Efficient Similarity Measures in NGS Genome Data Comparison for Phylogeny Reconstruction

by

Phanucheep CHOTNITHI

Dissertation

submitted to the Department of Informatics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

The Graduate University for Advanced Studies, SOKENDAI

March 2020

Committee

Advisor	Dr. Atsuhiro TAKASU Professor of National Institute of Informatics/SOKENDAI
Examiner	Dr. Keizo OYAMA Professor of National Institute of Informatics/SOKENDAI
Examiner	Dr. Kento AIDA Professor of National Institute of Informatics/SOKENDAI
Examiner	Dr. Kenro AIHARA Professor of National Institute of Informatics/SOKENDAI
Examiner	Dr. Tatsuya AKUTSU Professor of Kyoto University

Acknowledgements

I would like to thank the people who helped and supported me during my Ph.D. studies.

Firstly, I would like to express my sincere gratitude to my advisor Professor Takasu Atsuhiro for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge.

I want to thank all the members and internship students in Takasu Laboratory, especially Takenaka Akiko, for providing a motivating, fun, and positive working environment. Also, thanks to the people in NII for all the support and fruitful discussions about both research and life in Japan.

Abstract

In the study of molecular evolution, the researchers need to construct a phylogenetic tree to be analyzed in many applications. The phylogeny reconstruction requires a process of comparing and measuring the distance of genome sequences. With the growth of genomic data, next-generation sequencing (NGS) has become the mainstream format for genome sequence data due to its high-throughput sequencing. The NGS creates new challenges in many applications for genome sequence analysis. In the sequence comparison applications, the traditional multiple-sequence alignment approach is not a solution for NGS data anymore because of short-read assembly and computational resource problems. Therefore, alignment-free methods are more suitable for NGS data comparisons. Most of the alignment-free methods are k-mer based algorithms. However, the characteristics of NGS data might make those k-mer based methods non-optimal since the k parameter is a crucial factor in distance measurement and phylogenetic tree results. This thesis proposes novel approaches for parameter-free comparison of NGS short reads, which aims to eliminate the dependency of k parameters.

First, we propose an alignment-free sequence comparison based on the neighbor search result on the NGS read, namely d^{NS}, to reduce the effect of the overlap problem in the NGS data. Most of the alignment-free methods rely on a k-mer based distance measure. However, with the characteristic of NGS data, k-mer based distance measure might not be the optimal solution. NGS data contain a tremendous amount of overlap among NGS read fragments at random. This affects the distance between NGS data of each input species. Instead of calculating the distance between NGS sets based-on k-mer, d^{NS} defines distance based on the neighbor search results. We compared the proposed method with two existing methods. The results show that the method can distinguish the difference between diverse species better than baseline methods. It performs decently on the whole genome of close species with better robustness on different k parameter.

Second, we propose d^{RA}, which is an effective parameter-free comparison method of NGS short reads. In order to reduce the cost of tuning the parameter k, d^{RA} focu ses on k parameter-free approach. d^{RA} also provides an improvement from d^{NS} to calculate the

distance of NGS sets of close species. To measure the distance between the NGS sets, the method consists of two steps. First, with each NGS short read in a NGS set, we search for its corresponding alignment pair in the other NGS set. Then, the detected alignment pairs are used to calculate the distance between NGS sets without k parameters. We also conduct experiments to compare d^{RA} with existing methods. The experimental results show that the proposed method can measure more accurate distances for the dataset without any parameter involved.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

Phylogeny reconstruction is the process of studying the evolutionary relationships of a group of species. Such relationships can be represented by a branching diagram or a tree called *phylogenetic tree*. Each leaf node of the tree represents a species, and its edge represents the evolutionary relationship among species. Fig.1.1 shows the big picture of the phylogenetic tree of all lives on earth. Phylogenetic trees have been used in many applications in bioinformatics such as molecular evolution analysis, forensics, and medicine development. The very first step of phylogeny analysis is to construct an accurate phylogenetic tree for the genome sequences of interest. To construct the phylogenetic tree, we need to compute pairwise distances between the genome sequences of species. After that, the phylogenetic tree can be generated using a tree construction method on the distance matrix.

However, the type of genome sequence data used in bioinformatics has changed when next-generation sequencing (NGS) was introduced. NGS is a method used to transform data from genome samples into a digitized data sequence. It achieves a rapid throughput compared with traditional sequencing processes. Instead of one long sequence of genome data, NGS produces a large number of sequence fragments called *reads* per genome sample. NGS can be applied to various biological problems, including *de novo* whole-genome sequencing and RNA-seq.

In most genome sequence analysis applications, NGS data brings new challenges, where sequence comparison and phylogeny analysis are the main issues that many researchers are interested in. Typically, sequence-comparison algorithms use one long genome sequence, such as 16S rRNA, mitochondrial DNA (mtDNA), or the whole genome when measuring the distance between sequences. Then, clustering or classification algorithms are applied to the distance matrix to construct a phylogenetic tree. However, existing methods and algorithms

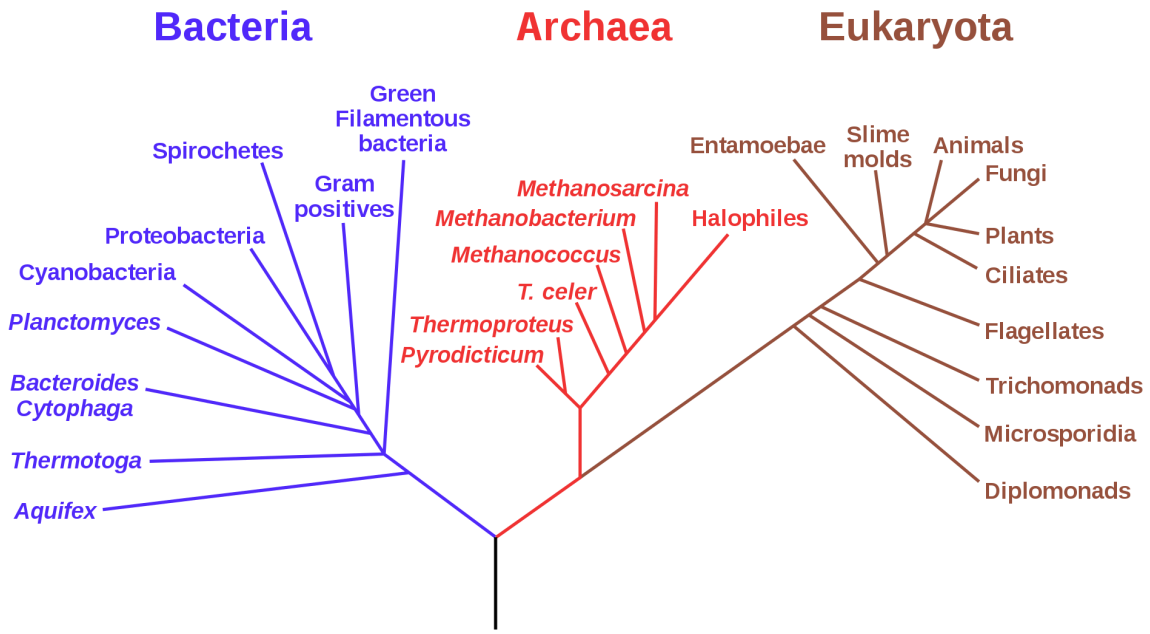


Figure 1.1: Phylogenetic tree of life

are no longer effective for this new type of genome data. The emergence of NGS short read data with their new form of genome sequences will challenge the approach of genome sequence analysis.

One of the traditional methods for sequence comparison is the multiple-sequence alignment (MSA), which is an *alignment-based* method. MSA reconstructs the short reads into one long sequence. In a process called *assemble*, NGS reads are mapped onto *template sequence*. The assembly process involves significant computational cost and has trouble in dealing with a large proportion of NGS short-read data. To assemble the genome without template sequences is very challenging because most of the reads are short and contain a large number of repeated genome data. Recently, the *alignment-free* approach for sequence comparison has attracted attention from researchers because of its processing efficiency compared with the alignment-based approach. This approach does not require an assembly process, and therefore scalable to a large number of NGS short reads. Most methods in the alignment-free approach rely on k-mer frequencies as the feature vector used to measure the distance between sequences. Since the alignment-based approach considers the difference of sequences in any position while the alignment-free approach infers the distance from the features of sequence, MSA is still more accurate than the alignment-free approach.

Many techniques have been proposed, focusing specifically on NGS short-read data. One of the most popular techniques is the k-mer based alignment-free methods, which calculate the distance between two NGS samples (or two DNA sequences) based on the

k-mer frequencies. The different values of parameter k in these methods could lead to the different results of the constructed phylogenetic trees. Since these methods rely on k-mers, they need to consider the random overlaps between NGS short reads. These overlaps affect the frequency of k-mers within NGS sets, which could lead to an inaccurate pairwise distance calculation. The random overlaps of the NGS short reads can cause differences between the k-mer frequency profiles of any two NGS sets, obtained from the same species sample. Since there is no ground-truth tree describing the natural relationships of the input species, deciding the value of k that could construct an optimal phylogenetic tree is not a trivial task. Therefore, it would be more efficient to construct the phylogenetic tree using a method that does not rely on k-mer.

1.2 Problem Definition

In this thesis, I focus on the comparison of NGS data sequences for phylogeny reconstruction. With the input as the NGS sets of the species, the phylogenetic tree of those input NGS sets can be reconstructed. This tree shows the evolutionary relationships between the input species according to their genome sequence distances. To construct an accurate phylogenetic tree, a reasonable distance measurement between the NGS sets is required.

Define $A = \{a_1, a_2, ..., a_n\}$ as a NGS set with n short reads. Each short read $a_i \in \Sigma^*$ is a genome sequence of four nucleotide characters $\Sigma = \{A, C, T, G\}$. With several NGS sets as the input, the distance matrix M contains every pairwise distance between each NGS set. Using the distance matrix M, we can reconstruct the phylogenetic tree result of the input NGS sets.

Many methods have been proposed to measure an accurate distance. Alignment-free approaches are considered as efficient methods for the task. Most of the alignment-free methods are based on the k-mer profile of the genome sequence. The k-mer profile of the genome sequence s can be defined as all possible substrings in s with length k. The parameter k is crucial for the distance measurement on these k-mer-based methods because it significantly affects the distance measurement result. Therefore, in this thesis, I address the problem of k dependency in the k-mer-based methods but still maintain the accuracy of the distance measurement.

1.3 Contribution

This thesis proposes two novel NGS data sequence comparison approaches for phylogeny reconstruction.

1.3.1 Alignment-free sequence comparison based on NGS short-reads neighbor search

First, I propose a novel assembly-free and alignment-free sequence comparison approach for NGS data called d^{NS}, based on the neighbor search. The main objective of d^{NS} is to reduce the effect of the overlaps among NGS short reads in sequence comparison. By using the neighbor search, the similar short reads that share the overlaps can be mapped into the same group. The proposed method uses the number of short reads included in the neighbor search corresponding to a set of queries to define the distance between NGS sets. d^{NS} also performs as a dimension reduction method of k-mer frequency vector for sequence comparison.

The experiment is conducted with two simulated NGS datasets. According to the experiment, d^{NS} is effective for the input sets, where each sequence is diverse from the other with high coverage. d^{NS} can construct the phylogenetic tree of a diverse species dataset with high accuracy, which indicates that d^{NS} can achieve sequence comparisons using NGS data. Also, d^{NS} is more robust because it concerns the effects of NGS short-read overlap on the k-mer frequency distribution than the other k-mer based alignment-free methods because it concerns various values of k. However, d^{NS} performs decently with closely related species datasets. Because this neighbor search-based alignment-free approach to sequence comparison is novel, there is plenty of scope for further development and possible improvements.

1.3.2 An effective parameter-free comparison of NGS short reads for phylogeny reconstruction

Second, I propose a novel sequence comparison approach that requires no k parameter adjustment while maintaining the accuracy of the result. Instead of assembling the NGS short reads and then aligning with each other to measure their distance, the novel proposed method, namely d^{RA}, is based on the alignment of NGS short reads directly. d^{RA} considers the information on the alignment of corresponding NGS short reads for the comparison of NGS data. The main idea is that if two assembled sequences are aligned with the other, there should be some alignment of their NGS short reads before assembly. By searching for the corresponding NGS short reads between each set and then calculating the distance from their alignment, this method can calculate the pairwise distance between NGS sets with no dependency on k parameter while maintaining excellent accuracy as the alignment-based approach. Since d^{RA} considers the alignment of short reads, an assembly is not required like the other alignment-free approaches. Because d^{RA} is a k-free approach, it can be applied even on NGS sets without benchmark trees, while other alignment-free approaches have difficulty

to adjust the k parameter in such NGS sets. Moreover, d^{RA} can improve the accuracy in the distance measurement by using the Gaussian mixture model.

I conducted experiments to compare d^{RA} with other alignment-free methods. The results show that the proposed method can provide an accurate distance measurement on three simulated NGS datasets to construct the phylogenetic tree compared with other alignment-free methods. The phylogenetic trees constructed from the method are more similar to the benchmark tree provided by other researchers while requiring no parameter adjustment. The experiment on the multiple simulated NGS sets from the same dataset is also conducted to evaluate the effect on different reads randomness and coverage. d^{RA} manages to calculate the accurate distance between closely related species, which is an improvement from d^{NS}.

1.4 Thesis Organization

The remaining chapters are organized as follows. Chapter 2 introduces the background of phylogeny reconstruction and NGS data. Chapter 3 surveys the past studies of alignment based and alignment-free sequence comparison. Chapter 4 describes alignment-free sequence comparison based on NGS short-reads neighbor search. Chapter 5 presents d^{RA} parameter-free comparison of NGS short reads for phylogeny reconstruction. Finally, chapter 6 summarizes the thesis.

Chapter 2

Background

2.1 Phylogeny Reconstruction

Phylogeny reconstruction [7] is the process to study the evolutionary history relationships of a group of species. The main aim of phylogeny reconstruction is to describe evolutionary relationships in terms of relative recency of common ancestry. A phylogenetic tree is a tree containing nodes that are connected by branches. Each branch represents the persistence of a genetic lineage through time, and each node represents the birth of a new lineage. If the tree represents the relationship among a group of species, then the nodes represent speciation events that cause the diversion among the group.

However, the individual leaves of the phylogenetic trees do not necessarily represent organisms. In another critical application of phylogenetic trees, we can study the evolution of genes, which can help us gain a deeper understanding of gene function. Specifically, we can learn about families of related genes.

Phylogenetic trees are not directly observed but are inferred from sequence or other data instead. Originally, phylogenetic trees are reconstructed using a variety of different algorithms. These algorithms all work by comparing a set of features of organisms and inferring the evolutionary distance between those organisms based on the similarity of their features. The features that are compared can be nearly anything observable, either from living organisms or fossilized representatives of extinct organisms. In the current era of phylogeny study, several researchers began to use genome sequences as the feature [32]. Using the genome sequences has several advantages over feature matrices derived from physical and behavioral traits, including that many more features can be observed. To construct the phylogenetic tree from genome sequences, the pairwise distance between sequences is calculated, and the resulting distance matrix is used for tree reconstruction [45].

Although there are many methods available, none of them guarantees that the constructed phylogenetic tree is, in fact, the "true" phylogenetic tree.

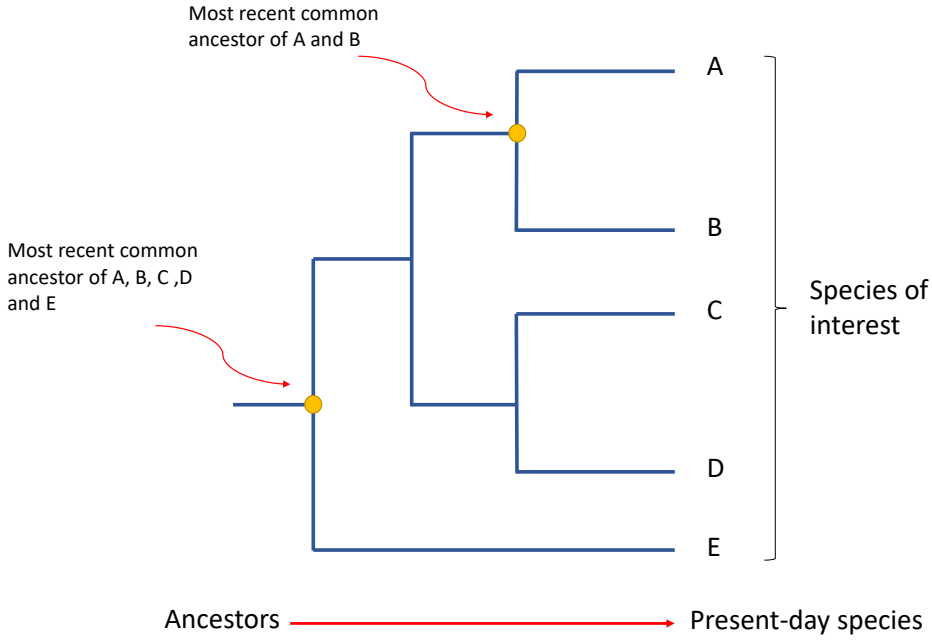


Figure 2.1: Example of phylogenetic tree

2.1.1 Distance Matrix Method

Distance calculation

A distance between a pair of objects is a measure of their dissimilarity. There is not one single definition of the distance between two objects. However, the underlying concept of a distance between two objects is the same.

Formally, a measure of dissimilarity d between two objects x and y is a distance if it meets the following four criteria for all x and y:

- $d(x,y) \geq 0$ (non-negativity)
- $d(x,y) = 0 \; if \; and \; only \; if \; x = y$ (identity of indiscernibles)
- $d(x,y) = d(y,x)$ (symmetry)
- $d(x,z) \leq d(x,y) + d(y,z)$ (triangle inequality)

When computing the distances between a number of objects n, the distance values are commonly represented in a *distance matrix* which contains all of those values. In calculating

the pairwise distance, there are two primary approaches that we could consider; *Alignment-based* and *Alignment-free* approach.

The alignment-based approach considers the order of the characters in the sequence by comparing a character at a position in one sequence only to the character at the corresponding position in the other sequence and then apply the evolutionary distance model to the hamming distance. The most traditional model is *Jukes-Cantor correction* (JC69) [23]. JC69 is typically applied to the Hamming distances between the sequences. The corrected genetic distance is computed as $d = -\frac{3}{4}\ln(1-\frac{4}{3}p)$, where p is the Hamming distance. There are also other commonly used models. The K80 [26] model assumes different rates for transitions and transversions. Both models predict equal frequencies of the four nucleotides. The assumption of equal base frequencies is relaxed in the HKY85 [19] model and the general time-reversible GTR [49] model.

On the other hand, the alignment-free approach considers the distance between sequences by the distance of the features of each sequence. The features that have been used are different depending on each method. Most of the alignment-free methods are based-on k-mer/word frequency. All substrings of length k in the sequence are counted as the feature vector. Each k-mer based method has its variation of methods to compare and calculate distance from these k-mer feature vectors. Other alignment approaches are based on the length of common substrings. While k-mer based methods consider fixed-length of substrings in the sequence, length of common substrings based methods consider variable length.

Phylogenetic reconstruction methods

After calculating the distances, we need to construct the phylogenetic tree from the distance matrix. Two well-known methods are *Unweighted Pair-Group Method with Arithmetic mean* [46] and *neighbor-joining* [1].

Unweighted Pair-Group Method with Arithmetic Mean (UPGMA): UPGMA is a generic hierarchical clustering algorithm. It is not specific to reconstructing biological trees, but rather is used for interpreting any distance matrix. It is relatively widely used for building phylogenetic trees, though its application in phylogenetics is usually restricted to building preliminary trees to "guide" the process of multiple sequence alignment.

