolutionary relationship of the query to the species in the multiple alignment.

The framework behind PhyLAT is the cornerstone of all the rest problems we study.

1.1.2 Multiple Alignment Improvement

The first problem is iterative improvement of multiple alignment quality using phylogenetic information.

Sequence alignment is a prerequisite to nearly all downstream comparative genomic analyses, including the identification of conserved sequence motifs, estimation of evolutionary distance between sequences, and inference of evolutionary history of genes and species. Errors in sequence alignment are found to have a significant negative effect on subsequent inference of sequence divergence, phylogenetic trees, and conserved motifs [76].

While there are many tools for constructing multiple alignment, there are few for refining existing multiple alignments. Existing tools may be fast and give a good initial multiple alignment, but the alignment quality can be improved by using more complex models, i.e., a biologically realistic and probabilistic model. It was shown that deletions in sequences will result in errors in several multiple sequence alignment tools using non-probabilistic scoring schemes, i.e., ClustalW, DIALIGN-T, MAFFT, MUSCLE, PROBCONS, T-COFFEE, and PRANK [55, 119]. It was also shown that iterative methods are more accurate than non-iterative methods [119].

We developed an optimization algorithm to iteratively refine the multiple alignment quality. It is a probabilistic model, using an iterative refining strategy. In each iteration, we take out one sequence from the multiple alignment, and realign it to the rest of the sequences using our phylogeny-aware alignment framework. We tested several strategies for picking sequences during iterations. Experiments show that our method is more accurate than existing methods on our test data.

1.1.3 Short Read Mapping

The second problem is phylogeny-aware short read mapping using multiple informant sequences.

Given a set of short reads from next-generation sequencing results, mapping them back to their orthologous locations in a reference genome is called short read mapping [90, 132].

This is a new problem arising with the development of next-generation sequencing techniques [124]. Because genomes from the same species are similar to each other in terms of DNA sequence and genome arrangement, it is relatively easy to map reads to a reference sequence from the same species.

Although in most cases, the reference genome is from the same species as the query reads, there are cases where interspecies mapping is necessary. One example is when such a reference genome is not available, i.e., no individuals of the same species have been sequenced and assembled before [126]. Another example is from metagenomics, where the reads can only be traced back to a set of species, or the species of the reads are totally unknown. In this case, the reads have to be classified according to their species, and then assembled within each species [8, 86]. Another example is RNA expression estimation. It has been shown that using read mapping to estimate the expression level is more accurate and repeatable than using microarray [87]. In many cases, some closely related species to the newly sequenced species have already been sequenced and assembled, which can provide useful information for classification and assembly of the newly sequenced reads [62, 63].

With the development of next-generation sequencing techniques, short reads are obtained in large volume every day. Most existing short read mapping tools either use a single reference genome, or are not designed for interspecies mapping because of these tools' limited ability to deal with interspecies levels of divergence between read and reference. As many new species are sequenced, methods for efficient and accurate interspecies mapping are needed. Such methods must use information from multiple informant species and do an alignment-based mapping procedure, but how to model mapping problem within this scenario is still an open problem.

We developed a model, PhyMap, to align a read to a multiple alignment allowing mismatches and indels. Because the number of reads is huge, we developed an efficient hashing technique to search for promising orthologous loci in the reference multiple alignment. This alignment framework can be easily extended to use as many informant species as we want. PhyMap computes local alignments of a query sequence against a fixed multiple-genome alignment of closely related species. PhyMap uses a known phylogenetic tree on the species in the multiple alignment to improve the quality of its computed alignments while also estimating the placement of the query on this tree. We show that our model can differentiate between orthologous and paralogous alignments. Furthermore, we compared our method with other popular short read mapping tools (BWA, BOWTIE and BLAST) on simulated data, and found that our method can map more reads accurately than any one of them.

1.1.4 Genome Rearrangement Inference

The third problem is inferring genomic order of query segments using informant sequences and phylogeny.

While many genomes have been sequenced and assembled into continuous sequences, for some species, their genomes are partly available or cannot be assembled into continuous sequences. One example is short read mapping. While read mapping can map reads to their orthologous locations in related species, different reference species will give different order of the mapped reads. Existing read assembly algorithms will produce segments of assembled reads, without inferring their orders [95, 107]. This is because the order of othologous segments in reference species is not the same as in the query species.

If a set of genes always appear together in a genomic block in both species, then this block is called synteny block. Genes across synteny blocks do not always appear together. Thus it is very difficult to assemble segments which do not belong to the same synteny blocks. Furthermore, breaks between synteny blocks are very common among species. For example, synteny relationships among 10 amniotes (human, chimp, macaque, rat, mouse, pig, cattle, dog, opossum, and chicken) were compared at <1 human-Mbp resolution. There are 2233 homologous synteny blocks (HSBs) [81].

To infer the order of a set of segments is NP-hard [22]. When multiple informant genomes are available, the rearrangement information in those genomes can actually be used to infer the rearrangement events or the order of the segments in the query species.

We give a probabilistic model of genome rearrangement. Based on that model, we develop an MCMC algorithm to infer the order of genome segments in a query species using orders of orthologous segments in informant species. Experiment results show that the MCMC algorithm can converge to the optimal solution under breakpoint distance measurement. It can find the ground-truth solution for small number of orthologous sequences. But it failed for large data sets. Note that the failures result from the discrepancy between the optimal goal function value which is the overall breakpoint distance and the ground-truth solution. The MCMC algorithm actually can find the optimal solution under the given goal function. We prove that using any Euclidean distance metrics as the goal function of the optimization process will result in the order of segments of the query species being the same as the order of one of the leaf nodes in the tree, which means the optimization program just picks one input as output under such case. We also tried a graph-based algorithm. The results show that while the distance measurement can be well approximated by breakpoint distance, the model of genome rearrangement needs to be refined.

1.2 Outline

This dissertation is organized as follows.

In chapter 2, we describe our model for aligning a single sequence to a multiple alignment using a phylogeny. This model is used in all subsequent problems as well. In the multiple alignment improvement problem, each sequence in the multiple alignment is taken out and realigned to the rest of the multiple alignment within this framework. In the short read mapping problem, each read is aligned to the multiple alignment within this framework, except that the aligned sequences are much shorter than in our homology search problem. In the genome rearrangement inference problem, the query sequences are first aligned to the informant sequences under this framework, then the order of the query sequences is inferred using the orders of the sequences of the informant species.

In chapter 3, we describe an optimization algorithm to iteratively refine multiple alignment quality. We did experiments using different re-aligning strategies. Our experimental results show that our alignment program can improve multiple alignment quality better than other alignment programs. We also show that our algorithm is robust regardless of the optimization order of the aligned sequences.

In chapter 4, we give a model to differentiate between orthologous and paralogous alignments. We describe our method for aligning a read to a multiple alignment allowing mismatches and indels. We give a theoretical proof that our model can differentiate between orthologous and non-orthologous alignments. We also give experimental results on simulated reads showing that our alignment algorithm PhyMAP can align short reads more accurately than other short read mapping tools.

In chapter 5, we describe a probabilistic model of genome rearrangement. Then we describe our MCMC optimization algorithm for finding the order of query sequences using the orders in informant species. We also prove that using any Euclidian distance measurement will result into a degenerated solution where the inferred order of the query sequences is the same as the order of one of the leaf nodes in the tree relating the informant species. We also give a graph-based method for the sequence order inference problem, which can represent the probability distribution of the order of the query sequences.

In chapter 6, we give a summary of the content of this dissertation. We briefly describe our works and results. We discuss prospects for future work.

Chapter 2

A Probabilistic Alignment Model for Multiple Alignment

In this chapter, we describe our framework for aligning a single query sequence to an existing multiple alignment. We will first introduce the background of the problem, then give detailed description of our model. This model is the basis for addressing all the problems studied in this thesis.

2.1 Background

The proliferation of high-throughput DNA sequencing has produced a huge amount of genomic DNA sequence, including genomes from many higher eukaryotes. Intensively studied clades of organisms – such as mammals, *Drosophila* fruit flies, and worms of the genus *Caenorhabditis* – are now typically represented in public databases by complete or partial genomes of multiple species. A group of closely related genomes can be combined into a large-scale multiple alignment of orthologous sequences [19, 20, 172].

Building a high-quality multiple-genome alignment requires a large investment of computational resources and curation time, particularly if the alignment will become a reference for future users. We would therefore like to amortize this investment by effectively utilizing information present in the alignment that is not readily available from its component genomes. Multiple-genome alignments are commonly used to interrogate a clade's evolutionary history

[37], often with the help of a phylogenetic tree on the component species, or to discover genomic loci of unusually high conservation [9] or unusually fast change [18]. However, they are rarely used to augment one of the most common operations in bioinformatics: aligning a new sequence to an existing reference.

In principle, using a reference multiple alignment, rather than any one of its component genomes, to align a query sequence should result in a more accurate alignment, since the aligner can use the pattern of conservation at each position to more accurately determine which query base corresponds to which multiple alignment column. Moreover, given a phylogenetic tree relating the species in the reference, an aligner should be able to use standard probabilistic models of evolution to compare the likelihoods of possible alignments, rather than resorting to an arbitrary scoring system. In fact, alignment could even infer the evolutionary relationship of the query to the reference, placing it on the tree of the reference's species.

In practice, however, most widely used alignment tools either cannot use reference multiple alignments or cannot do so in a phylogenetically aware way. BLAST [6] and other accelerated variants of Smith-Waterman [150] are widely used for pairwise sequence comparison, but these methods compare only individual sequences. PSI-BLAST [6] creates an alignment between a query and a profile computed from a database of individual reference sequences. However, the construction of the profile does not take into account phylogenetic information, so it does not weigh each reference sequence in a phylogenetic-aware way. HMMER [44, 45, 47] and SAM [70] *can* align a sequence to a preexisting multiple alignment, if it is generalized to a profile hidden Markov model, but even these tools use only statistical conservation at each position, rather than phylogenetic information, to perform alignments.

There are tools for *de novo* multiple sequence alignment using trees to improve alignment quality. The classic CLUSTAL [159] software uses a guide tree to align multiple sequences. More recent tools, such as POY [162], can align sequences given a tree or jointly compute a multiple alignment and a supporting phylogeny. However, these tools cannot as a rule incrementally update a multiple alignment and its associated tree starting from pre-existing references, which is the computation needed to align a query sequence to a multiple alignment database. Moreover, the high cost of *de novo* multiple alignment limits the computationally feasible methods that these tools can employ. PaPaRa [12], unlike the tools described above, uses a guide tree to map queries, in particular short reads, to an existing multiple alignment, but it does not score or improve the resulting alignments probabilistically given the phylogeny. *Practical implementation of high-throughput pairwise alignment between a query sequence and database of reference multiple alignments with phylogenetic information therefore remains an open problem.*

Classical results from phylogeny [173] give the theory needed to construct a maximum-likelihood alignment between a query sequence and a reference multiple alignment, provided that neither query nor reference contains gaps. This theory can be extended to allow the query to contain bases that are not homologous to any reference position or vice versa; for example, Siepel and Haussler describe such an approach for phylogenetic HMMs [145]. However, multi-genome reference alignments typically include columns with *both* bases and gaps, which may in fact be homologous to certain query bases. Finding a reasonable way to evaluate the likelihood of such putative homologies is a difficult problem. This fundamental issue, as well as assorted technical details needed to adapt any alignment algorithm to BLAST-like high-throughput use, make the construction of a fast, phylogenetically informed tool nontrivial.

2.2 The PhyLAT Alignment Framework

2.2.1 Overview

In this section, we describe PhyLAT (the Phylogenetic Local Alignment Tool) [155], a tool for rapidly aligning a query DNA sequence to a database of multi-genome reference alignments. PhyLAT combines BLAST-style seeding and extension heuristics with a EM-like, phylogenetically aware back-end alignment algorithm. We score alignments to references containing gaps using a model that is simplified enough for efficient implementation but disallows alignment hypotheses that are demonstrably impossible given the pattern of gaps in the reference. We show that PhyLAT produces results in protein-coding regions of mammalian genomes that are better supported by external evidence than the results of pairwise alignment, and that our tool can accurately infer the evolutionary relationship of the query to the species in the multiple alignment.

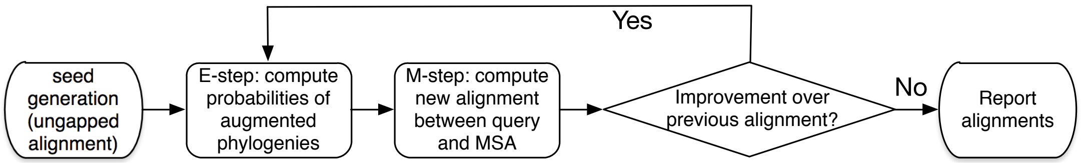

Yes
seed generation (ungapped alignment)
E-step: compute probabilities of augmented phylogenies
M-step: compute new alignment between query and MSA
Improvement over previous alignment?
No
Report alignments

Figure 2.1: Structure of PhyLAT algorithm. Not shown is the offline preprocessing of the database to compute its consensus and parameterize a mutation model at each position.

PhyLAT is built around an EM-like algorithm that simultaneously computes an alignment between a query sequence and a multiple alignment and predicts the placement of the query on the tree associated with the multiple alignment. The algorithm iteratively refines query alignment and branch placement until both have converged. To accelerate this core alignment algorithm, we adopt a BLAST-like seed generation and extension heuristics (see supplementary methods). We use the evolutionary consensus sequence of the multiple alignment to rapidly generate pairwise seed alignments, filter these seeds by E-value, and finally apply the core algorithm to each seed. The structure of the aligner is illustrated in Figure 2.1. The key contribution of PhyLAT is the use of the new underlying alignment model, which will be applied to later problems as well, and its efficient implementation.

Problem formulation for final alignment stage

Let M be a database composed of a multiple alignment of n orthologous DNA sequences. The species from which the DNA sequences are drawn are related by a phylogenetic tree τ, whose n leaves correspond to the n species. Each branch i in the tree has a length l_i, which is the evolutionary distance between the two endpoints of the branch. To convert these branches to transition probabilities, we use a mutation rate matrix Q, similar to the extended Tamura-Nei model [105, 157], that we estimate from the columns of M. We chose this Tamura-Nei-like model because it has a simple form with few parameters to estimate. In fact, PhyLAT can use arbitrary, non-time-reversible mutation models.

Given a query DNA sequence q, we want to find all high-scoring local alignments between q and M. We use a seed-and-extend procedure, described in the next section, to choose short substrings of q and subregions of M to align. For each such chosen pair, an alignment A is

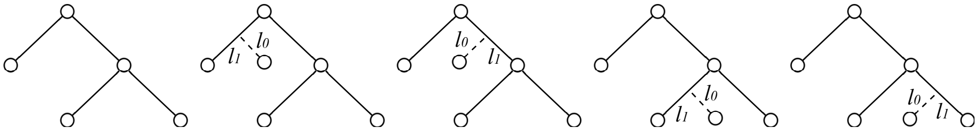


Figure 2.2: An example of augmented phylogenies. The phylogeny on the left is the original, while the rest are its four possible augmented phylogenies. Each augmented phylogeny is actually a family of trees with two parameters l_0 and l_1, which are the length of the new branch and its attachment site to its parent branch.

chosen to maximize a likelihood

$$\Pr(q, M|A, \tau). \tag{2.1}$$

Here, we assume that all possible alignments of q and M are *a priori* equally likely and choose the most likely one given the data and the tree.

Computing the complete-data likelihood for an alignment A requires that we know where the query is placed on τ relative to the sequences of M. We assume that we do not have this information; instead, we sum over all possible augmented tree topologies τ_i^* that add the query to a given branch on τ, as shown in Figure 2.2:

$$\Pr(q, M|A, \tau) = \sum_i \Pr(q, M, \tau_i^*|A, \tau).$$

For compactness of notation, we drop the explicit dependence of $Pr(q, M|A, \tau)$ on the fixed tree τ in subsequent sections.

EM computation of optimal local alignment

Alignment of a query to a reference starts with *seed generation*, which produces initial ungapped *seed alignments* between the query and one or more reference regions. Details of seed alignment generation are given later in this chapter. For each seed alignment, we perform gapped extension. Initially, we apply this final alignment stage to a region of the query and database of length 20. To allow for final alignments of varying lengths, we retry the computation on regions whose size is progressively doubled until doing so does not improve the final alignment score.

We now describe the EM algorithm used to compute the final local alignment A for given regions of q and M, as well as the probabilities of the augmented phylogenies τ_i^*. First, we define a set of indicator variables $\{x_i\}$ for each possible augmented phylogeny:

$$x_i = \begin{cases} 1 & \text{if augmented topology is } \tau_i^* \\ 0 & \text{otherwise.} \end{cases}$$

In this EM model, the known data are M and q (and the tree τ), while the latent variables are the x_i's. The unknown parameter of the model is the alignment A. The EM algorithm iteratively refines an initial guess $A^{(0)}$ at the alignment A while simultaneously inferring a distribution over the position of q in the phylogeny. The mth iteration starts with an alignment $A^{(m-1)}$ computed in the previous iteration. In the E-step of the iteration, the algorithm computes the expectation of each x_i:

$$\hat{x}_i = \Pr(x_i = 1 | q, M, A^{(m-1)}) \ . \tag{2.2}$$

In the M-step, the algorithm computes a new alignment $A^{(m)}$ to maximize the expected log-likelihood function:

$$A^{(m)} = \arg\max_A \sum_i \hat{x}_i \log \Pr(q, M | x_i = 1, A) \ . \tag{2.3}$$

Each iteration improves the likelihood of A and recomputes the distribution of the query's position in the tree. Finally, a local optimal point is reached, and the algorithm reports both a final alignment and an associated probability distribution over possible augmented tree topologies.

Assuming that the residues of q are stochastically independent, as are the columns of M, we can decompose the probability of the data given a tree placement and alignment as

$$\Pr(q, M|x_i = 1, A) = \prod_{j=1}^{|A|} \Pr(y[j], Z[j]|x_i = 1) \cdot \Pr_{y \notin q'}(y) \Pr_{Z \notin M'}(Z|\tau), \qquad (2.4)$$

where $y[j]$ and $Z[j]$ are a residue in q and column in M, respectively, from the jth column of alignment A, and q' and M' are the aligned regions of q and M respectively.

Computation of per-column probability

In both the E-step and the M-step, we need to compute the probability of an aligned query position and multiple alignment column given an augmented tree. The details of how this probability is computed determine the accuracy and efficiency of our algorithm. We introduce two key innovations for this task: treatment of alignment gaps in a way that is informed by the tree τ, and caching of subtree probabilities to accelerate the computation.

Further details of the per-column computation are given later in this chapter.

Treatment of gaps

An alignment of a query q to multiple alignment M may include gaps in either of q or M, or it may align a base of q to a column of M that contains both bases and gaps. To efficiently estimate the probabilities associated with such alignments, we need a gap model that is fast yet incorporates meaningful information about the alignment M. We consider two kinds of gaps. The first kind, the “local” gap, is assumed to arise as a series of single-base indels, while the second, the “global” gap, arises through a mutation that adds, deletes, or moves many contiguous bases at once. Local gaps are modeled using a single-base indel model, while global events may require a more complex model. In our work, we use the local model for sequence gaps of length ≤ 20; gaps longer than this are treated as missing data in the species where they occur. We note that this threshold was not empirically tuned to our test data but rather was an *a priori* estimate of the threshold between local and global indel events.

A very simple local indel model used in some work, including our own earlier work on PhyLAT [29], treats a gap as a fifth residue that can freely interconvert with A, C, G, and T. However, such treatment is inappropriate because, when we score an alignment column using a phylogeny, all observed residues are at the leaves in the tree. To compute the probability of the column, we sum over all possible labelings of the tree's internal nodes, which describe possible histories of insertion and deletion. Unfortunately, these labelings may include some histories that are biologically meaningless because they imply that aligned residues are nonhomologous. The models of [105] also have the problem of illegal labelings.

Other models exist that consider only legal indel histories for a given phylogeny [36, 41, 42]. However, the tools using these models are computationally expensive in practice because they enumerate all possible labelings. Moreover, these models consider only whether there is a base or gap at an inner node, disregarding the identity of the base. The model of [161] considers only legal labelings, but it still requires a time-reversible mutation model.

The current version of PhyLAT uses a gap model that recognizes that gaps cannot interconvert freely with residues in a phylogeny. Our model imposes two constraints. (1) Once a residue is deleted (converted to gap) on a branch, it cannot later be inserted, because the inserted residues are not homologous to the original residue. See Figure 2.3A. (2) If any internal node of the tree has a gap, then only one of its children can have a residue (insertion); the other one must have a gap. See Figure 2.3B. Note that once a residue is inserted, it can afterwards be deleted.

PhyLAT's per-column probability computation, while based on a simple mutation rate matrix Q (described in Methods) that nominally treats a gap as a fifth residue, sums over *only* those configurations of internal residues that are consistent with the two constraints given above. This excludes impossible indel histories that would otherwise contribute to the computed alignment probability.

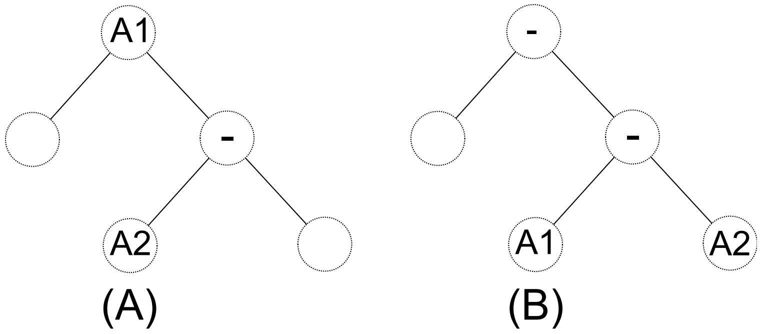


Figure 2.3: (A) If residue $A1$ is deleted and $A2$ is then inserted, $A1$ and $A2$ should not be considered homologous. This case is not allowed in our model. (B) If insertions occur on both child nodes, then residues $A1$ and $A2$ should not be considered homologous. This case is not allowed in our model.

2.2.2 Generation and refinement of seed alignments

PhyLAT identifies candidate alignments between the query and the database using a BLAST-like seeded alignment approach that is designed for pairwise alignment of two sequences. Each candidate local alignment becomes an input to the general problem described above.

To generate candidates, we first reduce the multiple alignment M to its most likely ancestral sequence, using the aforementioned Tamura-Nei-like mutation model. We first hash all 20-mers in the ancestral sequence, then go through the query to find all locations of 20-mers which are present in the hash table. We choose 20 because this length is suitable for whole-genome alignment and is also the default seed length in MegaBLAST [177]. We then perform ungapped extension at each seed location using the DNAPAM-50 scoring matrix [28, 40]. We chose DNAPAM-50 according to the distances among the species in the multiple alignment used in our experiments; other matrices could be used for other databases. We retain those seeds whose ungapped alignment scores pass an E-value threshold of 10 as determined by ungapped Karlin-Altschul statistics [68]. These seeds are passed to the next phase of gapped extension.

2.2.3 Computation of per-column probability

There are many scoring functions [168, 175], but none of them take into account phylogenetic information. Assuming that the residues of q are stochastically independent and so are the columns of M, we have

$$\Pr(q, M|A, \tau^*) = \prod_{j=1}^{|A|} \Pr(y[j], Z[j]|\tau^*) \cdot \prod_{y \notin q'} \Pr(y) \prod_{Z \notin M'} \Pr(Z|\tau) \qquad (2.5)$$

where q' and M' form the alignment in A, and τ^* is a tree obtained by adding q to a branch in the tree τ. When the placement of the query in the phylogeny is known, this information can be used as *a priori* knowledge in our algorithm. When this is unknown, we set the initial length of the branch leading to the query to 0.1, and set the placement of the new branch on the middle point of the branch where the new branch is placed.

Computation of per-column probabilities in the original phylogeny

In order to keep track of insertions and deletions in the phylogeny, for each node in the phylogeny, we define several probabilities of observing the residues at leaves due to different evolutionary histories. For convenience, we use the following definitions and notations.

Definitions and Notations: We use c and r to denote a node in the phylogeny and the residue at the node, respectively.
Z_r denotes the residues at leaves.
$Pr_{ID}(leaves = Z_r|c = r)$ denotes the probability of observing Z_r with only insertions followed by deletions and substitutions allowed.
$Pr_S(leaves = Z_r|c = r)$ denotes the probability of observing Z_r with only substitutions allowed.
$Pr_D(leaves = Z_r|c = r)$ denotes the probability of observing Z_r with only deletions and substitutions allowed.
$sub(X) = \{A, C, G, T\}$ for $X \in \{A, C, G, T\}$
$sub(X) = -$ for $X = -$
$ins(X) = \{A, C, G, T\}$ for $X = -$

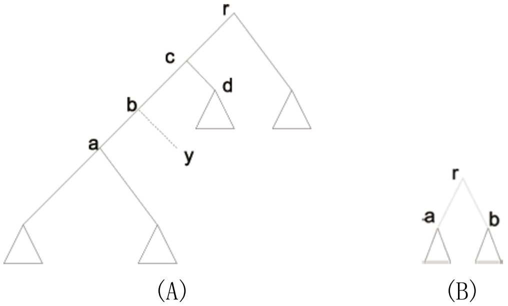


Figure 2.4: (A) An augmented phylogeny. (B) The original phylogeny. The probabilities at the node r can be computed from probabilities at its two children.

$ins(X) = \emptyset$ for $X \in \{A, C, G, T\}$
$del(X) = \{-\}$ for $X \in \{A, C, G, T\}$
$del(X) = \emptyset$ for $X = -$

We compute Pr_{ID}, Pr_S and Pr_D recursively as following.

$$\begin{aligned} & Pr_S(Z_r|r) \\ = & \sum_{a \in sub(r)} \sum_{b \in sub(r)} Pr_S(a|r) Pr_S(b|r) Pr_S(Z_a|a) Pr_S(Z_b|b) \end{aligned} \tag{2.6}$$

$$\begin{aligned} & Pr_D(Z_r|r) \\ = & \sum_{a\in sub(r)} \sum_{b\in sub(r)} Pr_S(a|r)Pr_S(b|r)Pr_D(Z_a|a)Pr_D(Z_b|b) \\ + & \sum_{a\in sub(r)} \sum_{b\in del(r)} Pr_S(a|r)Pr_D(b|r)Pr_D(Z_a|a)Pr_S(Z_b|b) \\ + & \sum_{a\in del(r)} \sum_{b\in sub(r)} Pr_D(a|r)Pr_S(b|r)Pr_S(Z_a|a)Pr_D(Z_b|b) \\ + & \sum_{a\in del(r)} \sum_{b\in del(r)} Pr_D(a|r)Pr_D(b|r)Pr_S(Z_a|a)Pr_S(Z_b|b) \end{aligned} \tag{2.7}$$

$$\begin{aligned} & Pr_{ID}(Z_r|r) \\ = & \sum_{a\in sub(r)} \sum_{b\in sub(r)} Pr_S(a|r)Pr_S(b|r)\cdot(Pr_{ID}(Z_a|a) \\ & Pr_M(Z_b|b) + Pr_M(Z_a|a)Pr_{ID}(Z_b|b) \\ & -Pr_M(Z_a|a)Pr_M(Z_b|b)) \\ + & \sum_{a\in ins(r)} \sum_{b\in sub(r)} \Pr_I(a|r)Pr_S(b|r)Pr_D(Z_a|a)Pr_S(Z_b|b) \\ + & \sum_{a\in sub(r)} \sum_{b\in ins(r)} Pr_S(a|r)\Pr_I(b|r)Pr_S(Z_a|a)Pr_D(Z_b|b) \\ + & \sum_{a\in sub(r)} \sum_{b\in del(r)} Pr_S(a|r)Pr_D(b|r)Pr_S(Z_a|a)Pr_S(Z_b|b) \\ + & \sum_{a\in del(r)} \sum_{b\in sub(r)} Pr_D(a|r)Pr_S(b|r)Pr_S(Z_a|a)Pr_S(Z_b|b) \\ + & \sum_{a\in del(r)} \sum_{b\in del(r)} Pr_D(a|r)Pr_D(b|r)Pr_S(Z_a|a)Pr_S(Z_b|b) \end{aligned} \tag{2.8}$$

Note that in the above equations r can be a residue or gap. We write so in order to give general forms. In specific cases, some terms in the equations will be empty.