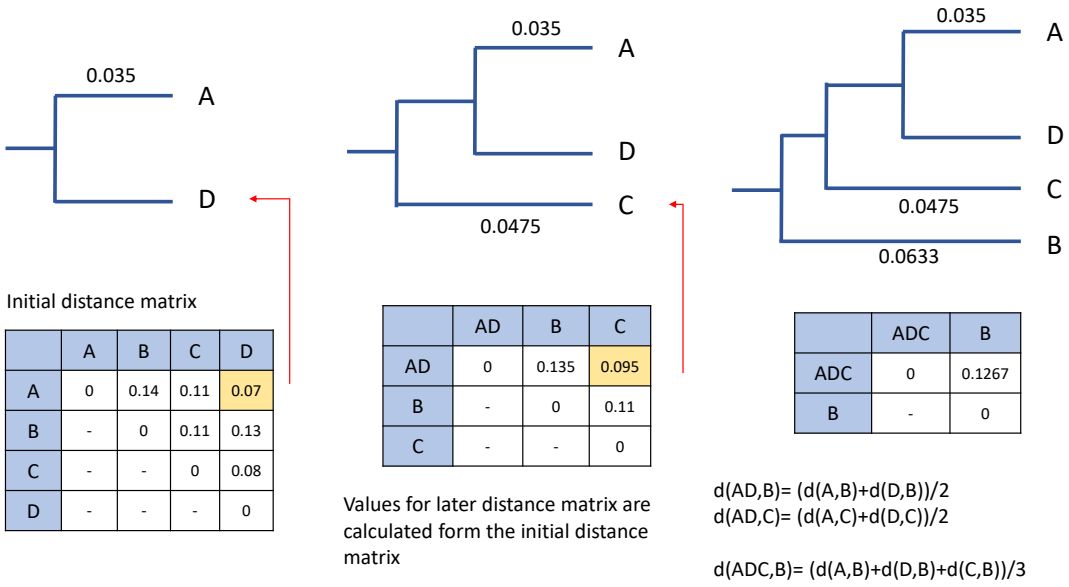
UPGMA starts with a distance matrix and works through the following steps to create a tree.

Step 1: Find the smallest non-zero distance in the matrix and define a species containing only those members. Draw that group, and set the total length of the branch connecting the leaves to the distance between the leaves. The distance between each leaf and the node connecting them should be half of the distance between the leaves.

Step 2: Create a new distance matrix with an entry representing the new group created in step 1.

Step 3: Calculate the distance matrix entries for the new group as the mean distance from each of the tip of the new group to all other leaves in the original distance matrix.

Step 4: If there is only one distance (below or above the diagonal) in the distance matrix, use it to connect the remaining unconnected groups, and stop. Otherwise, repeat step 1.



	A	B	C	D
A	0	0.14	0.11	0.07
B	-	0	0.11	0.13
C	-	-	0	0.08
D	-	-	-	0

	AD	B	C
AD	0	0.135	0.095
B	-	0	0.11
C	-	-	0

	ADC	B
ADC	0	0.1267
B	-	0

Figure 2.2: Example of the UPGMA method to construct a phylogenetic tree from the distance matrix

Neighbor-joining: The algorithm is also widely used to construct phylogenetic trees. Neighbor-joining starts with a completely unresolved tree, whose topology corresponds to the star network, and iterates over the following steps until the tree is completely resolved and all branch lengths are known as shown in Fig. 2.3:

Step 1: Calculate the matrix Q based on the input distance matrix.

$$Q(i,j) = (n-2)d(i,j) - \sum_{k=1}^{n} d(i,k) - \sum_{k=1}^{n} d(j,k)$$

where $d(i,j)$ is distance between species i and j, n is number of species

Step 2: Find the pair of distinct species i and j (i.e. with $i \neq j$) for which $Q(i,j)$has its lowest value. These species are joined to a newly created node, which is connected to the central node.

Step 3: Calculate the distance from each of the species in the pair to this new node using the following equation.

$$\delta(f,u) = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}\left[\sum_{k=1}^{n} d(f,k) - \sum_{k=1}^{n} d(g,k)\right]$$

and

$$\delta(g,u) = d(f,g) - \delta(f,u)$$

where f and g are the paired species and u is the newly created node.

Step 4: Calculate the distance from each of the species outside of this pair to the new node using the following equation.

$$d(u,k) = \frac{1}{2}\left[d(f,k) + d(g,k) - d(f,g)\right]$$

where u is the new node, k is the node which we want to calculate the distance to and f and g are the members of the pair just joined.

Step 5: Start the algorithm again, replacing the pair of joined neighbors with the new node and using the distances calculated in the previous step.

Strengths and weaknesses of distance methods

One advantage of distance methods (especially of neighbor-joining) is their computational efficiency. The clustering algorithm is fast because it does not need to compare as many trees under an optimality criterion. For this reason, neighbor-joining is useful for analyzing large data sets that have low levels of sequence divergence. Note that it might be essential to use a realistic substitution model to calculate the pairwise distances.

Distance methods perform poorly for very divergent sequences because large distances involve large sampling errors, and most distance methods (such as neighbor-joining) do not account for the high variances of large distance estimates. Distance methods are also sensitive to gaps in the sequence alignment.

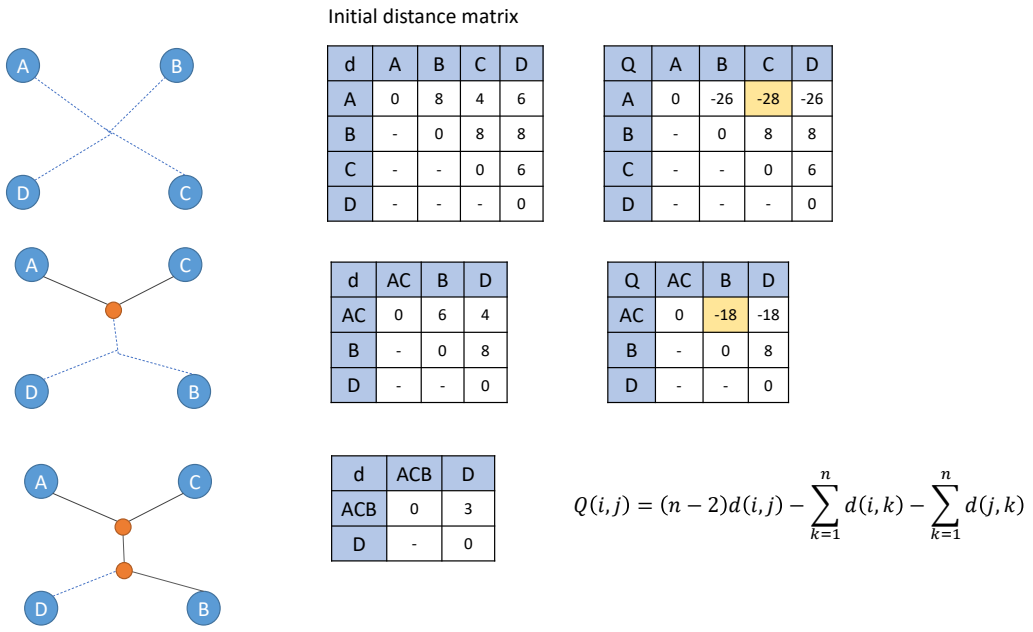


Figure 2.3: Example of the neighbor-joining method to construct a phylogenetic tree from the distance matrix

2.1.2 Other Phylogeny Reconstruction Methods

However, there are also the other phylogeny inference methods rather than distance methods which are *Maximum parsimony*, *Maximum likelihood*, and *Bayesian methods*.

Maximum parsimony method

The maximum parsimony method minimizes the number of changes on a phylogenetic tree by assigning character states to internal nodes on the tree. The character (or site) length is the minimum number of changes required for that site, whereas the tree score is the sum of character lengths overall sites. The maximum parsimony tree is the tree that minimizes the tree score. [16] and [18] developed an algorithm for finding the minimum number of changes on a binary tree (and for reconstructing the ancestral states to achieve the minimum). Parsimony has initially been developed for analyzing discrete morphological characters. Recently, it began to be applied to molecular data. The use of parsimony is still prevalent because it often produces consistent results and is computationally efficient.

A strength of the maximum parsimony method is its simplicity. Maximum parsimony is easy to describe and to understand, and it is compliant with mathematical analysis. The simplicity also helps in the development of efficient computer algorithms. However, a significant

weakness of parsimony is its lack of explicit assumptions, which makes it nearly impossible to incorporate any knowledge of the process of sequence evolution in tree reconstruction.

Maximum likelihood

Maximum likelihood uses derived states of meristic characters or quantitative characters to construct a tree based on the probabilities of character states changing on the tree. The probability of change is estimated from the data. Maximum likelihood trees are based on the probability that a particular model of character change and the observed character states would give rise to a particular tree. The tree with the highest probability, or likelihood, is the one favored. The first algorithm for maximum likelihood analysis of DNA sequence data was developed by [14]. Most models used in molecular phylogenetics assume independent evolution of sites in the sequence so that the likelihood is a product of the probabilities for different sites. The probability at any particular site is an average over the unobserved character states at the ancestral nodes. Likelihood and parsimony analyses are similar in this respect, although parsimony only uses the optimal ancestral states, whereas likelihood averages over all possible states.

One advantage of the maximum likelihood method is that all of its model assumptions are explicit so that they can be evaluated and improved. The main drawback of maximum likelihood is that the likelihood calculation and, in particular, tree search under the likelihood criterion are computationally demanding. Another drawback is that the method has potentially poor statistical properties if the model is misspecified.

Bayesian methods

The Bayesian method is similar to maximum likelihood but offers the possibility of efficiently combining different kinds of data (e.g., morphological and molecular) and offers the possibility of taking into account our confidence in relationships based on prior work. Bayesian inference relies on Bayes's theorem, which states that

$$P(T,\theta|D) = \frac{P(T,\theta)P(D|T,\theta)}{P(D)}$$

where $P(T,\theta)$ is the prior probability for tree T and parameter θ, $P(D|T,\theta)$ is the likelihood or probability of the data given the tree and parameter, and $P(T,\theta|D)$ is the posterior probability. The denominator $P(D)$ is a normalizing constant, as its role is to ensure that $P(T,\theta|D)$ sums over the trees and integrates over the parameters to one. The theorem states that the posterior is proportional to the prior times the likelihood, or the following information is the prior information plus the data information. Bayesian phylogenetic

inference relies on *Markov chain Monte Carlo* algorithms to generate a sample from the posterior distribution.

Bayesian methods can use realistic substitution models, as in maximum likelihood with prior probability allows the incorporation of information or expert knowledge. However, with the Markov chain involves, the method has heavy computation.

2.2 Application of phylogeny

Phylogeny is the evolutionary history of a characteristic, individual, population, species, or group of species. Phylogeny could be considered as the "facts" of evolution, and evolution is the central unifying principle of biology. As such, phylogeny has several primary uses within the biological sciences. Through phylogeny, we learn not only how the sequences came to be the way they are today, but also general principles that enable us to predict how they will change in the future.

Classification

Phylogeny is used extensively in biological classification. It is now the primary factor considered by taxonomists, although not the only one. Phylogenetics based on sequence data provides us with more accurate descriptions of patterns of relatedness than was available before the advent of molecular sequencing. Phylogenetics now informs the Linnaean classification of new species. In traditional Linnaean classification, non-phylogenetic groups are allowed. One glaring example is Class Reptilia, which is a paraphyletic group. If it was a true monophyletic group, meaning that it accurately reflects the phylogeny of all members and descendants, then it would include birds and mammals, both of which descended from reptiles.

Even so, Linnaean classification does give substantial weight to phylogeny when considering how organisms should be placed. Moreover, Linnaean classification is not the only option for taxonomists. A new proposed classification scheme, the PhyloCode, uses phylogeny exclusively. It does away with ranks, allowing an unlimited number of groups in an organism's classification which is useful for species. Their groups are the result of extensive radiation, therefore they have heavily branched phylogenies.

Identifying the origin of pathogens

Molecular sequencing technologies and phylogenetic approaches can be used to learn more about a new pathogen outbreak. This includes finding out about which species the pathogen

is related to each other and subsequently the likely source of transmission. This can lead to new recommendations for public health policy. For example, in the case of HIV (the virus responsible for AIDS, now the leading infectious cause of death worldwide), phylogenetic studies have revealed multiple sources of the disease in nonhuman primates and have also helped trace its transmission through human populations.

Conservation

Phylogenetics can help to inform conservation policy when conservation biologists have to make tough decisions about which species they try to prevent from becoming extinct. Conservation genetics for wildlife is an emerging challenge for humanity because it is generally accepted that the extinction of present species, even some of its populations, was caused by the massive expansion of a single species, the human (Homo sapiens). To conserve biodiversity, it is necessary not only to maximize the number of taxa that are saved today but also to guarantee the maintenance of high levels of biological diversity in the future. Hence, the phylogeny analysis is required to make an optimal solution.

2.3 Next Generation Sequencing

Deoxyribonucleic acid, commonly known as DNA, contains the blueprints of life. Within its structures are the codes required for the assembly of proteins and non-coding RNA. These molecular pieces of machinery affect all the biological systems that create and maintain life. By understanding the sequence of DNA, researchers have been able to explain the structure and function of proteins as well as RNA and have gained an understanding of the underlying causes of disease. Next Generation Sequencing (NGS) is a powerful technique that has enabled the sequencing of thousands to millions of DNA molecules simultaneously. This powerful tool is revolutionizing fields such as personalized medicine, genetic diseases, and clinical diagnostics by offering a high throughput option with the capability to sequence multiple individuals at the same time.

2.3.1 Genome sequence

The genome, carrier of this genetic information, is in most organisms deoxyribonucleic acid (DNA). In the case of some viruses, the genome contained in ribonucleic acid (RNA) instead. DNA is composed of two strands of nucleotides coiled around each other, linked together by hydrogen bonds and running in opposite directions. Each strand is composed of four complementary nucleotides – adenine (A), cytosine (C), guanine (G) and thymine (T) – with

an A on one strand always paired with T on the other, and C always paired with G. Hence the DNA sequence would be represented by the string of these four nucleotides characters.

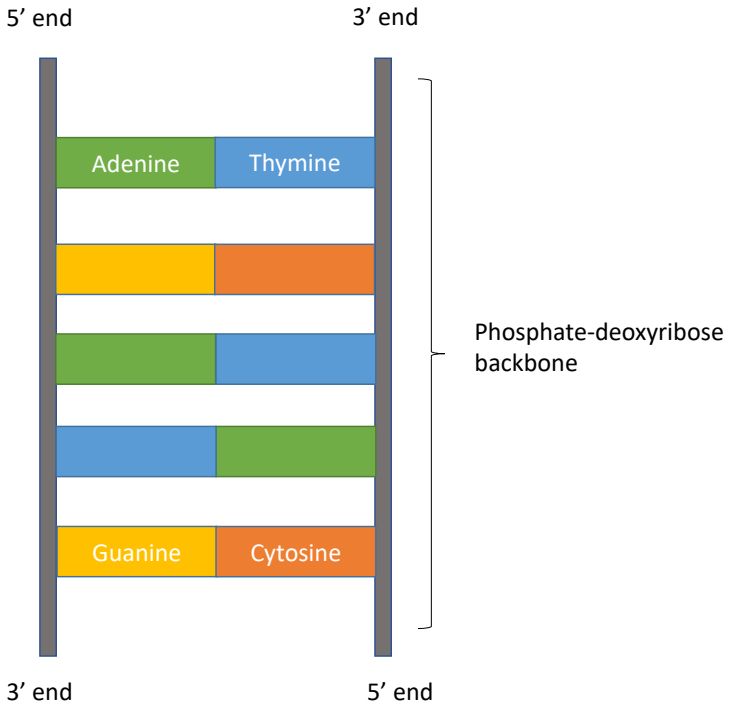


Figure 2.4: Structure of DNA

Knowledge of DNA sequences has become indispensable for basic biological research and in numerous applied fields such as medical diagnosis, biotechnology, forensic biology, virology, and biological systematics. Recently, the computer becomes an essential resource in every aspect of everyday life, including the study of the genome sequence. However, there is a process that needs to transform data from genome samples to a digitized data sequence called *Sequencing*.

2.3.2 DNA sequencing

DNA sequencing is the process of determining the sequence of nucleotide bases (A, T, C, and G) in a piece of DNA. Sequencing an entire genome (all of an organism's DNA) remains a complicated task. It requires breaking the DNA of the genome into many smaller pieces, sequencing the pieces, and assembling the sequences into a single long "consensus" sequence. Most established and well-known method for DNA sequencing is *Sanger sequencing*

Sanger sequencing method

Sanger sequencing is also known as the "chain termination method" The method was developed by two time Nobel Laureate Frederick Sanger and his colleagues in 1977, hence the name the Sanger Sequence. There are three main steps to Sanger sequencing.

Step 1: Generating *n* DNA fragments of varying lengths, each terminated with a labeled nucleotide, where *n* is the number of nucleotide bases in the target DNA sequence. This process can be done by combining the following ingredients:

- DNA primer: The starting point of DNA chain
- Nucleotides (dATP, dCTP, dGTP, dTTP)
- DNA polymerase: Ensyme used to chain nucleotides to the DNA sequence of interest
- The DNA sequence of interest
- Labeled terminate dideoxynucleotides (ddATP, ddCTP, ddGTP, ddTTP)

No nucleotide can be added to the DNA chain once a dideoxynucleotide has been incorporated so that each fragment will end with a labeled nucleotide. A much smaller amount of dideoxynucleotides is used than the number of regular nucleotides.

Step 2: Separating the *n* DNA sequences by length using *capillary gel electrophoresis*. The shorter fragments move faster than the longer fragments. The result is that the DNA pieces are fed into the third step from the shortest to the longest sequence.

Step 3: Using laser excites the label on the nucleotide at the end of each sequence. Each base is tagged with a different label, so the light emitted by each excited nucleotide can be tied to the correct base. The laser generates a chromatogram showing the fluorescent peak of each nucleotide. The chromatogram has the nucleotides in the correct order because of the electrophoresis.

Sanger sequencing gives a high-quality sequence for relatively long stretches of DNA (up to about 900 base pairs). It has typically been used in sequencing individual pieces of DNA, such as bacterial plasmids or DNA copied in PCR. However, Sanger sequencing is expensive and inefficient for larger-scale projects, such as the sequencing of an entire genome or metagenome (the "collective genome" of a microbial community). For tasks such as these, new, large-scale sequencing techniques are faster and less expensive. Hence the Sanger sequencing would be called *first generation sequencing*

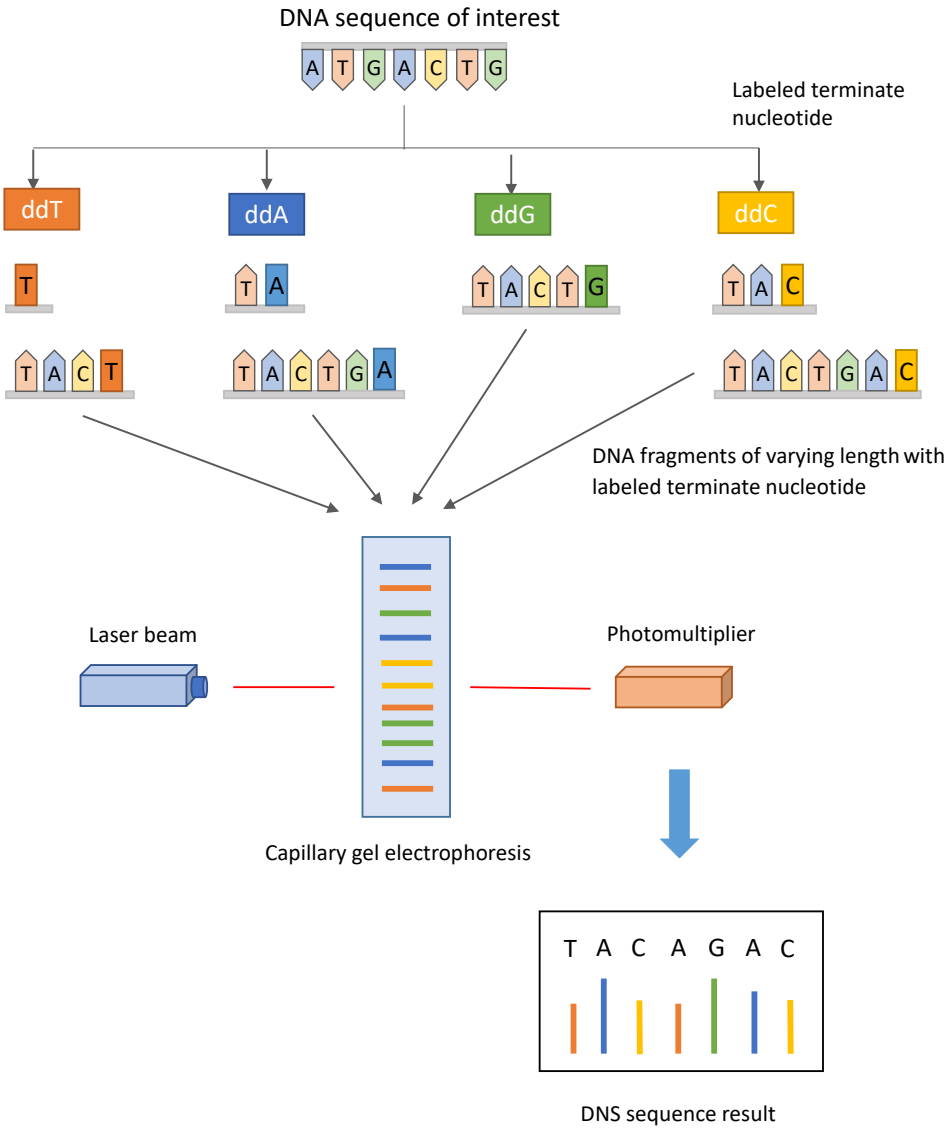


Figure 2.5: Process of sanger sequencing

Next Generation Sequencing method

Next Generation Sequencing methods have been introduced in the past decade that allows for massively parallel sequencing reactions. These systems are capable of analyzing millions or even billions of sequencing reactions at the same time. Although different machines have been developed with various technical details, they all share some following common characteristics.

- All Next Generation Sequencing platforms require a library obtained either by amplification or ligation with custom adapter sequences. These adapter sequences allow for library hybridization to the sequencing chips and provide a universal priming site for sequencing primers.

- Each library fragment is amplified on a solid surface (either beads or a flat silicon derived surface) with covalently attached DNA linkers that hybridize the library adapters. This amplification creates clusters of DNA, each originating from a single library fragment; each cluster will act as an individual sequencing reaction.

  The sequence of each cluster is optically read (either through the generation of light or fluorescent signal) from repeated cycles of nucleotide incorporation. Each machine has a unique cycling condition.

- Each machine provides the raw data at the end of the sequencing run. This raw data is a collection of DNA sequences that were generated at each cluster. This data could be further analyzed to provide more meaningful results.

There are several methods of sequencing reaction for different NGS platforms.

Illumina sequencing

In Illumina sequencing, 100-150bp reads are used. Somewhat longer fragments are ligated to universal adaptors and annealed to a slide using the adaptors. PCR is carried out to amplify each read, creating a spot with many copies of the same read. They are then separated into single strands to be sequenced.

Step 1: The slide is flooded with nucleotides and DNA polymerase. These nucleotides are fluorescently labeled, with the color corresponding to the base. The terminator also included, so that only one base is added at a time.

Step 2: An image is taken of the slide. In each read location, there will be a fluorescent signal indicating the base that has been added.

Step 3: The slide is then prepared for the next cycle. The terminators are removed, allowing the next base to be added, and the fluorescent signal is removed, preventing the signal from contaminating the next image.

Step 4: The process is repeated, adding one nucleotide at a time and imaging in between.

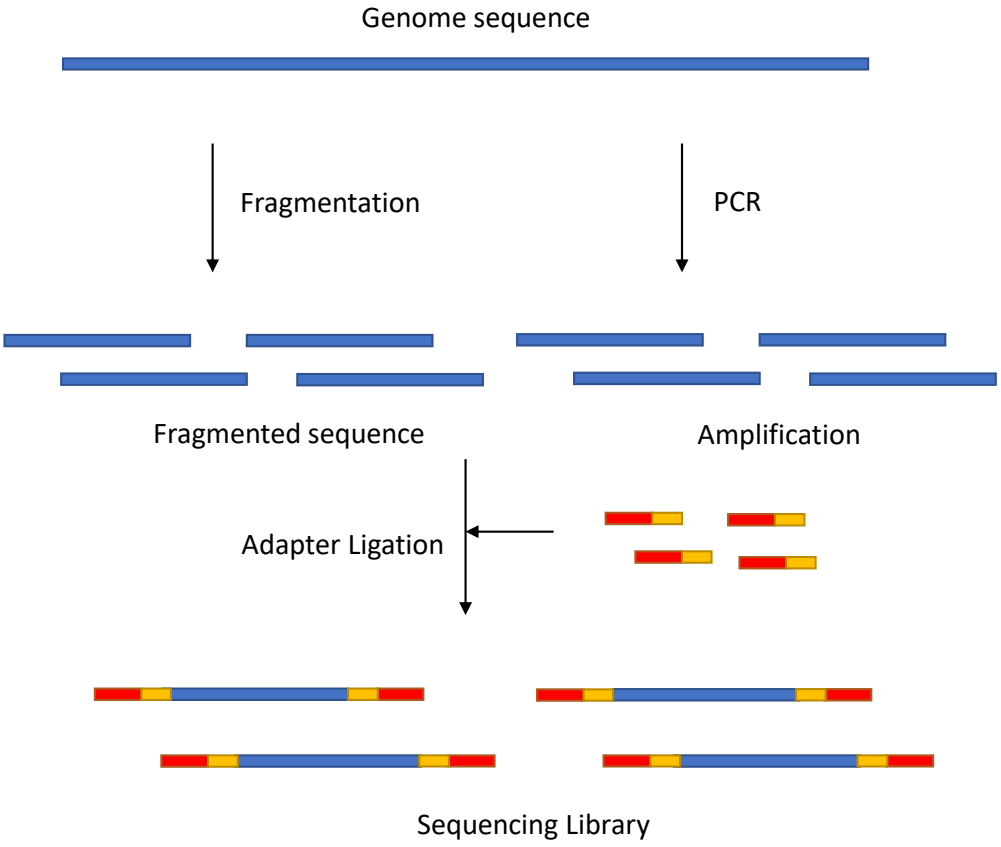


Figure 2.6: Genome sequence library consruction

454 sequencing

Roche 454 sequencing can sequence much longer reads than Illumina. Like Illumina, it sequences multiple reads at once by reading optical signals as bases are added. As in Illumina, the DNA or RNA is fragmented into shorter reads, in this case, up to 1kb. Universal adaptors are added to the ends, and these are annealed to beads, one DNA fragment per bead. The fragments are then amplified by PCR using adaptor-specific primers.

Step 1: Each bead is then placed in a single well of a slide. So each well will contain a single bead, covered in many PCR copies of a single sequence. The wells also contain DNA polymerase and sequencing buffers.

Step 2: The slide is flooded with one of the four NTP species. Where this nucleotide is next in the sequence, it is added to the sequence read. If that single base repeats, then more bases will be added. So if we flood with Guanine bases, and the next in a sequence is

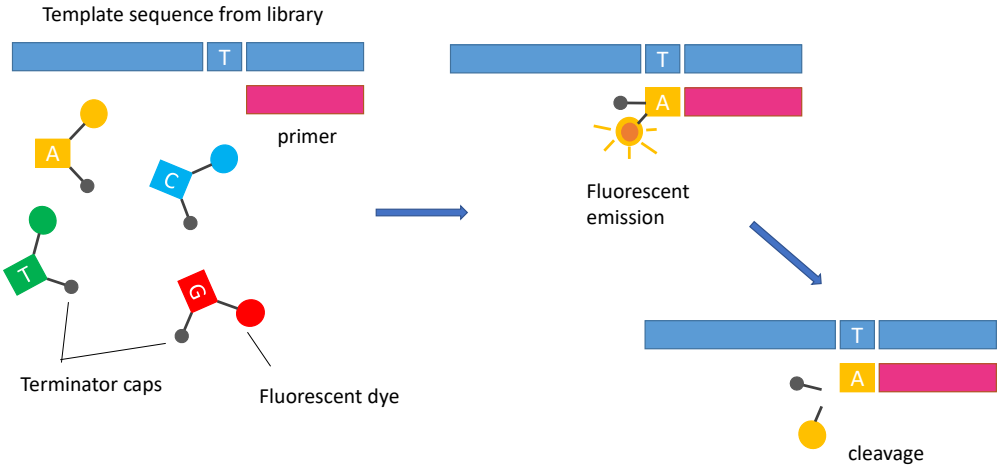


Figure 2.7: Illumina sequencing

G, then one G will be added. However, if the next part of the sequence is GGGG, then four Gs will be added.

Step 3: The addition of each nucleotide releases a light signal. These locations of signals are detected and used to determine which beads the nucleotides are added to.

Step 4: This NTP mix is washed away. The next NTP mix is now added and the process repeated, cycling through the four NTPs.

This kind of sequencing generates graphs for each sequence read, showing the signal density for each nucleotide wash. The sequence can then be determined computationally from the signal density in each wash. All of the sequences read gotten from 454 will be different lengths, because different numbers of bases will be added to each cycle.

Ion Torrent: Proton / PGM sequencing

Unlike Illumina and 454, Ion Torrent and Ion Proton sequencing do not make use of optical signals. Instead, they exploit the fact that the addition of a dNTP to a DNA polymer releases an H+ ion.

As in other kinds of NGS, the input DNA or RNA is fragmented, this time 200bp. Adaptors are added, and one molecule is placed onto a bead. The molecules are amplified on the bead by emulsion PCR. Each bead is placed into a single well of a slide.

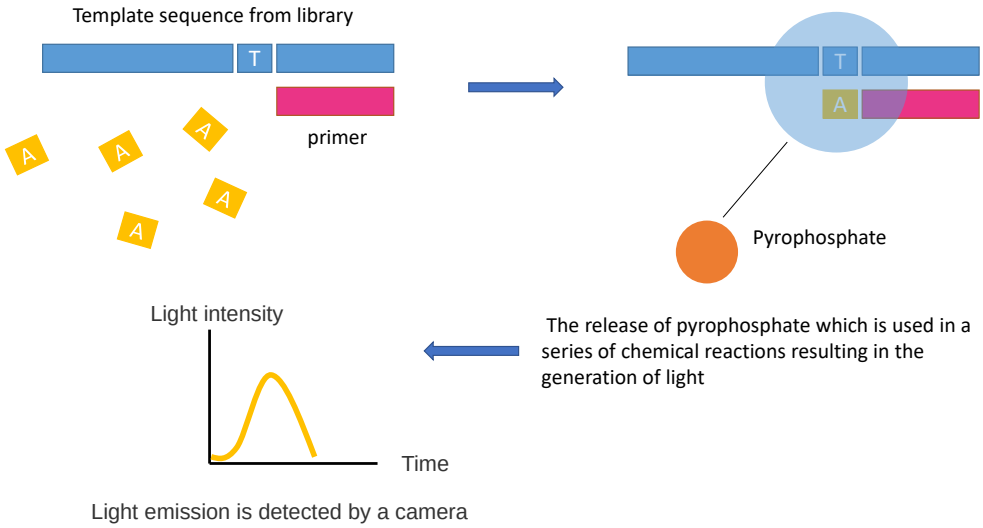


Figure 2.8: 454 sequencing

Step 1: Like 454, the slide is flooded with a single species of dNTP, along with buffers and polymerase, one NTP at a time. The pH is detected in each of the wells, as each H+ ion released will decrease the pH. The changes in pH allow us to determine if that base, and how many thereof, was added to the sequence read.

Step 2: The dNTPs are washed away, and the process is repeated cycling through the different dNTP species.

Step 3: The pH change, if any, is used to determine how many bases (if any) were added with each cycle.

Useful term

Next Generation Sequencing (NGS) is a growing field of study, with the first machine is marketed in 2005. However, in less than a decade, NGS has become a cornerstone of molecular biology and genetics. As such, being familiar with its technical terms will help in better understanding the available literature and becoming a member of its ever-expanding community. In this section, the most common terms used in this field are explained:

- **Next Generation Sequencing**: NGS is a sequencing method where millions of sequencing reactions are carried out in parallel, increasing the sequencing throughput.

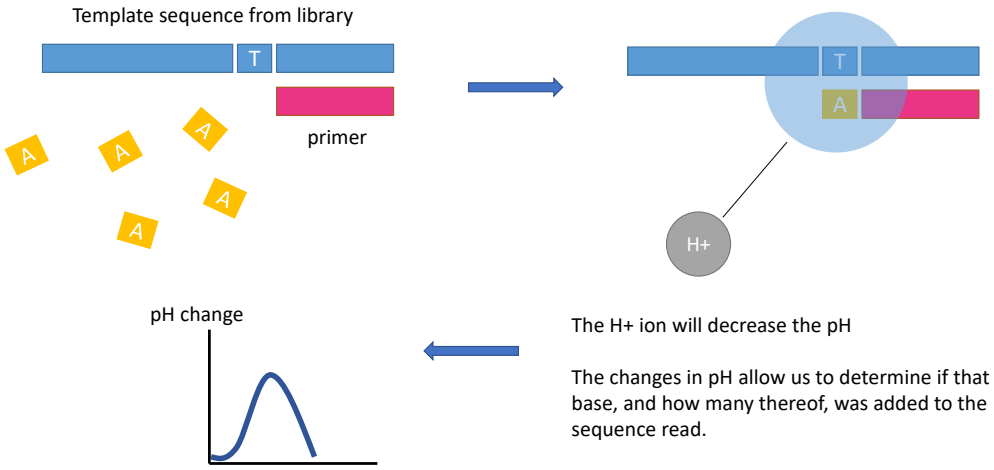


Figure 2.9: Ion Torrent: Proton / PGM sequencing

- **Reads**: The output of an NGS sequencing reaction. A read is a single uninterrupted series of nucleotides representing the sequence of the template.
- **Read Length**: The length of each sequencing read. This variable is always represented as an average read length since individual reads have varying lengths.
- **Coverage**: The number of times a particular nucleotide is sequenced. Due to the error-prone sequencing reactions, random errors could occur. Therefore, 30x coverage is typically required to ensure that each nucleotide sequence is accurate.
- **Deep Sequencing**: Sequencing where the coverage is greater than 30x. Deep sequencing is used in cases where dealing with rare polymorphisms, in which only a subset of the sample expresses the mutation. This method increases the range, complexity, sensitivity, and accuracy of the result.
- **Paired-End Sequencing**: Sequencing from both ends of a fragment while keeping track of the paired data. With this method, the sequencing reaction will commence from one end of the fragment. Once completed, the fragment is denatured, and a sequencing primer is hybridized to the reverse side adapter. The fragment is then sequenced again. Using this method will allow either further confirmation of the accuracy of the sequence, or it could be used to increase the overall read length.

- **Adapter**: Unique sequences used to cap the ends of fragmented DNA. The adapter's functions are as follows: 1) allow hybridization to the solid surface; 2) provide priming location for both amplification and sequencing primers; and 3) provide barcoding for multiplexing the different samples in the same run.
- **Library**: A collection of DNA fragments with adapters ligated to each end. Library preparation is required before a sequencing run. Our next knowledge base will delve into the different sample and library preparation methods available.
- **Reference sequence/genome**: A fully sequenced and mapped genome used for the mapping of sequence reads.
- **De Novo Assembly**: Assembly of the sequence reads to generate a reference sequence.
- **Specificity**: The percentage of sequences that map to the intended targets out of total bases per run.
- **Homopolymer**: A stretch of single nucleotide bases, such as AAAA or GGGGGG

2.4 Sequence Assembly

Sequence assembly is a process of aligning and merging fragments or reads retrieved from the NGS process to reconstruct the original sequence. The problem of sequence assembly can be compared to taking many copies of a book, passing each of them through a shredder with a different cutter, and piecing the text of the book back together just by looking at the shredded pieces which are a challenging task.

There are two type of sequence assembly, *de novo* and *mapping* assembly. De novo assemble short reads to create full-length sequences, without using a template. On the other hand, mapping assembly requires a template sequence to build a sequence that is similar but not necessarily identical to the template sequence.

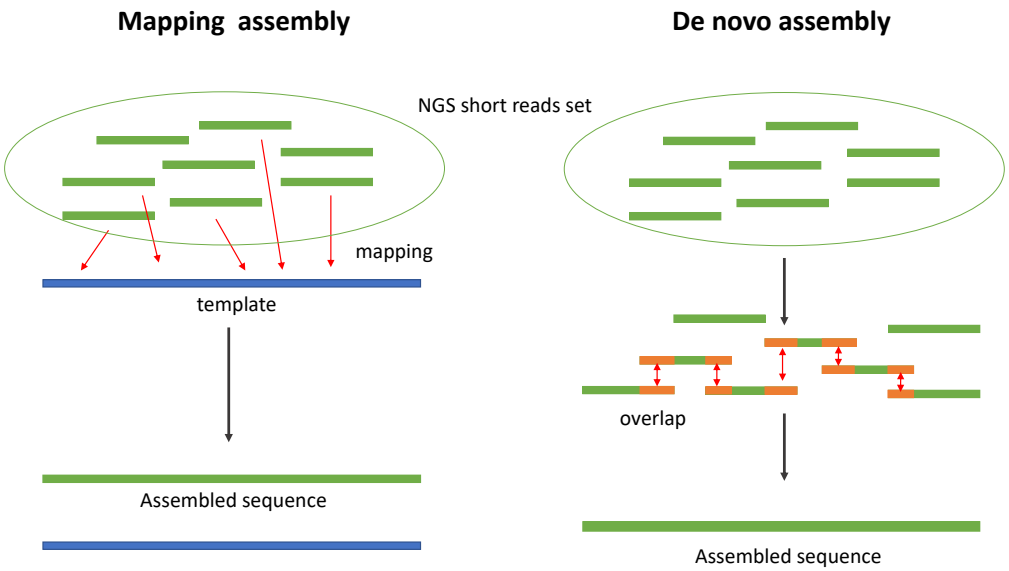


Figure 2.10: Difference between mapping and de novo assembly

Chapter 3

Related Work

3.1 Multiple Sequence Alignment

The sequence comparison is a well-known problem for genome sequence analysis. Many researchers have proposed the sequence comparison methods in past decade. The most traditional method is the alignment-based multiple sequence alignment (MSA). There are several research that propose the methods and tools for efficient MSA such as the *Clustal* series [20], [50], [44], *T-coffee* [33], *MAFFT* [25] and *MUSCLE* [10].

MSA is a sequence alignment of three or more genome sequences, generally a protein, DNA, or RNA sequences. In many cases, the input set of query sequences is assumed to have an evolutionary relationship by which they share a lineage and are descended from a common ancestor. From the resulting MSA, sequence homology can be inferred, and phylogenetic analysis can be conducted to assess the sequences shared evolutionary origins.

For the set S of m sequences $S_i, i = 1, ..., m$, MSA does the alignment on set of sequences S by inserting any amount of gaps needed into each of the S_i sequences of S until the modified sequences, $S_i^{'}$, all conform to length $L \geq max(|S_i||i = 1, ..., m)$ and no values in the sequences of S of the same column, consists of only gaps. However, solving MSA requires a huge amount of computational resources. The complexity of dynamic programming on MSA grow significantly with the number of sequences, which is $O(n^m)$ where n is the length of a sequence and m is a number of sequences. To solve the problem, several heuristic approaches of multiple sequence alignment were proposed.