Computation of per-column probability in augmented phylogeny

To compute the probability for an augmented phylogeny, we decompose the tree into three parts: the part above the augmented branch, the augmented brach with its neighboring branches, and the part below the augmented branch. These probabilities are precomputed and dynamically combined when computing the probability of the augmented phylogeny.

If $r \in \{A, C, G, T\}$,

$$\begin{aligned}
& Pr_{ID}(y, Z_r|r) \\
= \; & \sum_{c\in\Sigma}(Pr_D(Z_r^c|r)(\sum_{d\in del(c)} Pr_D(d|c)Pr_S(Z_d|d) \\
& + \sum_{d\in sub(c)} Pr_S(d|c)\Pr(Z_d|d)) \\
& (\sum_{b\in del(c)} (Pr_D(b|c)Pr_S(y|b) \sum_{a\in sub(b)} (Pr_S(a|b)Pr_S(Z_a|a))) \\
+ \; & \sum_{b\in sub(c)} (Pr_S(b|c)Pr_D(y|b)\sum_{a\in\Sigma}(Pr_D(a|b)Pr_D(Z_a|a)))))
\end{aligned}$$

If $r = -$,

$$
\begin{aligned}
& Pr_{ID}(y, Z_r|r) \\
= \; & \sum_{c \in \{A,C,G,T\}} [Pr_{ID}(Z_r^c|r)(\sum_{d \in \Sigma} \Pr(d|c) Pr_D(Z_d|d)) \\
& \cdot \sum_{b \in \Sigma} (\Pr(b|c) Pr_D(y|b) \sum_{a \in \Sigma} (Pr_D(a|b) Pr_D(Z_a|a)))] \\
+ \; & Pr_S(Z_r^c|c)((\sum_{d \in \{A,C,G,T\}} \Pr(d|c) Pr_D(Z_d|d)) \cdot \Pr(b = -|c) \\
& \Pr(a = -|b) \Pr(y = -|b) Pr_S(Z_a|a)) \\
+ \; & Pr_S(Z_r^c|c)(\Pr(d = -|c) Pr_S(Z_d|d)(\sum_{b \in \{A,C,G,T\}} \Pr(b|c) \\
& Pr_D(y|b) \sum_{a \in \Sigma} \Pr(a|b) \Pr(Z_a|a))) \\
+ \; & Pr_S(Z_r^c|c)(\Pr(d = -|c) Pr_S(Z_d|d) \Pr(b = -|c) \Pr(y|b) \\
& \Pr(a = -|b) Pr_S(Z_a|a)) \\
+ \; & Pr_S(Z_r^c|c)(\Pr(d = -|c) Pr_S(Z_d|d) \Pr(b = -|c) Pr_S(y|b) \\
& (\sum_{a \in \{A,C,G,T\}} \Pr(a|b) Pr_D(Z_a|a))) \\
+ \; & Pr_S(Z_r^c|c)(\Pr(d = -|c) Pr_S(Z_d|d) \Pr(b = -|c) Pr_S(y|b) \\
& \Pr(a = -|b) Pr_{ID}(Z_a|a))
\end{aligned}
$$

In the equation above, $\Pr(Z_r^c|r)$ denotes the probability of observing all the leaves in the tree rooted at r and considering the inner node c as a leaf. This is pre-computed using equations in the previous section. In this equation, the first term in the summation gives the probability of an insertion before c. The second term gives the probability of an insertion at (c, d). The third term gives the probability of an insertion at (c, b). The fourth term gives the probability of an insertion at (b, y). The fifth term gives the probability of an insertion at (b, a). The sixth term gives the probability of an insertion after a.

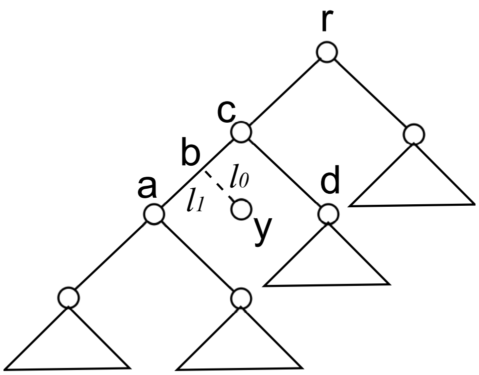


Figure 2.5: Computation of per-column probability. The probabilities of subtrees rooted at a and d are precomputed. We also precompute the probability of the tree rooted at r considering c as a leaf. During each EM iteration, we optimize l_0 and l_1. To compute the probability of the whole tree, we just need to combine the precomputed probabilities with the probabilities of edges (c, d), (c, b), (b, a), and (b, y).

Accelerating the per-column computation

Computing the probability associated with a leaf labeling of a large phylogeny, especially using the enhanced treatment of gaps described above, can be computationally expensive. To avoid enumerating all possible labelings, we developed a caching technique and a dynamic programming algorithm that can reduce the computational cost exponentially. To minimize the cost of this computation, we decompose the augmented phylogeny into subtrees for which we may precompute probabilities for every possible leaf labeling in a bottom-up fashion, then cache the probabilities for all such labelings in tables stored in the nodes of the augmented phylogeny. To compute a per-column probability, we combine the cached probabilities, which depend on the residues in the alignment column, with terms describing the contribution of the query residue, as illustrated in Figure 2.5.

Computational complexity

Subtree caching trades a nontrivial (but manageable) space cost for a substantial speedup gained by not having to recompute probabilities from scrach for each alignment column. The precomputed values are stored in tables of total size $O(|\Sigma|^{n+1})$ where n is the number

	-	A	C	G	T
-	$\cdot$	$\gamma\pi_A$	$\gamma\pi_C$	$\gamma\pi_G$	$\gamma\pi_T$
A	$\gamma\pi_-$	$\cdot$	$\beta\pi_C$	$\alpha\pi_G$	$\beta\pi_T$
C	$\gamma\pi_-$	$\beta\pi_A$	$\cdot$	$\beta\pi_G$	$\alpha\pi_T$
G	$\gamma\pi_-$	$\alpha\pi_A$	$\beta\pi_C$	$\cdot$	$\beta\pi_T$
T	$\gamma\pi_-$	$\beta\pi_A$	$\alpha\pi_C$	$\beta\pi_G$	$\cdot$

Table 2.1: Mutation rate matrix Q. π_r is the background frequency for residue r in M.

of species in the tree. Using a bottom-up approach to fill the tables for each node in the tree, precomputation requires only constant time per table cell.

During the EM algorithm, the per-column probability computation for each augmented tree uses the precomputed values plus a computation at the augmented branch. In the E-step, the algorithm runs over all the columns in the alignment A for each possible augmented tree, so the time cost is just $O(n \cdot |A|)$. In the M-step, the algorithm uses banded linear-space Smith-Waterman. The time cost is $O(W \cdot |q|)$, where W is the width of the band and q is the query.

2.2.4 Mutation model

The mutation matrix used in our method is shown in Table 2.1. The three free parameters α, β, and γ correspond to instantaneous rates of transitions, transversions, and indels. Although our mutation rate matrix is time-reversible, our algorithm to estimate the parameters in this model does not rely on this property, nor does our probabilistic model for computing alignments between q and M. Hence, our algorithms can be directly used on general non-time-reversible models [174].

2.3 Experimental Results

We have implemented PhyLAT in the C++ language, using the TAO optimization package [10] to estimate maximum-likelihood values for the edge-length parameters l_0 and l_1 shown in Figure 2.2. In this section, we interrogate the result quality of PhyLAT.

We tested PhyLAT's accuracy on three queries: human chromosome 22, *C. elegans* chromosome 3, and *D. melanogaster* chromosome 4, each aligned to a database of multiple genome alignments for related species. Here, we present only the human results, which are representative of our tool's qualitative performance vs. competing aligners; the other experiments' results are described in our supplementary material. We aligned human chromosome 22 (assembly hg19, GRCh37) against a whole-genome alignment of five mammals (shown in Figure 2.6) from the UCSC genome database [20, 72], which was assembled using human chromosome 22 as the reference sequence. We also tested the accuracy of tree placement by aligning opossum to a different five-species tree from the UCSC database.

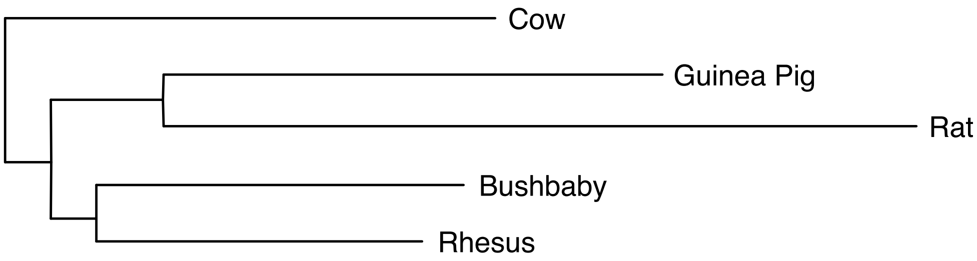


Figure 2.6: Phylogeny of the species in the multiple alignment. Branch lengths are proportional to evolutionary distances.

Accuracy of DNA alignment of human chromosome 22

Currently, there are no good methods to evaluate absolute accuracy of arbitrary multiple alignments of DNA sequence [73, 76, 127]. However, protein-coding regions are generally stable and can be translated to protein and aligned by a protein aligner, producing alignments of generally higher quality than those obtained from DNA alone. We therefore

#sequences	#alignments	#orthologous	%orthologous	%identical
2	982	797	81.16%	53.65%
3	383	316	82.51%	46.89%
4	553	477	89.49%	52.89%
5	726	704	96.97%	59.38%
Total	2644	2294	86.76%	53.83%

Table 2.2: Effect of using multiple alignment on improving orthology detection. *# sequences*: number of species present in the aligned multiple alignments. *# alignments*: number of PhyLAT alignments. *# orthologous*: number of orthologous PhyLAT alignments. *% orthologous*: # orthologous/# alignments. *% identical*: percentage of PhyLAT columns containing identical bases. The more database species present at a locus, the greater the percentage of alignments involving orthologous sequences.

validated PhyLAT's alignment quality by examining local alignments involving annotated coding sequences. We used the UCSC database's reference alignment between human and the other five species as our ground truth for orthology relationships among our sequences.

Table 2.2 illustrates one benefit of using multiple species for recovering alignments between the query and *orthologous* sequences in the database. We divided the alignments found by PhyLAT according to the number of species (up to five) with sequence at the locus of the alignment in the database. The more species present in the database at a given locus, the higher the probability that an alignment at that locus aligns the query to orthologous sequences. We note that alignments with more species present are not systematically better-conserved than those with fewer species; indeed, aligned regions with only two aligned sequences had higher identity on average than those with three or four. Nevertheless, the fraction of query sequences aligned to their orthologous regions in the multiple alignment increased monotonically with the number of species present.

Validation of DNA alignment by protein alignment

To estimate the likely accuracy of PhyLAT's alignments in coding DNA, we extracted and translated the sequences it aligned, then used a protein multiple aligner to realign them, and finally checked whether the DNA alignment inferred from the aligned proteins matched

PhyLAT's alignment. Because protein alignment uses information not available to a DNA aligner, we expect that it will yield more accurate results in general; hence, concordance between the DNA and protein alignments acts as a proxy for the (unknown) absolute accuracy of the DNA alignment.

From PhyLAT's DNA alignments involving orthologous sequences, we first extracted those portions that covered protein-coding regions (as annotated in the UCSC database). For each such alignment between a DNA query q and a multiple alignment of k DNA sequences $s_1 \ldots s_k$, PhyLAT's output induces pairwise protein alignments A_i between the translation of q and that of each s_i. We compared the induced alignments A_i to alignments A_i' obtained by first translating q and s_i independently, then aligning the two resulting protein sequences using a protein-specific alignment tool. A codon in a query was considered "accurately aligned" to the database if and only if, for $1 \leq i \leq k$, A_i agreed with the corresponding, independently derived A_i' over that codon. We repeated this experiment using four different protein aligners – ClustalW [159], DIALIGN [112], Muscle [48], and T-Coffee [117] – and obtained substantially similar results with each. Additional validation would be possible by comparing our results to, e.g., structural superposition of the aligned proteins. However, such superpositions are already known to agree closely with protein aligners' output on mammalian proteins [14], so we did not pursue this extra validation step.

The first part of Table 2.3 shows PhyLAT's accuracy on our test set using ClustalW as the protein aligner. Over 97% of query codons aligned by the algorithm were accurately aligned to the database by our measure. Moreover, PhyLAT's alignments covered more than 99% of all annotated codons in the multiple alignment, so this accuracy applies to essentially all the coding sequence that could possibly be aligned.

We further subdivided the protein-coding region of the database to identify regions where the protein multiple alignment induced by the DNA multiple alignment of $s_1 \ldots s_k$ was inconsistent with the result obtained by independently translating each of $s_1 \ldots s_k$, then aligning the resulting sequences using a protein multiple aligner. Such regions are more likely to be misaligned in the database, which in turn provides bad information to PhyLAT's aligner. For the 84% of codon positions in the database that were consistent by the above criterion, PhyLAT's accuracy was well over 99%.

An alternative to PhyLAT's approach would be to align the query to a single, representative DNA sequence instead of a DNA multiple alignment. For example, one might align our human query to one of the multiple alignment's component species' genomes, or to the evolutionary consensus of these genomes given the tree. We therefore investigated whether such pairwise alignments, as realized by the widely used BLAST software (v2.2.23+) [30], could match the accuracy and coverage obtained by PhyLAT.

It was not computationally feasible to BLAST the entirety of human chromosome 22 at once against a database sequence as long as our multiple alignment. Instead, for each homologous PhyLAT alignment of query segment q and database segment M, we extracted the collinear block B in UCSC's multispecies multiple alignment that contained q and M. We then used BLAST to align q to each individual sequence in B, or to its evolutionary consensus. If BLAST returned more than one local alignment between a query and a block, then we retained all such alignments. Finally, we evaluated the collection of induced BLAST alignments in protein coding regions of the query using the same accuracy and coverage measures described above. Note that accuracy for a pairwise BLAST alignment of two coding DNA sequences is determined by agreement with a single pairwise protein alignment between them, whereas for PhyLAT, *all* induced pairwise alignments must agree with the protein aligner's results.

The second part of Table 2.3 shows the results of using BLAST pairwise alignments, rather than PhyLAT's approach, on our human to mammalian alignment task. For species other than rhesus, the closest to the human query, per-codon accuracy of the pairwise alignments was inferior to PhyLAT's. Aligning to the consensus actually lowered accuracy compared to two-species alignments. Moreover, the pairwise alignment sets covered fewer codons in the original multiple alignment than did PhyLAT's output. This lower coverage arises because not all species had sequence at every point in the reference multiple alignment. Hence, even aligning human to rhesus, which produced alignments to nearly 100% of the codons in the rhesus sequence, yielded less than 92% coverage of all codons in the multiple alignment.

Overall, PhyLAT produced alignments with accuracy comparable to using BLAST to search against the best single reference species from the multiple alignment, while offering substantially improved coverage because of the availability of multiple species to cover assembly or homology gaps left by any one species' genome.

We also compared PhyLAT with other commonly used multiple alignment tools, including POY, MAFFT [71], MUSCLE , CLUSTAL, and PaPaRa [12]. Because these programs produce multiple alignments, which include all input sequences, it is not proper to feed the whole reference sequences and query to them to produce genome-scale multiple alignments. Instead, we use homologous segments from the reference sequences and the query where PhyLAT finds alignments. The results are shown in Table 2.3. Note that because we do global alignments on the input, these aligners aligned all the input codons.

Tree placement of human sequences

Another measure of PhyLAT's accuracy is whether it placed each query sequence in its correct location on the tree of the species in the database. For the local alignments of orthologous sequences in our test set, EM should place the human query in its accepted location relative to the other, non-human mammalian species with a high posterior probability while assigning low probabilities to incorrect placements.

Although some methods are available for comparing two trees with branch lengths [122], there is not an acknowledged standard on correct branch lengths for a given phylogeny. We therefore assessed only whether the most likely placement of the query in each local alignment was topologically correct, i.e. was the human sequence placed on the branch leading to rhesus, or to some other branch?

Figure 2.7 shows how many alignments placed the human sequence on each branch of the phylogeny with highest probability. 88.7% of alignments correctly placed the human sequence adjacent to rhesus. If we add in "almost correct" placements (defined as branch placement adjacent to the correct one), the fraction of such placements rises to 95.2%.

We also compared PhyLAT's accuracy with that of tools whose results include a branch placement for the query sequence, including POY, RAxML [151], EPA [11], pplacer [104] and PaPaRa [12]. Because all these programs need multiple alignments to do prediction, we used only orthologous informant sequences and queries as the input. Because gene trees may be different from species trees, in order to assess branch placement accuracy, we also divided the placements into two categories: those whose informant trees matched the trees built by PHYLIP [49] from MAFFT alignments of the sequences, and those whose informant trees

		#aligned	#accurate	accuracy	#total	coverage	coverage
						in species	in MA
PhyLAT	whole MA	16404	15956	97.27%	16445	-	99.75%
	consistent MA	13487	13414	99.46%	-	-	-
	inconsistent MA	2917	2542	87.14%	-	-	-
BLAST	Cow	12384	11607	93.73%	15129	81.86%	75.31%
	Guinea Pig	12306	11667	94.81%	16159	76.16%	74.83%
	Bushbaby	10732	9748	90.83%	12527	85.67%	65.30%
	Rhesus	15066	14700	97.57%	15077	99.93%	91.61%
	Rat	11503	10597	92.12%	15787	72.86%	69.95%
	consensus	16174	14420	89.16%	16445	-	98.35%
POY	whole MA	16445	14556	88.51%	16445	-	100.00%
	consistent MA	13502	12423	92.01%	-	-	-
	inconsistent MA	2943	2133	72.48%	-	-	-
MAFFT	whole MA	16445	15600	94.86%	16445	-	100.00%
	consistent MA	13502	13122	97.19%	-	-	-
	inconsistent MA	2943	2442	82.98%	-	-	-
MUSCLE	whole MA	16445	15181	92.31%	16445	-	100.00%
	consistent MA	13502	12911	95.62%	-	-	-
	inconsistent MA	2943	2270	77.13%	-	-	-
CLUSTAL	whole MA	16445	15238	92.66%	16445	-	100.00%
	consistent MA	13502	13044	96.61%	-	-	-
	inconsistent MA	2943	2194	74.55%	-	-	-
PaPaRa	whole MA	16445	15296	93.01%	16445	-	100.00%
	consistent MA	13502	13091	96.96%	-	-	-
	inconsistent MA	2943	2205	74.92%	-	-	-

Table 2.3: Comparison among PhyLAT alignments, BLAST pairwise alignments, and alignments of other phylogeny-aware tools. *# aligned*: total # of codons in query aligned to the database; *# accurate*: number of aligned codons in previous column that are aligned the same by DNA and protein aligners; *accuracy*: ratio of accurate to aligned codons; *# total*: total number of codons present in the indicated sequence; *coverage in species*: total number of codons in species' sequence covered by query alignments; *coverage in MA*: total number of codons in entire multiple alignment database covered by query alignments.

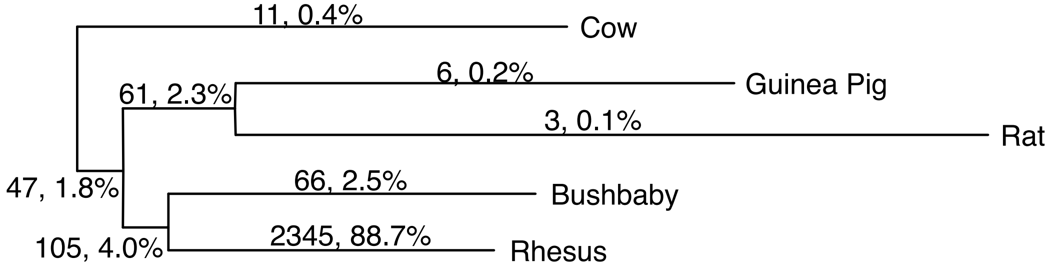


Figure 2.7: Tree placements for all human query sequences. The correct location of the query is on the branch leading to rhesus. Each branch is labeled with the number of queries placed on that branch, as well as the percentage of all queries that this number represents.

	#correct in congruent	#correct in incongruent	#total correct	overall accuracy
PhyLAT	1452	641	2093	91.72%
POY	1	0	1	0.04%
RAxML	669	599	1268	55.57%
EPA	650	634	1284	56.27%
pplacer	731	695	1426	62.49%

Table 2.4: Tree placement of orthologous human query sequences. There are 1558 congruent informant trees and 641 incongruent informant trees.

were incongruent with their PHYLIP trees. The results are shown in Table 2.4. PhyLAT's placement accuracy was substantially greater, both absolutely and relative to its competitors, when the informant sequences matched the supplied phylogeny. We note that POY has only one correct placement; this is because it builds an entirely new tree on the input sequences instead of just inserting the query species into the existing tree of informant species. We provided the informant phylogeny as input restrictions on the tree topology, but POY used it only as a starting tree and failed to produce output trees consistent with these restrictions.

We further investigated how confident PhyLAT typically was about its branch placements. A confident placement has the vast majority of the probability mass, with little probability assigned to other hypotheses, while an low-confidence placement distributes the probability more equally across branches. We computed the entropy for the posterior placement distribution of each query, summarizing these entropies in a histogram in Figure 2.8. For most correct branch placements, PhyLAT was highly confident about its predictions, while

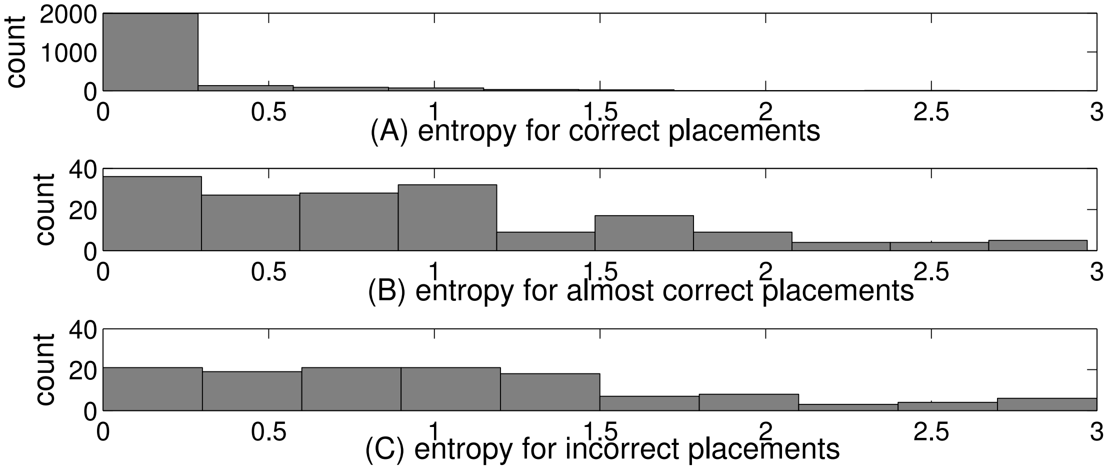


Figure 2.8: Histograms of entropies (in bits) of posterior branch placement distributions. Because there are 8 possible values for the branch placement, the entropy is in interval [0,3]. The smaller the entropy, the more concentrated the probability distribution, and the more confidence PhyLAT has in the branch placement.

confidence for incorrect predictions was typically lower. For almost-correct placements, the ratio of high- to low-confidence placements is close to even. We could detect and reject most incorrect placements, with relatively few false rejections, by rejecting any placement with an entropy over 0.25 bits.

The absolute accuracy of tree placement for our experiments on *C. elegans* and *D. melanogaster* was considerably lower than for our mammalian alignment – between 40 and 50%. However, as in the mammalian case, PhyLAT's results were more accurate than those of competing tools that gave placement information in their output. Details may be found in our supplementary material.

Tree placement of the opossum species

In spite of the fact that phylogenetic relations of existing species have been explored extensively, many relations remain missing, and many are being modified constantly. In a recent update of the phylogenetic tree of 46 species from the UCSC database, 35 species changed their tree placements [130]. One example of such a change was the movement of opossum from relatively near the root of the mammalian phylogeny to a location much closer to other marsupials such as wallaby.

As a further test of PhyLAT on a different data set, we aligned opossum chromosome X to a 5-species multiple alignment from the UCSC genome database. PhyLAT produced 931 local alignments, with branch placements of the opossum queries as shown in Figure 2.9. Assuming, as in the revised UCSC tree, that the correct placement is on the branch to wallaby, 63.3% of queries were placed correctly, while 81.3% were placed correctly or almost correctly. In contrast, the number of queries placed at opossum's old location in the tree was only 6.3%. This example shows that PhyLAT's placement probabilities can be useful for discovering inconsistencies with an accepted phylogeny.

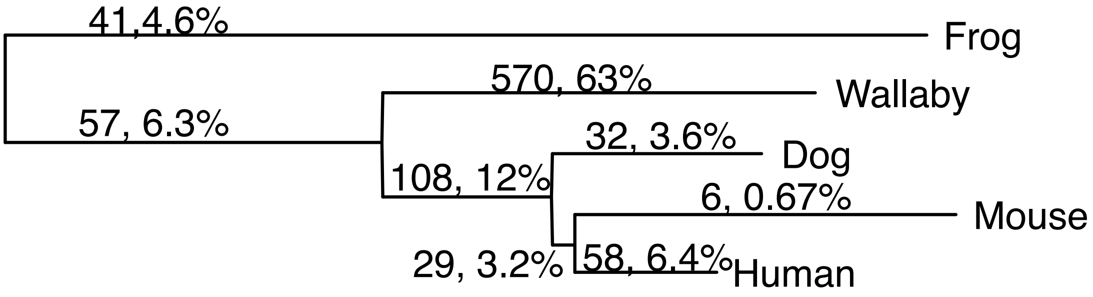


Figure 2.9: Tree placement for opossum. Branch lengths are proportional to evolutionary distances. The correct location of the query is on the branch leading to wallaby. Opossum had until recently been placed on the branch leading to the parent of wallaby. Each branch is labeled with the number of queries and the fraction of all queries placed on it.

2.4 Conclusion

This chapter introduced our PhyLAT alignment framework. This framework will be used in all of our rest three problems studied in the following chapters. The core of our alignment framework is using a multiple alignment as a reference sequence and using a phylogeny as the basis for the scoring system. These two concepts, multiple alignment and phylogeny, are important and fundamental in bioinformatics field, but they are seldom dealt with together. Our model combines them together. With this biologically realistic model, we developed the EM optimization algorithm. Our algorithm can efficiently align a sequence to a multiple alignment and simultaneously predict the branch location of the query sequence. Experiments strongly suggest that our framework is better than other methods in terms of alignment accuracy and tree placement accuracy [155].

Chapter 3

Multiple Alignment Improvement

3.1 Problem Introduction

Sequence alignment is a prerequisite to nearly all downstream comparative genomic analyses, including the identification of conserved sequence motifs, estimation of evolutionary distance between sequences, and inference of evolutionary history of genes and species. Errors in sequence alignment are found to have a significant negative effect on subsequent inference of sequence divergence, phylogenetic trees, and conserved motifs [76].

While there are many tools for constructing multiple alignment, there are few for refining existing multiple alignments. Existing tools may be fast and give a good initial multiple alignment, but the alignment quality can be improved by using more complex model, i.e., a biological realistic and probabilistic model. It was shown that deletions in sequences will result in errors in several multiple sequence alignment tools using non-probabilistic scoring schemes, i.e., ClustalW, DIALIGN-T, MAFFT, MUSCLE, PROBCONS, T-COFFEE, and PRANK [55, 119]. It was also shown that iterative methods are more accurate than non-iterative methods [119]. Here we gave a method for iteratively refining a multiple alignment by using phylogenetic information in a probabilistic scoring scheme.

Our work in Chapter 2 shows that simultaneously aligning a query sequence to a multiple alignment reference and inferring the query's tree placement will give more accurate results on both problems than doing separately [155]. But aligning one sequence to a reference is different from improving a multiple alignment. The most important difference is that these two tasks require very different computational resources. Aligning a query to a multiple alignment assumes that the multiple alignment is accurate; thus it is a pairwise alignment whose complexity is quadratic. But for multiple alignment improvement problem, the whole alignment is subject to be optimized. The computational complexity is exponential in the number of sequences. We cannot simply enumerate all possible cases. There will also be suboptimal alignments with locally maximal scores.

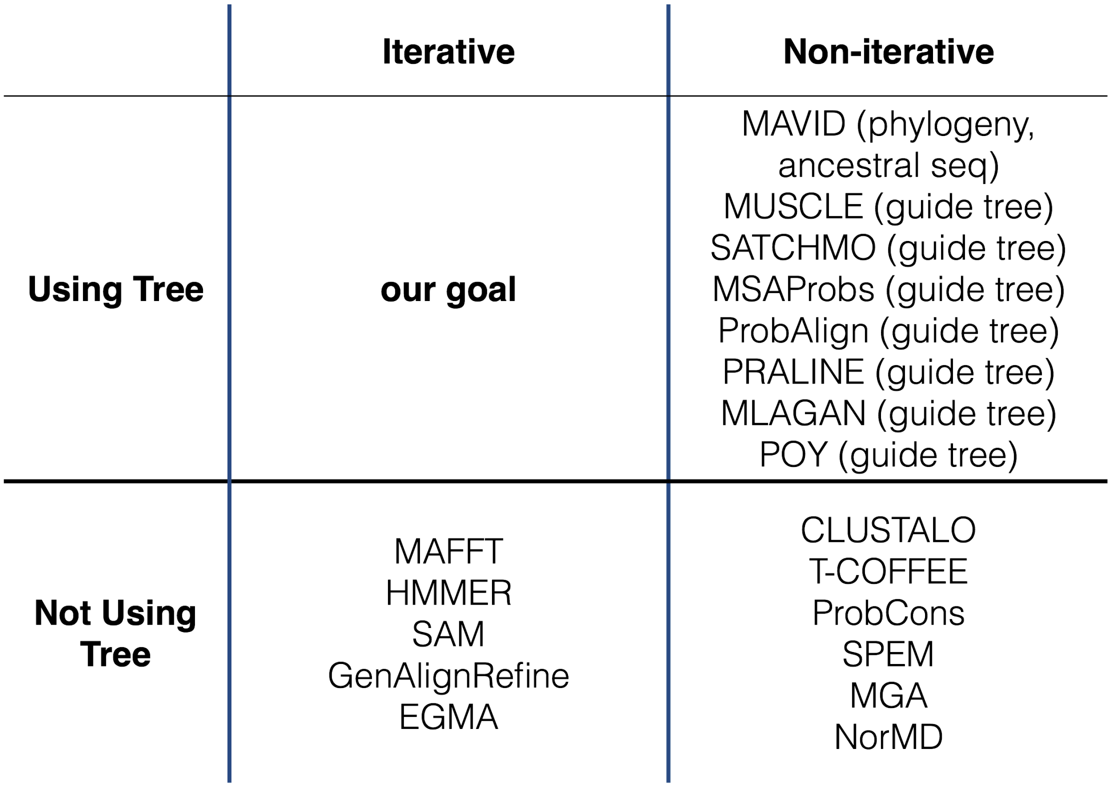
We developed an iterative algorithm which combines the power of iterative realignment process and the power of a phylogeny-based probabilistic scoring system. The iterative process makes our algorithm very steady and insensitive to the realignment order, while the probabilistic scoring scheme make the detection of column-wise homology much more sensitive than other algorithms which do not use phylogenetic information or probabilistic scoring systems like ClustalO, MAFFT, and MAVID [24, 71, 146].

3.2 History, Applications, and Existing Tools

Before fast sequencing techniques were developed, there are few whole genomes available. For example, the genomes online database (GOLD), which is a comprehensive resource for accessing information related to completed and ongoing genome projects world-wide, had only six complete genomes and a handful of ongoing genome projects when it was establisted in 1997. Now it has 60853 genome projects and 58153 complete genomes [16]. With the fast increasing number of genomes available, building reliable multiple sequence alignment is needed. Genome rearrangement events are common between different species. For example, it was shown that 9% of human/mouse homology may be attributed to small rearrangements [26]. Thus people developed glocal alignment algorithms which try to align all parts of genomes from different species. Traditional local alignment and global alignment algorithms cannot do this. With glocal alignment algorithms, researchers can find homologous segments from different species, even if the segments are not in the same order. Thanks to glocal alignment algorithms like MAVID and MultiZ [20, 24], UCSC genome database [51] contains multiple alignments for 67 vertebrates.

With a better multiple sequence alignment database, other downstream researches, e.g. conserved element finding, gene prediction, and phylogeny reconstruction, can be improved. It was shown that using multiple alignment as informant, one can achieve higher sensitivity in finding gene-coding areas [27]. Multiple alignment database can also be used to find transcription factor binding sites [120]. Another application of multiple sequence alignment is ancestral sequence inference [24]. Higher-quality multiple sequence alignment database will allow more accurate detection of homology and better understanding of genome rearrangement events in query sequences or newly sequenced genomes [26]. Our work in chapter 2 showed that there are errors in these multiple alignments in coding regions [155].

Figure 3.1 lists popular multiple alignment algorithms and classifies them according to their scoring scheme and alignment procedure. We can see that most algorithms are non-iterative and therefore are unable to correct errors occurred in previously aligned alignments. This might lead to lower-quality alignments. Those algorithms which are iterative do not use probabilistic scoring schemes. Most of them use empirical parameters for scoring the alignments, resulting in scores which are not meaningful in the sense of probability. HMM-based models suffer from high time complexity and extra efforts for estimating large number of parameters. Iterative algorithms are not commonly applied to improve genome-scale multiple alignments. Our goal lies in the category of non-HMM iterative probabilistic algorithms.



	Iterative	Non-iterative
Using Tree	**our goal**	MAVID (phylogeny, ancestral seq) MUSCLE (guide tree) SATCHMO (guide tree) MSAProbs (guide tree) ProbAlign (guide tree) PRALINE (guide tree) MLAGAN (guide tree) POY (guide tree)
Not Using Tree	MAFFT HMMER SAM GenAlignRefine EGMA	CLUSTALO T-COFFEE ProbCons SPEM MGA NorMD

Figure 3.1: Existing multiple alignment algorithms

Gibbs sampling has also been applied to multiple alignment [83, 97]. There are variations of Gibbs sampling when it is applied to multiple alignment. Sampling can be applied to a local part of an alignment or a whole sequence in the alignment. Sampling can also be applied to leaves in a phylogeny or internal nodes. One problem with Gibbs sampler is it may still be trapped at local optima. Another problem is it performs well only when there is a clear block of ungapped alignment shared by all of the sequences, and performs poorly on general sets of test cases when compared with global alignment methods [117, 160]. Futhermore, it was shown that Gibbs sampling could never align more than 10 sequences in an experiment trying to align kinase sequences [117]. The common background probability distribution of letters in alignment is easily affected by background noise. Gibbs sampling has been coupled with higher-order probabilistic distribution [158] to refine the probability distribution. However, this introduces extra time complexity without guaranteed improvements.

Phylogeny-aware scoring scheme has also been explored in the Gibbs sampling framework [144]. However, such methods have several problems. First, they do not use point mutation model. This leads to incomplete use of the phylogenetic information in the phylogeny. Second, they do not use variable branch lengths in phylogeny. This also reduced the effectiveness of the phylogenetic information. Third, they only use time-reversible models, which makes the scoring scheme restrictive and inaccurate. Fourth, They use fixed phylogeny, which means the structure of the phylogeny must be known. Thus they cannot correct errors in tree structures. Fifth, when computing alignment scores they include invalid indel histories. This will also result in inaccurate alignment scores. Sixth, they use unrooted phylogeny, which is not realistic. Seventh, they do not use affine gap penalty, which also leads to inaccurate scores. [59, 66, 82, 93, 99, 100, 106, 115, 128, 144, 154, 168]. Last but not the least, appropriate parameters in Gibbs-sampling-based alignment algorithms can only be found with experience gained from careful analysis of the sampling results [106].

For our problem, we want to improve existing multiple alignments where all the aligned sequences are supposed to be orthologous, possibly containing indels. We want to use a biologically more realistic model, i.e., rooted phylogeny, non-time-reversible mutation model, affine gap penalty, only valid indel history, and variable-length phylogeny. In this case, a phylogeny-aware probabilistic global alignment algorithm may outperform existing Gibbs-sampling-based methods.

3.3 Problem Formulation

Given a multiple alignment M of n sequences $s_1 ... s_n$ from species $1 ... n$, and a phylogeny τ associated with the n species, we want to find the optimal multiple alignment by iteratively refining M. In each iteration, we take out a sequence s_i from M. By taking out s_i, the reduced alignment is M_i, and the reduced phylogeny is τ_i. We realign s_i to M_i using τ_i, constraining species i on its original branch in τ allowing its distance to its parent and children variable. In other words, we want to find a new alignment A_i^* of M_i and s_i and a new phylogeny τ_i^*, which maximize the following probability:

$$Pr(M_i, s_i | A_i^*, \tau_i^*), \tag{3.1}$$

or equivalently the following:

$$\log Pr(M_i, s_i | A_i^*, \tau_i^*). \tag{3.2}$$

We iteratively do this for each sequence in M until this probability does not increase. We observe that the alignment maximizing the sum 3.2 does not change if we subtract from this sum the constant

$$C' = \log(Pr(s_i) \cdot Pr(M_i | \tau_i)). \tag{3.3}$$

Considering each column in M_i independent, maximizing 3.2 is to maximize

$$\begin{aligned} \log Pr(M_i, s_i | A_i^*, \tau_i^*) - C' &= \log \frac{Pr(M_i, s_i | A_i^*, \tau_i^*)}{Pr(s_i) \cdot Pr(M_i | \tau_i)} && (3.4) \\ &= \sum_j \log \frac{Pr(y_j, Z_j | \tau_i^*)}{Pr(y_j) \cdot Pr(Z_j | \tau_i)}. && (3.5) \end{aligned}$$

where y_j and Z_j are the jth column in alignment A_i^*.

For the details of computation of the tree probabilities, please see Chapter 2.

Because the resulting alignment may not be the correct one, it is preferable to measure the uncertainty in the given alignment, and possibly give suboptimal alignments as well. Because we use probabilistic scoring scheme, paths in our Smith-Waterman-like alignment process can be easily viewed in a probabilistic way. We can give suboptimal alignments by tracing back along the suboptimal paths.

3.4 Results

We developed our iterative probabilistic multiple alignment improvement algorithm called PhyMAIT, which stands for **Phy**logenetic **M**ultiple **A**lignment **I**mprovement **T**ool. To improve a given multiple alignment, PhyMAIT iteratively takes out a sequence from a multiple alignment according to a given strategy and realigns the sequence back to the rest of the multiple alignment. During each iteration of the improvement, one sequence is picked out and realigned to the rest of the multiple alignment. Different strategies can be used for picking out sequences. The strategy for picking out a sequence can be any user-defined strategy. It can be as simple as Round-Robin or a very complex one. We used three strategies. The first one is first picking out those sequences which are more distant from the rest of the species. The rationale behind this is that the most-distant sequence may be aligned the least accurate because it is the least similar sequence to the rest. The second one is first picking out those sequences which are less distant from the rest of the species. The rationale behind this is that the closest sequences are the core of the alignment whose correctness may have more impact on the rest of the alignment. The third one is randomly picking out sequences. The rationale behind this is that random walk may avoid being trapped at local optimal solutions. The ideal case is that all these strategies give the same result when they converge to the same optimal alignment. We will discuss the effect of using different picking strategies in section 3.5. This process iterates until the quality of the multiple alignment does not change. During each iteration, an EM-like algorithm is employed to simultaneously compute an alignment between the taken-out sequence and the remaining multiple alignment and optionally estimate the placement of the taken-out species on the tree associated with the remaining multiple alignment. The EM algorithm iteratively refines the alignment and branch placement until both have converged. The structure of PhyMAIT is illustrated in Figure 3.2.

We first tested PhyMAIT's accuracy on three multiple alignment databases: a multiple alignment database of 6 mammalian sequences, a multiple alignment database of 6 worm sequences, and a multiple alignment database of 6 fruitfly sequences. All the three databases are from UCSC database [20, 72]. The phylogenies relating the species in the three databases are shown in Figure 3.3, Figure 3.4 and Figure 3.5. The mammalian multiple alignment was

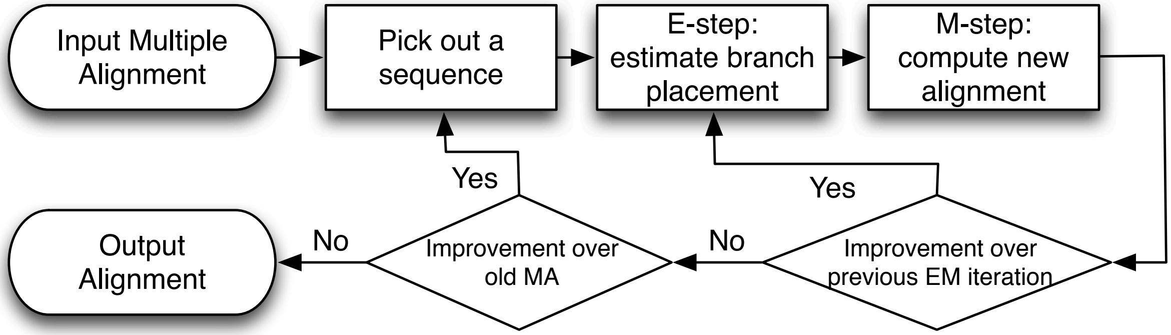

Input Multiple Alignment
Pick out a sequence
E-step: estimate branch placement
M-step: compute new alignment
Yes
Yes
Output Alignment
No
Improvement over old MA
No
Improvement over previous EM iteration

Figure 3.2: Structure of PhyMAIT algorithm. Not shown is the offline preprocessing of the database to parameterize a mutation model at each position.

assembled using human chromosome 22 as the reference sequence. The worm multiple alignment was assembled using *C.elegans* chromosome 3 as the reference sequence. The fruitfly multiple alignment was assembled using *D.melanogaster* chromosome 4 as the reference sequence.

	date assembled	alignment length	#codons
mammals	Dec 2013	23133033	184317
worms	Oct 2010	6253062	814609
fruitflies	Aug 2014	1701564	56444

Table 3.1: Datasets used in PhyMAIT experiments. Alignment length is the total length of the multiple alignment database. #codons is the number of codons used in protein validation process.

Accuracy of DNA alignment

Currently, there are no good methods to evaluate absolute accuracy of arbitrary multiple alignments of DNA sequence [73, 76, 127]. However, protein-coding regions are generally stable and can be translated to protein and aligned by a protein aligner, producing alignments of higher quality than those obtained from DNA alone. We therefore validated PhyMAIT's

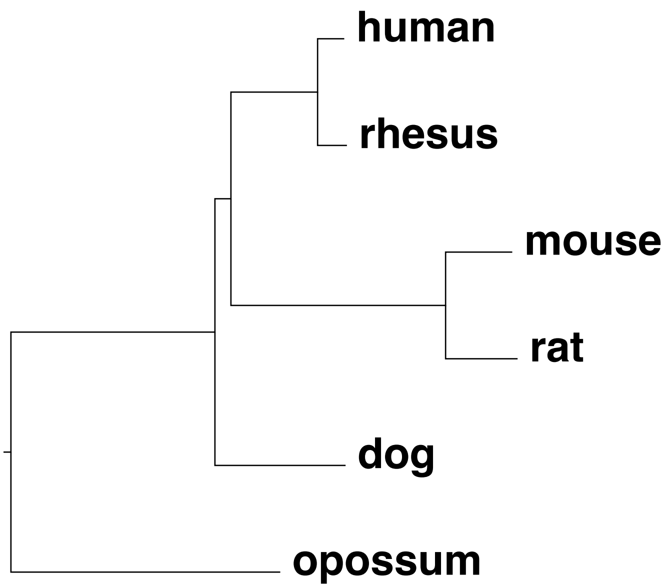


Figure 3.3: Mammalian tree.

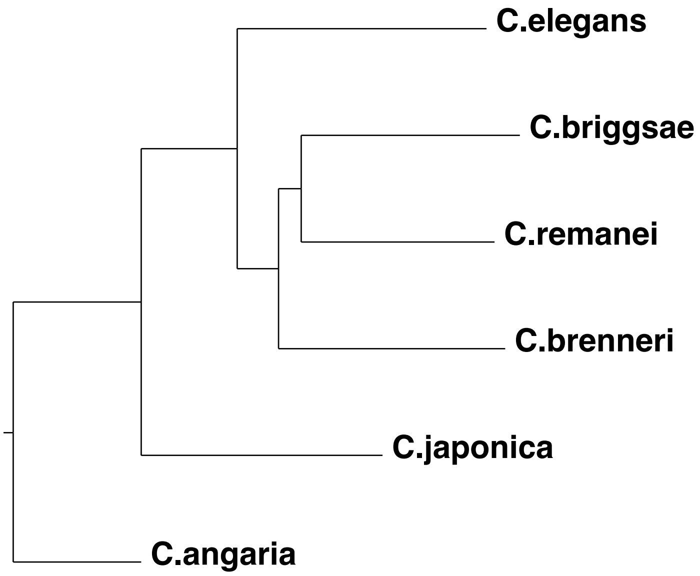


Figure 3.4: Worm tree.

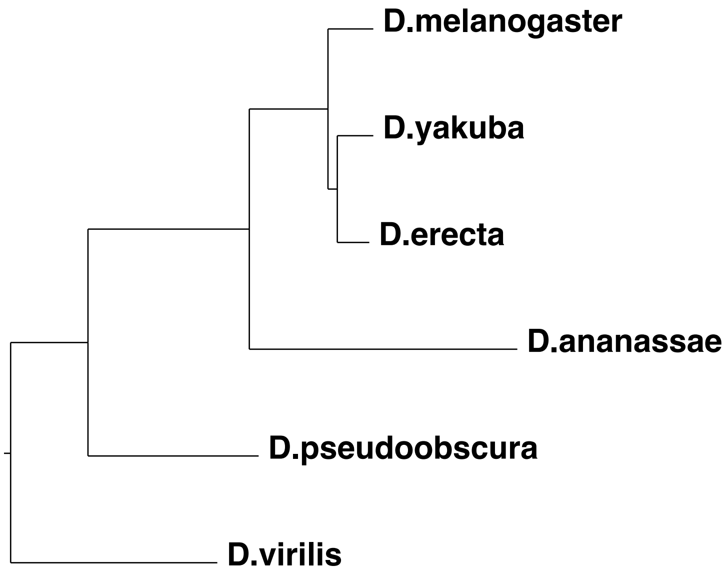


Figure 3.5: Fruitfly tree.

alignment quality by examining alignments involving annotated coding sequences. We used the exon alignments in the UCSC database as our ground truth for orthology relationships among our sequences.

Validation of DNA alignment by protein alignment

To validate our results, we realign the sequences from coding regions using protein alignment programs. If a base is aligned to the same bases in both DNA alignment and its corresponding protein alignment, then it is considered aligned correctly, or wrong otherwise. To estimate the likely accuracy of PhyMAIT's alignments in coding DNA, we used exon DNA alignments from the UCSC database as input and used their protein alignments as the standard. We checked whether the DNA alignments of PhyMAIT are consistent with the protein alignments. Because protein alignment uses information not available to a DNA aligner, we expect that it will yield more accurate results in general; hence, concordance between the DNA and protein alignments acts as a proxy for the (unknown) absolute accuracy of the DNA alignment.

		MAF	PhyMAIT	ClustalO	MAFFT	MAVID
human	c	93.85%	99.19%	93.03%	94.26%	94.10%
	i1	5.63%	0.52%	6.37%	5.13%	5.09%
	i2	0.52%	0.29%	0.60%	0.61%	0.81%
worm	c	91.02%	97.64%	90.37%	91.09%	90.79%
	i1	7.97%	1.91%	8.42%	7.95%	8.35%
	i2	1.01%	0.45%	0.81%	0.96%	0.86%
fly	c	92.41%	97.13%	91.82%	92.01%	91.96%
	i1	7.14%	2.59%	7.79%	7.63%	7.25%
	i2	0.45%	0.28%	0.39%	0.36%	0.79%

Table 3.2: Comparison of coding alignments from PhyMAIT, CLUSTAL Omega, MAFFT and MAVID, and the original UCSC MAF alignments. Lines "c", "i1" and "i2" denote consistently aligned codons, non-frame-shifted inconsistently aligned codons, and frame-shifted inconsistently aligned codons, respectively.

For each input alignment of k DNA sequences $s_1 \dots s_k$, PhyMAIT's output induces multiple protein alignment A of the translation of the k sequences. We compared the induced alignment A to alignment A' obtained by first translating the k sequences independently, then aligning the resulting protein sequences using a protein-specific alignment tool. A codon was considered "accurately aligned" in the DNA multiple alignment if and only if it was aligned to the same codons in A and A'. We created these ground-truth protein alignments using four different protein aligners – ClustalO [146], DIALIGN [112], Muscle [48], and T-Coffee [117] – and obtained substantially similar results with each. Additional validation would be possible by comparing our results to, e.g., structural superposition of the aligned proteins. However, such superpositions are already known to agree closely with protein aligners' output [14], so we did not pursue this extra validation step.

Table 3.2 shows PhyMAIT's accuracy on our test set using ClustalO as the protein aligner. Overall, PhyMAIT aligns more accurately than the original MAF file and other competitors.

We further examined the inconsistent parts of the alignments. We classified the inconsistently aligned bases into two classes. The first class contains those bases which were not aligned to the same bases as in the protein alignments without ORF shifting. The second class contains those bases which were not aligned to the same bases as in the protein alignments with ORF shifting. See Figure 3.8.

We validated our method on protein-coding regions of the multiple alignments. The human multiple alignment contains 125848 codons. The *C. elegans* multiple alignment contains 578447 codons. The *Drosophila* multiple alignment contains 90269 codons. Table 3.2 shows the alignment accuracies of PhyMAIT and other competitors including both iterative and non-iterative methods. We noted that these accuracies are not high considering we are aligning coding regions. We checked those CDS sequences from UCSC database, and found they actually contain ORF-shifted regions.

Impact of Adding Additional Species to Alignment

We also tested PhyMAIT's and the competing tools' ability to exploit additional genomes to improve the accuracy of induced alignments. Starting from our fruit fly MAF alignment set, we extracted from each coding multiple alignment a subset of just three species. These reduced alignments formed our "exp3" data set. We then progressively augmented these alignments to produce sets "exp4" through "exp8" with 4-8 species. The species used in each experiment are shown in the tree of Figure 3.6.

Ideally, the quality of alignments obtained by the various tools should improve as more species are added to the alignment, since deeper alignments provide more information to resolve ambiguities of which base should align to which. On the other hand, more species provide a more stringent test by our accuracy metric, since a "consistent" codon must be aligned consistently in *all* informant species.

Table 3.3 shows the accuracies of each program on sets exp3 through exp8, using the procedure described in the previous experiment for each set. All programs obtained at least some improvement in accuracy as more species were added, from exp3 to exp7. (The original aligner used to build the MAF alignments only obtained improvements for up to six species.) However, PhyMAIT was both the most accurate tool for every number of informant species *and* showed the largest gains in accuracy with additional species (6%, compared to only about 3% for MAFFT and MAVID). This result suggests that PhyMAIT is more able than the competing tools to make appropriate use of additional species.

Interestingly, the accuracy of every program *declined* from set exp7 to set exp8. The source of the decline is not immediately clear; it may be that the eighth species, *D. willistoni*,

		MAF	PhyMAIT	ClustalO	MAFFT	MAVID
exp3	c	88.37%	92.01%	86.60%	89.15%	88.73%
	i1	9.72%	6.32%	11.05%	8.93%	9.34%
	i2	1.91%	1.67%	2.35%	1.92%	1.93%
exp4	c	89.12%	93.63%	86.99%	90.31%	89.72%
	i1	9.14%	5.03%	11.72%	8.82%	9.39%
	i2	1.74%	1.34%	1.29%	0.87%	0.89%
exp5	c	90.06%	95.53%	88.74%	91.45%	90.51%
	i1	8.93%	3.98%	10.12%	7.96%	8.77%
	i2	1.01%	0.49%	1.14%	0.59%	0.72%
exp6	c	92.41%	97.13%	91.82%	92.01%	91.96%
	i1	7.14%	2.59%	7.79%	7.63%	7.25%
	i2	0.45%	0.28%	0.39%	0.36%	0.79%
exp7	c	91.79%	98.16%	92.03%	92.22%	92.18%
	i1	7.45%	1.63%	7.61%	7.45%	7.22%
	i2	0.76%	0.21%	0.36%	0.33%	0.60%
exp8	c	91.47%	98.01%	91.10%	91.89%	91.52%
	i1	7.63%	1.32%	7.37%	6.77%	7.05%
	i2	0.90%	0.67%	1.53%	1.34%	1.43%

Table 3.3: Comparison of alignments from PhyMAIT, Clustal Omega, MAFFT, and MAVID versus original MAF alignments with increasing number of species. Experiment exp3 to exp8 has 3-8 species respectively, with one species added in each experiment.