3.1.1 Clustal W

The main idea of *Clustal W* is aligning the multiple sequences based-on their pairwise alignments. First, *Clustal W* performs all pairwise alignments and calculates the alignment

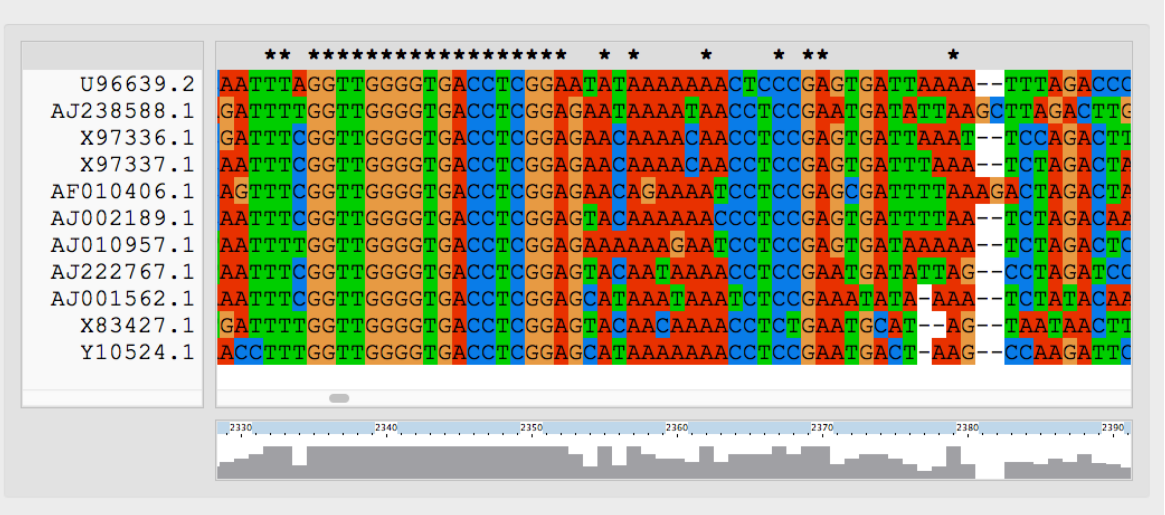


Figure 3.1: Example of multiple sequence alignment

score of each pair, which stored the alignment score matrix. Next, they use the alignment score matrix to construct the phylogenetic tree of the sequences. Finally, align the sequences in the order defined by the phylogeny tree: from the leaves towards, the root. Although *Clustal W* manages to provide good multiple sequence alignment result, there are some problems with the method. The alignment result strongly depends on the initial pairwise sequence alignment. The errors from the initial alignments also carry through the whole process.

3.1.2 T-coffee

T-coffee shares the same approach with *Clustal W*, which is initially based on pairwise alignments of the input sequences. However, *T-coffee* also considers adding the additional sequence to align with the existing pairwise alignment to find the optimal alignment between sequences. First, constructing the primary library. Each pair of sequences is aligned using ClustalW. In these alignments, each pair of aligned residues is associated with a weight equal to the average identity among matched residues within the complete alignment. Then, extend the library of each sequence pair by aligning the additional sequences to the primary library. In this step, the weights of each position on the alignment can be calculated and stored as an extended library. In the last step, do the progressive alignment using the tree but using the weights from the extended library for scoring the alignment.

3.1.3 MUSCLE

MUSCLE method is different from *Clustal W* and *T-coffee*. While both *Clustal W* and *T-coffee* start with the pair-wise alignment of sequences, *MUSCLE* is instead based-on *k*-mer frequency.

Step 1: Build quick approximate sequence similarity tree without pairwise alignment but compute distances by computing the number of *k*-mer between any pair of sequences.

Step 2: Compute the progressive multiple sequence alignment according to the phylogenetic tree constructed from *k*-mer distance matrix.

Step 3: Compute the pairwise distance from the alignment of the step. 3

Step 4: Compute the progressive multiple sequence alignment again but based on the phylogenetic tree constructed from the pairwise distance of the alignment

Step 5: Refine the alignment by iteratively partitioning the sequence into two groups and merging the aligning multiple alignments from the two groups

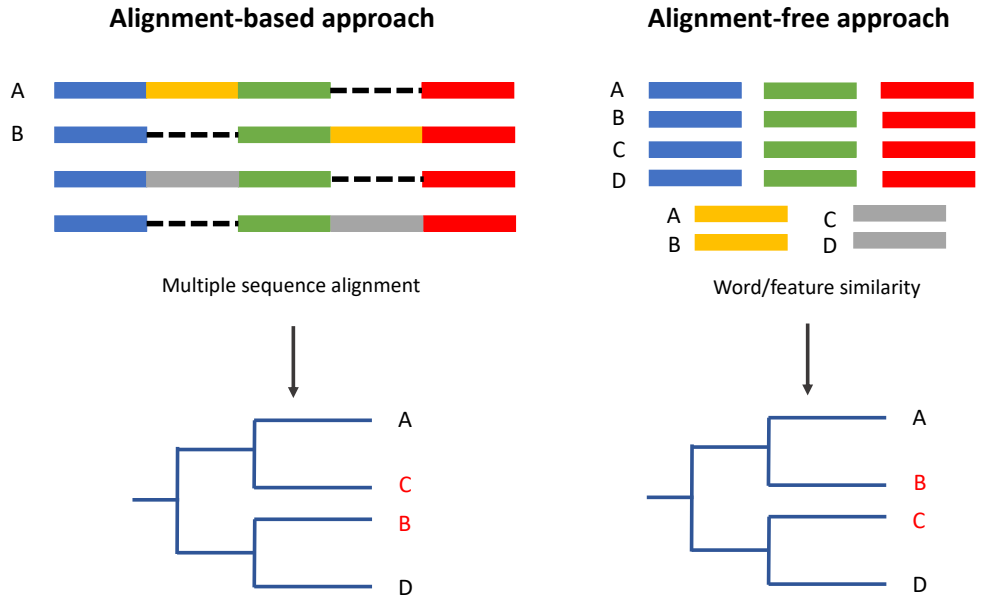


Figure 3.2: Difference between Alignment-based and Alignment-free approach

3.2 Alignment-Free Approach

Recently, with the growth of the sequencing technique, MSA is limited because of its low efficiency for the large genome comparison. The alignment-free approach has been introduced to solve the problem of MSA. Many alignment-free methods are proposed with different techniques. Like the name *Alignment-free* suggested, the approach compares or calculates the distance between genome sequences without the alignment.

Several Alignment-free methods has been proposed with different algorithm and technique. We can classify the alignment-free methods based-on their approach. First category is the methods based on k-mer frequency and occurrence. Most of alignment-free methods are in category, e.g., *FFP* [54], *MASH* [34], *spaced-word* [28], *CVTree* [37], d_2^S [47], and *skmer* [42]. Second, the method based-on length of common substrings such as, *ACS* [52] and *kmacs* [29].

3.2.1 FFP: Feature frequency profiles approach

A feature frequency profile (FFP) approach is the most straightforward method and also establish the k-mer based alignment-free approach to compare genome sequence. A sliding window of size k is run through the sequence of length n from position 1 to $n-k+1$ and counts the number of all $N = 4^k$ possible k-mers where four is the number of DNA bases and $N = 20^k$ for protein sequences. The normalized k-mer frequency vector V is constructed for the sequence by the counted k-mer. The distance between two sequences A and B is defined by the Jensen–Shannon (JS) divergence between their respective k-mer frequency vector V_A and V_B.

The Jensen–Shannon (JS) divergence is calculated by following equation:

$$JS(V_A, V_B) = \frac{1}{2}KL(V_A, V_M) + \frac{1}{2}KL(V_B, V_M)$$

where,

$$V_{Mi} = \frac{V_{Ai} + V_{Bi}}{2}$$

for $i = 1, ..., N$ and KL is the Kullback-Leibler Divergence

$$KL(V_A, V_M) = \sum_{i=1}^{N} V_{Ai} log_2 \frac{V_{Ai}}{V_{Mi}}$$

3.2.2 CVTree: CV alignment-free method

For a fixed length k, count separately the number of substrings of length k, k-1, k-2 on each input sequence. The initial CV is the number of k-mer frequency, which is $N = 4^k$ total dimensions for DNA sequences and $N = 20^k$ for protein sequences in lexicographic order. Calculate the subtraction score for the k-mer a_i:

$$a_i(\alpha_1\alpha_2\ldots\alpha_k) \equiv \frac{f(\alpha_1\alpha_2\ldots\alpha_k) - f^0(\alpha_1\alpha_2\ldots\alpha_k)}{f^0(\alpha_1\alpha_2\ldots\alpha_k)}$$

where $f(\alpha_1\alpha_2\ldots\alpha_k)$ is the frequency of k-mer $\alpha_1\alpha_2\ldots\alpha_k$ and $f^0(\alpha_1\alpha_2\ldots\alpha_k)$ is the predicted frequency of the k-mer calculated by using a (k-2)-th Markov assumption.

Let $CV_A = (a_1, a_2, \ldots, a_N)$ and $CV_B = (b_1, b_2, \ldots, b_N)$ be the CVs for the species A and B, respectively. Finally, calculate the distance matrix for the modified CV:

$$D(A,B) = (1 - C(CV_A, CV_B))/2$$

where

$$C(CV_A, CV_B) = \frac{\sum_{i=1}^{N} a_i \times b_i}{\sqrt{\sum_{i=1}^{N} a_i^2 \times \sum_{i=1}^{N} b_i^2}}$$

3.2.3 d_2^S k-mer statistical alignment-free method

d_2^S statistics is a modified version of D_2, D_2^*, and D_2^S statistics [14], [15]. They apply to NGS data by considering the random processes of NGS data in terms of D_2, D_2^*, and D_2^S to model the correct k-mer distribution of NGS data. NGS short reads are small fragments from the original long sequence, which means that the method of sampling those reads will affect the k-mer frequency distribution. Another characteristic of NGS data relevant to d_2^S statistics is that an NGS short read can originate from the forward or reverse strand of the original genome, requiring consideration of not only the k-mer distributions of short-read data themselves but also their complementary sequences.

Suppose that M reads of length β are sampled from a genome of length n. Let X_w and Y_w be the numbers of occurrences of word pattern w in the M pairs of reads from the first genome and the second genome, respectively. They define $\widetilde{X}_w^2 = X_w - M(b-k+1)(p_w + p_{\bar{w}})$ with $\widetilde{Y}_w^2$ being defined analogously. Let $w = w_1 w_2 \ldots w_k$ and $p_w = p_{w_1} p_{w_2} \ldots p_{w_k}$, with $\bar{w}$ being the complement of word w. Consider two genome sequences taking L letters $(0, 1, ..., L-1)$ at each position. For the null model, they assume that the two genomes are independent, and both are generated by models with p_l being the probability of taking state l, $l = 0, 1, ..., L-1$.

d_2^S can be calculated by:

$$d_2^S = \frac{1}{2}\left(1 - \frac{D_2^S}{\sqrt{\sum_{w\in A^k} \widetilde{X}_w^2/\widetilde{Z}_w^2}\sqrt{\sum_{w\in A^k} \widetilde{Y}_w^2/\widetilde{Z}_w^2}}\right)$$

where

$$D_2^S = \frac{\widetilde{X}_w \widetilde{Y}_w}{\widetilde{Z}_w}$$

and

$$\widetilde{Z}_w = \sqrt{\widetilde{X}_w^2 + \widetilde{Y}_w^2}$$

3.2.4 Spaced word: Fast alignment-free sequence comparison using spaced-word frequencies

While most alignment-free algorithms compare the k-mer frequency profile of sequences, Spaced Word uses a pattern of *match* and *don't care* positions. The occurrence of a spaced word in a sequence is then defined by the characters at the match positions only, while the characters at the don't care positions are ignored. Instead of comparing the frequencies of k-mer in the input sequences, this approach compares the frequencies of the spaced words according to the pre-defined pattern.

Let $w = (w', P)$ be a spaces word with weight k and length l such that $p_1 < ... < p_k$ is the position of '1' in P. w' is the sequence of length l and P is the *pattern* defined by $P = \{0,1\}^k$. The '1' in P is referred to *match* position and '0' is referred to *don't care* positions. The spaced word w is considered to occur in sequence S at position i if

$$S[i + p_j - 1] = w'[j]$$

for all $1 \leq j \leq k$

The distance $d(A,B)$ can be calculated from the frequency vector of spaced word with the corresponding pattern between A, B using the Jensen–Shannon divergence. However, This approach can be generalized by considering a set of patterns. Then, the final distance is the average of all $d(A,B)$ calculated form each pattern.

3.2.5 MASH: Fast genome and metagenome distance estimation using MinHash

MASH is a fast method that uses the MinHash bottom sketch strategy for estimating the Jaccard index of the k-mers occurrence of two input sequences. By applying the MinHash sketch, MASH can reduce large sequences and sequence sets to small, representative sketches, from which global mutation distances can be rapidly estimated. Mash estimates the ratio of k-mer matches to the total number of k-mers of the sequences. MASH can be used to estimate the evolutionary distances between the compared sequences.

3.2.6 Skmer: Assembly-free and alignment-free sample identification using genome skims

This method also based on k-mer like many alignment-free methods. Generally, Skmer is the improvement of Mash [34], where Jaccard index (J); a similarity measure between any two sets (In this case, k-mer occurrence) defined as the size of their intersection divided by the size of their union; is estimated efficiently using a hashing procedure. Then the similarity is used to estimate the genomic distance between two genomes. The problem of Mash is that its similarity is impacted by many factors such as coverage, sequencing error, and data length. Skmer aim at solving all the effect of these factors on the final similarity. There are two steps on Skmer: first step is using k-mer frequency profiles to estimate the sequencing error and the coverage. Let M_i be the number of k-mer observed i times in the genome-skim. Let $h = argmax_{i\geq 2} M_i$. By defining $\xi = \frac{M_{h+1}}{M_h}(h+1)$, k-mer coverage ($\lambda$) and the sequencing error rate (ε) can be calculated by this equation:

$$\lambda = \frac{M_1}{M_h}\frac{\xi^h}{h!}e^{-\xi} + \xi(1-e^{-\xi})$$

$$\varepsilon = 1-(\xi/\lambda)^{1/k}$$

In next step, they use the hashing technique of Mash to compute Jaccard index J and then compute the final genomic distance using the equation:

$$D = 1-\left(\frac{2(\zeta_1 L_1+\zeta_2 L_2)J}{\eta_1\eta_2(L_1+L_2)(1+J)}\right)^{1/k}$$

where for $i \in \{1,2\}$, $\eta_i = 1-e^{-\lambda_i(1-\varepsilon_i)^k}$ and $\zeta_i = \eta_i + \lambda_i(1-(1-\varepsilon_i)^k)$, and L_i is the estimated genome length.

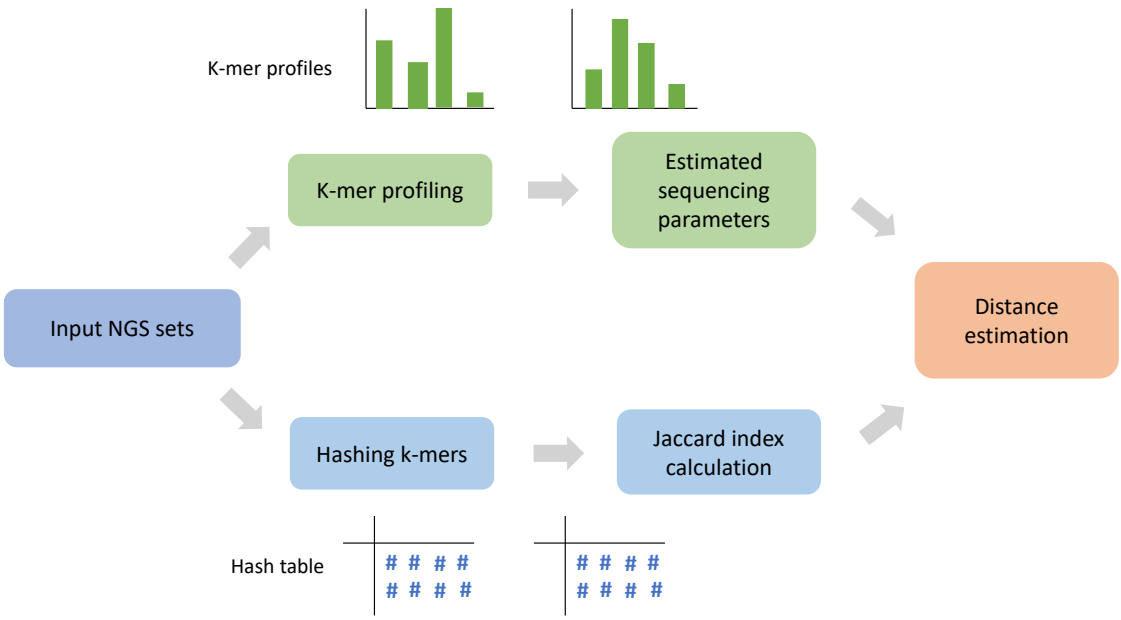


Figure 3.3: The overview of Skmer pipeline

3.2.7 ACS: Average common substring approach

The average common substring (ACS) is the alignment-free method based-on computing the average lengths of longest common substrings. They used these average lengths between the sequences to construct phylogenetic trees from an efficient algorithm.

Let $A = (a_1, ..., a_n)$ and $B = (b_1, ..., b_m)$ be sequences of lengths n and m. For any position i, $r(i)$ is the length of longest substring in A that *exact matches* with substring in B at some position j. Then the average length of every $r(i)$ as the measure $L(A,B) = r(i)/n$. The distance $d(A,B)$ is defined as follow:

$$d(A,B) = \frac{log(m)}{L(A,B)} - \frac{log(n)}{L(A,A)}$$

where $L(A,A) = n/2$

As the measure, $d(A,B)$ is not symmetric, the final distance $ACS(A,B)$ between two sequences A and B is calculated by the following.

$$ACS(A,B) = \frac{d(A,B) + d(B,A)}{2}$$

3.2.8 kmacs: k-mismatch average common substring approach

This approach is a generalization version of the ACS. They share the same approach of defining the distance between genome sequences from the lengths of longest common substrings. However, kmacs estimates for each position i of the first sequence the longest substring starting at i and matching a substring of the second sequence with up to k mismatches. It defines the average of these values as a measure of similarity between the sequences and turns this into a symmetric distance measure like ACS. Kmacs does not compute exact k-mismatch substrings, since this would be computational too costly, but approximates such substrings instead.

Chapter 4

Alignment-free Sequence Comparison based on NGS Short-reads Neighbor Search

4.1 Introduction

Sequencing is a process that transforms data from genome samples into a digitized data sequence. Nowadays, a tremendous amount of sequencing methods have been proposed. The traditional sequencing process produces a long sequence for a DNA sample. However, this sequencing process is just available for a small portion of DNA sequence per sample such as mitochondrial DNA (mtDNA) or prokaryote DNA. It cannot deal with the sequence of the whole genome due to the massive amount of DNA sequence. Recently, next-generation sequencing (NGS) [30] has been introduced to achieve high-throughput sequencing compared to traditional processes. By using a different technique to the sequencing, NGS provides a large number of sequence fragments called *reads*, per genome sample instead of one long sequence of genome data.

The data sequence retrieved from the sequencing process is used in sequence comparison and phylogeny reconstruction processes to generate a phylogenetic tree, such tree is a key for analysis in a vast amount of studies in biology fields. Typically, sequence comparison algorithms use one long genome sequence, such as 16S rRNA in mitochondrial DNA (mtDNA), to measure the distance between each sequence into a distance matrix [55, 9, 53]. Then, clustering or classification algorithms are applied to the distance matrix to construct a phylogenetic tree that shows evolutionary relationships among sequences. To construct the tree, it is essential that an accurate and efficient sequence comparison method is required.

The emergence of NGS short reads data with the new form of genome sequences brought new challenges for sequence comparison [36, 5]. The alignment-based methods such as the multiple-sequence alignment (MSA) have trouble dealing with a large proportion of NGS short reads data. Moreover, when NGS was introduced, the differences between NGS short reads data and the long sequence data need to be considered. Assembly, a procedure to reconstruct the NGS short reads into the long sequence, is required when working with NGS data. In the assembly procedure, NGS short reads are mapped onto a template sequence, which consumes significant computational cost. On the other hand, to assemble the genome without template sequences is very challenging since the reads are mostly short and contain a large number of repeated genome sequences.