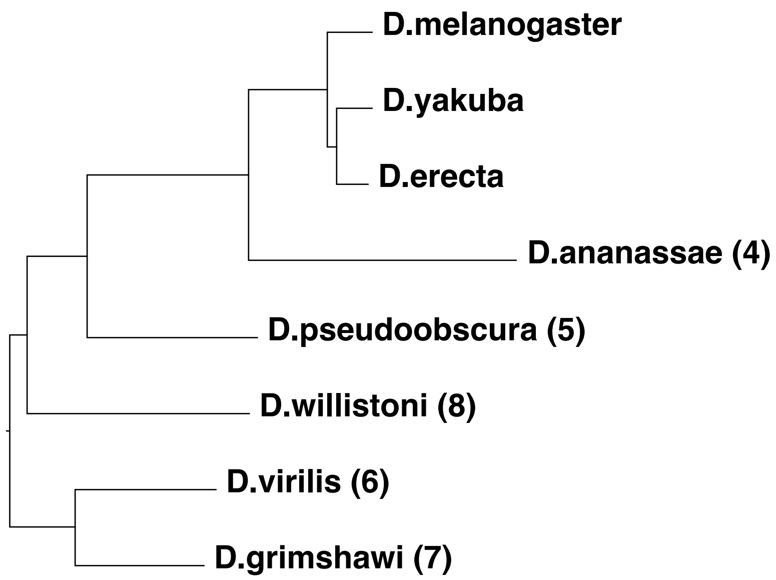


Figure 3.6: Order of addition of species for experiments exp3 to exp8. Non-numbered species are present in all experiments.

is unusually difficult to align or has inaccurate sequence or annotation that corrupted the protein alignments.

Validation on Simulated Non-Coding DNA

While the coding experiments of the previous section used real alignment data and offered a readily available source of ground truth, they have the disadvantage of focusing on some of the best-conserved, and hence easiest-to-align, regions in the genomes of interest. Ideally, we would be able to conduct similar experiments on the non-coding parts of the aligned genomes to evaluate each tool's performance in a wider variety of conditions. Unfortunately, there is no readily available ground truth for such experiments; hence, we instead turn to simulated data to further exercise the capabilities of the tools.

We started with the six-species mammalian phylogeny of Figure 3.5 and used the ROSE simulator [152] to produce orthologous sequences according to this phylogeny. Because the data is simulated, the actual homology relationships among bases are known, and so ground truth is readily available. ROSE allows the overall rate of evolution to be adjusted; we chose

	indel rate	mutation rate
low evolutionary rate	0.000080	0.01000
mid evolutionary rate 1	0.000185	0.01175
mid evolutionary rate 2	0.000290	0.01350
mid evolutionary rate 3	0.000395	0.01525
high evolutionary rate	0.000500	0.01700

Table 3.4: Parameters of ROSE used to generate simulated data with different evolutionary rates.

low and high rates that roughly matched the observed mutation rates of coding regions and conserved LINE repeat elements in the real mammalian MAF alignments, respectively, and then interpolated several more rates in between to observe behavior for a range of rates. We generated roughly a million multiple alignment positions for each rate tested.

We used ROSE to generate 1000 sets of aligned sequences, each of length roughly 1000, for each choice of mutation parameters tested; exact lengths varied due to the introduction of indels. The ancestral sequence at the root of each phylogeny was randomly generated with equal base frequencies; ROSE then introduced simulated mutations (substitutions and indels) along the branches of the mammalian phylogeny used for the previous experiment. Substitutions were generated according to an HKY model with parameters TransitionBias = 2.5 and TTration = 2.5.

Evolutionary rate parameters were chosen to match the observed indel frequencies of coding DNA in our mammalian alignments at the lowest rate and to match the observed frequencies in conserved LINE elements at the highest rate. We then chose three equally spaced intermediate mutation rates. The exact parameters used for each evolutionary rate are given in Table 3.4.

As before, we applied each of MAFFT, MAVID, and Clustal Omega to align these sequences *de novo*. We then used the Clustal Omega alignment (which typically had the *lowest* accuracy of the three competitor programs) as input to PhyMAIT to produce a refined result. In these experiments, we report accuracy as the fraction of alignment positions in which all aligned bases are actually homologous.

Figure 3.7 shows the accuracy of the four programs on our simulated alignments. PhyMAIT consistently outperforms the competing tools, despite starting from the least accurate initial alignments (Clustal Omega's). The results at the lowest evolutionary rate roughly reproduce the observations on real coding alignments, as expected. Even at the highest evolutionary rate tested, PhyMAIT was still able to align roughly 90% of positions correctly; in contrast, the competitors' accuracies ranged from 50 to 75%.

We note that no change was required to PhyMAIT's settings to deal with the different data sets shown here. Instead, its initial parametrization step, with its inferred branch length multipliers at each position, adjusted its model appropriately for each experiment.

Computational Efficiency

PhyMAIT uses several techniques to reduce computational cost. We use a compact, column-oriented storage format for multiple alignments to reduce cache misses. Moreover, we use customized phylogenetic caching techniques, described in [155], to store per-column probabilities, which greatly reduces the cost of probability computations.

As an indicator of relative performance, PhyMAIT took 31 minutes to compute the optimized fruit fly alignments used in the experiment of Table 3.2 on an Intel Xeon 2.4GHz processor, while the three competing tools took between 9 and 21 minutes. Moreover, the refinement task is easily parallelized across alignments, so PhyMAIT, like other alignment tools, can easily be distributed across multiple processors.

The time complexity of PhyMAIT lies mainly in the E-step of its iterative realignment algorithm, in particular evaluating alternative augmented tree topologies and optimizing the additional branch lengths for each. With increasing numbers of informant species, this cost comprises the greatest part of the running time. The space requirements of PhyMAIT's caching strategy scales exponentially with the number (but not the genome length!) of species in the alignment; however, the space cost is only tens of megabytes for aligning up to ten species.

Overall, the increased resource requirements of PhyMAIT are feasible and likely worthwhile given its observed and implied ability to improve widely used multiple alignments, even in regions of comparatively low conservation.

3.5 Discussion

3.5.1 Order of Sequence Re-Alignment

In order to test the effect of picking order on the accuracy of resulting alignment, we tested several strategies. The first is picking the species with the minimum distance to all other species first. The second is picking the species with the maximum distance to all other species first. The third is picking randomly. The rationale behind the first strategy is that the most-distant species may be aligned most inaccurately. Thus by realigning it first, we can correct the most errors in the resulting alignment. The rationale behind the second strategy is that the central species is part of the core of the alignment. Thus by improving the core alignment, the whole alignment would be improved. The rationale behind the third strategy is that random walk can avoid trap at local optimal solutions.

All strategies gave similar results, which means the final alignment converged to the same local optimal solution. We can view each step of realignment as a step in the global optimization process. The initial alignment can be viewed as the initial solution. By iteratively picking out a sequence and realigning it to the rest of the alignment, the initial solution is improved. Each time only one sequence is picked out, so the search step is limited. This may be the reason why all three strategies converged to the same local optimal solution: the limited search step in each iteration limits the ability to jump out of local optimal solutions.

3.5.2 Optimal Number of Species

We also tested the effect of number of species on the accuracy of the resulting alignment. Theoretically, the more species we have, the more homology information we can use to improve the alignment. However, in practice, more species may not give better resulting alignment. This may due to several possible reasons. First, the quality of the extra sequence may not be good enough for providing accurate information. For example, it may contain sequencing errors. Second, the extra sequence may not be orthologous to the rest alignment. This is an error in the initial alignment, which cannot be corrected within our framework. Third, the branch in the phylogeny relating the extra species to the rest species may not be accurate. Thus the computation of column score in the alignment is not accurate, which will affect the accuracy of the resulting alignment. Fourth, the extra species may be too distant from the rest species. Thus the orthology information may be lost due to saturation of mutations. In this case, the extra sequence is just like a random sequence, which cannot provide useful information but only noise.

3.6 Conclusion

High-quality DNA multiple-genome alignments are an important substrate for modern genome annotation and exploration. Evidence for the presence of conserved functional elements and signals, particularly in non-coding regions, as well as overall conclusions about the evolutionary history at different genomic loci, rest on the accuracy of the underlying sequence alignments. To ensure that the biological community has the highest-quality alignments possible, we investigated whether iterative refinement of existing DNA alignments, using knowledge of their underlying evolutionary relationships, is likely to yield measurable improvements in alignment quality.

We have described PhyMAIT, an efficient, phylogenetically aware tool for improving DNA multiple sequence alignments of orthologous genomes. PhyMAIT demonstrably improves the quality of multiple alignments even in coding regions – the regions most likely to be correctly aligned *a priori* – from the widely-used UCSC Genome Database. Moreover, results on simulated data suggest that equal or greater improvement is likely to be achieved on a variety of alignable non-coding sequences, depending on their degree of conservation. In our tests, PhyMAIT outperformed several competing tools and was better able to exploit additional informant genomes to improve alignment quality. Our work strongly suggests both that better genomic DNA alignments are achievable in practice, and that existing, widely-used genomic alignments both can be substantially improved and provide a usable basis for refinement. We plan to apply PhyMAIT on a larger scale to improve the multiple alignment tracks for several assemblies, starting with improving the UCSC *Drosophila* alignments to support an ongoing project to find novel regulatory motifs on the fourth chromosome of *D. melanogaster*.

A number of opportunities exist to improve PhyMAIT's performance and utility. First, our scoring model's assumption that successive columns in the multiple alignment are stochastically independent is not realistic. It could be useful to add a dependence model between adjacent bases/columns. The theory of phylogenetic HMMs provides one clear avenue to such improvements.

Second, because PhyMAIT is based on an underlying probabilistic model, it may be possible to augment its computation to evaluate posterior confidence in the refined multiple alignments. Confidence estimates, e.g. the likelihood that a particular position is correctly aligned, would be helpful for judging the reliability of alignments and identifying where multiple, alternative hypotheses are plausible. If a region of importance has low posterior confidence, it might even be worth acquiring additional sequence data from other informant species to resolve ambiguities. Currently, the realignment step of PhyMAIT naturally provides a posterior distribution over augmented tree topologies as a result of its EM algorithm, but it is not immediately clear how to estimate a similar distribution over multiple alignments. One possibility is to modify the iterative selection and realignment of sequences into a true MCMC method, which would allow sampling from the posterior distribution of alignments. Whether such an approach can be made efficient enough to support robust confidence estimates is a topic for future work.

Finally, it could be useful to consider alternative trees for the underlying multiple alignment. While PhyMAIT already considers multiple hypotheses for the tree placement of and branch length to the sequence being realigned, it does not use information from one realignment step to modify the tree for subsequent steps. It may be that one can improve the relative branch lengths, or even alter the tree topology, simultaneously with improving a set of alignments. If a large improvement in alignment score occurs with a major change in topology or branch length, this could indicate an error in orthology assignment or an region of localized rapid evolution in some subclade of species. While it is possible to sum probabilities over multiple tree hypotheses, the increased computational cost of using multiple trees makes it imperative to be careful not to consider too many such alternatives. Heuristics for limiting the search space of alternatives could help to guide the process.

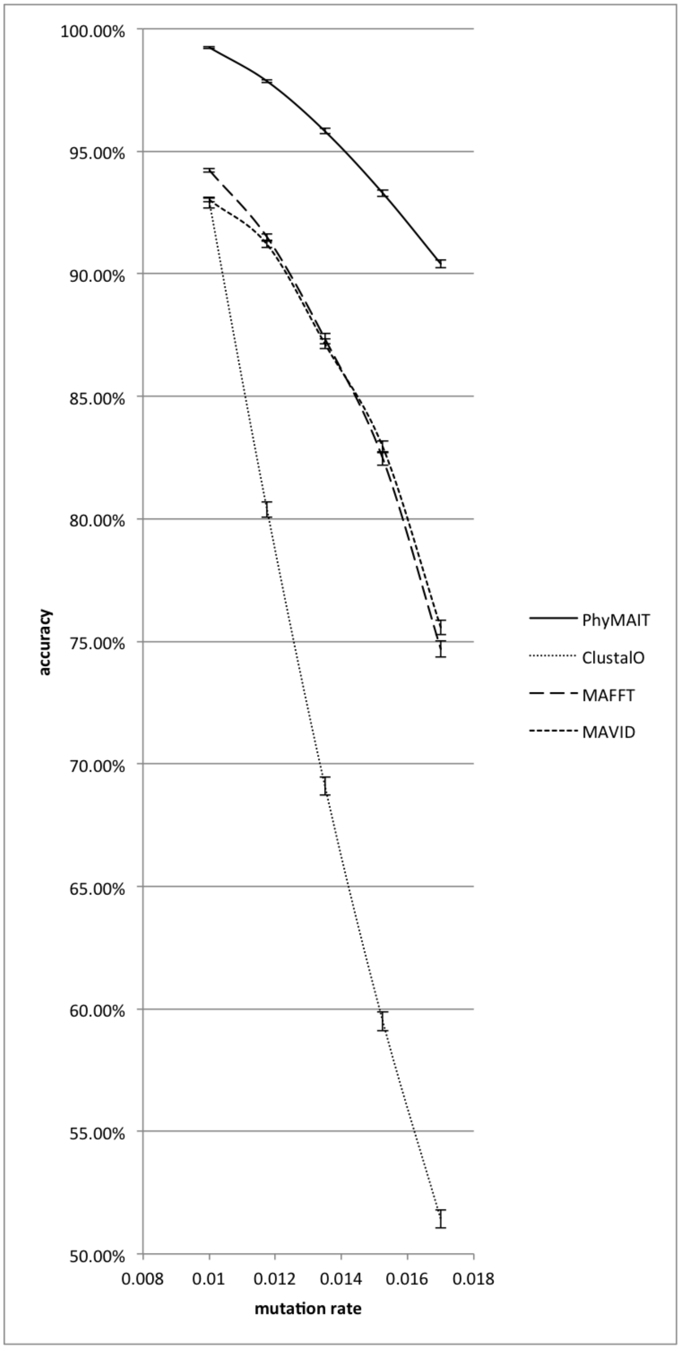


Figure 3.7: Accuracy comparison of PhyMAIT, CLUSTAL Omega, MAFFT, and MAVID using simulated data. Error bars show 95% confidence intervals on each sample of 1000 simulated alignments.

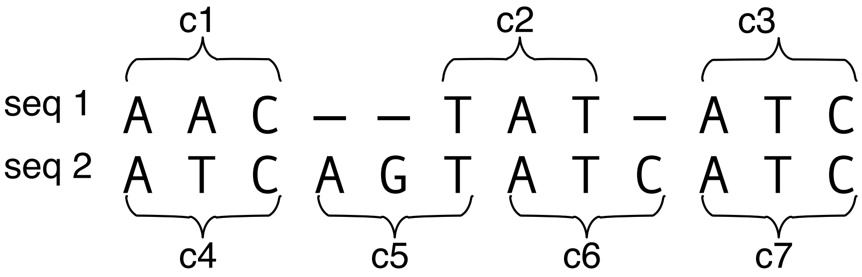


Figure 3.8: Example of ORF-shifted bases in alignment.

Chapter 4

Short Read Mapping

4.1 Problem Introduction

Given a set of short reads from next-generation sequencing results, mapping them back to their orthologous locations in a reference genome is called short read mapping [90, 132]. This is a new problem arising with the development of next-generation sequencing techniques [124]. Because genomes from the same species are similar to each other in terms of DNA sequence and genome arrangement, it is relatively easy to map reads to a reference sequence from the same species.

Although in most cases, the reference genome is from the same species as the query reads, there are cases where interspecies mapping is necessary. One example is when such a reference genome is not available, i.e., no individuals of the same species have been sequenced and assembled before [126]. Another example is from metagenomics, where the reads can only be traced back to a set of species, or the species of the reads are totally unknown. In this case, the reads have to be classified according to their species, and then assembled within each species [8, 86]. Another example is RNA expression estimation. It has been shown that using read mapping to estimate the expression level is more accurate and repeatable than using microarray [87]. In many cases, some closely related species to the newly sequenced species have already been sequenced and assembled, which can provide useful information for classification and assembly of the newly sequenced reads [62, 63].

With the development of next-generation sequencing techniques, short reads are obtained in large volume every day. Most existing short read mapping tools either use a single reference genome, or can only do intraspecies mapping. As many new species are sequenced, methods for efficient and accurate interspecies mapping are needed. Such methods must use information from multiple informant species and do an alignment-based mapping procedure, but how to model mapping problem within this scenario is still an open problem.

We use a phylogenetic-aware short read mapping algorithm for doing interspecies mapping. It was shown that multiple alignment of phylogenetically diverse sequences is substantially better than pairwise alignment at capturing orthologous sequences [102]. We use a multiple alignment of several reference sequences from different species and a phylogenetic tree of those species. We assume the originating species of the reads is unknown (if it is known or partially known, then it can be considered as user's *prior*). By simultaneously aligning the reads to the multiple alignment and calculating the posterior probability of each branch placement, we can find their orthologous positions and the most likely tree placements.

Our work in Chapter 2 show that by using a multiple alignment as reference, othologous queries can be detected more accurately than using a single reference sequence [155]. By simultaneously aligning queries to the reference and inferring tree placements, both alignments and tree placements may be inferred more accurately than doing them separately.

4.2 History, Applications, and Existing Tools

Since the publication of the first human genome [77, 163], there has been revolutionary progress in the genome sequencing techniques. With next-generation sequencing techniques, the human genome can be sequenced at a 10-fold coverage in a single run, resulting 30 Gb DNA sequence data, yet with less than $1000 cost, comparing with 3 billion dollars and 13 years at the time of the Human Genome Project! This huge amount of short read data brings many new opportunities and challenges.

One application of read mapping is read assembly [57, 126, 147, 156, 164]. While traditional read assembly algorithms work well on longer reads, reads from the next-generation sequencing techniques are much shorter, making it more complex and time-consuming to assemble them into a continuous sequence. *Ab initio* read assembly algorithms are much slower than mapping. For example, Bowtie (version 1 [78] and 2 [79]) can map 30 million reads per hour using 3.2 Gb memory, while SSAKE [169] can only assemble 4 million reads per hour using 32 Gb memory. *Ab initio* assemblers cannot give a continuous assembled sequence, leaving hundreds to thousands of contigs [176], because they cannot determine the order of assembled segments. Because two genomes from the same species or closely related species are similar to each other [25], read mapping can accelerate read assembly by mapping newly sequenced reads to one or more existing reference genomes. Because the reference genomes also provide information on genome rearrangement, this information can be used to determine the order of assembled contigs of the new genome.

Another application is variation discovery, which is to find the variations among genomic sequences [1, 7, 34, 54, 57, 60, 64, 74, 84, 85, 88, 94]. One example is targeted resequencing [126]. Targeted resequencing typically investigates a few genes across a large population. All the genes are sequenced individually and then compared with each other. Recognition of functional variants is at the center of NGS data analysis and bioinformatics [101]. The goal of variation detection is to detect genomic variations between two or more genomes or functional elements. Such variations can be insertions, deletions, SNPs, or genome rearrangements. Using a reference sequence from the same species as the reads will not give genome rearrangement information. When such a reference genome is not available, closely related species can be used.

Another application scenario for short read mapping is gene expression estimation. RNA-Seq [87] is a new technology using short reads from RNA sequences to estimate gene expression level. Studies have shown that expression estimates using RNA-Seq are highly reproducible [103] and more accurate than microarray results [113, 114]. Because reads are short, they are often mapped to more than one gene or isoform. In this case, accuracy of expression estimation will be lowered. Using multiple informant genomes will help placing the reads onto their correct locations.

Another application is from metagenomics. In metagenomics, genomes from different species are sequenced. To assemble them, one must first classify them into their own species. One way is to map the reads to sequences from different taxa related by a phylogenetic tree [12]. Reads mapped to the same node are considered coming from the same phylotype.

Another application is functional prediction. Given a database of functional elements, one can map the newly sequenced reads to the database and predict the function of the new sequence. The reason for using read mapping instead of assemble-and-align method is because mapping takes much shorter time than assembly and uses less memory [78, 169].

There are many existing short read mapping tools [90, 132], but they are not suitable to our problem because of several issues. First, they are not designed for interspecies mapping [4, 61, 125, 143] . Secondly, they cannot use multiple reference sequences [35, 39, 58, 98, 141, 142, 149]. Thirdly, they can handle only limited number of indels and gaps [78, 78, 89, 92]. Fourthly, they cannot handle long reads produced by newer next-generation sequencing machines [91] . Fifthly, they do not use biologically realistic alignment models [12, 170].

By using our phylogeny-aware alignment model, we have a probabilistic scoring scheme that incorporates more biologically relevant information. Furthermore, we do not need to align the query to each inner node in the tree, which reduces the time complexity. Our model also uses the posterior probability of the branch placement of the query to reflect the actual evolutionary history of the query and the informant species, which enables us to infer the query species and also helps to improve the alignment score quality.

4.3 Problem Formulation

Given a query sequence, we want to find its orthologous region in a reference sequence. A clear distinction between orthologous and paralogous sites is critical for the construction of a robust evolutionary classification of genes and reliable functional annotation of newly sequenced genomes. Finding homologous parts of the query in the reference is relatively easy. Because non-homologous regions are independent of the query, they can be considered as random sequences. The score distribution of random alignments empirically follows a Gumbel distribution [5]. Homologous alignments usually have high scores, which are at the tail of the Gumbel distribution. Thus standard seeding-alignment methods can find homologous regions, but distinguishing orthologous from paralogous parts is difficult, because they can be very similar. Here we want to use several informant sequences to help differentiate ortholog from paralog. We assume that the reference and the informants are contained in a multiple alignment, and the species are related by a phylogenetic tree. The branch lengths of the tree can be different at different locations in the multiple alignment. For simplicity, we make the following assumption.

Assumption 1. *Each column in the multiple alignment is composed of only orthologous bases.*

For many parts in our experiments, we use data from the UCSC (University of California Santa Cruz) Genome Bioinformatics Site [51]. The database contains the reference sequence and working draft assemblies for a large collection of genomes, as well as multiple alignments built with those reference sequences. Though actual multiple alignments from the database may contain non-homologous alignments, their goal is to construct orthologous alignments. In fact, they use species tree to guide the multiple alignment, which means the sequences are considered orthologous rather than paralogous.

The rationale of our method is that the evolutionary tree of orthologous sites is in accordance with the taxa phylogeny, while the evolutionary tree of paralogous sites is not [165]. Furthermore, orthologous regions and paralogous regions are under different evolutionary restrictions, which means they follow different evolutionary model in terms of indel rates and mutation rates. Thus if we score paralogous sites using the evolutionary model of orthologous sites, it will give lower scores.

Our problem is given multiple matches between the query and the multiple alignment, find the orthologous one.

We use the following notations:

τ - phylogeny at orthologous location without query

τ^* - phylogeny at orthologous location with query

τ' - phylogeny at paralogous location without query

$\tau^{*\prime}$ - phylogeny at paralogous location with query

A typical duplication scenario is shown in Figure 4.1, R, Q and I standing for reference, query and informants. Base Y is duplicated at the root. Q is orthologous to R and I, and paralogous to R' and I'. Aligning Q to R and I using phylogeny τ^* will place Q on the correct

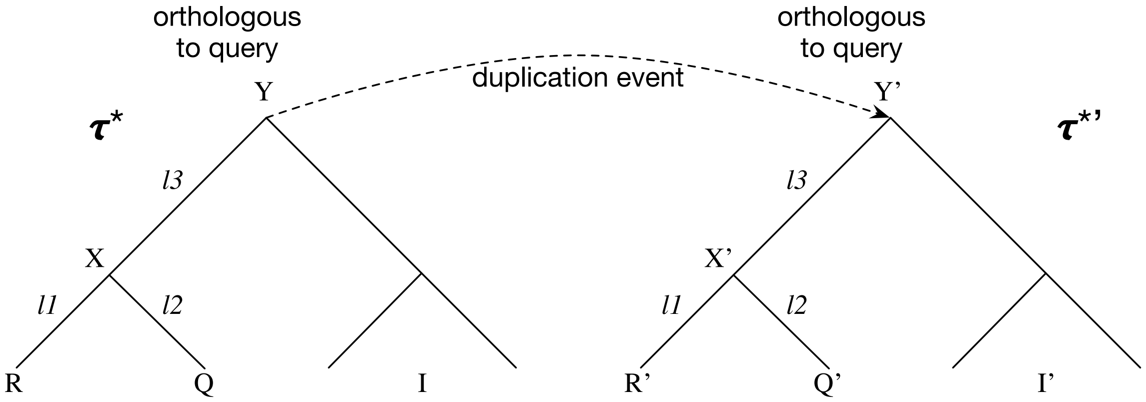


Figure 4.1: Orthology and paralogy

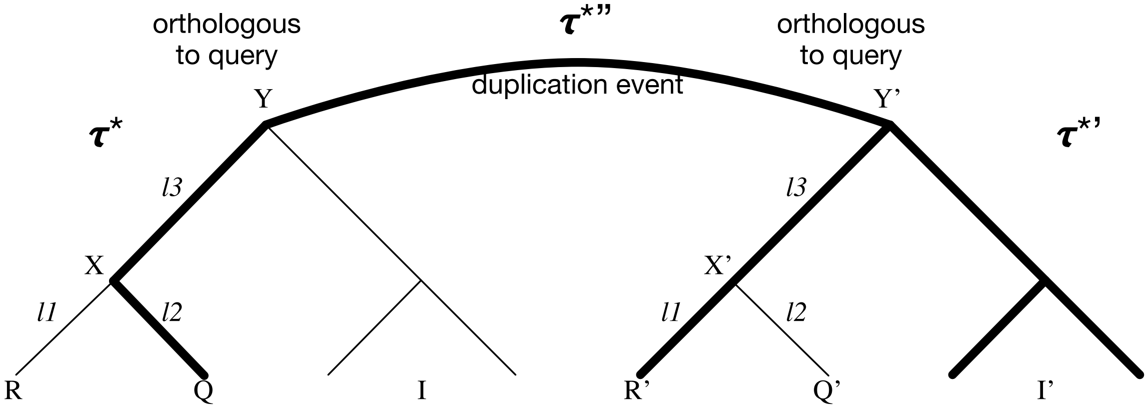


Figure 4.2: Paralogous query and reference sequences

branch, while aligning Q to R' and I' using phylogeny $\tau^{*\prime}$ will place Q at the root Y' with a lower score.

We want to differentiate between orthologous and paralogous alignments by their scores. So their score distributions must be different. Next we examine their score distributions.

We are given one query sequence and multiple orthologous loci in the multiple alignment, at most one of which is truly orthologous. When aligning Q to the orthologous part using τ^*, considering aligning a single base, the probability and score of the alignment are:

$prob_{orth} = Pr(R, Q, I|\tau^*)$, $score_{orth} = \frac{Pr(R,Q,I|\tau^*)}{Pr(R,I|\tau)Pr(Q)}$.

Please refer to equation 2.1 for the definition of $Pr(R, Q, I|\tau^*)$ and the conversion from the tree probability on the lefthand side to the score representation on the righthand side.