Recently, the alignment-free methods for sequence comparison have attracted attention from researchers because of its processing efficiency compared with the alignment-based method. These methods have an advantage over MSA in the assembly process because they do not require an assembly process, hence they are scalable to large numbers of NGS short reads. Most alignment-free methods rely on k-mer frequencies to measure the distance [53]. Several alignment-free methods have been proposed to focus specifically on NGS short reads data. CVTree [56, 37], d_2^S [47] have shown good results for distance measurements and phylogeny reconstruction with both NGS short reads data and long genome sequences. However, these methods significantly depend on a parameter k; the different value of k could lead to the different phylogenetic tree results. Hence, researchers have trouble determining which k value would construct the best tree that is closest to the natural evolutionary relation between their input species. Moreover, alignment-free methods remain less accurate than MSA.

In this chapter, I propose a novel approach for an assembly-free and alignment-free sequence-comparison method for NGS data called d^{NS}. The main aim of d^{NS} is to reduce the effect of the overlap among NGS short reads in sequence comparisons. By grouping similar short reads, we can assume that reads sharing the same overlap are likely to fall into the same group. Using a statistical assessment of the number of short reads included in the neighbor search with a set of queries, the method provides information about the similarity between NGS sets. I performed experiments with two simulated NGS datasets. According to the results using 29 mammalian mtDNA sequences [35, 4], d^{NS} performed well when reconstructing the phylogenetic tree of a diverse-species dataset, which indicates that d^{NS} can achieve sequence comparisons using NGS data. For a 29-member $Escherichia/Shigella$ whole-genome dataset [57], d^{NS} outperformed d_2^S and matched the performance of $CVTree$. In addition, the results showed that d^{NS} is more robust with respect to various values for k than d_2^S and $CVTree$, which indicates that d^{NS} is robust against the effects of NGS

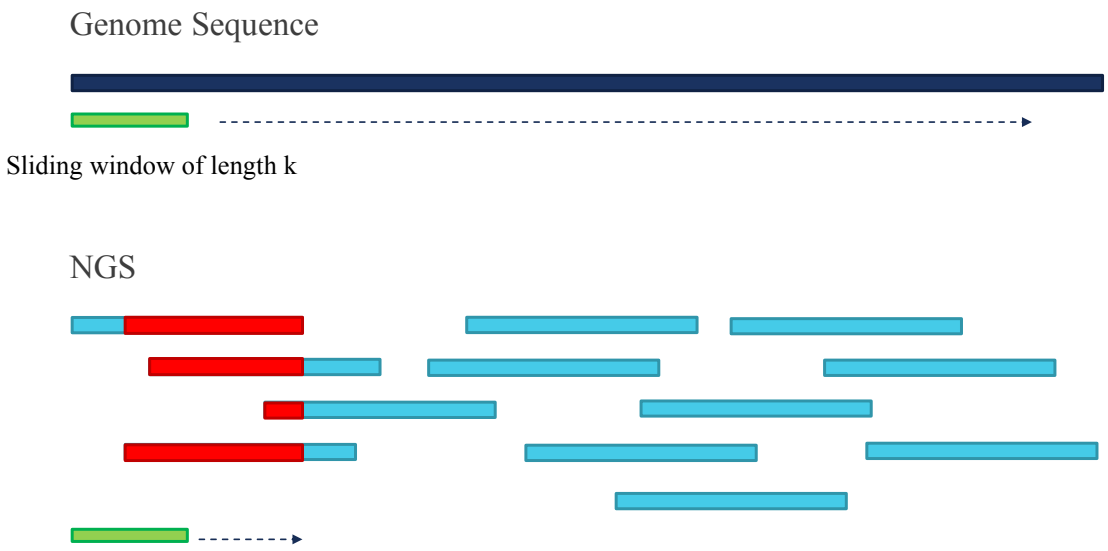


Figure 4.1: The sliding window for k-mer frequency count in genome sequence and NGS

short-read overlap on the k-mer frequency distribution. Because this neighbor-search-based alignment-free approach to sequence comparison is novel, there is plenty of scope for further development and possible improvements.

4.2 Proposed Method

The NGS data comprise a huge quantity of short reads that contain overlapping data. Particularly for whole-genome sequences, the number of overlaps and repeats can grow dramatically. Most existing research on alignment-free methods adopts k-mer frequencies to specify the profile of a sequence and when obtaining distances in NGS sets. However, the random overlap of short reads in NGS data will affect the distribution of k-mer frequencies. This is the crucial problem I focus on in this research. As shown in Fig. 4.1, the k-mer frequency profile is calculated from the sliding window of size k running through the sequence of length n from position 1 to $n-k+1$ and counts the number of all $N = 4^k$ possible k-mers. In the case of NGS data, the frequency of k-mers of each NGS read is counted one by one. The number of k-mer counted by the sliding windows are redundant because of the overlap among NGS reads shown as the red part in Fig. 4.1 if I consider counting k-mer on the first NGS read. Hence the k-mer frequency profiles used in distance calculation are not the optimal profiles, which leads to the mistake in the distance measurement.

Because the overlap and the repeating data cause the problem, the key idea is to reduce their effect by grouping similar short reads and use another feature vector instead of a k-mer

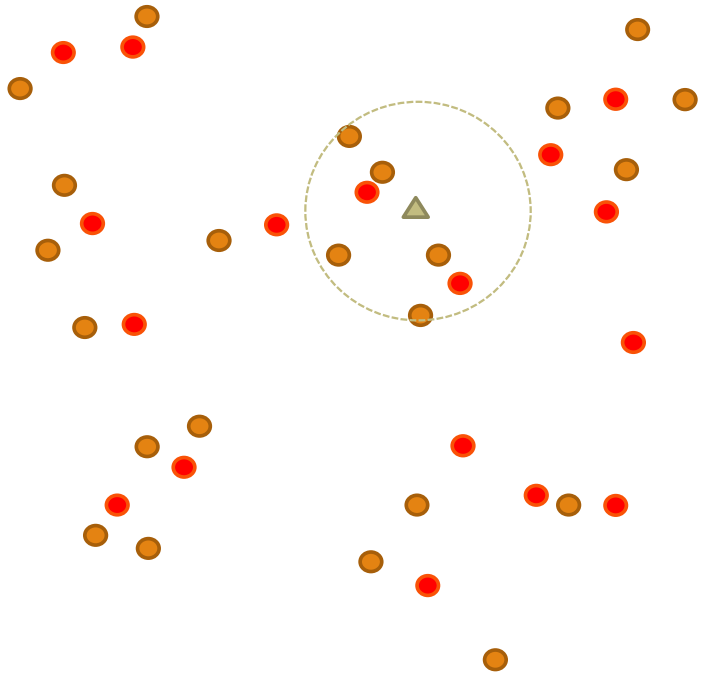


Figure 4.2: Neighbor search in NGS short reads

frequency vector. I can then use a statistical approach to calculate the evolutionary distance between NGS short reads. Fig. 4.2 shows a feature space spanned by the k-mers. (As mentioned above, the dimensionality of the space is 4^k, but, for readability, shown in a 2-D space.) Each circle represents an NGS short read. Circles of the same color indicate that the corresponding NGS short read comes from the same genome sequence. Every NGS short read is mapped to this feature space. For a given query short read r, its set of neighbors is defined as the set of short reads whose distance from r is within a predefined threshold — the circle in Fig. 4.2 encloses the neighborhood of the short read represented by the triangle.

The assumption is that the short reads that are placed near each other in the feature space will have a high probability of sharing overlapping data. I define the difference between any two NGS sets by comparing the number of neighbor-search results that correspond to the same collection of search queries on their NGS short reads. Because this method does not consider k-mer frequencies in the similarity measures of NGS sets, any overlap effects on the final distance matrix are reduced. The approach also provides the dimension reduction to the feature vector that represents each NGS sets. For the k-mer based methods, the dimension of the feature vector used in distance measurement is 4^k, for DNA sequence. With this approach, the dimension of the feature vector would be equal to the number of search queries. The method consists of 2 steps: neighbor searching and d^{NS} distance calculation [6].

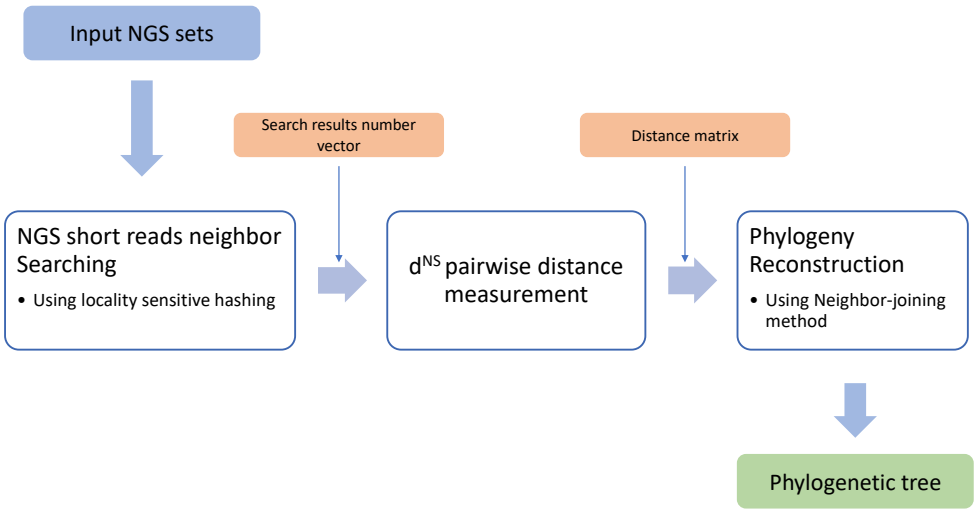


Figure 4.3: The pipeline process to construct phylogenetic tree using d^{NS}

4.2.1 Locality-sensitive hashing (LSH) for neighbor searching

I use locality-sensitive hashing (LSH) [17] for the neighbor searching step because of its lightweight nature. Minhash [3] was originally used to compare the similarity between documents. This algorithm provides a fast approximation of the Jaccard similarity between two sets by using their Minhash signatures and counts the number of components of the signatures that are equal.

Let h be the hash function for mapping an integer to another different integer, with no collisions. Apply n hash functions in $H = h_1, h_2, ..., h_n$ to the set of integers. For each h_i from $i = 1$ to n, the minimum hash value produced by h_i will be assigned to the ith component of the Minhash signature. I use this process to obtain the Minhash signature of an NGS short read. The set of k-mers that appear in an NGS short read are transformed into a set of integers to enable the hash functions to be applied. These hash functions are randomly generated with various values for the parameters that produce different hash functions. LSH is a process for finding a group of items whose Minhash signature is similar to a query's signature. It separates the Minhash signature into a series of bands, each comprising a set of rows. For example, 200 Minhash signatures might be separated into 20 bands of 4 rows each. Each band is then hashed to a *bucket*. If two sets have the same Minhash signature in a band, they will be hashed to the same bucket, and will, therefore, be considered candidate pairs. In

our approach, utilizing LSH with Minhash enables us to search for similar NGS short reads easily. The flow of this step is shown in Fig. 4.4

However, d^{NS} could adopt alternative neighbor-search algorithms because the distance measurements in d^{NS} are based on the results of the neighbor search, rather than its method.

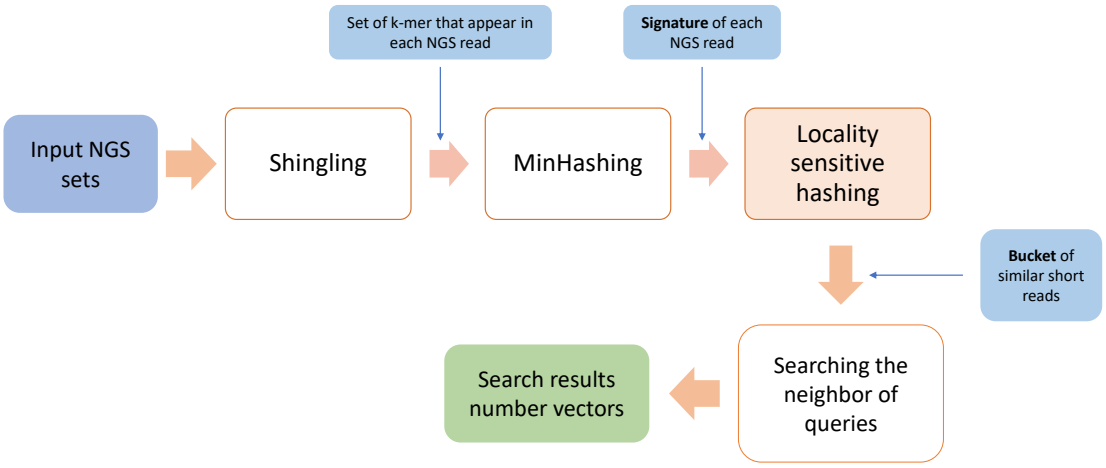


Figure 4.4: Neighbor searching process using locality-sensitive hashing

4.2.2 d^{NS} pairwise distance measurment

Denote $d^{NS}(X,Y)$ as the pairwise distance between NGS sets X and Y, where $X = \{x_1, x_2, ...x_n\}$ and n is the number of NGS short reads of X. Similarly, $Y = \{y_1, y_2, ...y_m\}$ and m is the number of NGS short reads of Y. For a query sequence q, let R_X^q denote the number of neighbors of q in X. $d^{NS}(X,Y)$ can then be calculated as follows:

$$d^{NS}(X,Y) = (D(X,Y) + D(Y,X))/2, \tag{4.2.1}$$

where

$$D(X,Y) = \sum_{i=1}^{n} \left(1 - \frac{min\left(\frac{R_X^{x_i}}{n}, \frac{R_Y^{x_i}}{m}\right)}{max\left(\frac{R_X^{x_i}}{n}, \frac{R_Y^{x_i}}{m}\right)} \right) \times \left(\frac{R_X^{x_i}}{\sum_{i=1}^{n} R_X^{x_i}} \right). \tag{4.2.2}$$

$D(X,Y)$ is a divergence measurement calculated by summation of the rational difference between the number of neighbors in NGS sets X and Y for all NGS short reads $x_1, x_2, ...x_n \in X$. The *min* to *max* ratio of two normalized values $\frac{R_X^{x_i}}{n}$ and $\frac{R_Y^{x_i}}{m}$ in Eq. (4.2.2) indicates the rational similarity between those two values. If the normalized numbers of neighbors for X and Y are the same, this term will be equal to 1. Subtracting the term from 1 makes it a divergence

measurement. For each short read in X and Y, the distance is weighted by the normalized number of the neighbors for that query. Because $D(X,Y)$ is an asymmetric function, I define the distance $d^{NS}(X,Y)$ as the average value of $D(X,Y)$ and $D(Y,X)$. According to the Eq. (4.2.2), the set of queries is the set of all NGS short reads in X for $D(X,Y)$, and Y for $D(Y,X)$. However, I can generalized the equation by considering any set of NGS short reads as the queries by following equation:

$$d^{NS}(X,Y) = \sum_{i=1}^{j} \left(1 - \frac{min\left(\frac{R_X^{q_i}}{n}, \frac{R_Y^{q_i}}{m}\right)}{max\left(\frac{R_X^{q_i}}{n}, \frac{R_Y^{q_i}}{m}\right)} \right) \times \left(\frac{R_X^{q_i} + R_Y^{q_i}}{\sum_{i=1}^{j}(R_X^{q_i} + R_Y^{q_i})} \right). \tag{4.2.3}$$

when $Q = \{q_i, q_2, .., q_j\}$ is the set of queries.

Traditionally, I can apply well-known distance and similarity measurements to the feature vector such as Euclidean distance, Cosine similarity, and Jensen–Shannon divergence. However, these traditional metrics I mentioned do not consider the different weights for each dimension. d^{NS} measures the distance by considering the normalized number of short reads result as a weight for each query (dimension) on the feature vector. Because I assume that each query should contribute to the pairwise distance differently. However, in the case of the k-mer frequency vector, each dimension is referred to individual k-mer, which all equally significant to represent the sequence.

4.3 Evaluations and Results

4.3.1 Experiment setup

Table 4.1: Size and the total sequences length of two datasets

	Size (MB)	Total sequences length
29 mammalian mtDNA	0.5	482,127
29 *Escherichia/Shigella*	144	141,962,164

Two datasets, comprising 29 mammalian mtDNA sequences [35, 4] and 29 *Escherichia/Shigella* [57] genomes shown in Table 4.1 were used to evaluate d^{NS} by comparing it with two existing k-mer-based alignment-free methods, namely *CVTree* and d_2^S. Because both datasets were originally made up of long sequences, I used a tool called *MetaSim* [39] to simulate

NGS short reads from long genome sequences. I used three error models, namely 454, Empirical(Illumina), and Sanger, which enabled us to simulate the NGS high-throughput sequencing results from three different NGS platforms. These sequenced the actual samples into NGS data. In the following discussion, the term “Exact" refers to the non-error case in simulating NGS short reads from long genomic sequences. I used sampling depths of $1\times$, $5\times$, $10\times$, and $30\times$, where the sampling depth means the average number of occurrences of the character at each position in the original sequences appearing in the NGS set. The length of NGS short reads was set to 100, with a default parameter for the error distribution for each model. For the parameter k, I considered using k values in the range 6 to 10. Although a larger k should give a better result, the processing time to map each NGS short read to the feature space would increase significantly. I planned our experiments to use this range of k values for several reasons. One reason was that *CVTree* and d_2^S proponents have suggested it as a suitable range. Second, for d^{NS}, k values out of this range would affect the efficiency of the neighbor-search process. Table 4.2 shows the size and the total number of short reads and total sequences length of simulated NGS short reads set of all two datasets.

MSA was used as the benchmark method for comparison with the alignment-free methods to evaluate their performance on phylogeny reconstruction. I used the *ClusterOmega* tool [44], followed by the *dnadist* tool in the PHYLIP package [15], on aligned sequences from MSA to calculate distance matrices.

For a distance matrix, either from MSA or from an alignment-free method, I used the *neighbor* tool in the PHYLIP package to construct a phylogenetic tree using the neighbor-joining method [41]. I used the popular Robinson–Fould distance (RF) [40] for evaluation, as described in [51]. The RF value can be calculated by counting the internal nodes that appear in one tree but not in the other. A small RF value means that the shape of the trees is close to the benchmark tree. The values for RF range from 0, meaning two trees are exactly the same, to $2(n-3)$ where n is the number of leaf nodes.

4.3.2 Experimental results

The 29 mammalian mtDNA sequences

The 29 mammalian mtDNA sequences are a well-studied dataset, being widely used for the evaluation of existing sequence comparison methods. The MSA tree for this dataset is, therefore, a reliable benchmark for our experiments. Because the evolutionary relationships between each species in this dataset are diverse, a sequence comparison method should be able to reconstruct a phylogenetic tree almost identical to that for MSA to offer confidence in the performance of the method.

I applied three alignment-free methods, namely d^{NS}, $CVTree$, and d_2^S, to simulated NGS short-read data. I compared the resultant phylogenetic trees with the benchmark tree obtained from MSA with mtDNA sequences. At a sampling depth of $1\times$, the phylogenetic trees obtained from the three alignment-free methods were very different from the MSA benchmark tree because of the shallow sampling depth. All of the methods included d^{NS}, cannot provide accurate phylogenetic trees. Since the number of short reads is too small, the information needed to calculate the accurate pairwise distance between NGS sets is not enough. As shown in Fig. 4.5, there is no phylogenetic tree that has RF distance to the benchmark tree less than 22. This means that the shape of every tree is quite different from the benchmark tree. However, the phylogeny tree which has a minimum RF distance is tree constructed by d_2^S, which shows the accuracy of d_2^S is better than $CVTree$ and d_2^S.