The probability is the likelihood of the phylogeny τ^*. The score is the likelihood ratio for two hypotheses: the first one being the sequences are correctly aligned, the second being the sequences are independent. The score will be used to pick the orthologous alignment. When aligning Q to paralogous parts using τ'^*, the probability and score of the alignment are:

$prob_{para} = Pr(R', Q, I'|\tau^{*\prime\prime})$, $score_{para} = \frac{Pr(R',Q,I'|\tau^{*\prime})}{Pr(R',I'|\tau')Pr(Q)}$.

Note that the phylogenies used in calculating probability is different from the one used in calculating score. Because when we align the query to the paralogous site, we do not know whether it is orthologous or paralogous. We compute the score as if the query is orthologous to the reference. But to calculate the theoretical score distribution, we should use the real probability of this event.

4.4 Results and Discussion

We did both theoretical computation of alignment score distributions for orthologous and paralogous alignments and experiments on real data. The structure of our algorithm is shown in Figure 4.3.

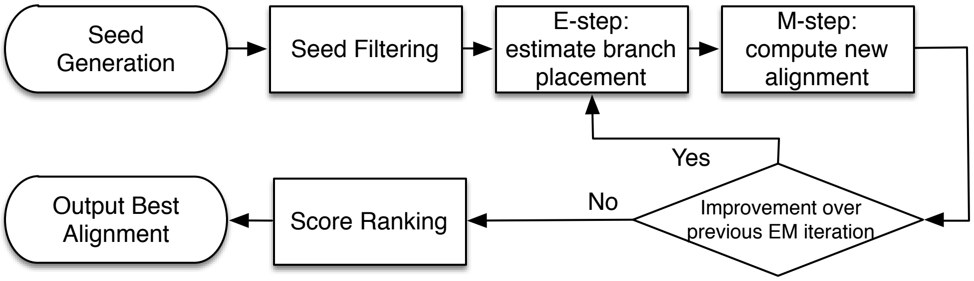

Seed Generation
Seed Filtering
E-step: estimate branch placement
M-step: compute new alignment
Yes
No
Improvement over previous EM iteration
Output Best Alignment
Score Ranking

Figure 4.3: Structure of PhyMap

4.4.1 Theoretical Comparison of Alignment Score Distributions

We did both theoretical computation and experimental researches. We compared the theoretical score distributions of orthologous alignments and non-orthologous alignments using our PhyMap model with informant species, our PhyMap model without informant species, and Smith-Waterman dynamic programming model, which uses similar dynamic programming algorithm as our PhyMap but does not use non-probabilistic scoring system and does not use tree model.

Based on the real phylogeny of human and 4 related species, we compute the score distributions for various alignment lengths ranging from 1 to 64. To generate an alignment of length K, we generate each column of the alignment independently. For each column, we know the phylogeny with branch lengths and the evolutionary model, so we can compute the distribution of assignment of bases in the tree which are generated by the evolutionary model. Similarly, to generate a paralogous alignment column, we need to take into account duplication events in the phylogeny. We assume the duplication event does not incur any mutations. We also assume the duplicated site has the same evolutionary rate as the original site.

Our computations show that there is a big difference between distributions of scores of orthologous alignments and paralogous alignments using informant species, as shown in Figure 4.4. Using PhyLAT with informant sequences, the score distributions of orthologous alignments and paralogous alignments have increasing difference as the alignment length increases. For Smith-Waterman alignments, the difference is much harder to detect even as the query length increases.

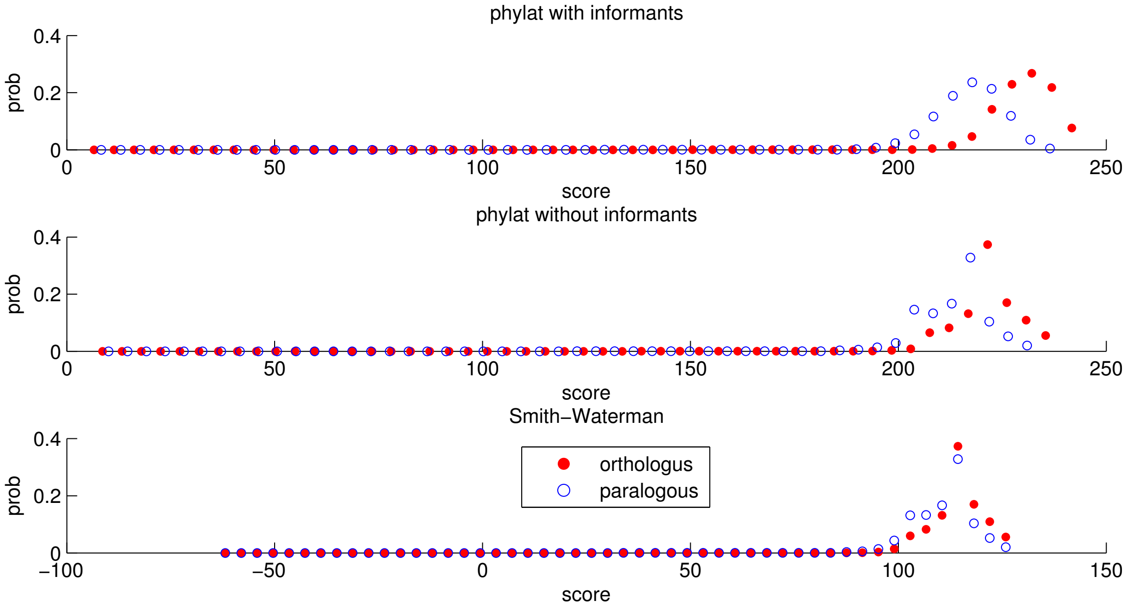


Figure 4.4: Distributions of scores of orthologous/paralogous alignments of length 64. The first column shows score distributions of PhyLAT using informant sequences. The second column shows score distributions of PhyLAT not using informant sequences. The third column shows score distributions of Smith-Waterman alignments of the query and the reference sequence.

Our results are preliminary evidence that adding the informants provides substantially more information about the correct alignment than methods that do not use the phylogeny and associated information. Thus our model can differentiate between orthologous and paralogous alignments better than non-phylogenetic pairwise models.

4.4.2 Experimental Results with Simulated Data

Furthermore, we compared our method with other popular short read mapping tools (BWA, BOWTIE and BLAST V2) on simulated data. To prepare the input sequences, we choose an multiple alignment of human chromosome 22 and four related mammalian species from

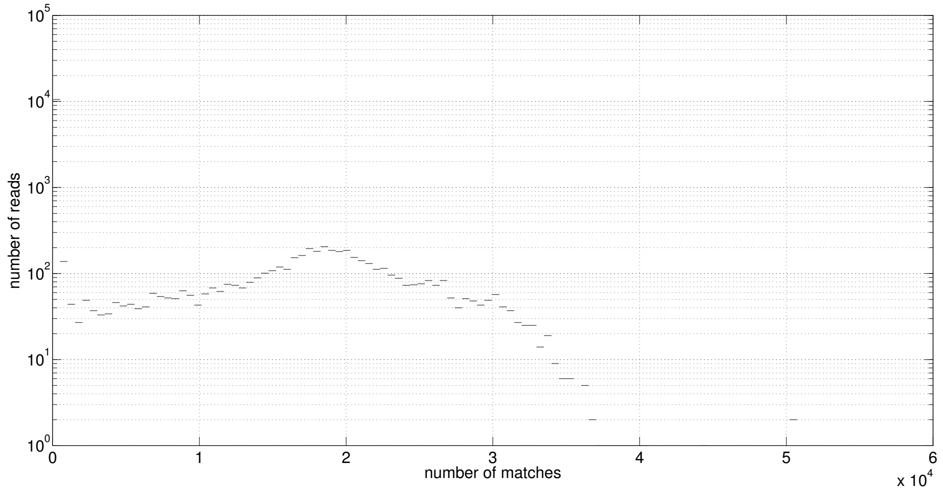


Figure 4.5: Histogram of number of matches for human query reads.

UCSC genome database as raw data, so we know that the sequences are orthologous to each other. The five species and their relating phylogeny is shown in Figure 4.6. We use human chromosome 22 as the query species. To generate reads, we first cut the human chromosome 22 into reads of length 100. For PhyMap we use the remaining multiple alignment of the four informant species as the reference sequence. For other competitor alignment tools, we use rhesus sequence as the reference sequence because rhesus is the closest single informant species. Then we use BLAST to generate seed matches between query reads and reference sequences. There are 21689 reads. Total number of seeds is 128138682. Each read has 5908 seeds on average. Figure 4.5 shows the histogram of number of seeds per read for the experiment with human query reads. We can see that the distribution is a long-tail distribution where majority of the reads have less than 1000 matches, while some reads have more than 40000 matches. After generating matches, we run PhyMap and the competitor alignment tools on the top 10 seeds to get full alignments with scores. Then we pick the alignment with the highest score as the orthologous alignment for each seed.

We found that PhyMap can correctly map more reads than competitor tools, i.e., BWA, BOWTIE, and BLAST. See Table 4.1. We chose BLAST because most other tools were designed specifically for intra-species mapping. Interspecies mapping has to be able to handle higher rates of indels and mismatches. We can see that short read mapping tools like

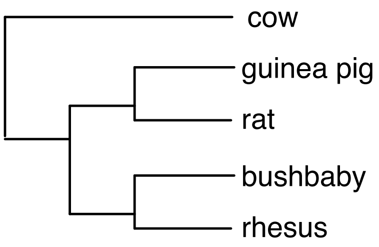


Figure 4.6: Mammalian tree.

BWA and BOWTIE can only map a small fraction of reads. This is because they cannot properly handle indels or mutations in short reads. We used Blast to find promising mapping locus for a given read, then used PhyMap to find the orthologous one. This is because BLAST is very good at fast locating promising matches while it lacks the ability to further refine those matches to a level we need for the short read mapping task. While Blast can map most of the reads, but many of the highest-scoring matches given by Blast are not actually orthologous matches. This supports our belief that by incorporating phylogenetic information and informant sequences, we can achieve higher accuracy in orthology mapping. The running time for PhyMap is 7.28 hours. The running time for BLAST, BWA and BOWTIE are 2.15 hours, 1.21 hours and 1.09 hours respectively. We can see that while PhyMap uses longer time than other mapping tools, it achieves much higher accuracies than them.

Table 4.1: Comparison of BWA, BOWTIE, BLAST, and PhyMap on simulated data. Reads are simulated from human chromosome 22. #correct: number of correctly mapped reads. #wrong: number of wrongly mapped reads. #unmapped: number of unmapped reads. Accuracy: #correct/#total.

	#correct	#wrong	#unmapped	accuracy
BWA	5795	1626	14268	26.72%
BOWTIE	8593	3394	9702	39.62%
BLAST	15813	5855	21	72.91%
PhyMap	19545	2123	21	90.11%

Chapter 5

Genome Rearrangement Inference

5.1 Problem Introduction

The problem we discuss in this chapter is inferring the order of a set of sequences of a query species given the phylogeny and sequence orders of orthologous informant sequences. While many genomes have been sequenced and assembled into continuous sequences, for some species, their genomes are partly available or cannot be assembled into continuous sequences. While read mapping can map reads to their orthologous locations in related species, different reference species will give different order of the mapped reads. This is because the order of orthologous segments in reference species is not the same as in the query species.

If a set of genes always appear together in a genomic block in both species, then this block is called synteny block. Genes across synteny blocks do not always appear together. Thus it is very difficult to assemble segments which do not belong to the same synteny blocks. Furthermore, synteny blocks are very common among species. For example, synteny relationships among 10 amniotes (human, chimp, macaque, rat, mouse, pig, cattle, dog, opossum, and chicken) were compared at < 1 human-Mbp resolution. There are 2233 homologous synteny blocks (HSBs) [81]. Existing read assembly algorithms will produce segments of assembled reads, without inferring their orders [95, 107]. When multiple informant genomes are used in read mapping, the rearrangement information in those informant genomes can actually be used to infer the rearrangement events in the new genome. How to use rearrangement information in orthologous sequences to infer genomic order of query sequences is still an open problem.

Because we have multiple informant sequences combined in a multiple alignment, and the multiple alignment is divided into blocks of orthologous sequences, a set of blocks with available genomic orders in informant sequences along with a phylogenetic tree should provide a lot of information for inferring the order of the segments at any node in the phylogeny, including leaf nodes which represent existing species. This is shown in Figure 5.1.

Our study showed that simultaneously aligning a query to a multiple alignment reference and inferring the query's branch placement gives more accurate results to both problems than doing them separately [155]. It also provides a biologically realistic and probabilistic model for aligning blocks of sequences. With more accurate alignments, tree placements, and the genome rearrangement information contained in the multiple alignment, we attempt to make progress on the genomic order inference problem.

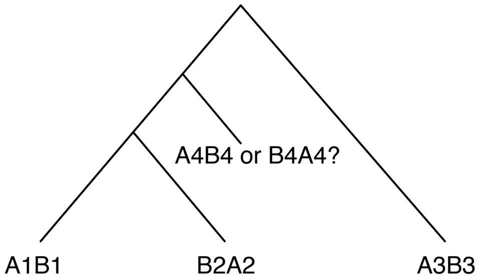


Figure 5.1: Infer block order with phylogeny. Species 1,2,and 3 are informant species. Species 4 is query. A1, A2, A3, and A4 are orthologous segments. B1, B2,B3,and B4 are orthologous segments. Because the order of A and B are known for the informant species, we can infer a distribution over the possible orderings of A and B at the query species, with the help of the phylogeny.

5.2 History, Applications, and Existing Tools

Genome rearrangement has been studied since the early 20th century [153]. While there are many existing methods for inferring phylogenies from genomic sequences or functional elements, there is less research on inferring sequence order from phylogenetic information. For the problem of inferring phylogeny from sequences, different methods use different kinds of sequences. They can be nucleotide or amino acid sequence data [96, 99, 128], genes [148], SNPs [116, 121], or gene order [17, 110, 111, 134]. Although they use different kinds of data, all the methods try to optimize a goal function of some kind of *sequence difference* in the tree. For those methods using sequence data, they compute sequence difference based on mutation models of DNA or protein sequences. For those methods using gene order, they try to minimize the total number of evolutionary events, like insertions, deletions, duplications, and inversions. Although evolutionary model of a single nucleotide or amino acid has been well studied, the evolution pattern of a whole genome is less studied. Due to the computational intractability of theoretical models for whole-genome evolution, algorithms which try to measure the evolutionary distance between two or more sequences of blocks or genes in genomes often resort to approximate measurements of the real evolutionary distance, such as breakpoint distance.

The general computational problem of reconstructing a phylogeny from gene order data is NP-hard [32, 108, 123]. Yet it is well studied. Many heuristic algorithms have been developed [109]. Existing models can be divided into three classes, i.e., distance-based, maximum parsimony, and maximum likelihood. Wang *et. al.* [167] proposed a method for inferring phylogeny from gene orders of equal gene content. Sankoff [134] gave a method using consensus gene order, but for unequal gene content, it is impossible to infer the consensus gene order. Sankoff [134] also gave a method for phylogeny reconstruction using binary phylogeny model and breakpoint distance as measurement of distance between two gene orders.

For the inverse problem, inferring the order of a set of sequences of a query species given the phylogeny and sequence orders of orthologous informant sequences, there is very limited research. Existing methods for related problems can be classified into two categories. The first category concentrates on the problem of estimating the evolutionary distance between two gene orders [166]. The second category concentrates on finding the gene order minimizing the sum of its distances to a set of existing orders. For those works in the second category, they either infer a set of conserved intervals without specifying their order or give just a single order of all the genes without giving alternatively solutions [13]. Existing methods for inferring a sequence in a phylogeny or inferring gene orders in a phylogeny have their limitations which make them not suitable to inferring sequence order for existing species. For example, in ancestral gene order inference, if we want to reshape the phylogeny such that an existing species becomes the ancestor, then we must assume the global rearrangement model is time-reversible. Just similar to a reversible nucleotide substitution model, this time-reversibility is unrealistic in the biological sense. Another limitation of these methods is that they usually ignore the length or genomic positions of the sequences or genes. Instead, the sequences or genes are treated just as an order. Furthermore, these algorithms use simple non-biologically-meaningful models, lacking models of genome rearrangement. Catchen [33] used a mapping between two gene orders to roughly infer their ancestral gene order, but it only considered gene clusters up to a pre-set limited size. It did not infer an order of all genes appearing in children species.

We need a genome rearrangement model which (1) gives plausible probabilities associated with rearrangement events, and (2) is computationally tractable to explore the space of hypotheses. Without such a model, there are two issues. First, without such a model, we cannot measure the evolutionary distance and probability precisely. Measurements like breakpoint distance are a good approximation, but far from precise. Second, without such a model, it is hard to efficiently explore the vast solution space. For example, if there are 100 segments, then there will be $10! = 3.6 \times 10^6$ possible rearrangements of the segments. Furthermore, if there are 10 species in the phylogeny, there will be 9 inner nodes. Each inner nodes will have 10^6 possibilities. Then there will be $10^{6^9} = 10^{54}$ possible rearrangements for the whole tree. If we have such a model, then we can sample the possible orders in a probabilistic way.

5.2.1 Measurement of distance

To infer sequence order at a node in a phylogeny given sequence orders at other nodes, one needs a measurement of evolutionary distance between gene orders. Evaluating the evolutionary distance between genomes in terms of gene order rearrangements has been intensively studied since the early 90's. This is partly due to its important applications in comparative genomics [23]. From an algorithmic point, it can roughly be defined as follows: given a set A of gene families, two genomes G and H, represented as sequences of signed elements (genes) from A, and a set of evolutionary operations that operate on segments of genes (like reversals, transpositions, insertions, duplications, deletions for example), what is the minimum number of operations needed to transform G into H? This number is often called the *edit distance*. From this definition, it is obvious that there are many possible measurements, depending on the types of operations allowed. In practice, due to the complexity of edit distance, people usually use approximate measurements. Here we talk about two of the most popular ones.

Breakpoint Distance Bhutkar [17] used number of shared neighboring gene pairs (NGP). Sankoff [134] used breakpoint distance. A breakpoint is a pair of adjacent markers in one genome that are not adjacent in another. Although solving for the tree with the fewest breakpoints is not the same as solving for the tree optimizing a weighted combination of rearrangements, it preserves some of that information and is much faster. This is because breakpoint model is a very simple model and its computation has linear time complexity. So the number of breakpoints between two genomes can be rapidly computed for a pair of genomes. Blanchette *et al.* introduced a heuristic method involving solving multiple traveling salesman problems to infer breakpoint patterns at inner nodes of a tree, until there is no change in the number of breakpoints in the tree [21]. Cosner and Moret *et al.* have used the binary presence/absence coding for breakpoints [38, 110]. Gallut *et al.* have used a modified breakpoint coding with states that represent a marker with its two neighbors, assuming unordered parsimony change between these states [52, 53]. In inferring hypothetical ancestors within the tree, they retain only those combinations of states that would yield a full genome.

Sorting by Reversals Another measurement of rearrangement distance is number of reversals. David Sankoff [136, 138] proposed a model of sorting by reversals. Given the orders of genes in two genomes $\pi = \pi_1\pi_2...\pi_n$ and $\sigma = \sigma_1\sigma_2...\sigma_n$. A reversal $\rho(i, j)$ of an interval $\pi_i...\pi_j$ is the permutation $\pi_j...\pi_i$. The reversal distance between two permutations is define as the minimum number of reversals needed to convert one permutation to the other. Though this is a more biologically realistic model than breakpoint distance, but it is much harder to compute. Given a set of permutations $\pi^1, ..., \pi^k$, the problem of finding a permutation σ such that the sum of the distances $\sum_{i=1,k} d(\pi^i, \sigma)$ is minimized is called Multiple Genomic Distance Problem. In the case where the distance is reversal distance, Caprara showed that both the signed and the unsigned sorting-by-reversal problems are NP-hard [31, 32]. Berman and Kaplan devised fast algorithms for sorting signed permutations by reversals [15, 67] . The Kaplan algorithm has quadratic time complexity by bypassing the equivalent transformations step of the Hannenhalli-Pevzner algorithm and exploring the properties of the interleaving graph of gray edges rather than the interleaving graph of cycles.

Since Multiple Genomic Distance is difficult in the case of reversal distance, most genomic molecular evolution studies are based on breakpoint distance. However, the Multiple Genomic Distance problem in this formulation is also NP-hard. Sankoff suggested heuristics for this problem [134].

5.3 Problem Formulation

We are given the following information:

- n permutations $\pi^1...\pi^n$ of genome segments from n informant species. Each π^i is a permutation $\pi^i = \pi^i_1...\pi^i_m$. Each segment π^i_j has a genomic location $\phi(\pi^i_j)$ and a length $l(\pi^i_j)$. For $1 \leq j < k \leq m$, $\phi(\pi^i_j) < \phi(\pi^i_k)$. π^i_k is orthologous to π^j_k for all $k \in \{1...m\}$ and $i, j \in \{1...n\}$ and $i \neq j$.
- A set of query segments $\{q_1, ..., q_m\}$ from a query species. Each segment q_i has one and only one orthologous segment in $\{\pi^k_j | j \in \{1...m\}\}$ for $k \in \{1...n\}$.
- A phylogeny relating the n informant species and the query species. Optionally the phylogeny can have branch lengths representing the evolutionary distances between the species.

The output is the true order of the query segments $\{q_1...q_m\}$ in its genome.

Probabilistic Rearrangement Model Assume we have K segments of queries, each query q^k having been aligned to a multiple alignment M^k in the database. M^k is composed of N informant sequences, sequence for species i being s^k_i. All segments from species i are ordered as $s_i^{i_0}, ..., s_i^{i_K}$ in the genome of species i. Given ordered segments from species i and species j, i.e., $S_i = s_i^{i_0}...s_i^{i_K}$ and $S_j = s_j^{j_0}...s_j^{j_K}$, there is a probability with which S_i evolves into S_j, denoted by $P_t(S_i, S_j)$. Our problem is to find an order of $q^0...q^k$ and an order for each inner node to maximize the probability of observing all the leaves, i.e.,

$$\prod_{e\in\text{inner nodes}} P_t(S_e, S_{\text{left child}}) \cdot P_t(S_e, S_{\text{right child}}),$$

where left and right child are the left and right child of e, and S_e is considered as missing data. Note that in our formulation, we preserve as much information as possible. For example, we can also use the length of the segments as parameters in our model. We can also use the genomic distances between segments as model parameters. How much information we use depends on two things: (1) how much complexity and computational work will they incur; (2) how will they affect the accuracy of the results.

When the nodes are residues or bases, the phylogeny gives the information on indels. Similarly, here the phylogeny gives the information of genome rearrangement events in the informant species and the query. The difference is that for base/residue phylogeny, there are well-established mutation models, but for genome rearrangement there are not.

To parameterize our model, we have insertion rate, deletion rate, inversion rate, translocation rate and inverted translocation rate. We do not have coalescence rate or separation rate. Modeling them may give us more powerful model, but will also increase the complexity of the model, and make it harder to efficiently explore the full solution space. For each type of rearrangement event, we use constant rate. These rates can optionally be estimated from existing data, such as the TICdb database [118], which describes the genomic location of 1,225 translocation breakpoints in human tumors, corresponding to 247 different genes.

A translocation is determined by three variables, which are the starting position where the translocation happens, the length of the translocated sequence and the distance between its translocation starting and ending locations. For simplicity, we can ignore the first variable, which is the length of the translocated sequence. We also ignore the effect of the starting point of the translocation on the translocation probability. We only consider the distance between the starting and ending points of the translocation event. In other words, two translocations with the same translocation distances are considered to have the same probability. If we normalize the whole sequence range to $[0, 1]$, a translocation distance is just a real number in $[0, 1]$ which the translocation probability depends on. We assume the translocation distance x follows a beta distribution:

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{(\beta-1)}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}d_u}$$

The beta distribution is a family of continuous probability distributions defined on the interval (0, 1) parameterized by two positive shape parameters, typically denoted by α and β. See Figure 5.2 for examples. The beta distribution is suited to the statistical modeling of proportions in applications where values of proportions equal to 0 or 1 do not occur. For our problem, we can integrate over an interval δ around x.

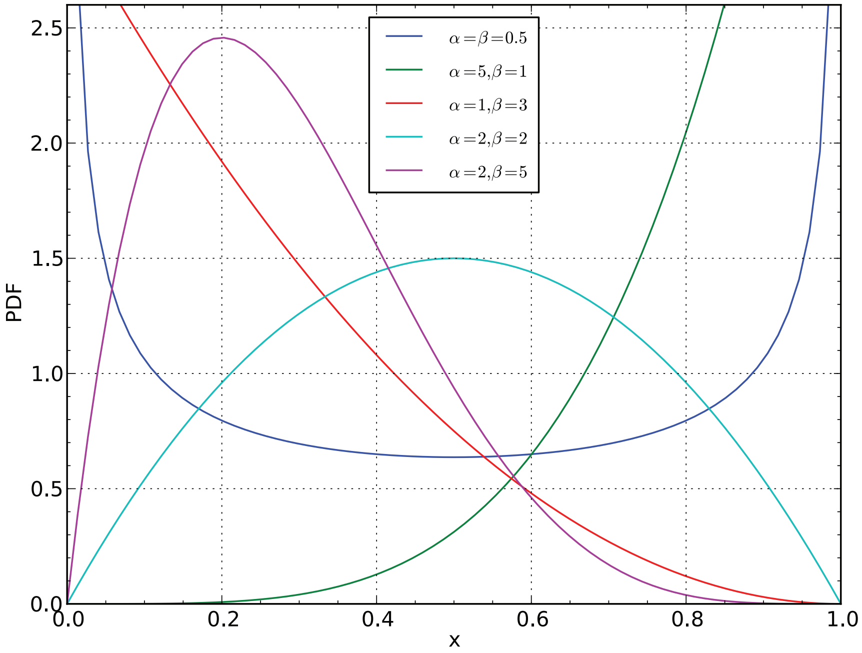


Figure 5.2: Examples of beta distribution. The one with $\alpha = 1$ and $\beta = 3$ has the trend suitable for our problem.

We assume a constant rate for all types of events across the entire genome. Future work may consider non-constant rates due to effects such as 3D folding of chromatin.

Parsimonious Methods v.s. MCMC Methods As we discussed previously, there are two disadvantages of using non-probabilistic rearrangement models. First, it might not capture as much information about alternative orderings of sequences. Second, it will make it very hard to efficiently explore the vast solution space. For parsimonious models, they are subject to the common disadvantages of their kind. For example, a parsimonious model considers the evolutionary history with the minimum number of events the *only* correct history, but this is often not the case. It is probable that parsimony methods will do better in the case of gene orders than they do for aligned molecular sequences, because the chance of parallel change or reversion is much lower in the gene order case.

Sankoff *et al.* were the first to make a non-parsimonious parametric model of rearrangement of a map (using only inversions and not allowing deletion or duplication) [137]. Sankoff *et al.* have made a start on statistical analysis of probabilistic models by using only the breakpoints information and computing phylogenetic invariants for a model of breakpoints independently arising and disappearing in a model of unsigned inversions (”reversals”) [135]. This implies that probabilistic models can be carried out on phylogeny-related inferences. To search for the optimal solution under such probabilistic models, Markov chain Monte Carlo methods are among the most popular and most powerful ones. An MCMC algorithm was previously used to infer mitochondrial gene order that is assumed to change by inversions [80]. Their Bayesian MCMC considered both gene orders and phylogenies. MCMC algorithm was also applied to the problem of inference of ancestral gene order in bacteria [43].