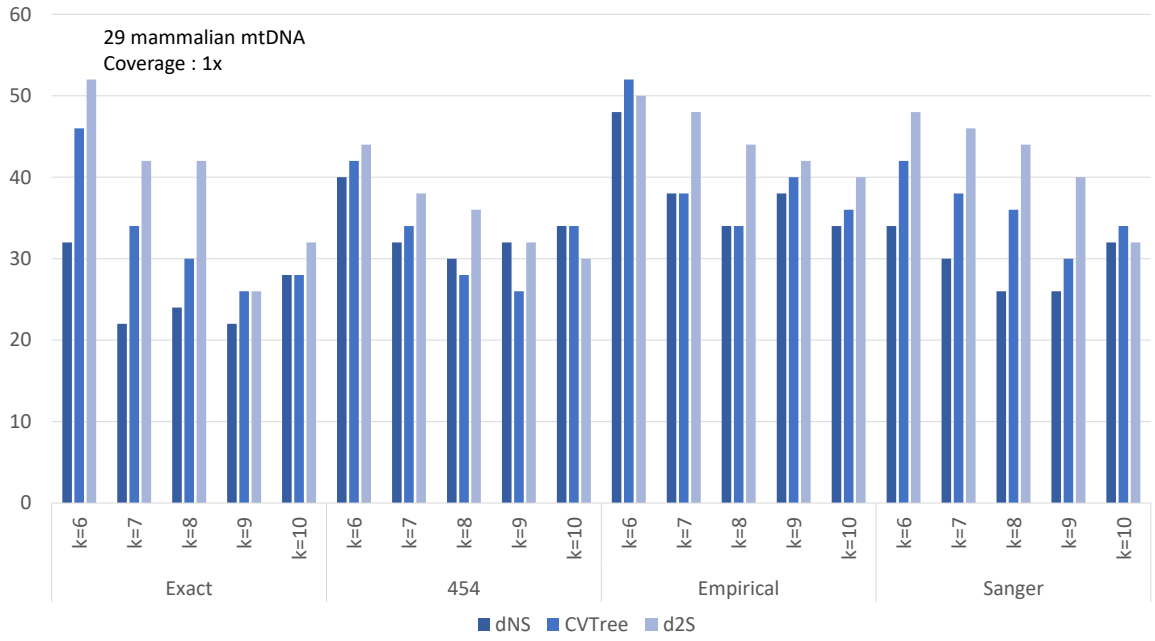


Figure 4.5: The RF of phylogenetic tree results for each method on NGS reads of 29 mammalian mtDNA sequences with a sampling depth of $1\times$

Fig. 4.6 shows the RF between the MSA benchmark tree and the phylogenetic tree obtained by d^{NS}, $CVTree$, and d_2^S on four types of NGS reads, using coverage of $5\times$ and various k parameter values. The figure shows that the overall phylogenetic tree results are closer to the benchmark than the results from GS sets with coverage 1x. Moreover, d^{NS} constructs more accurate tree than either $CVTree$ or d_2^S in most cases. Fig. 4.7 and Fig. 4.8 shows similar results of the RF between the MSA benchmark tree and the phylogenetic tree with NGS sets coverage are 10x and 30x, respectively. However, the RF distances shown in figures tend to get lower when the coverage higher. As shown in Fig. 4.8, the phylogeny

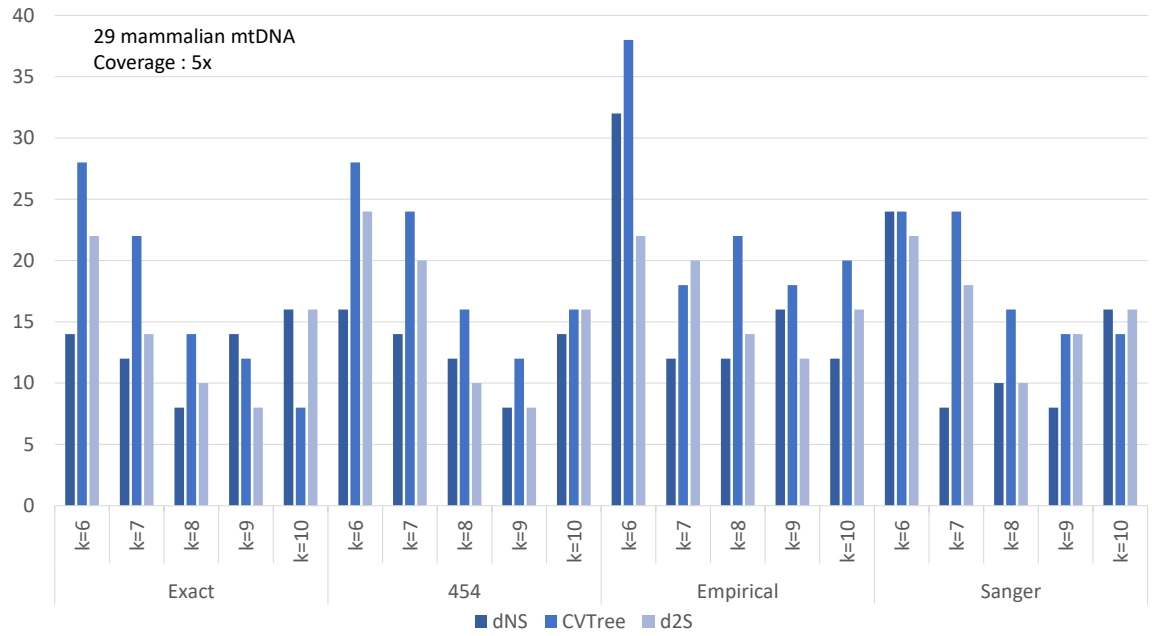


Figure 4.6: The RF of phylogenetic tree results for each method on NGS reads of 29 mammalian mtDNA sequences with a sampling depth of 5×

tree which has minimum RF distance is tree constructed by d_2^S in the Exact error model with k=8. This result means that the phylogenetic tree obtained by d^{NS} is almost the same as the benchmark with just one mistake.

Table 4.3 summarizes the most accurate result for each alignment-free method shown in Fig. 4.6. Note that RF can be up to 52 for this dataset. The best RF result in the table among all three methods is 8, which means that the rational distance between the tree obtained via alignment-free methods and the benchmark tree is 8/52 = 0.154. I can, therefore, consider that d^{NS} and the other two alignment-free methods all perform well using this dataset. d^{NS} produced the best result among the alignment-free methods across all NGS error models. Regarding the sampling depth, I found no significant differences between 10× and 30× sampling, as shown in Fig. 4.9 for d^{NS}. The same result was found for $CVTree$ and d_2^S [51]. Since most of the real NGS sets usually have low coverage. I consider discussing the result of Fig. 4.6 with the coverage 5x instead of the result from 4.7 with coverage 10x or 4.8 30x.

I investigated how parameter values affect the performance of d^{NS}. Fig. 4.9 shows the result of d^{NS} on NGS reads of 29 mammalian mtDNA sequences with a parameter setup that included four NGS error models, k values from 6 to 10, and sampling depths of 1×, 5×, 10×, and 30×. With a sampling depth of 1× for any NGS error model, d^{NS} could not produce an accurate phylogenetic tree for this dataset. The reason could be that the numbers of queries used in the neighbor search are too small to retrieve good distance measurements.

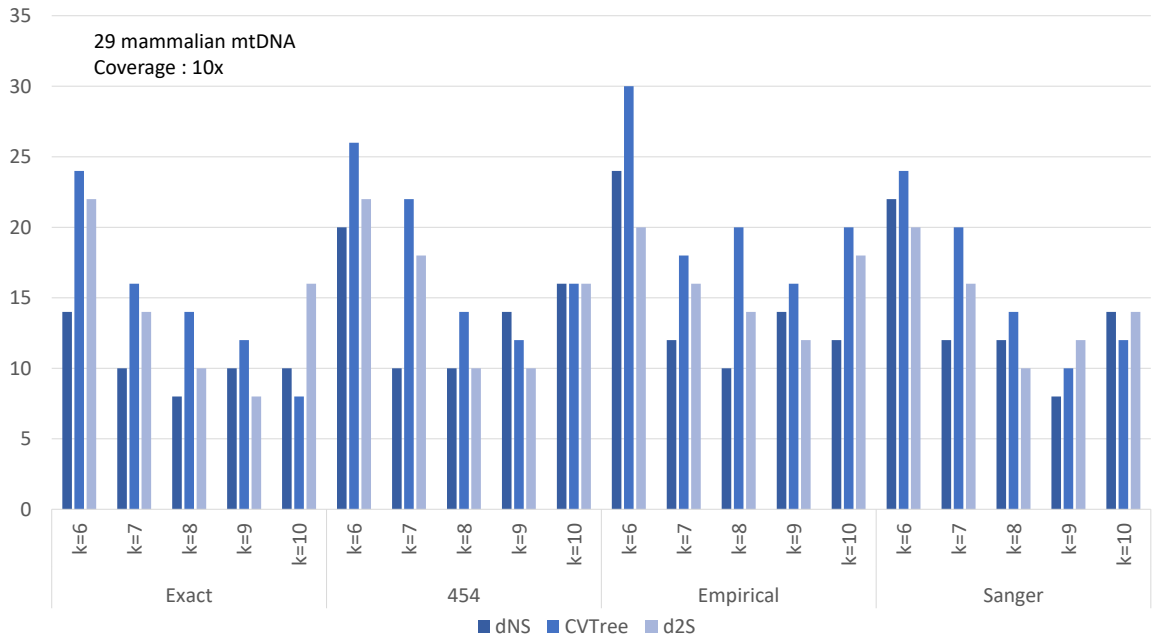


Figure 4.7: The RF of phylogenetic tree results for each method on NGS reads of 29 mammalian mtDNA sequences with a sampling depth of 10×

According to this result for d^{NS}, I can infer that a more suitable value for the k parameter would be 8 or 9.

The 29 *Escherichia/Shigella* whole-genome sequences

I used this dataset to evaluate the performance of d^{NS} on the whole genomes of species that are close to each other in evolutionary terms. The 29 whole-genome sequences come from two main genera, namely *Escherichia* and *Shigella*, which are from the same *Enterobacteriaceae* family in the *Bacteria* kingdom. Because the dataset is large, MSA's lack of scalability prevents it from being applied. I obtained the benchmark tree for this dataset from [57]. This involved concatenating the alignments of the 2034 core genes of the *Escherichia/Shigella* genomes, then using a maximum-likelihood method to construct the phylogenetic tree for this dataset.

With the close evolutionary relationship between the *Escherichia* and *Shigella* species, all alignment-free methods tested in this experiment failed to obtain an accurate RF result when comparing their resultant trees with the benchmark tree. Fig. 4.10 and Fig. 4.11 show the RF distance results for NGS reads of 29 *Escherichia/Shigella* whole-genome sequences with a sampling depth of 1x and 5x, respectively. The RF distance results are worse compared with the results of the 29 mammalian mtDNA dataset. This situation means

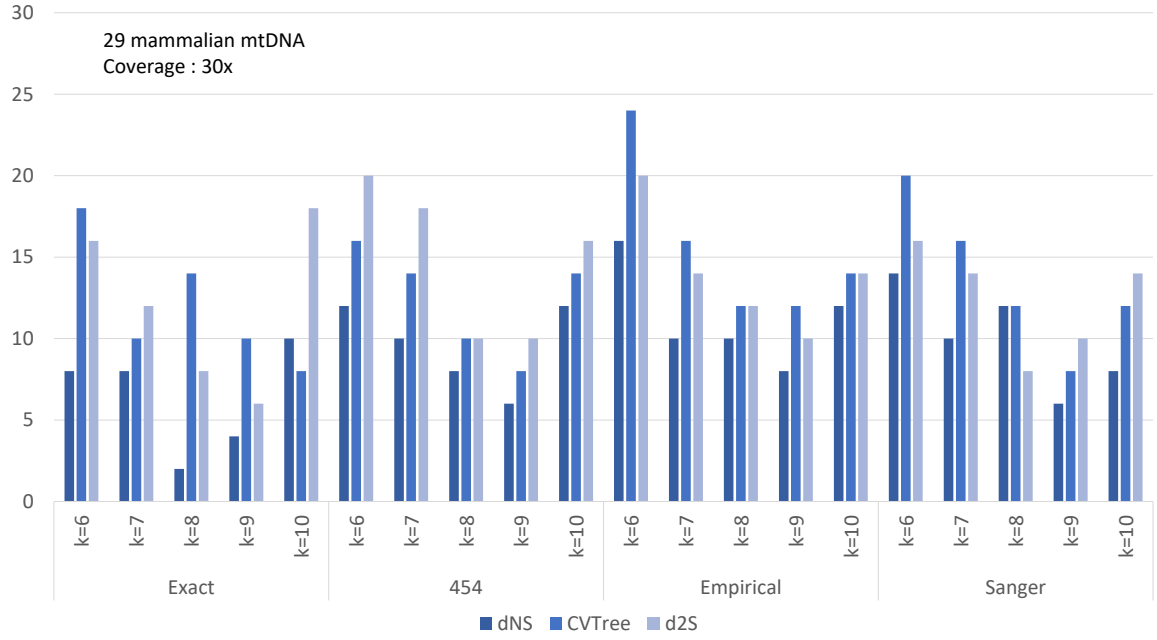


Figure 4.8: The RF of phylogenetic tree results for each method on NGS reads of 29 mammalian mtDNA sequences with a sampling depth of 30×

that d^{NS} and the other methods are not good enough to calculate the distance to construct the accurate phylogenetic tree for the closely related species. In other words, the methods cannot distinguish the small differences between sequences of closely related species. However, d^{NS} still construct better tree results most of the times depends on the k parameter.

As shown in Table 4.4, the best RF value was 16, with the rational distance between the result tree and the benchmark tree being 16/52 = 0.3. The performances of all three methods were below a satisfactory level. There was no significant difference among the d^{NS}, $CVTree$, and d_2^S methods. d_2^S performed better for the Exact error model, whereas d^{NS} and $CVTree$ performed better for the other error models.

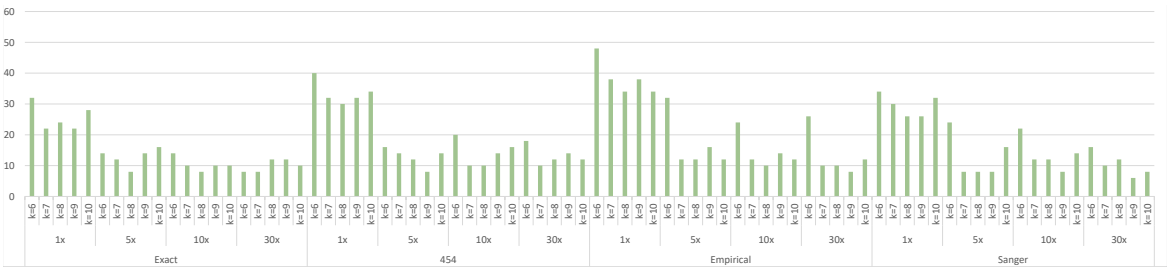


Figure 4.9: The RF of phylogenetic tree results for d^{NS} on NGS reads of 29 mammalian mtDNA sequences using four NGS error models, k = 6-10, and sampling depths of 1×, 5×, 10×, and 30×

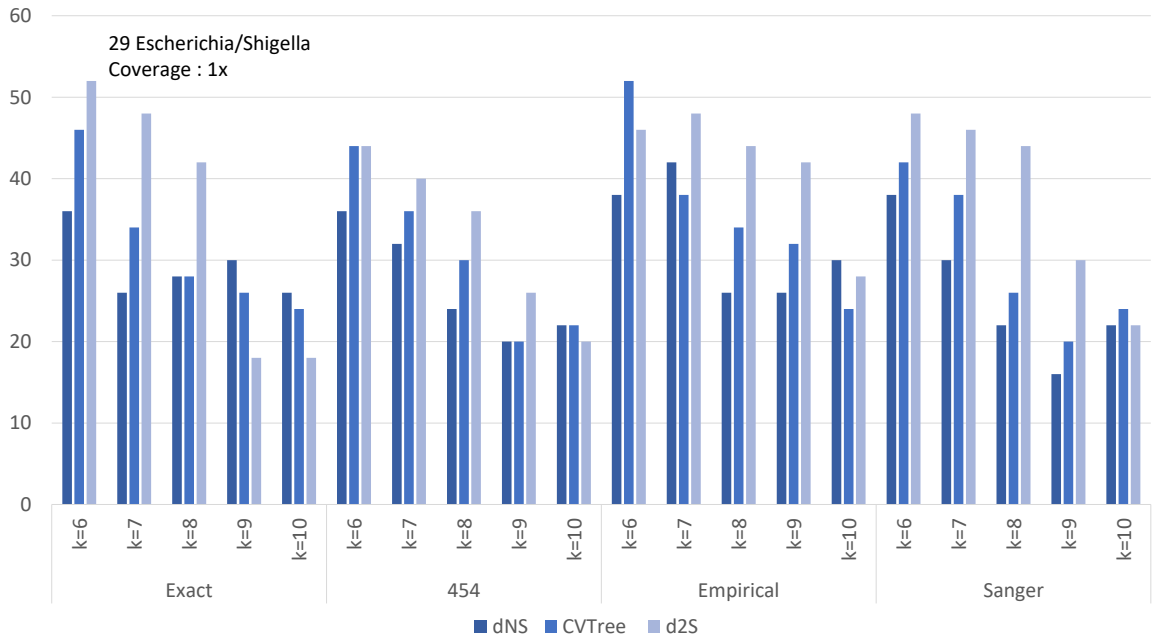


Figure 4.10: The RF results for NGS reads of 29 *Escherichia/Shigella* whole-genome sequences with a sampling depth of 1×

A point to note is that d^{NS} appears more robust with respect to variations in the k parameter than *CVTree* or d_2^S, as shown in Fig. 4.10. For most k, and for each error model, d^{NS}'s phylogenetic tree is more accurate than those of the other methods, with the RF value being at the same level. For example, although d_2^S performs best on the Exact model with RF of 18 when $k = 9$ and 10, the RF values are much bigger for other k values. Robustness against the parameter k is beneficial because it makes parameter tuning easier and I can optimize the processing efficiency by choosing a smaller value for k. The reason for this effect is that the k parameter does not directly affect how d^{NS} calculates the distance between each species. It uses the k value only for constructing the feature space. Because of limited computing resources, I examined only the case of the 1× and 5x sampling depth.

In further discussion, d^{NS} also performs as the dimension reduction methods. Instead of comparing the k-mer frequency vector with 4^k dimensions, the feature vector of d^{NS} is equal to the total number of NGS short reads. For the mammalian mtDNA dataset, the total number of NGS short reads is not higher than the k-mer frequency vector with 4^k dimensions. However, for the *Escherichia/Shigella* dataset, the total number of NGS short reads are much more significant. Hence the d^{NS} could be worst in terms of the dimension number because the dimension of d^{NS} feature vectors is much higher than the k-mer frequency vector. However, the number of features in the d^{NS} vector can be defined by defining the number of neighbor search queries.

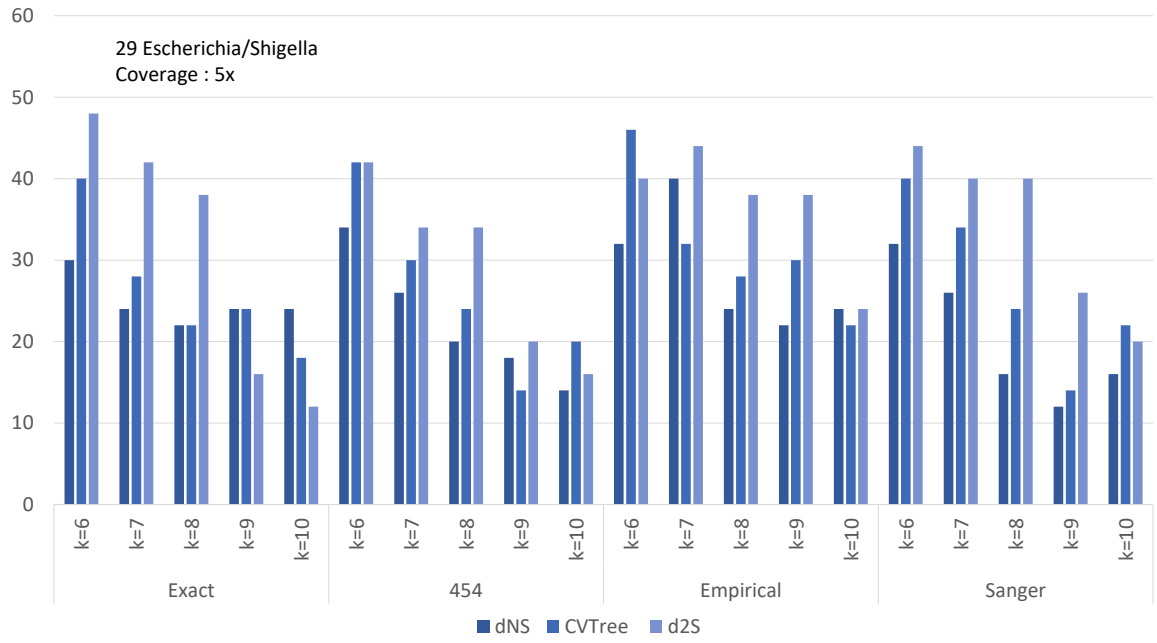


Figure 4.11: The RF results for NGS reads of 29 *Escherichia/Shigella* whole-genome sequences with a sampling depth of 5×

Fig. 4.12 shows the RF distance results for the varied number of queries with k=6 to 10. The size of query 100% means using all of NGS short reads in the dataset as queries. According to Fig. 4.12, the phylogenetic tree result slightly gets worse when the query size gets lower. Since using just 1% of the whole short reads as queries on this dataset still provides a similar result with using 100%. d^{NS} can surely provide dimension reduction to the large dataset to calculate the distance. With the typical k-mer profile, although we could obtain very high dimension data to represented each NGS sets, they are very sparse and few of them are relevant to actual evolutionary distance. The reason behind the good efficiency of d^{NS} even using a small amount of NGS short reads as the queries is the concept of quality information over the quantity. By using the process of dimension reduction is able to retrieve the necessary information to calculate the distance between NGS sets. Because of using the number of the neighbor of the query as the information for distance measurement, up to a certain number of queries, the overall information we retrieved is not increasing due to the shared neighbor among them.

The main aim of this research is to introduce a novel approach to performing NGS data comparisons. It is expected that the computational efficiency of *CVTree* and d_2^S would exceed that of d^{NS} in its current implementation. All of the experiments are conducted by using Intel Core i7-4980HQ 2.8 GHz processor which includes four independent processors with 16 GB DDR3L SDRAM. Table 4.5 confirms that d^{NS}'s runtime is slower than the others. However,

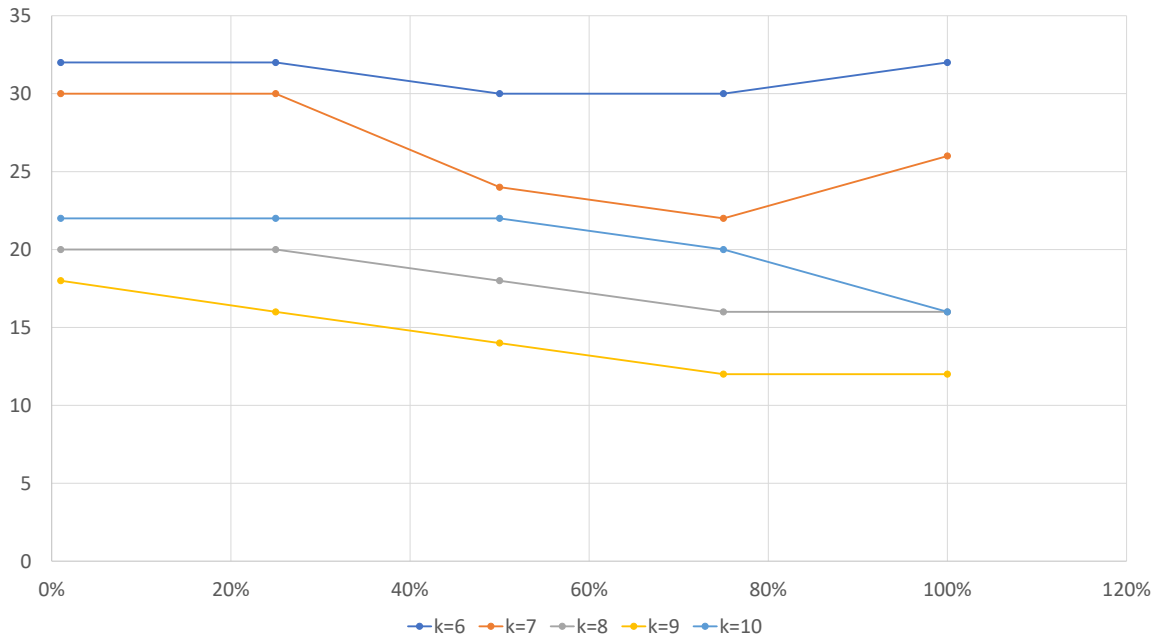


Figure 4.12: The RF results for NGS reads of 29 *Escherichia/Shigella* whole-genome with different query size sequences with a sampling depth of 1×

the k parameter value does not affect the runtime of d^{NS}, unlike those for *CVTree* and d_2^S. In particular, d_2^S's runtime grows dramatically between $k = 6$ and $k = 10$. It is an advantage that the k value has little effect on the runtime of our proposed method. In addition, the runtime of d^{NS} shows linear growth with varying sampling depth, as shown in Fig. 4.13

According to pseudo-code shown in Algorithm 1 the complexity of the method is $O(nl + qml)$ where n is the total number of the short reads, q is a number of queries, m is an upper-bound number of neighbor result, and l is the length of short reads. At the first step of neighbor searching, it requires $O(nl + qml)$ to construct the MinHash signature and buckets for every n short read with length l and searching for every q query neighbor in the bucket. The last step is the pairwise distance calculation, which uses $O(q)$ times. The functions called in Algorithm 1 can be done by linear time.

4.4 Conclusion

In conclusion, I propose a novel approach for an alignment-free method d^{NS} that is focused on NGS short-read data. It is based on neighbor searching. Its main advantage is that it is an accurate alignment-free sequence-comparison method for reconstructing a phylogenetic tree more consistently than other k-mer-based alignment-free methods. Although it might lose

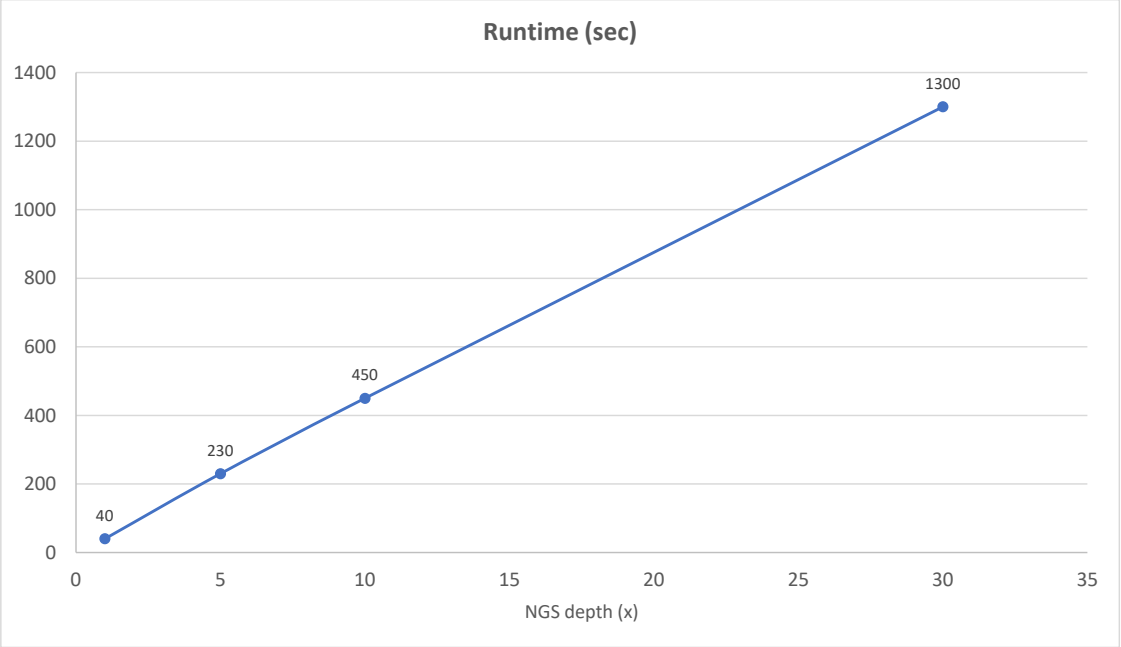


Figure 4.13: Computational runtime (seconds) for d^{NS} on the 29 mammalian mtDNA dataset with NGS sampling depths of $1\times$, $5\times$,$10\times$, and $30\times$

significant information in the NGS data when ignoring the k-mer frequencies, the method can specify the distance between NGS sets with reasonable accuracy when a sufficient number of queries are used.

Algorithm 1: Algorithm for d^{NS} to calculate pair-wise distance

Input : Set of NGS short reads of m species $S = \{R_1, \ldots, R_m\}$ where $R_i = \{r_{i,1}, \ldots, r_{i,o}\}$, the total number of reads is $n = m * o$

Output : A Distance matrix denoted by D with $m \times m$ dimensions;

b is bucket structure for LSH;

foreach $R_i \in S$ **do**

  foreach $r_{i,j} \in R_i$ **do**

    kmers = CountKmerProfile($r_{i,j}$);

    key = MinHashLSH(kmers);

    b[key].push($r_{i,j}$)

  end

end

$Q \subset$ all NGS short reads as query;

foreach $q_i \in Q$ **do**

  neighbor=searchNeighbor(q,b);

  D=updateDistance(*neighbor*,D);

end

Table 4.2: Size and the total number of short reads and total sequences length of NGS short reads set of all two datasets

	Error model	Coverage	Size (MB)	Total number of short reads	Total sequences length
29 mammalian mtDNA	Exact	1x	1.6	5,800	580,000
		5x	8	29,000	2,900,000
		10x	16	58,000	5,800,000
		30x	47.5	174,000	17,400,000
	454	1x	1.7	5,800	583,387
		5x	8.5	29,000	2,916,864
		10x	17	58,000	5,832,977
		30x	50	174,000	17,500,757
	Empirical	1x	1.7	5,800	608,800
		5x	8	29,000	2,944,000
		10x	17	58,000	6,088,000
		30x	46	174,000	17,264,000
	Sanger	1x	1.6	5,800	580,032
		5x	8	29,000	2,900,063
		10x	16	58,000	5,799,621
		30x	47.5	174,000	17,399,840
29 *Escherichia/Shigella*	Exact	1x	420	1,500,000	150,000,000
		5x	2100	7,500,000	750,000,000
	454	1x	445	1,500,000	148,676,934
		5x	2225	7,500,000	743,241,873
	Empirical	1x	450	1,500,000	154,000,000
		5x	2250	7,500,000	770,000,000
	Sanger	1x	430	1,500,000	150,000,089
		5x	2150	7,500,000	750,000,241

Table 4.3: Best RF Result for any K parameter on NGS short read of 29 mammalian mtDNA sequence with sampling depth of 5x

	d^{NS}	$CVTree$	d_2^S
Exact	**8**	**8**	**8**
454	**8**	12	**8**
Empirical	**12**	18	**12**
Sanger	**8**	14	10

Table 4.4: Best RF Result for any K parameter on NGS short read of 29 *Escherichia*/*Shigella* whole-genome sequence with sampling depth of 1x

	d^{NS}	$CVTree$	d_2^S
Exact	26	26	**18**
454	**20**	**20**	**20**
Empirical	26	**24**	28
Sanger	**16**	20	22

Table 4.5: Computational runtime for each alignment-free method (second)

	d^{NS}	$d_2^S(k=6)$	$d_2^S(k=10)$	$CVTree(k=6)$	$CVTree(k=10)$
29 mammalian mtDNA	230	4	780	2	5
29 *Escherichia*/*Shigella* whole genome	8600	30	1050	25	180

Chapter 5

An effective parameter-free comparison of NGS short reads for phylogeny reconstruction

5.1 Introduction

In this chapter, I propose a novel sequence comparison approach that requires no k parameter adjustment while maintaining the accuracy of the result. Although the d^{NS} in Chapter 4 provides a reasonable distance measurement, it is not accurate for the input set of closely related species. The goal of this research is to develop a novel sequence comparison approach that requires no k parameter adjustment while maintaining the accuracy of the result. I utilize the information on short reads alignment for comparison of NGS data. Instead of assembling the NGS short reads then aligning the result with the other sequences to measure their distance, I propose a new method, namely d^{RA}, that is based on the alignment of NGS short reads themselves. The main idea is that if the sequence ends up aligned with others after assembly, their NGS short reads before assembly should also be aligned. By searching for the corresponding NGS short reads between each set and then calculating the distance from their alignment, this method allows the distance to be calculated with no dependency on the k parameter and to maintain the same accuracy as the alignment-based approach. The method also has no requirement for assembly, like the alignment-free approach.

I have compared the method with alignment-free methods and found that my novel read alignment approach can provide a more accurate distance measurement on three simulated NGS datasets to construct the phylogenetic tree than other alignment-free methods. The phylogenetic trees constructed using the new method are similar to the benchmark tree

obtained by other researchers while requiring no parameter adjustment. I also conducted experiments on multiple simulated NGS sets from the same dataset to evaluate the effect on different reads' randomness and coverage. The approach delivers similar measured distances among each set.

In summary, this chapter makes the following contributions:

- I propose a novel sequence comparison approach, namely d^{RA}, which requires no k parameter while maintaining the accuracy of the result. Because d^{RA} is a k-free approach, it can be applied even on NGS sets without benchmark trees, whereas it is difficult to adjust the k parameter for other alignment-free approaches in such NGS sets.

- I utilize the Gaussian mixture model to improve the accuracy of the distance measurement of the approach.

- I conducted experiments on three real datasets to measure the accuracy of the proposed approach in comparison with other alignment-free approaches. Along with the accuracy, I also measured the consistency of pairwise distance computation to evaluate better the effectiveness of the proposed method. The experimental results indicate that d^{RA} provides higher accuracy while maintaining consistency compared with other baseline methods.

- I also conducted experiments to evaluate the efficiency of the proposed approach. From empirical evidence, d^{RA} mostly outperformed other alignment-free approaches. In some cases, although d^{RA} takes longer processing time, it offers better accuracy and consistency than baseline approaches.

5.2 Proposed Method

According to [13], lack of alignment makes it more difficult to extract all of the possible information about evolutionary distances between species from k-mer-based methods because they only use differences in the presence/absence of k-mers. For example, if k-mer contains multiple substitutions, it is counted as one k-mer difference, which is the same as a k-mer that contains only one substitution. Thus, a lower k is more sensitive to the evolutionary distances than a larger k. However, the lower k causes the homoplasy problem, which is popularly considered as "noise" in the phylogenetic tree reconstruction [48, 13]. Therefore, the parameter k affects distance measurement and needs to be appropriately set.

Moreover, larger k in the k-mer-based methods can deal with the homoplasy problem but is not sensitive to the evolutionary distances because it causes more loss of evolutionary information. Hence, k-mer-based methods require large datasets with vast amounts of data to provide accurate distances and balance the effect of long k-mer [13].

To solve these problems in the k-mer-based methods, I propose a novel approach to eliminate the dependency of the k parameter so that the method works well with not only large datasets but also small datasets. To maintain the accuracy of the method as much as possible, I take advantage of the alignment aspect of MSA because the alignment method evaluates the evolutionary distances based on the mutation that causes the substitution directly. When working with the NGS short reads data, many methods need to use an assembly process, but this is time-consuming and has the problem of lack of suitable reference sequences. Hence, the method focuses on the alignment approach without assembly on NGS data. Then, the problem becomes how to approximate the distance between NGS short reads sets without assembly. To tackle this problem, I propose a method to combine the distance of each alignment pair into an accurate pairwise distance using the Gaussian mixture model. I call my method the *short read alignment approach* or d^{RA}.

To define the method, I consider the relationship between the alignment of assembled sequences and the NGS short reads without assembly. With two NGS sets $A = \{a_1, a_, \cdots, a_n\}$ and $B = \{b_1, b_2, \cdots, b_m\}$, let S_A and S_B be the sequences assembled from NGS sets A and B, respectively. According to the MSA, S_A and S_B are aligned into $Align_A$ and $Align_B$ by inserting some gaps. The distance between these two sequences can be calculated from those aligned sequences, as shown in Fig. 5.1.

To replicate the alignment of the $Align_A$ and $Align_B$ on the NGS short reads without assembly, I assume that, for some NGS short read, $a \in A$, a could be considered as inexact match (the matching process which allows some mismatch and gaps) with the substring of $Align_B$ because the gaps are allowed in the alignment. Given that $Align_A$ and $Align_B$ are aligned with each other, a is also an inexact match with the substring of $Align_B$. In other words, some NGS short reads $a \in A$ are an inexact match with some NGS short reads $b \in B$. With this information, we could establish the relationship of the alignment with NGS short reads directly without assembly.

Any NGS short reads $a \in A$ could be considered an inexact match, the matching process which allows some mismatch and gaps, with the substring of $Align_A$, because the gaps are allowed in the alignment. Since $Align_A$ and $Align_B$ are aligned with each other, a is also inexact match with the substring of $Align_B$. In other words, any NGS short reads $a \in A$ is inexact match with some NGS short reads $b \in B$. With this information, we could replicate the alignment of the $Align_A$ and $Align_B$ on the NGS short reads without assembly.

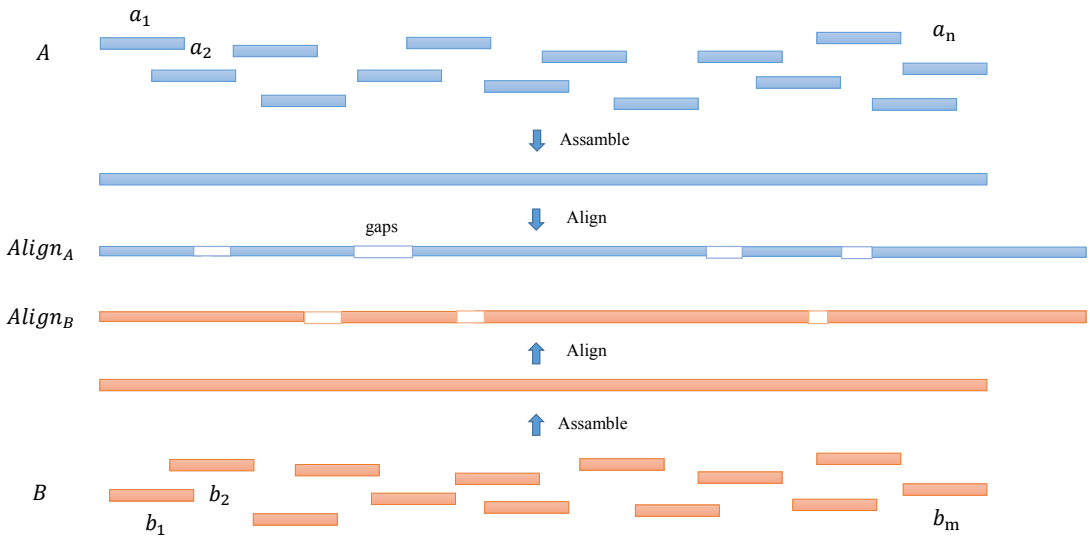


Figure 5.1: The traditional method to align two NGS short reads data

For a pair of strings x and y, let $d(x,y)$ denote the normalized unit cost edit distance, i.e., $d(x,y)$ is calculated by the edit distance with all costs of operation being equal to 1 between x and y divided by $max(|x|,|y|)$. Consider two aligned sequences $Align_A = align_{A1}...align_{An}$ and $Align_B = align_{B1}...align_{Bn}$ with the distance between them equal to $d(AlignA, AlignB)$. Assume that the probability that the substitution, insertion, and deletion occur is independent and uniform in $Align_A$ and $Align_B$. Therefore, with any corresponding substrings $s_a = align_{Ai}...align_{Aj}$ and $s_b = align_{Bi}...align_{Bj}$ where $1 \leq i, j \leq n$, the normalized unit cost edit distance $d(s_a, s_b) \approx d(Align_A, Align_B)$.