Our algorithm is composed of two steps. The first step is an optional step. We estimate the rearrangement rates using maximum likelihood method. In the second step, we estimate the order of the target sequences using MCMC algorithm. The reason we do not use an EM algorithm for this is because we do not believe the rearrangement rates will be affected much after inserting the target species into the phylogeny.

Metropolis-Hastings Markov Chain Monte Carlo (MCMC) Sampling Method The following is an introduction to the general MCMC algorithm using genealogy inference as an example. Since the correct genealogy is not known, especially in more complex cases such as those with rearrangement, the estimate should be based on a good sample of possible genealogies. To make the sampling as efficient as possible, only genealogies that are reasonably concordant with the data are chosen. Undirected random sampling (Monte Carlo sampling) is not efficient since the number of possible genealogies skyrockets as the number of sampled individuals increases. MCMC sampling, by imposing preferences on the random walk, allows movement through the space of possible genealogies in a purposeful fashion.

The Metropolis-Hastings algorithm [131] can draw samples from any probability distribution $P(x)$, provided you can compute the value of a function $f(x)$ which is proportional to the density of P. The lax requirement that $f(x)$ should be merely proportional to the density, rather than exactly equal to it, makes the Metropolis-Hastings algorithm particularly useful, because calculating the necessary normalization factor is often extremely difficult in practice. Two things are necessary to build up a Metropolis-Hastings MCMC coalescent sampler. First, a mathematical statement of how the parameters are expected to affect the shape of the genealogy is needed. In other words, we must have a way to assess the fitness of a genealogy given the parameters. Second, the relative fit of the data to the various genealogies must be assessed so that the sampler can concentrate on genealogies that explain the data well. This is the goal of phylogeny estimation as well; therefore similar methods may be used. Likelihood methods are the most appropriate in this situation because they are accurate and flexible and because they can tell not only which genealogy is better, but also by how much. The fit of data to a genealogy can be expressed as the probability of the data, assuming an appropriate model of molecular evolution, with respect to any given genealogy. Combining the two parts, we can calculate the likelihood of the parameters Θ given the data D by summing over all possible genealogies G, i.e. $L(\Theta) = \sum_G P(D|G)P(G|\Theta)$.

Unfortunately, the whole summation is not possible in any but trivial cases. To overcome this problem, the Metropolis-Hastings sampler generates a biased set of genealogies driven by an assumed value Θ_0 of the parameters, and then it corrects for that bias in evaluating the likelihood. The result is a relative likelihood:

$$L(\Theta)/L(\Theta_0) = \Sigma_{G^*}(P(D|G^*)P(G^*|\Theta))/(P(D|G^*)P(G^*|\Theta_0))$$

Here Σ_{G^*} is a sum over genealogies selected in proportion to $P(D|G)P(G|\Theta_0)$. If an infinitely large sample could be generated, then this approximation would give the same results as the straightforward likelihood. In practice, a sufficiently large sample must be considered so that the region of plausible genealogies is well explored. The algorithm will only efficiently explore the right region if Θ_0, which acts as a guide, is close to the true, unknown Θ. Kuhner proposed a strategy (used in LAMARC [75]) to make short runs of the program in order to obtain a preliminary estimate of Θ, and then feed that estimate back in as Θ_0. The final run will then have Θ_0 close to Θ, and will be more efficient (and less biased) than the earlier ones. The program generates its sample of genealogies by starting with some arbitrary or user-supplied genealogy and proposing small rearrangements to it. The choice of rearrangements is guided by $P(G|\Theta_0)$.

Once a rearranged genealogy has been produced, its plausibility is assessed ($P(D|G)$) and compared to the plausibility of the previous genealogy. If the new genealogy is superior, it is accepted. If it is inferior, it still has a chance to be accepted: for example, genealogies that are ten times worse are accepted one time in ten that they occur. This behavior helps keeping the sampler from walking up the nearest "hill" in the space of genealogies and sticking there, even if there are better regions elsewhere. Given sufficient time, all areas of the space will be searched, though proportionally more time will be spent in regions where $P(D|G)P(G|\Theta_0)$ is higher.

Once a large sample of genealogies has been produced, it is then used to construct a likelihood curve showing $\mathrm{L}(\Theta)/\mathrm{L}(\Theta_0)$ for various values of Θ, which is normally displayed as a log-likelihood curve. The maximum of this curve is the maximum likelihood estimate of Θ; the region within two log-likelihood units of the maximum forms an approximate 95% confidence interval. Typically, the strategy is to run 5-10 short chains of a few thousand genealogies each, to get a good starting value of Θ, and then 1-2 long chains to generate the final estimate.

The most difficult part of creating such a Metropolis-Hastings sampler is working out a way to make rearrangements guided by $P(G|\Theta_0)$: this is particularly challenging in cases with rearrangement, where the genealogy becomes a tangled graph. A flowchart of the MCMC process for the genealogy inference problem is shown in Figure 5.3.

In our problem, we need to compute the probability $P_t(S_i, S_j)$ for each edge (S_i, S_j) in the tree. We only want to infer the order at the query species node, but in order to compute the probability of the tree, we also need to infer the orders at inner nodes. In order to do this, we need a mapping from $\{s_i^k\}$ to {insertion, deletion, translocation, stay} to specify which segments are translocated and which segments remain at the same location. For example, in Figure 5.4, species a and b each have three segments, but there are at least two possible mappings shown in the figure. We choose the most likely one to compute $P_t(S_a, S_b)$. In other words,

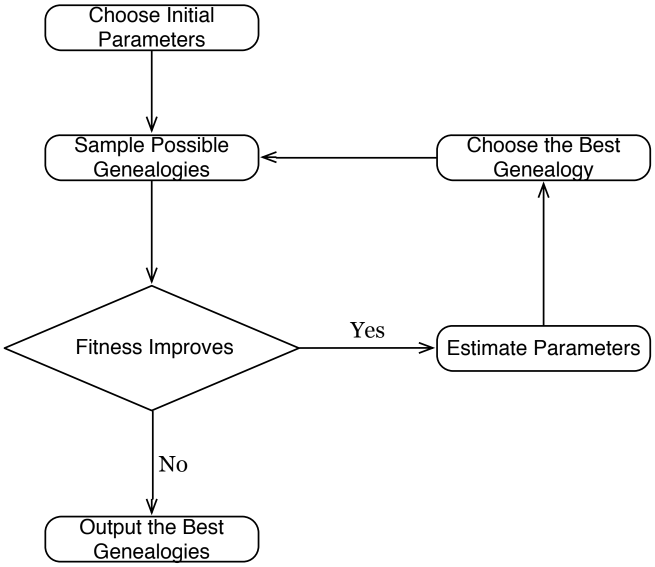

Choose Initial Parameters
Sample Possible Genealogies
Choose the Best Genealogy
Fitness Improves
Yes
Estimate Parameters
No
Output the Best Genealogies

Figure 5.3: Flowchart of Metropolis-Hastings MCMC algorithm.

$$P_t(S_i, S_j) = \underset{\textbf{mapping } m}{\arg\max}\ P_t(S_i, S_j, m).$$

Our experiment data from the UCSC database contains starting and ending genomic positions for each sequence segment. So we know the orientation of the segments. To take this into account, our model also contains inversion event. However, considering inversion will make our problem much harder. Specifically, instead of needing a mapping from each segment s_i to the set of rearrangement events, we need a rearrangement history to compute the probability $P_t(S_i, S_j, h)$, where S_i is the ancestral permutation, S_j is the offspring permutation, and h is the evolutionary history. There are two possible solutions to this problem. One is using an approximate cost function instead of the probabilistic function. Another is to estimate the evolutionary history and compute the probability from the history. Here we used the first solution, which is breakpoint distance.

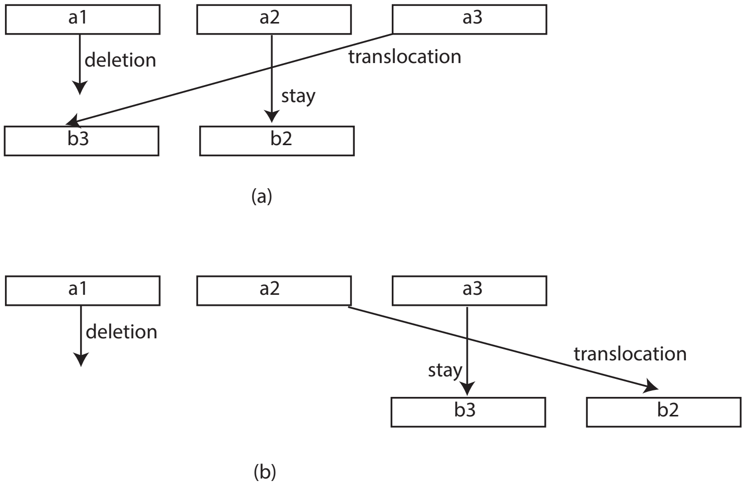


Figure 5.4: Two possible scenarios of rearrangement mapping.

5.4 Results and Discussion

5.4.1 MCMC-based method

The main part of the MCMC optimization process is as follows. First, we assign a random permutation to each inner node and the query node in the tree. In each iteration of the MCMC process, we simulate a random rearrangement event at each node. If the event decreases the cost of the tree, then it is accepted. Otherwise, the rearrangement event is accepted with its probability. Then we re-estimate the evolutionary rate parameters based on sampled permutations. Then we use the new parameters to sample new permutations. This process continues until the fitness of the permutations does not improve. In other words, we stop when the overall cost of the tree cannot be decreased. Figure 5.5 shows the flowchart of the MCMC algorithm we used.

The rearrangement model we used is as follows. There are five kinds of rearrangement events we considered, i.e., insertion, deletion, inversion, translocation and inverted translocation. To make a rearrangement event, we first pick a rearrangement type according to its probability, then we choose the location of the rearrangement event according to the distribution of the event, then perform the rearrangement event on the given permutation. The probability of each rearrangement type is set equally. Note that this will only affect the efficiency of searching the solution space, not the computation of the goal function.

We did experiments using *Drosophila melanogaster* as the query species and 9 others as informant species from the UCSC database. The phylogeny of the query and informant species is shown in Figure 5.6. The database is composed of blocks of orthologous sequences from these species and is generated using MultiZ. Because the query sequences have genomic positions with them in the database, we know the ground truth of the order of the query sequences. Note that the database only contains orthologous sequence blocks. It does not contain any inference results on sequence orders. We originally wanted to also use human chromosome 22 and *C. elegans* chromosome 3 as query species and their closely-related species as informant species, but the UCSC genome database does not have mappings of the segments of these species with genome rearrangement events in the mappings.

We first extracted those alignment blocks in the maf file containing these species and removed other species. Then we removed unequal gene content. Then we have a mapping of orthologous segments from these species and their genomic order in each species. We made three test data sets from our input. A small test data set contains 4 species and 5 segments for each species. A mid test data set contains 10 species and 20 segments. A large test data set contains 10 species and 135 segments. The phylogeny of the species in the small test data set is shown in Figure 5.7. The phylogeny in the mid and the large test data sets are the same as in Figure 5.6.

The results are shown in Table 5.1 and Table 5.2. The difference between these tables is that Table 5.1 shows the value of the fitness function after the run, while Table 5.2 actually measures the accuracy vs ground truth. Note that for the same data set, the initial values in the ”before” column are different for different rows. This is because the initial solution is randomly generated. From Table 5.1 we can see that the goal function value decreases tremendously as we increase the number of iterations in the MCMC algorithm. The goal function value tends to converge as the number of iterations increases. However, from Table 5.2 we can see that for all three experiments, even if we greatly increase the number of MCMC iterations, the breakpoint distance between the resulting query permutation and the ground-truth query permutation is not improved.

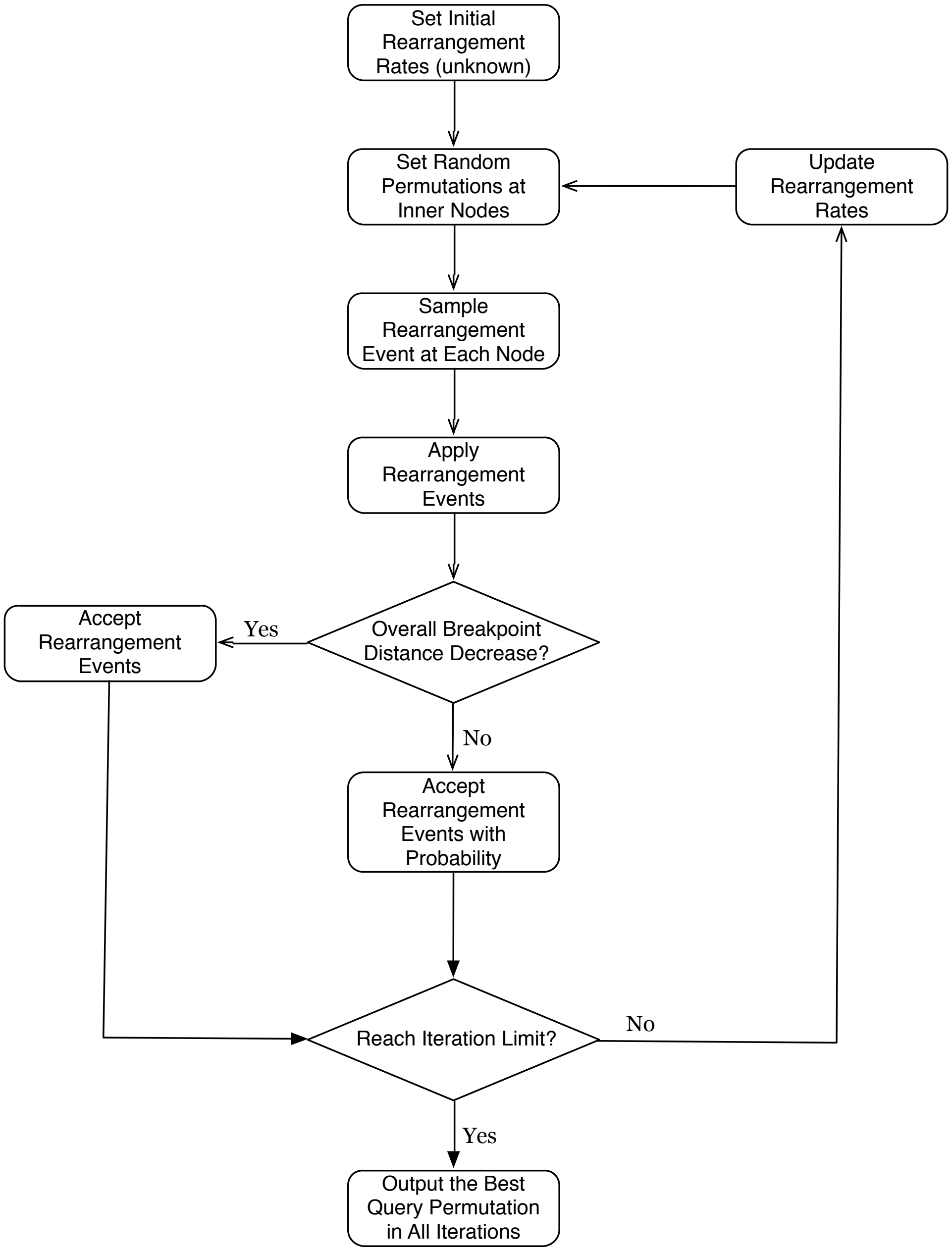

Set Initial Rearrangement Rates (unknown)
Set Random Permutations at Inner Nodes
Update Rearrangement Rates
Sample Rearrangement Event at Each Node
Apply Rearrangement Events
Overall Breakpoint Distance Decrease?
Yes
Accept Rearrangement Events
No
Accept Rearrangement Events with Probability
Reach Iteration Limit?
No
Yes
Output the Best Query Permutation in All Iterations

Figure 5.5: Flowchart of the MCMC algorithm used to infer query sequence order.

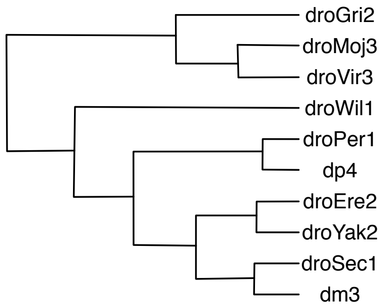


Figure 5.6: Phylogeny of *Drosophila melanogaster*(dm3) and 9 informant species

The inconsistency between the decrease of the goal function value and the decrease of the distance between the optimized sequence order and the real order indicates that our goal function may not be a good one. In other words, using breakpoint distance does not give the solution we want. As for the MCMC algorithm, it is indeed effective in optimizing the goal function. Then we analyzed the property of breakpoint distance and found that it is a kind of Euclidean distance [65]. Actually, we proved that using any Euclidian distance measurement in optimizing the permutations will give an output permutation of the query segments in the same order as the query's sibling. If the sibling is an input informant permutation, in this situation, the output of the query's permutation will just converge to the input permutation. See Theorem 1.

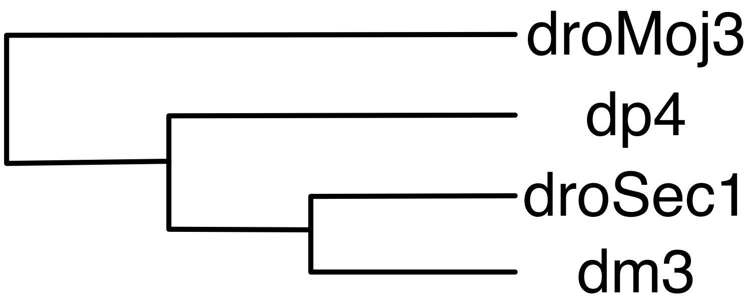


Figure 5.7: Phylogeny of species in small test data set.

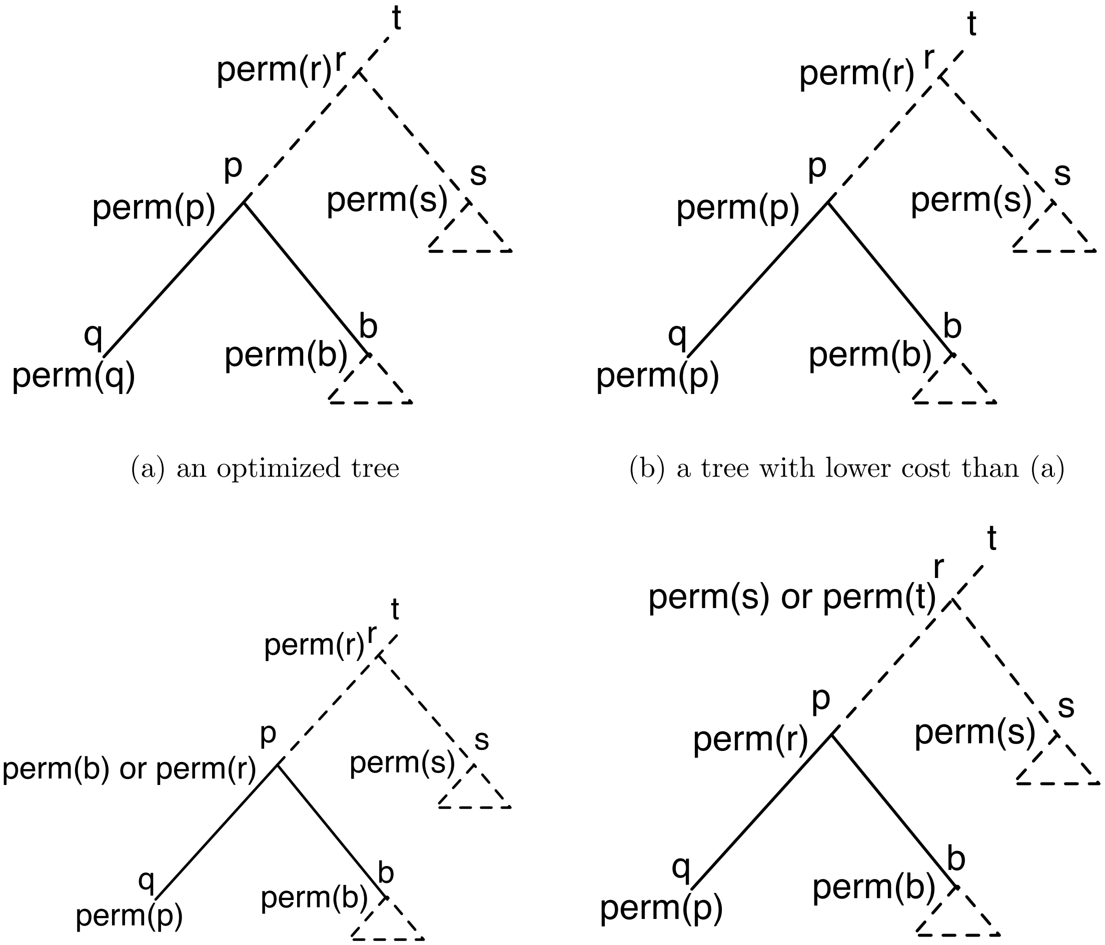
Theorem 1. *Using any Euclidean distance measurement in optimizing permutations in a tree will give a query permutation the same as one of the leaves in the tree.*

Table 5.1: Optimization results of overall breakpoint distance in tree for small, mid and large size experiments. The ”before” columns contain breakpoint distances of the initial tree. The ”after” columns contain the breakpoint distances of the optimized tree.

		small		mid		large	
		before	after	before	after	before	after
#iterations	10	17	8	309	274	2376	2376
	100	16	6	307	207	2373	2286
	1000	14	4	305	109	2381	1942
	10000	8	4	319	41	2369	1359

Table 5.2: Optimization results of breakpoint distance between query's initial permutation and true permutation for small, mid and large size experiments. The ”before” columns contain the breakpoint distances between the initial query permutation and the query's true permutation. The ”after” columns contain breakpoint distances between the optimized query permutation and the query's true permutation.

		small		mid		large	
		before	after	before	after	before	after
#iterations	10	2	1	18	17	131	131
	100	2	4	19	18	130	130
	1000	2	4	19	17	133	134
	10000	2	3	16	18	134	127



(a) an optimized tree

(b) a tree with lower cost than (a)

(c) a tree with lower cost than (b)

(d) a tree with lower cost than (c)

Figure 5.8: Illustration of optimization with additive cost function.

Proof. The situation here is like in the problem of optimal lifted alignment, where given a labeling of sequences at the leaves of a phylogenetic tree, a lifted alignment (in which each internal node is labeled with one of the leaf sequences) is close to the best possible.

Let d be the distance measurement, which means it satisfies the following three conditions for any permutation x and y:

- $d(x, y) \geq 0$ and $d(x, y) = 0$ if and only if $x = y$
- $d(x, y) = d(y, x)$
- $d(x, y) + d(y, z) \geq d(x, z)$

Without loss of generality, we assume the tree structure is as in Figure 5.8a. For any assignment in the tree, if the permutation at q is not the same as the permutation at p, then we can set permutation at q to perm(p) to reduce the cost. So perm(q) must be equal to perm(p), as shown in Figure 5.8b. Then at node p, we will argue that perm(p) is either perm(r) or perm(b). If it's not, then we can set perm(p) to perm(b) or perm(r) and get a lower-cost tree, as shown in Figure 5.8c. If perm(p)=perm(b), then perm(q)=perm(p)=perm(b) and proof is done. Otherwise, perm(p) must be equal to perm(r), as shown in Figure 5.8d. Similarly, for node r, the permutation must be either perm(s) or perm(t). We can reroot the tree at node q. We showed that the permutation at a node must be the same as one of the two permutations at its children nodes. Thus, permutation at q must be equal to a permutation of the leaves in the tree. □

Breakpoint distance has been extensively used in other researches as the metric of gene order distance [3, 21, 22, 50, 123, 134, 171]. We also used breakpoint distance as an approximation to the real sequence order distance. We want to know how does this affect our results.

Corollary 2. *Using breakpoint distance as optimization goal function will give an optimization result in which the query segments' order is the same as one of the segment orders of the input species.*

Proof. Because breakpoint distance satisfies the definition of a mathematical distance, according to theorem 1, the segments' order at the query node will be one of the leaf nodes' segment orders. □

5.4.2 Graph-algorithm-based method

From the above experimental results we have several observations. While the breakpoint distance may not be a good choice in our goal function, MCMC algorithm is indeed effective in optimizing the goal function, even though the size of the search space is exponential to the number of input sequences. Thus MCMC algorithm still has the potential to solve the problem as long as we can find a proper goal function. We proved that using any Euclidean distance as the optimization goal function will lead to degenerated solution where the inferred segment order will be equal to one of the input segment orders. An alternative is to use a non-distance based scoring scheme.

The property in Euclidean distance definition that leads to degenerated solution is the property that $d(x, x) = 0$. In a non-distance based metrics, we want $d(x, x) > 0$. One instance of such metrics is probability. For example, solutions to inferring the probability of one segment order given another order can be classified into exact solutions and approximate solutions. To get an exact solution, one needs to compute the probability distribution of the recombined order given the original order. Because the solution space size is exponential to the input and the number of segments and ways of rearrangements are large, it is impossible to compute the exact distribution of the recombined order. Another choice is to estimate the most likely rearrangement history between two orders and use its probability as the distance between the two orders. However, due to the large search space, such a path must be estimated from a heuristic search or randomized search, which will make the estimated path highly unstable from one run to another, which will make the estimated probability inaccurate. Yet another option for tackling the exponential solution space is that we decompose the whole solution space into orthogonal subspaces.

In this section, we explore such possibilities. We give another formulation of the rearrangement inference problem, which is a combinatorial optimization representation. For each node in the tree, we compute the probability of one segment being present before another segment by combining such probabilities in its two child nodes. For example, in Figure 5.9, the order of segments a and b is given at the three leaves A, B, and C. At node D, the probability of a being present before b is defined by

$$P(a < b \text{ in } D) \equiv \alpha * P(a < b \text{ in } A) + (1-\alpha) * P(a < b \text{ in } B),$$

where α is the weight of the left child and $1-\alpha$ is the weight of the right child. The weights can be defined in terms of branch lengths. We weigh the left and right probabilities of sequence a preceding sequence b by right branch length and left branch length respectively. Formally we have:

$$Pr(a < b|D) = \frac{Pr(a < b|A) \cdot l_{DB} + Pr(a < b|B) \cdot l_{DA}}{l_{DB} + l_{DA}}.$$

For node A, because it is a leaf, so $P(a < b \text{ in } A) = 1$. Finally we will have a distribution on the order of a and b in the root node E. In our original problem, the query node is a leaf. In this case, we can reroot the tree. In our example, assume all branches have the same length. Then we will have the following probabilities:

$$\begin{aligned}
P(a < b \text{ in } A) &= 1.0, \\
P(a > b \text{ in } A) &= 0.0, \\
P(a < b \text{ in } B) &= 0.0, \\
P(a > b \text{ in } B) &= 1.0, \\
P(a < b \text{ in } C) &= 0.0, \\
P(a > b \text{ in } C) &= 1.0, \\
P(a < b \text{ in } D) &= 0.5 * P(a < b \text{ in } A) + 0.5 * P(a < b \text{ in } B) = 0.5, \\
P(a > b \text{ in } D) &= 0.5 * P(a > b \text{ in } A) + 0.5 * P(a > b \text{ in } B) = 0.5, \\
P(a < b \text{ in } E) &= 0.5 * P(a < b \text{ in } D) + 0.5 * P(a < b \text{ in } C) = 0.25, \\
P(a > b \text{ in } E) &= 0.5 * P(a > b \text{ in } D) + 0.5 * P(a > b \text{ in } C) = 0.75,
\end{aligned}$$

After computing the probability distribution at each node in the tree, we want to find an order at the query node which maximizes the overall probability of the order. Let's examine

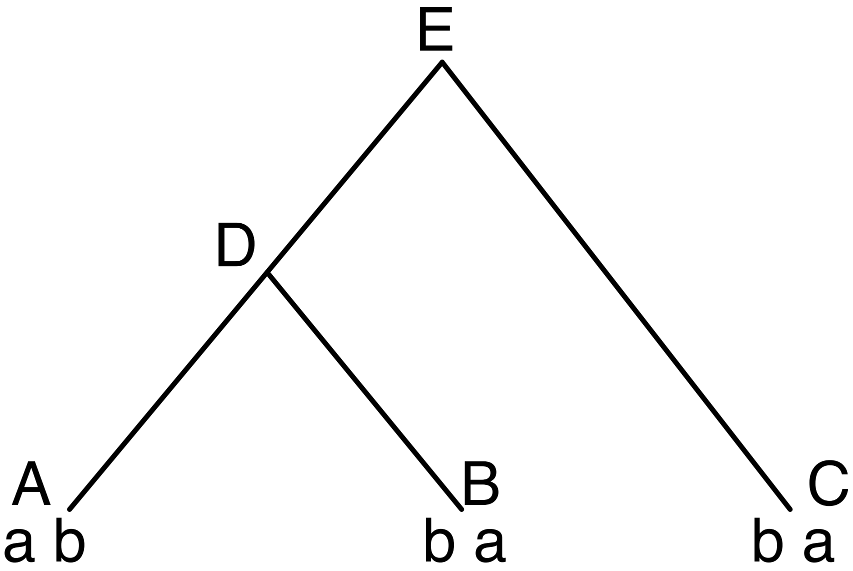


Figure 5.9: A phylogeny with three species and two sets of homologous sequences.

another example, shown in Figure 5.10. In this example, there are three segments at each node. At each node, we have $P(x < y)$ for any pair of x and y. From the probability distribution at the root node, we build a graph in the following process. Each segment is converted to a node. Each pair of nodes has two directional edges. The edge from node x to y has weight $P(x < y)$. The resulting graph for the root node is shown in Figure 5.11. The edges in solid lines represent a solution, which is a directional Hamilton path in the graph. Our goal is to find the Hamilton path with the maximum probability. To convert the product of probabilities into a sum of edge weights, we can take *log* on each probability. Then finding an Hamilton path with maximum probability becomes finding an path with maximum weight. We use MCMC algorithm to find such a path.

There are several advantages of this method. First, it represents all possible paths in a single graph. Remember that all other algorithms can only infer a single order of the input segments. This graph-based representation gives essentially a probability distribution of the order of the input segments. Second, though finding a max-weight Hamilton path is still NP-hard, but this reduces the original search problem by exponential scale. Because

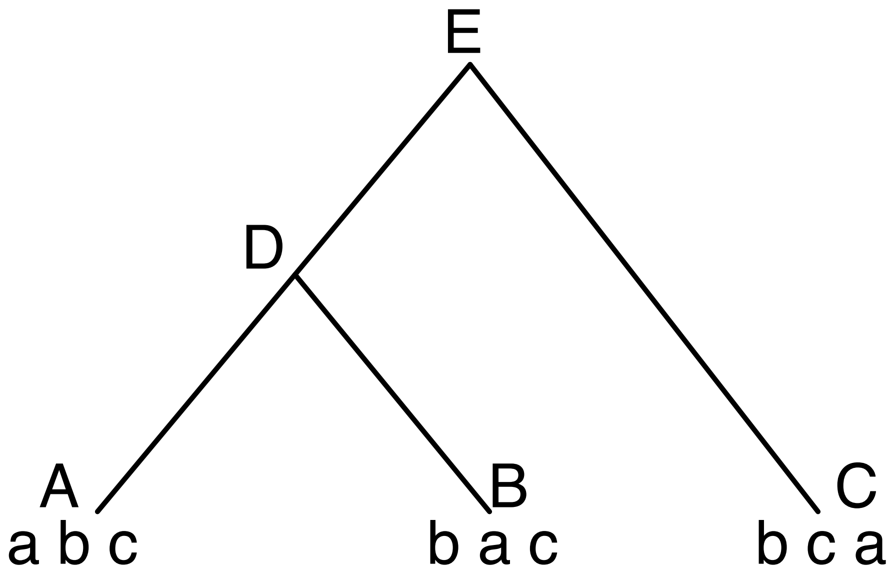


Figure 5.10: A phylogeny with three species and three sets of homologous sequences.

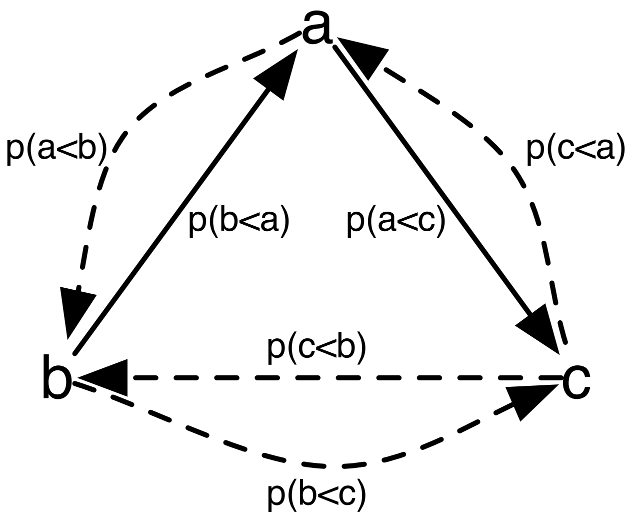


Figure 5.11: Conversion from rearrangement inference problem to Hamilton path problem.

in the original problem, we need to consider all possible rearrangements at each node. Assume there are m species and n segments, then there are $O(n!)$ permutations at each node, which is $O(n^n)$. There are m species, so there are $2m$ nodes in the tree. Thus there are $O((n^n)^{2m}) = O(n^{2mn})$ possible cases in the tree. In our graph-based formulation, there are only $n!$ different Hamilton paths. Furthermore, finding a max-weight Hamilton path is a classical combinatorial optimization problem and there exists many existing solutions to this problem. Note that even if we just compute a single output order, the search space is much smaller than the search space of the original problem. Given any path, we can compute its probability in O(n) time, compared with in the original problem where we need to sample possible mutations at each inner node.

Using our three test data sets, the experimental results are shown in Table 5.3 and Table 5.4. The difference between these tables is that Table 5.3 shows the length of the initial and final path, while Table 5.4 actually measures the accuracy vs ground truth. Note that in both table the "before" values in a column are different for different rows. This is because the initial permutation is randomly chosen.

Table 5.3: Optimization results of longest path in graph for small, mid and large size experiments. The "before" columns contain the initial distances. The "after" columns contains the optimized path length. These are all log distances.

		small		mid		large	
		before	after	before	after	before	after
#iterations	10	-26.63	-1.28	-400.35	-189.94	-2929.03	-2973.06
	100	-24.84	-1.28	-355.13	-157.89	-3085.46	-2883.31
	1000	-25.80	-1.28	-400.66	-169.88	-2976.59	-2864.47
	10000	-47.02	-1.28	-394.79	-165.51	-3065.70	-2956.32

We can see that except for the small data set, the graph-based algorithm still failed to get closer to the ground truth.

Experimental Results on Simulated Data Next we want to separate our algorithm to the problem from the underlying model. We tested our algorithm on simulated permutations generated using our own rearrangement model. First we fix the phylogeny, then we assign a random permutation at the root. Next, from the root to the leaves, we simulate

Table 5.4: Optimization results of query permutation for small, mid and large size experiments. The "before" columns contain the breakpoint distances between the original query permutation and the query's true permutation. The "after" columns contain breakpoint distances between the optimized query permutation and the query's true permutation.

		small		mid		large	
		before	after	before	after	before	after
#iterations	10	4	1	18	17	131	129
	100	3	1	15	17	134	130
	1000	3	1	17	16	128	127
	10000	2	1	16	17	133	132

rearrangement events sampled by our rearrangement model. Finally we run our algorithm on the simulated data to infer the query permutation. Our experimental results showed that our algorithm can correctly infer the permutation at the query node for small data sets and works to some extent for mid-size data sets, but failed for large data sets. The results are shown in Table 5.5 (in terms of path length) and Table 5.6 (in terms of breakpoint distance to the ground truth permutation).

Table 5.5: Optimization results of longest path in graph for small, mid and large size experiments. The "before" columns contain the initial distances. The "after" columns contains the optimized path length.

		small		mid		large	
		before	after	before	after	before	after
#iterations	10	-12.80	-0.59	-330.25	-121.71	-3039.66	-3016.64
	100	-14.02	-0.59	-326.36	-153.36	-2977.25	-2885.40
	1000	-12.79	-0.59	-373.58	-129.02	-2972.08	-2856.95
	10000	-37.06	-0.59	-375.32	-166.01	-3002.51	-2841.58

To further validate our assumption, we investigated the relationship between breakpoint distance and branch length of the simulated data. We computed breakpoint distances and branch lengths between all pairs of leaf nodes in the tree. The results are shown in Table 5.7. The relationship between breakpoint distance and path length is shown in Figure 5.12.

Table 5.6: Optimization results of query permutation for small, mid and large size experiments. The "before" columns contain the breakpoint distances between the original query permutation and the query's true permutation. The "after" columns contain breakpoint distances between the optimized query permutation and the query's true permutation.

		small		mid		large	
		before	after	before	after	before	after
#iterations	10	2	0	18	9	132	131
	100	3	0	15	9	130	126
	1000	2	0	18	9	130	125
	10000	4	0	17	8	131	124

Table 5.7: Examples of breakpoint distances and branch lengths in phylogeny.

dm3	droSec1	0	0.041684
dm3	droYak2	0	0.077513
dm3	droEre2	0	0.078749
dm3	dp4	1	0.196726
dm3	droPer1	1	0.19749
dm3	droWil1	2	0.250317
dm3	droVir3	2	0.274038
dm3	droMoj3	2	0.289755
dm3	droGri2	2	0.281076
...	...	...	...
droGri2	dm3	2	0.281076
droGri2	droSec1	2	0.28681
droGri2	droYak2	2	0.298465
droGri2	droEre2	2	0.299701
droGri2	dp4	2	0.310848
droGri2	droPer1	2	0.311612
droGri2	droWil1	4	0.331701
droGri2	droVir3	3	0.160424
droGri2	droMoj3	0	0.176141

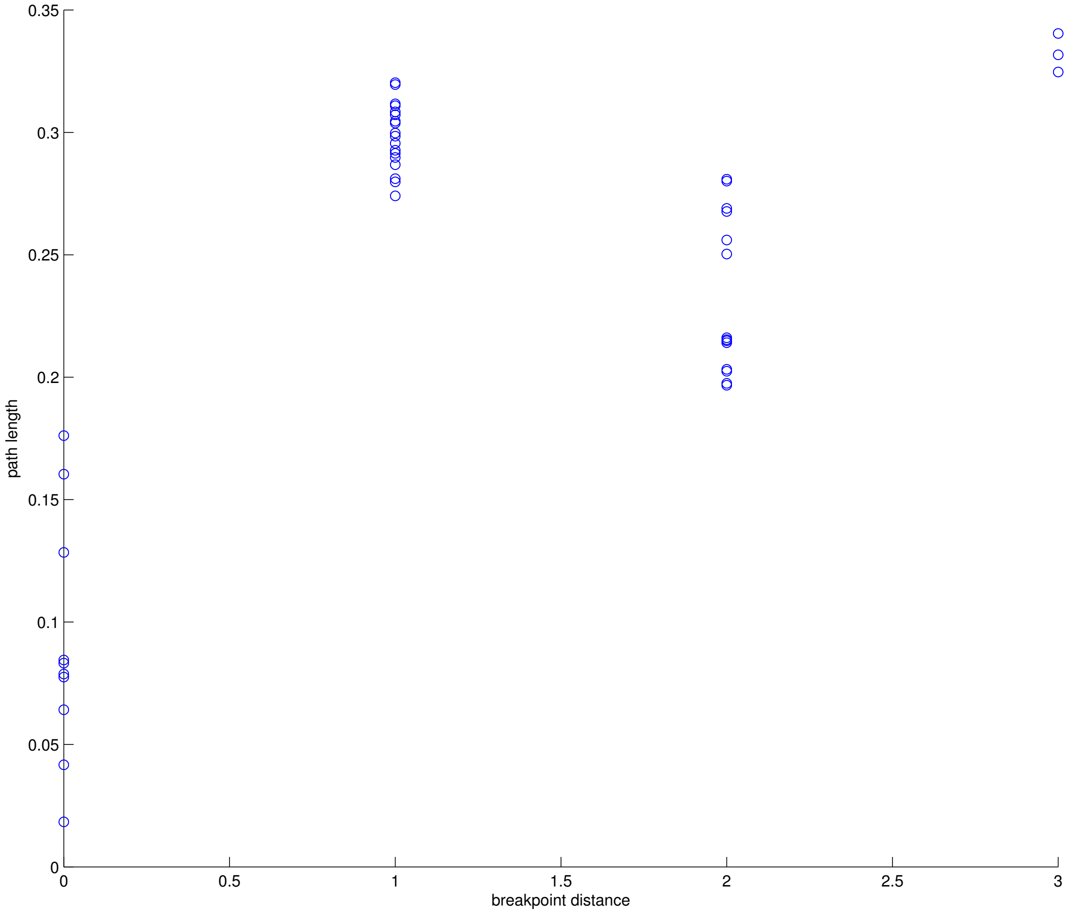


Figure 5.12: Relationship between breakpoint distance and path length of simulated data.

We can see that with simulated data, the optimization result is much closer to the truth than using the real data. The only difference between the simulated data and real data is that the simulated data is generated according to our rearrangement model while the real data is not. This implies that our model is indeed inaccurate for the problem. From the relationship between breakpoint distance and path probability, we can see that the breakpoint distance is a good approximation of the path length in the probability graph. This means using the probability of path as a goal function is a reasonable metric.

5.4.3 Conclusion so far

We give a probabilistic model of genome rearrangement. Based on that model, we develop an MCMC algorithm to infer the order of genome segments in a query species using orders of orthologous segments in informant species. Experiment results show that the MCMC algorithm can find the solution on small data set with breakpoint distance measurement. But it failed for large data sets. The failures result from the discrepancy between the optimal goal function value which is the overall breakpoint distance and the ground-truth solution. The MCMC algorithm actually can find the optimal solution under the given goal function. We prove that using any Euclidean distance metrics as the goal function of the optimization process will result in the order of segments of the query species being the same as the order of one of the leaf nodes in the tree, which means the optimization program just picks one input as output under such case. We also tried a graph-based algorithm. The results show that while the distance measurement can be well approximated by breakpoint distance, the model of genome rearrangement needs to be refined.

Chapter 6

Conclusions and Future Work

The contributions of this dissertation include several aspects. First, we formulated four problems. The first one is sequence similarity search using a multiple alignment database and a phylogeny-based scoring system. The second one is multiple alignment improvement. The third one is interspecies short read mapping. The fourth one is genome rearrangement inference. Second, we gave solutions to the problems. For the sequence similarity search problem, we integrated the multiple alignment database and the phylogeny-based probabilistic scoring scheme in an EM framework. Our method successfully find good matches between a query sequence and a database. Our method performs better than competitors. In the multiple alignment problem, we embedded the multiple-alignment-phylogeny framework into an iterative optimization process. We customize the sequence similarity search algorithm to fit in this process. Our method can successfully correct errors in multiple alignment database. Our method also performs better than other competitor algorithms. For the interspecies mapping problem, we tackle the problem from both theoretical computation aspect and experimental aspect. We have successfully mapped substantially more reads to their orthologous locations in reference sequence than competitor programs. For the genome rearrangement inference problem, we successfully applied MCMC algorithm to the breakpoint distance optimization problem, though the breakpoint distance metrics may not be a proper one. We also formulated the problem as a graph problem and developed a partially working algorithm for it which can find solutions to small and mid-size data sets on simulated data. We pointed to possible future directions and solutions. Last but not least, we formulated these problems in a common biological model and algorithmic framework. With the fast accumulation of newly assembled whole genomes and increasing understanding of the problems in bioinformatics, more problems can be related to each other and solved in the same framework. This dissertation is an example of this kind of work.

6.1 Sequence Similarity Search

In Chapter 2 we studied the sequence similarity search problem. We introduce PhyLAT, the Phylogenetic Local Alignment Tool, to compute local alignments of a query sequence against a fixed multiple-genome alignment of closely related species. PhyLAT uses a known phylogenetic tree on the species in the multiple alignment to improve the quality of its computed alignments while also estimating the placement of the query on this tree. It combines a probabilistic approach to alignment with seeding and expansion heuristics to accelerate discovery of significant alignments. We provide evidence, using alignments of human chromosome 22 against a 5-species alignment from the UCSC Genome Browser database, that PhyLAT's alignments are more accurate than those of other commonly used programs, including BLAST, POY, MAFFT, MUSCLE, and CLUSTAL. PhyLAT also identifies more alignments in coding DNA than does pairwise alignment alone. Finally, our tool determines the evolutionary relationship of query sequences to the database more accurately than do POY, RAxML, EPA, or pplacer.

Several opportunities exist to improve PhyLAT's performance and utility. First, our assumption that successive residues in the query or successive columns in the multiple alignment are stochastically independent is not realistic. It could be useful to add a dependence model between adjacent bases/columns to improve the accuracy of alignment.

Second, PhyLAT assumes that its query is a single DNA sequence. It would be useful to handle queries that are themselves multiple alignments. However, there are unresolved computational complexity issues with this extension. In particular, we cannot simply enumerate all simultaneous branch placements of all species in the query multiple alignment with respect to the database multiple alignment. Some efficient way must be found to form a hypothesis about how the query species and database species relate within a single tree.

Third, we need to develop efficient approaches to estimate the statistical significance of gapped alignments in PhyLAT without resorting to expensive simulations. Karlin and

Altschul's theory for ungapped alignments is not applicable to gapped alignments [69]. However, many kinds of alignments involving gaps were empirically demonstrated to follow EVD [6, 46, 133, 140]. To derive empirical parameters of KA statistics, tens of thousands of alignments need to be generated and scored. This process is computationally expensive but may not give parameters accurate enough for computing statistical significance for new alignments, especially when the composition of new data is different from that in simulation. In [133], a rescaling technique was explored to use a standard score distribution to estimate statistical significance of profile alignments of new profiles. In Aleksandar's work [2], the island method was applied to collect more optimal scores from a single simulated alignments.

Finally, it would be useful to consider alternative database trees during alignment, e.g. to accommodate the possibility that a query is not being aligned to an orthologous locus in the database. While it is possible to sum probabilities over multiple tree hypotheses, the increased computational cost of using multiple trees makes it imperative to be careful not to consider too many such alternative trees. Heuristics for picking likely trees would help to guide the search.

6.2 Multiple Alignment Improvement

In Chapter 3 we discussed is multiple alignment improvement. Multiple alignments are often the initial input of down-stream analysis. The accuracy of multiple alignments will directly impact the results of the analysis. Though there are many tools for generating multiple alignments, there are few tools for improving existing multiple alignments. The reason we need an improvement tool besides multiple alignment generation tools is that we can use fast algorithms to generate initial multiple alignment and then use more complex models to fine tune the alignments. We developed PhyMAIT, which is a phylogeny-aware multiple alignment improvement tool. It uses iterative optimization to improve multiple alignments. It also incorporates phylogenetic information and uses probabilistic alignment framework to accurately align sequences. We tried several strategies for the alignment process. Our experiment results show that our algorithm can improve the quality of existing multiple alignments.

One future work is the acceleration of the multiple alignment process. Because alignments can be as long as thousands of columns with each column containing dozens of bases from different species. Redoing the whole alignment during each iteration is very time consuming. After a few iterations, some parts of the alignment may be very steady and change little. In such cases, we have good reasons to ignore those regions and focus on only the regions with less stability.

6.3 Short Read Mapping

In Chapter 4 we discussed short read mapping problem. We identified the need for inter-species mapping. Before our work, there is only intra-species mapping. There are several possible use cases for interspecies mapping. One example is metagenomics. With the development of genome sequencing techniques, short reads are generated at a unprecedented pace. Often it is the case that a species' genome is sequenced for the first time. In this case, no reference sequence of the same species is available. We then can use one or more closely-related species' sequences as the reference sequence. We formulated the short read mapping problem in the context of multiple sequence alignment with phylogenetic information. We then gave an algorithm to do short read mapping. Our experiment results show that our algorithm can align short reads more accurately than other intra-species mapping tools.

One thing to explore is how to speed up the short read mapping process. Because the number of short reads generated from some applications are very large, acceleration of the alignment process is very important. One solution is parallel computation of the alignment phase. Because the alignment of different reads are irrelevant of each other, so they can be done separately on different machines. This is a perfect use case for Map-Reduce framework. During the Map phase, short reads are hashed into a key-value pair $< readid, readsequence >$. During the map phase, each read will be aligned to the multiple alignment where there is a seed. The output of the map phase are pairs of $< alignedreadposition, < read, alignmentscore >>$. Note that the second element is a tuple of the read sequence and score. The output of the map phase will be grouped by aligned read position, so reads mapped to the same position will be grouped together. Then during the reduce phase, we can use scores to decide which read is the true orthologous sequence.

6.4 Genome Rearrangement Inference

In Chapter 5 we discussed is genome rearrangement inference. With new genomes being sequenced everyday, assembling the sequences is a huge task. Often when people try to assemble the sequences, they can not fully resolve the order of the sequences. However, with the help of other informant sequences and their phylogenetic relationships with the target sequence, inferring the order of sequences of the new genome may become easier. We defined a model of genome rearrangement, which can account for inversion, translocation and inverted translocation. This is a probabilistic model. We tried to use breaking point distance as the measurement of distance between different orders of sequences. We found that it did not work. We proved that using any Euclidean measurement in the optimization algorithm will result in a degenerated solution. However, our algorithm did find the optimal or near-optimal solutions under the condition of using breakpoint measurement. This is a hint that MCMC algorithm will work for finding the evolution history of sequences, which can be used together with a probabilistic measurement to find the order of sequences in target genome.

Because this problem is very open, there are still a lot of things to do. One is improving the genome rearrangement model. This is a more biological problem than algorithmic. It is also the most important part in this problem. Starting with a wrong model, whatever complex algorithms we use, we will not likely to get the correct results. The model contains two parts. One is what kinds of events we want to model. Examples are transitions, inversions, indels and translocations. Currently we just consider translocations, inversions and inverted translocations. In other words, we only considered equal-gene-content cases. We filtered the input data so they only contain the same set of genes. This is a very strong restriction. Another is the parameters of the model. One example is the transition rate. Different sets of species may have different rates. Even for the same set of species, rates on different branches may be different. Another possible future work is modeling this problem in a different way. Current models, including our model, can only give a single output of the optimized order of the input sequences. Because MCMC is a randomized algorithm, the output may be different each time. However, compared with the number of possible solutions, this is still a very small number. How to evaluate the reliability of the output with respect to other possible solutions is very important. One possible solution is finding a representation which can express all the possible solutions in parallel. We modeled the problem using a graphical model, whose nodes represent the genes and edges represent the relative probabilistic order. Then we find the order with the largest probability by finding the Hamilton path in the graph with the largest weight. Given any order, we can also calculate its probability by calculating the weight of the corresponding Hamilton path. Though finding the optimal Hamilton path is also an NP-hard problem, but it reduces the complexity of the original problem by orders of magnitude. Furthermore, there exists many approximation algorithms for this problem.

References

[1] C.A. Albers, G. Lunter, D.G. MacArthur, G. McVean, W.H. Ouwehand, and R. Durbin. Dindel: Accurate indel calls from short-read data. *Genome research*, 21(6):961–973, 2011.

[2] P. Aleksandar. Island method for estimating the statistical significance of profile-profile alignment scores. *BMC Bioinformatics*, 10(1):112, 2009.

[3] M.A. Alekseyev and P.A. Pevzner. Breakpoint graphs and ancestral genome reconstructions. *Genome research*, 19(5):943, 2009.

[4] C. Alkan, J.M. Kidd, T. Marques-Bonet, G. Aksay, F. Antonacci, F. Hormozdiari, J.O. Kitzman, C. Baker, M. Malig, O. Mutlu, et al. Personalized copy number and segmental duplication maps using next-generation sequencing. *Nature genetics*, 41(10):1061–1067, 2009.

[5] SF Altschul and W. Gish. Local alignment statistics. *Methods in enzymology*, 266:460–480, 1996.

[6] SF Altschul, TL Madden, AA Schaffer, J. Zhang, Z. Zhang, W. Miller, and DJ Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389, 1997.

[7] V. Bansal and O. Libiger. A probabilistic method for the detection and genotyping of small indels from population-scale sequence data. *Bioinformatics*, 27(15):2047–2053, 2011.

[8] E. Bao, T. Jiang, I. Kaloshian, and T. Girke. SEED: efficient clustering of next-generation sequences. *Bioinformatics*, 27(18):2502–2509, 2011.

[9] G. Bejerano, M. Pheasant, I. Makunin, S. Stephen, W.J. Kent, J.S. Mattick, and D. Haussler. Ultraconserved elements in the human genome. *Science*, 304(5675):1321, 2004.

[10] Steve Benson, Lois Curfman McInnes, Jorge Moré, Todd Munson, and Jason Sarich. TAO user manual (revision 1.9). Technical report.

[11] S.A. Berger, D. Krompass, and A. Stamatakis. Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood. *Systematic biology*, 60(3):291, 2011. original paper of EPA.

[12] Simon A. Berger and Alexandros Stamatakis. Aligning short reads to reference alignments and trees. *Bioinformatics*, 27(15):2068–2075, 2011. original paper of PaPaRa.

[13] Anne Bergeron, Mathieu Blanchette, Annie Chateau, and Cedric Chauve. Reconstructing ancestral gene orders using conserved intervals. In *Algorithms in Bioinformatics*, pages 14–25. Springer, 2004.

[14] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, TN Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[15] Piotr Berman and Sridhar Hannenhalli. Fast sorting by reversal. In *Combinatorial Pattern Matching*, pages 168–185. Springer, 1996.

[16] A. Bernal, U. Ear, and N. Kyrpides. Genomes online database (gold): a monitor of genome projects world-wide. *Nucleic Acids Research*, 29(1):126–127, 2001.

[17] Arjun Bhutkar, William M Gelbart, Temple F Smith, et al. Inferring genome-scale rearrangement phylogeny and ancestral gene order: a drosophila case study. *Genome Biol*, 8(11):R236, 2007.

[18] C.P. Bird, B.E. Stranger, M. Liu, D.J. Thomas, C.E. Ingle, C. Beazley, W. Miller, M.E. Hurles, and E.T. Dermitzakis. Fast-evolving noncoding sequences in the human genome. *Genome Biology*, 8(6):R118, 2007.

[19] M. Blanchette. Computation and analysis of genomic multi-sequence alignments. *Annu. Rev. Genomics Hum. Genet.*, 8:193–213, 2007.

[20] M. Blanchette, W.J. Kent, C. Riemer, L. Elnitski, A.F.A. Smit, K.M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E.D. Green, et al. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome research*, 14(4):708, 2004. original paper of TBA and multiz.

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

[22] G. Blin, E. Blais, P. Guillon, M. Blanchette, and N. El-Mabrouk. Inferring gene orders from gene maps using the breakpoint distance. *Comparative Genomics*, pages 99–112, 2006.

[23] Guillaume Blin, Guillaume Fertin, and Cedric Chauve. The breakpoint distance for signed sequences. In *1st Conference on Algorithms and Computational Methods for biochemical and Evolutionary Networks (CompBioNets' 04)*, volume 3, pages 3–16. King's College London publications, 2004.

[24] N. Bray and L. Pachter. MAVID: constrained ancestral alignment of multiple sequences. *Genome Research*, 14(4):693, 2004. Original paper of MAVID, a tool for multiple alignment. Uses ML ancestral sequence to align progressively.

[25] R.J. Britten. Divergence between samples of chimpanzee and human dna sequences is 5%, counting indels. *Proceedings of the National Academy of Sciences*, 99(21):13633, 2002.

[26] M. Brudno, S. Malde, A. Poliakov, C.B. Do, O. Couronne, I. Dubchak, and S. Batzoglou. Glocal alignment: Finding rearrangements during alignment. *Bioinformatics*, 19(90001):54–62, 2003.

[27] M. Brudno, A. Poliakov, A. Salamov, G.M. Cooper, A. Sidow, E.M. Rubin, V. Solovyev, S. Batzoglou, and I. Dubchak. Automated whole-genome multiple alignment of rat, mouse, and human. *Genome research*, 14(4):685–692, 2004.

[28] J. Buhler, U. Keich, and Y. Sun. Designing seeds for similarity search in genomic DNA. In *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 67–75. ACM, 2003.

[29] J. Buhler and R. Nordgren. Toward a Phylogenetically Aware Algorithm for Fast DNA Similarity Search. *Lecture Notes in Computer Science*, 3388:15–29, 2005.

[30] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T.L. Madden. BLAST+: architecture and applications. *BMC bioinformatics*, 10(1):421, 2009.

[31] A. Caprara. Sorting by reversals is difficult. In *Proceedings of the first annual international conference on Computational molecular biology*, page 83. ACM, 1997.

[32] Alberto Caprara. Formulations and hardness of multiple sorting by reversals. In *Proceedings of the third annual international conference on Computational molecular biology*, pages 84–93. ACM, 1999.

[33] Julian M Catchen, John S Conery, and John H Postlethwait. Inferring ancestral gene order. In *Bioinformatics*, pages 365–383. Springer, 2008.

[34] K. Chen, J.W. Wallis, M.D. McLellan, D.E. Larson, J.M. Kalicki, C.S. Pohl, S.D. McGrath, M.C. Wendl, Q. Zhang, D.P. Locke, et al. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nature Methods*, 6(9):677–681, 2009.

[35] Y. Chen, T. Souaiaia, and T. Chen. PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. *Bioinformatics*, 25(19):2514, 2009.

[36] L. Chindelevitch, Z. Li, E. Blais, and M. Blanchette. On the inference of parsimonious indel evolutionary scenarios. *Journal of Bioinformatics and Computational Biology*, 4(3):721–744, 2006.

[37] P. Cliften, P. Sudarsanam, A. Desikan, L. Fulton, B. Fulton, J. Majors, R. Waterston, B.A. Cohen, and M. Johnston. Finding functional features in Saccharomyces genomes by phylogenetic footprinting. *Science*, 301(5629):71, 2003.

[38] Mary E Cosner, Robert K Jansen, Bernard ME Moret, Linda A Raubeson, Li-San Wang, Tandy Warnow, and Stacia Wyman. An empirical comparison of phylogenetic methods on chloroplast gene order data in campanulaceae. In *Comparative Genomics*, pages 99–121. Springer, 2000.

[39] A. Cox. Eland: efficient local alignment of nucleotide data. *unpublished, http://bioit. dbi. udel. edu/howto/eland*, 2006.

[40] J. David and S.F. Altschul. Improved Sensitivity of Nucleic Acid Database Searches Using Application-Specific Scoring Matrices. 1991.

[41] A.B. Diallo, V. Makarenkov, and M. Blanchette. Finding maximum likelihood indel scenarios. *Lecture Notes in Computer Science*, 4205:171, 2006.

[42] A.B. Diallo, V. Makarenkov, and M. Blanchette. Exact and heuristic algorithms for the indel maximum likelihood problem. *Journal of Computational Biology*, 14(4):446–461, 2007.

[43] Xavier Didelot, Daniel Lawson, Aaron Darling, and Daniel Falush. Inference of homologous recombination in bacteria using whole-genome sequences. *Genetics*, 186(4):1435–1449, 2010.

[44] S.R. Eddy. Multiple alignment using hidden Markov models. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, volume 3, pages 114–120, 1995.

[45] SR Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.

[46] S.R. Eddy. A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PLoS Computational Biology*, 4(5):e1000069, 2008.

[47] S.R. Eddy, G. Mitchison, and R. Durbin. Maximum discrimination hidden Markov models of sequence consensus. *Journal of Computational Biology*, 2(1):9–23, 1995.

[48] R.C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792, 2004.

[49] J. Felsenstein. Phylip-phylogeny inference package (version 3.2). 1989.

[50] Z. Fu and T. Jiang. Computing the breakpoint distance between partially ordered genomes. In *Proceedings APBC*, pages 237–246, 2007.

[51] P.A. Fujita, B. Rhead, A.S. Zweig, A.S. Hinrichs, D. Karolchik, M.S. Cline, M. Goldman, G.P. Barber, H. Clawson, A. Coelho, et al. The ucsc genome browser database: update 2011. *Nucleic acids research*, 39(suppl 1):D876, 2011.

[52] Cyril Gallut and Véronique Barriel. Cladistic coding of genomic maps. *Cladistics*, 18(5):526–536, 2002.

[53] Cyril Gallut, Veronique Barriel, and Regine Vignes. Gene order and phylogenetic information. In *Comparative Genomics*, pages 123–132. Springer, 2000.

[54] H. Ge, K. Liu, T. Juan, F. Fang, M. Newman, and W. Hoeck. FusionMap: detecting fusion genes from next-generation sequencing data at base-pair resolution. *Bioinformatics*, 27(14):1922, 2011.

[55] T. Golubchik, M.J. Wise, S. Easteal, and L.S. Jermiin. Mind the gaps: evidence of bias in estimates of multiple sequence alignments. *Molecular biology and evolution*, 24(11):2433–2442, 2007.

[56] O. Harismendy, P.C. Ng, R.L. Strausberg, X. Wang, T.B. Stockwell, K.Y. Beeson, N.J. Schork, S.S. Murray, E.J. Topol, S. Levy, et al. Evaluation of next generation sequencing platforms for population targeted sequencing studies. *Genome Biol*, 10(3):R32, 2009.

[57] L.D.W. Hillier, G.T. Marth, A.R. Quinlan, D. Dooling, G. Fewell, D. Barnett, P. Fox, J.I. Glasscock, M. Hickenbotham, W. Huang, et al. Whole-genome sequencing and variant discovery in C. elegans. *Nature methods*, 5(2):183–188, 2008.

[58] S. Hoffmann, C. Otto, S. Kurtz, C.M. Sharma, P. Khaitovich, J. Vogel, P.F. Stadler, and J. Hackermüller. Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Computational Biology*, 5(9):e1000502, 2009.

[59] I. Holmes and W.J. Bruno. Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics*, 17(9):803, 2001.

[60] F. Hormozdiari, C. Alkan, E.E. Eichler, and S.C. Sahinalp. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome research*, 19(7):1270–1278, 2009.

[61] F. Hormozdiari, F. Hach, S.C. Sahinalp, E.E. Eichler, and C. Alkan. Sensitive and fast mapping of di-base encoded reads. *Bioinformatics*, 27(14):1915, 2011.

[62] D. Huson, D. Richter, S. Mitra, A. Auch, and S. Schuster. Methods for comparative metagenomics. *BMC bioinformatics*, 10(Suppl 1):S12, 2009.

[63] D.H. Huson, A.F. Auch, J. Qi, and S.C. Schuster. MEGAN analysis of metagenomic data. *Genome research*, 17(3):377–386, 2007.

[64] O. Isakov, S. Modai, and N. Shomron. Pathogen detection using short-RNA deep sequencing subtraction and assembly. *Bioinformatics*, 27(15):2027–2030, 2011.

[65] Arash Jamshidpey, Aryo Jamshidpey, and David Sankoff. Sets of medians in the non-geodesic pseudometric space of unsigned genomes with breakpoints. *BMC genomics*, 15(Suppl 6):S3, 2014.

[66] J.L. Jensen and J. Hein. Gibbs sampler for statistical multiple alignment. *Statistica Sinica*, 15(4):889, 2005.

[67] Haim Kaplan, Ron Shamir, and Robert E Tarjan. A faster and simpler algorithm for sorting signed permutations by reversals. *SIAM Journal on Computing*, 29(3):880–892, 2000.

[68] S. Karlin and SF Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences*, 87(6):2264, 1990.

[69] S. Karlin and SF Altschul. Applications and statistics for multiple high-scoring segments in molecular sequences. *Proceedings of the National Academy of Sciences*, 90(12):5873, 1993.

[70] K. Karplus, C. Barrett, and R. Hughey. Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846, 1998.

[71] K. Katoh, K. Misawa, K. Kuma, and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*, 30(14):3059, 2002. Original paper of MAFFT. Use FFT to find homologous regions. Use arbitrary score matrix.

[72] W.J. Kent, R. Baertsch, A. Hinrichs, W. Miller, and D. Haussler. Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proceedings of the National Academy of Sciences of the United States of America*, 100(20):11484, 2003.

[73] J. Kim and S. Sinha. Indelign: a probabilistic framework for annotation of insertions and deletions in a multiple alignment. *Bioinformatics*, 23(3):289, 2007.

[74] J.O. Korbel, A.E. Urban, J.P. Affourtit, B. Godwin, F. Grubert, J.F. Simons, P.M. Kim, D. Palejev, N.J. Carriero, L. Du, et al. Paired-end mapping reveals extensive structural variation in the human genome. *Science*, 318(5849):420, 2007.

[75] Mary K Kuhner. Lamarc 2.0: maximum likelihood and bayesian estimation of population parameters. *Bioinformatics*, 22(6):768–770, 2006.

[76] S. Kumar and A. Filipski. Multiple sequence alignment: in pursuit of homologous DNA positions. *Genome Research*, 17(2):127, 2007.

[77] E.S. Lander, L.M. Linton, B. Birren, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[78] B. Langmead, C. Trapnell, M. Pop, and S.L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*, 10(3):R25, 2009.

[79] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357–359, 2012.

[80] 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(4):681–693, 2002.

[81] D.M. Larkin, G. Pape, R. Donthu, L. Auvil, M. Welge, and H.A. Lewin. Breakpoint regions and homologous synteny blocks in chromosomes have different evolutionary histories. *Genome research*, 19(5):770, 2009.

[82] N. Lartillot. Conjugate gibbs sampling for bayesian phylogenetic models. *Journal of computational biology*, 13(10):1701–1722, 2006.

[83] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Chem. Rev*, 93:741, 1993.

[84] S.Q. Le and R. Durbin. SNP detection and genotyping from low-coverage sequencing data on multiple diploid samples. *Genome research*, 21(6):952–960, 2011.

[85] S. Lee, F. Hormozdiari, C. Alkan, and M. Brudno. MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. *Nature methods*, 6(7):473–474, 2009.

[86] H. Leung, SM Yiu, B. Yang, Y. Peng, Y. Wang, Z. Liu, J. Chen, J. Qin, R. Li, and F.Y.L. Chin. A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio. *Bioinformatics*, 27(11):1489, 2011.

[87] B. Li, V. Ruotti, R.M. Stewart, J.A. Thomson, and C.N. Dewey. Rna-seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4):493–500, 2010.

[88] G. Li, J. Gelernter, H.R. Kranzler, and H. Zhao. M3: an improved snp calling algorithm for illumina beadarray data. *Bioinformatics*, 28(3):358–365, 2012.

[89] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[90] H. Li and N. Homer. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in bioinformatics*, 11(5):473, 2010.

[91] H. Li, J. Ruan, and R. Durbin. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome research*, 18(11):1851, 2008.

[92] R. Li, C. Yu, Y. Li, T.W. Lam, S.M. Yiu, K. Kristiansen, and J. Wang. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966, 2009.

[93] X. Li and W.H. Wong. Sampling motifs on phylogenetic trees. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9481, 2005.

[94] H. Lin, Z. Zhang, M.Q. Zhang, B. Ma, and M. Li. ZOOM! Zillions of oligos mapped. *Bioinformatics*, 24(21):2431, 2008.

[95] Y. Lin, J. Li, H. Shen, L. Zhang, C.J. Papasian, H. Deng, et al. Comparative studies of de novo assembly tools for next-generation sequencing technologies. *Bioinformatics*, 27(15):2031–2037, 2011.

[96] Y. Lin, V. Rajan, and B.M.E. Moret. Fast and accurate phylogenetic reconstruction from high-resolution whole-genome data and a novel robustness estimator. *Journal of Computational Biology*, 18(9):1131–1139, 2011.

[97] J.S. Liu, A.F. Neuwald, and C.E. Lawrence. Bayesian Models for Multiple Local Sequence Alignment and Gibbs Sampling Strategies. *Journal of the American Statistical Association*, 90(432), 1995.

[98] G. Lunter and M. Goodson. Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome research*, 21(6):936–939, 2011.

[99] G. Lunter, I. Miklós, A. Drummond, J. Jensen, and J. Hein. Bayesian coestimation of phylogeny and sequence alignment. *Bmc Bioinformatics*, 6(1):83, 2005.

[100] GA Lunter, I. Miklos, YS Song, and J. Hein. An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *Journal of Computational Biology*, 10(6):869–889, 2003.

[101] E.R. Mardis. The impact of next-generation sequencing technology on genetics. *Trends in Genetics*, 24(3):133–141, 2008.

[102] E.H. Margulies, C.W. Chen, and E.D. Green. Differences between pair-wise and multi-sequence alignment methods affect vertebrate genome comparisons. *Trends in Genetics*, 22(4):187–193, 2006.

[103] J.C. Marioni, C.E. Mason, S.M. Mane, M. Stephens, and Y. Gilad. Rna-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research*, 18(9):1509–1517, 2008.

[104] F. Matsen, R. Kodner, and E.V. Armbrust. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC bioinformatics*, 11(1):538, 2010.

[105] G. McGuire, M.C. Denham, and D.J. Balding. Models of sequence evolution for DNA sequences containing gaps. *Molecular Biology and Evolution*, 18(4):481, 2001.

[106] D. Metzler, R. Fleißner, A. Wakolbinger, and A. von Haeseler. Assessing variability by joint sampling of alignments and mutation rates. *Journal of molecular evolution*, 53(6):660–669, 2001.

[107] J.R. Miller, S. Koren, and G. Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327, 2010.

[108] Bernard ME Moret, Jijun Tang, Li-San Wang, and Tandy Warnow. Steps toward accurate reconstructions of phylogenies from gene-order data. *Journal of Computer and System Sciences*, 65(3):508–525, 2002.

[109] Bernard ME Moret, Jijun Tang, and Tandy Warnow. Reconstructing phylogenies from gene-content and gene-order data. *Mathematics of Evolution and Phylogeny*, pages 321–352, 2005.

[110] Bernard ME Moret, Li-San Wang, Tandy Warnow, and Stacia K Wyman. New approaches for reconstructing phylogenies from gene order data. *Bioinformatics*, 17(suppl 1):S165–S173, 2001.

[111] B.M.E. Moret and T. Warnow. Advances in phylogeny reconstruction from gene order and content data. *Methods in enzymology*, pages 673–700, 2005.

[112] B. Morgenstern, K. Frech, A. Dress, and T. Werner. DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics*, 14(3):290, 1998.

[113] A. Mortazavi, B.A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621–628, 2008.

[114] U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder. The transcriptional landscape of the yeast genome defined by rna sequencing. *Science*, 320(5881):1344, 2008.

[115] L.A. Newberg, W.A. Thompson, S. Conlan, T.M. Smith, L.A. McCue, and C.E. Lawrence. A phylogenetic Gibbs sampler that yields centroid solutions for cis-regulatory site prediction. *Bioinformatics*, 23(14):1718, 2007.

[116] K.C. Nixon. WinClada, version 1.00. 08. *Published by the author, Ithaca, New York*, 2002.

[117] C. Notredame, D.G. Higgins, and J. Heringa. T-coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000.

[118] F.J. Novo, I.O. de Mendíbil, and J.L. Vizmanos. Ticdb: a collection of gene-mapped translocation breakpoints in cancer. *BMC genomics*, 8(1):33, 2007.

[119] P. Nuin, Z. Wang, and E. Tillier. The accuracy of several multiple sequence alignment programs for proteins. *BMC bioinformatics*, 7(1):471, 2006.

[120] I. Ovcharenko, G.G. Loots, B.M. Giardine, M. Hou, J. Ma, R.C. Hardison, L. Stubbs, and W. Miller. Mulan: multiple-sequence local alignment and visualization for studying function and evolution. *Genome Research*, 15(1):184, 2005.

[121] G. Pandya, M. Holmes, J. Petersen, S. Pradhan, S. Karamycheva, M. Wolcott, C. Molins, M. Jones, M. Schriefer, R. Fleischmann, et al. Whole genome single nucleotide polymorphism based phylogeny of Francisella tularensis and its application to the development of a strain typing assay. *BMC microbiology*, 9(1):213, 2009.

[122] N.D. Pattengale, E.J. Gottlieb, and B.M.E. Moret. Efficiently computing the robinson-foulds metric. *Journal of Computational Biology*, 14(6):724–735, 2007.

[123] Itsik Pe'er and Ron Shamir. The median problems for breakpoints are np-complete. *Elec. Colloq. on Comput. Complexity*, 71(5), 1998.

[124] E. Pettersson, J. Lundeberg, and A. Ahmadian. Generations of sequencing technologies. *Genomics*, 93(2):105–111, 2009.

[125] L. Pireddu, S. Leo, and G. Zanetti. SEAL: a distributed short read mapping and duplicate removal tool. *Bioinformatics*, 27(15):2159–2160, 2011.

[126] M. Pop, A. Phillippy, A.L. Delcher, and S.L. Salzberg. Comparative genome assembly. *Briefings in bioinformatics*, 5(3):237, 2004.

[127] A. Prakash and M. Tompa. Measuring the accuracy of genome-size multiple alignments. *Genome Biology*, 8(6):R124, 2007.

[128] B.D. Redelings and M.A. Suchard. Joint bayesian estimation of alignment and phylogeny. *Systematic Biology*, 54(3):401–418, 2005.

[129] J.S. Reis-Filho. Next-generation sequencing. *Breast Cancer Research*, 11(Suppl 3):S12, 2009.

[130] B. Rhead, D. Karolchik, R.M. Kuhn, A.S. Hinrichs, A.S. Zweig, P.A. Fujita, M. Diekhans, K.E. Smith, K.R. Rosenbloom, B.J. Raney, et al. The ucsc genome browser database: update 2010. *Nucleic acids research*, 38(suppl 1):D613–D619, 2010.

[131] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[132] M. Ruffalo, T. LaFramboise, and M. Koyutürk. Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics*, 27(20):2790–2796, 2011.

[133] R. Sadreyev and N. Grishin. COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. *Journal of molecular biology*, 326(1):317–336, 2003.

[134] David Sankoff and Mathieu Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *Journal of Computational Biology*, 5(3):555–570, 1998.

[135] David Sankoff and Mathieu Blanchette. Probability models for genome rearrangement and linear invariants for phylogenetic inference. In *Proceedings of the third annual international conference on Computational molecular biology*, pages 302–309. ACM, 1999.

[136] David Sankoff, Robert Cedergren, and Yvon Abel. [26] genomic divergence through gene rearrangement. *Methods in enzymology*, 183:428–438, 1990.

[137] David Sankoff and Martin Goldstein. Probabilistic models of genome shuffling. *Bulletin of mathematical biology*, 51(1):117–124, 1989.

[138] David Sankoff, Guillame Leduc, Natalie Antoine, Bruno Paquin, B Franz Lang, and Robert Cedergren. Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proceedings of the National Academy of Sciences*, 89(14):6575–6579, 1992.

[139] M.P. Sawicki, G. Samara, M. Hurwitz, and E. Passaro Jr. Human genome project. *The American journal of surgery*, 165(2):258–264, 1993.

[140] A.A. Schaffer, Y.I. Wolf, C.P. Ponting, E.V. Koonin, L. Aravind, and S.F. Altschul. IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. *Bioinformatics*, 15(12):1000, 1999.

[141] M.C. Schatz. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics*, 25(11):1363, 2009.

[142] K. Schneeberger, J. Hagmann, S. Ossowski, N. Warthmann, S. Gesing, O. Kohlbacher, and D. Weigel. Simultaneous alignment of short reads against multiple genomes. *Genome Biol*, 10:R98, 2009.

[143] Z. Shu-Qi, W. Jun, Z. Li, L. Jiong-Tang, G. Xiaocheng, G. Ge, and W. Liping. Boat: Basic oligonucleotide alignment tool. *BMC Genomics*, 10, 2009.

[144] R. Siddharthan, E.D. Siggia, and E. Van Nimwegen. Phylogibbs: a gibbs sampling motif finder that incorporates phylogeny. *PLoS Computational Biology*, 1(7):e67, 2005.

[145] A. Siepel and D. Haussler. Combining phylogenetic and hidden Markov models in biosequence analysis. *Journal of Computational Biology*, 11(2-3):413–428, 2004.

[146] F. Sievers, A. Wilm, D. Dineen, T.J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, 7(1), 2011. clustal omega.

[147] K.S. Small, M. Brudno, M.M. Hill, and A. Sidow. A haplome alignment and reference sequence of the highly polymorphic Ciona savignyi genome. *Genome biology*, 8(3):R41, 2007.

[148] R.L. Small, R.C. Cronn, and J.F. Wendel. Las johnson review no. 2. use of nuclear genes for phylogeny reconstruction in plants. *Australian Systematic Botany*, 17(2):145–170, 2004.

[149] A. Smith, Z. Xuan, and M. Zhang. Using quality scores and longer reads improves accuracy of Solexa read mapping. *BMC bioinformatics*, 9(1):128, 2008.

[150] TF Smith and MS Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[151] A. Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688, 2006.

[152] J. Stoye, D. Evers, and F. Meyer. Rose: generating sequence families. *Bioinformatics*, 14(2):157, 1998.

[153] AH Sturtevant and CC Tan. The comparative genetics ofdrosophila pseudoobscura andd. melanogaster. *Journal of Genetics*, 34(3):415–432, 1937.

[154] M.A. Suchard and B.D. Redelings. Bali-phy: simultaneous bayesian inference of alignment and phylogeny. *Bioinformatics*, 22(16):2047–2048, 2006.

[155] H. Sun and J.D. Buhler. Phylat: a phylogenetic local alignment tool. *Bioinformatics*, 28(10):1336–1344, 2012.

[156] SR Swindell and TN Plasterer. SEQMAN. Contig assembly. *Methods in molecular biology (Clifton, NJ)*, 70:75, 1997.

[157] K. Tamura and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Molecular Biology and Evolution*, 10(3):512, 1993.

[158] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. De Moor, P. Rouze, and Y. Moreau. A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling. *Bioinformatics*, 17(12):1113–1122, 2001.

[159] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673, 1994.

[160] J.D. Thompson, F. Plewniak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, 27(13):2682–2690, 1999.

[161] J.L. Thorne, H. Kishino, and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *Journal of Molecular Evolution*, 33(2):114–124, 1991.

[162] A. Varón, L.S. Vinh, and W.C. Wheeler. POY version 4: phylogenetic analysis using dynamic homologies. *Cladistics*, 26(1):72–85, 2010. Compute alignment and phylogeny simultaneously.

[163] J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt, et al. The sequence of the human genome. *science*, 291(5507):1304, 2001.

[164] J.P. Vinson, D.B. Jaffe, K. O'Neill, E.K. Karlsson, N. Stange-Thomann, S. Anderson, J.P. Mesirov, N. Satoh, Y. Satou, C. Nusbaum, et al. Assembly of polymorphic genomes: algorithms and application to Ciona savignyi. *Genome research*, 15(8):1127–1135, 2005.

[165] DP Wall, HB Fraser, and AE Hirsh. Detecting putative orthologs. *Bioinformatics*, 19(13):1710–1711, 2003.

[166] Li-San Wang and Tandy Warnow. Estimating true evolutionary distances between genomes. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 637–646. ACM, 2001.

[167] L.S. Wang, T. Warnow, B.M.E. Moret, R.K. Jansen, and L.A. Raubeson. Distance-based genome rearrangement phylogeny. *Journal of molecular evolution*, 63(4):473–483, 2006.

[168] T. Wang and G.D. Stormo. Combining phylogenetic data with co-regulated genes to identify regulatory motifs. *Bioinformatics*, 19(18):2369–2380, 2003.

[169] R.L. Warren, G.G. Sutton, S.J.M. Jones, and R.A. Holt. Assembling millions of short DNA sequences using SSAKE. *Bioinformatics*, 23(4):500–501, 2007.

[170] D. Weese, A.K. Emde, T. Rausch, A. Döring, and K. Reinert. RazerS-fast read mapping with sensitivity control. *Genome Research*, 19(9):1646, 2009.

[171] O. Westesson and I. Holmes. Accurate Detection of Recombinant Breakpoints in Whole-Genome Alignments. *PLoS Computational Biology*, 5(3), 2009.

[172] T.J. Wheeler and J.D. Kececioglu. Multiple alignment by aligning alignments. *Bioinformatics*, 23(13):i559, 2007.

[173] Z. Yang. A space-time process model for the evolution of DNA sequences. *Genetics*, 139(2):993, 1995.

[174] Z. Yang, N. Goldman, and A. Friday. Comparison of models for nucleotide substitution used in maximum-likelihood phylogenetic estimation. *Molecular Biology and Evolution*, 11(2):316, 1994.

[175] G. Yona and M. Levitt. A unified sequence-structure classification of protein sequences: combining sequence and structure in a map of the protein space. In *Proceedings of the fourth annual international conference on Computational molecular biology*, page 317. ACM, 2000.

[176] D.R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome research*, 18(5):821–829, 2008.

[177] Z. Zhang, S. Schwartz, L. Wagner, and W. Miller. A greedy algorithm for aligning DNA sequences. *Journal of Computational Biology*, 7(1-2):203–214, 2000.