Because some NGS short read $a \in A$ can be an inexact match or alignment with some NGS read $b \in B$ when A and B are the NGS sets, we could consider the alignment part between a and b as the corresponding substring of $Align_A$ and $Align_B$. For example, in Fig. 5.2, the NGS short read $a_2 \in A$ is the alignment pair of $b_2 \in B$ with the alignment part shown as the region between the red lines. However, only one alignment pair is not enough to approximate an accurate distance $d(Align_A, Align_B)$. With the collection of the alignment pairs between NGS short reads of A and B, the concatenation of the alignment parts that represent the longer corresponding substrings of $Align_A$ and $Align_B$ would provide more accurate distance approximation of the $d(Align_A, Align_B)$.

The proposed method consists of two main steps. First, searching for the "*alignment pair*", corresponding NGS short reads of each read from any NGS sets with the other sets (species). The second step is combining the distance of alignment pairs into the pairwise distance between each two input NGS sets.

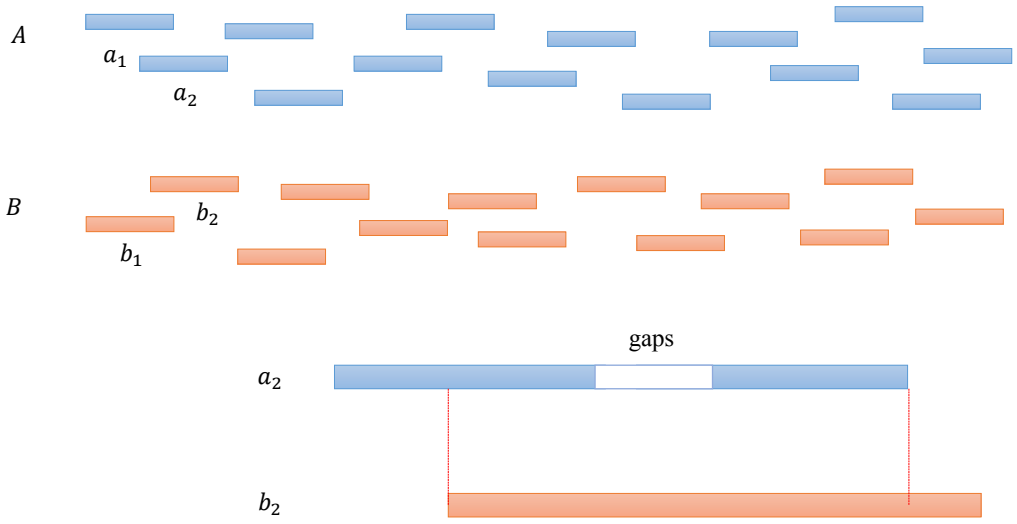


Figure 5.2: The relationship of the alignment between 2 NGS short reads without assembly

5.2.1 Alignment pair searching

At this step, searching for the alignment pair from each NGS sets for every NGS short reads that are required in the next step. The set of alignment pairs between A and B is denoted by $P(A,B)$ as follows:

$$P(A,B) = \bigcup_{i=1}^{n} \underset{(a_i,b)\in\{a_i\}\times B}{argmin} \; d(a_i,b)$$

.

The alignment pairs in $P(A,B)$ are the pairs of NGS short reads $a \in A$ and $b \in B$ with minimum edit distance. Fig. 5.4 shows an example of the alignment pair searching. According to the figure, there are several NGS sets A, B to Z. Each set consists of the NGS short reads $A = \{a_1, a_2, ..., a_l\}$, $B = \{b_1, b_2, ..., b_m\}$ until $Z = \{z_1, z_2, ..., z_n\}$ when l, m, and n are the sizes of sets A, B, and Z, respectively. Then search for the alignment pair of each NGS short reads of all other sets. For example, with $a_1 \in A$, I first search the alignment pair of a_1 in B and find b_1. Then, continue searching in the other sets until the last set Z. In Z, we find the alignment pair is z_n as shown in the figure. Given that the size of each set is not the same, some short reads might be aligned to more than one read. If $|A| > |B|$ then some short read in A could be aligned with the same short read in B. For example, short read a_2 and a_i are paired with b_2.

The overall process of alignment pair searching is very similar to the *maximum weight bipartite graph matching* problem. For this problem, A bipartite graph $G = (U,V,E)$ is a graph whose vertices can be divided into two disjoint sets U and V such that each edge

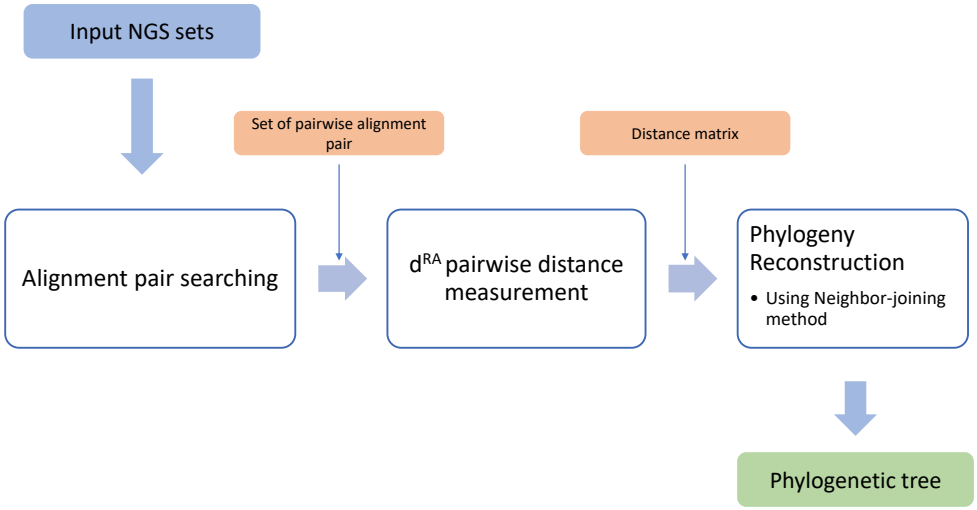


Figure 5.3: The pipeline process to construct phylogenetic tree using d^{RA}

$(u_i, v_j) \in E$ connects a vertex $u_i \in U$ and one $v_j \in V$. If each edge in graph G has an associated weight w_{ij}, the graph G is called a weighted bipartite graph. In a bipartite graph $G = (U, V, E)$, a matching M of graph G is a subset of E such that no two edges in M share a common vertex. If the graph G is a weighted bipartite graph, the maximum weighted bipartite matching is a matching whose sum of the weights of the edges is maximum.

However, applying maximum weight bipartite graph matching to this alignment pair searching is not optimal for several reasons. First, the weighted matrix E is needed to be fully constructed before applying the maximum weight bipartite graph matching algorithm. In alignment pair searching case, the set of vertex U and V represents the set of NGS short reads in A and B. Therefore each edge $(u_i, v_j) \in E$ with weight w_{ij} represents the alignment between reads $a_i \in A$ and one $b_j \in B$ with similarity $1 - d(a_i, b_j)$. The pairwise distance of every short read between NGS set A and B must be calculated. This process could be computationally heavy for alignment pair searching. The other reasons are that the result for the maximum weighted bipartite matching graph is a bijection, but the number of short reads for each NGS set is not the same. Hence there would be some short reads that are not paired with others which could contain distance information for the pairwise distance measurement.

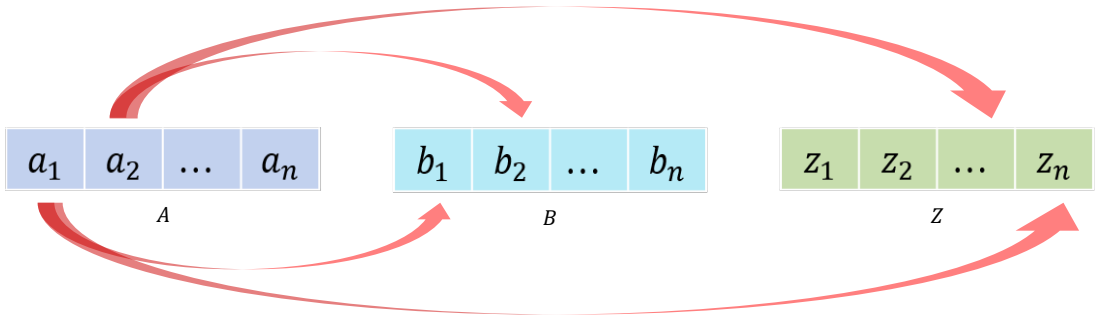


Figure 5.4: Alignment pairs searching

5.2.2 Pairwise distance measurement

After retrieving a collection of alignment pairs from the alignment pair searching step, I use the distance of alignment pairs to calculate the final pairwise distance between NGS sets. For the distance measurement, I consider each alignment pair as a part of the overall alignment between two NGS sets. Hence, we could estimate the distance between any NGS sets by combining the alignment pairs corresponding to those sets with the following equation:

$$d^{RA} = (D(A,B) + D(B,A))/2, \tag{5.2.1}$$

$$D(A,B) = \sum_{(a,b)\in P(A,B)} \left(-\frac{3}{4} ln \left(1 - \frac{4}{3} d(a,b) \right) \right) * w_{s(a,b)}. \tag{5.2.2}$$

I define $D(A,B)$ as the pair-wise distance between two NGS sets A and B (Eq. 5.2.1). So $D(A,B)$ can be calculated by the summation of the Jukes-Cantor distances of any corresponding alignment pair in $P(A,B)$ with the weight of $w_{s(a,b)}$. The Jukes-Cantor model estimates the evolutionary distance between DNA sequences by considering the mutation rate of the nucleotide. The model assumes that all four nucleotides A, C, T, and G have the same probability of appearing in the sequence and the same mutation rate. Given that the alignment set $P(A,B)$ is not equal to $P(B,A)$, $D(A,B)$ is asymmetric. So, I define the distance measurement d^{RA} as an average of $D(A,B)$ and $D(B,A)$.

The weight $w_{s(a,b)}$ is from the assumption that each individual alignment pair distance should not contribute to the final pair-wise distance equally. The significance of the alignment pairs increases exponentially to the similarity of the alignment pair [31]. Hence, the relationship between the weight $w_{s(a,b)}$ and the similarity $s(a,b) = 1 - d(a,b)$ is defined as follows:

$$w_{s(a,b)} = \frac{exp\,(s\,(a,b))}{\sum_{(a,b)\in P(A,B)} exp\,(s\,(a,b))}. \tag{5.2.3}$$

However, there are some cases in which the alignment pairs retrieved in the searching step are not the corresponding alignment pairs from the alignment of the assembly sequences. These noncorresponding alignment pairs should contribute to the pairwise distance significantly less than the corresponding pairs.

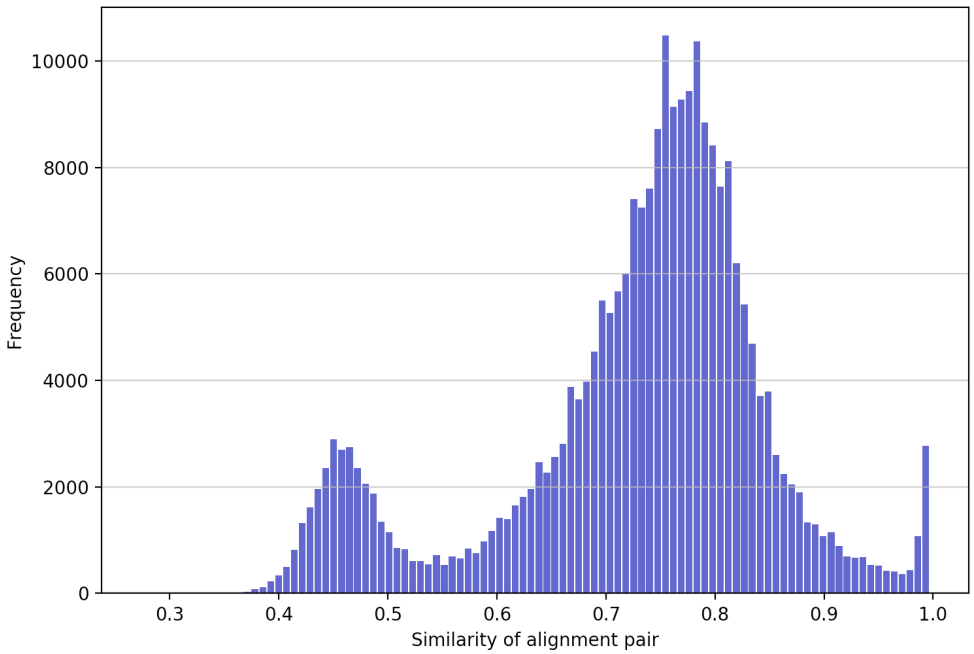


Figure 5.5: Similarity of alignment pair

I assume that the distribution of frequency of the all alignment pairs $(a,b) \in P(A,B)$ according to their similarity is a bimodal distribution. The bimodal distribution consists of two modes (peaks). In this case, the distribution of the first mode with less similarity is referred to as *noncorresponding* alignment pairs, and the second mode with more similarity as *corresponding* alignment pairs. The alignment pairs with high similarity have more probability of being the corresponding pairs. Fig. 5.5 shows an example of the bimodal distribution.

To prove the assumption of the distribution of lower similarity is referred to *noncorresponding* alignment pairs, I've conducted experiments on two random generated NGS sets, each set consists of ten thousand short reads. These two NGS sets are not related to one another. The distribution of the similarity of their alignment pair is shown in Fig. 5.6. The result shows that even unrelated NGS sets the alignment pair searching still provide the pair

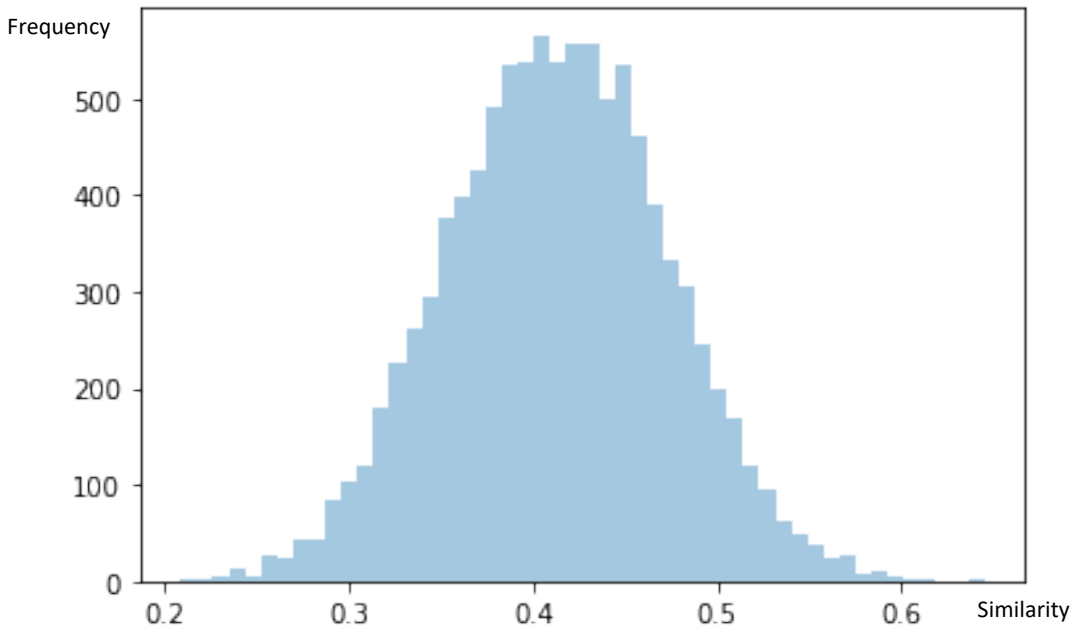


Figure 5.6: Distribution of the alignment pair between unrelated NGS sets

with some similarity between them but with low similarity than the actual related alignment pairs.

For the alignment pair $(a,b) \in P(A,B)$, let $Prob(a,b)$ denote the probability that the pair (a,b) is the corresponding pair. $Prob(a,b)$ can be calculated by learning the bimodal distribution using the Gaussian mixture model with an expectation maximization algorithm [8, 2]. To reduce the significance of non-corresponding alignment pairs, weight $w_{s(a,b)}$ was redefined according to the $Prob(a,b)$ by the following equation:

$$w_{s(a,b)} = \frac{exp\,(s\,(a,b)) * Prob(a,b)}{\sum_{(a,b)\in P(a,b)} exp\,(s\,(a,b)) * Prob(a,b)}. \tag{5.2.4}$$

5.3 Experiment and Evaluation

5.3.1 Experiment setup

Datasets

To evaluate the proposed method d^{RA}, I use three datasets, 29 *mammalian mtDNA* sequences [35, 4], 29 *Escherichia/Shigella* [57], and 18 *Drosophila* genomes [42]. The 29 *mammalian mtDNA* dataset consists of the mitrocondrial DNA sequence within 29 mammal species. The 29 *Escherichia/Shigella* dataset consists of the entire genome sequences of 29 species of

bacteria in the family of *Escherichia* and *Shigella*. The last dataset is the 18 species of fly (insect) or *Drosophila*. The statistics of all three datasets are shown in Table 5.1.

Table 5.1: Size and the total sequence lengths of the three datasets

	Size (MB)	Total sequence lengths
29 *mammalian mtDNA*	0.5	482,127
29 *Escherichia/Shigella*	144	141,962,164
18 *Drosophila*	3,110	3,109,816,396

Experiment Procedure

Given that all three datasets were initially long sequences, I used a tool called *ART* [22] to simulate NGS short reads from the long genome sequences. I used two error models; namely, *454* and *Illumina*, to simulate the NGS high-throughput sequencing results from two different NGS platforms. These methods produced the actual samples as NGS short reads data. The *454* model produces various lengths of NGS short reads and has a high chance of sequencing errors on homopolymer sequences, which include multiple consecutive duplicate characters. Meanwhile, the *Illumina* model provides fixed-lengths of NGS short reads and has no problem with the homopolymer sequences.

I conducted experiments with various values of coverage on each dataset. Coverage is the average time of occurrence of nucleotides at each position in the original sequences that appear in the NGS sets. For example, the coverage value 5x means that the NGS short reads overlap five times according to each position in the original sequences. I could say that the NGS set with 1x coverage is the NGS set with no overlap. The length of NGS short reads was set to 150 bps, with a default parameter for the error distribution for each model.

The size and the total number of original datasets and the simulated NGS sets are summarized in Table 5.2. Because, in practice, researchers usually get low coverage data in the sequencing process, I also conducted experiments on low coverage NGS data in this paper. I simulated four *454* and four *Illumina* NGS sets with 5x coverage in the 29 *mammalian mtDNA* dataset and 1x coverage in the 29 *Escherichia/Shigella* dataset.

In the 18 *Drosophila* dataset, I simulated four *Illumina* NGS sets with 0.1x coverage of the dataset. Because the 18 *Drosophila* dataset is the entire genome sequence dataset, it contains a massive amount of repeated sequences and homopolymer sequences. As noted above, using

the *454* model, it is possible to have sequencing errors on homopolymer sequences; thus, I did not simulate the *454* NGS sets with this dataset.

With the simulated NGS short reads data, I applied the proposed method d^{RA} to calculate a distance matrix. The phylogenetic tree was then constructed according to the calculated distance matrix.

Table 5.2: Size and the total number of short reads and total sequence lengths of NGS short reads set of all three datasets

	NGS set	Size (MB)	Total number of short reads	Total sequences length
29 *mammalian mtDNA* (5x)	454_1	5	8,540	1,982,139
	454_2	5	8,571	1,990,340
	454_3	5	8,618	1,989,688
	454_4	5	8,631	1,994,046
	illumina_1	5	16,010	2,401,500
	illumina_2	5	16,010	2,401,500
	illumina_3	5	16,010	2,401,500
	illumina_4	5	16,010	2,401,500
29 *Escherichia/Shigella* (1x)	454_1	260	499,945	111,949,666
	454_2	260	499,782	112,024,738
	454_3	260	499,285	111,918,934
	454_4	260	499,634	111,956,774
	illumina_1	304	946,150	141,922,500
	illumina_2	304	946,169	141,925,350
	illumina_3	304	946,151	141,922,650
	illumina_4	304	946,177	141,926,550
18 *Drosophila* (0.1x)	illumina_1	681	1,908,519	286,277,850
	illumina_2	680	1,907,719	286,157,850
	illumina_3	680	1,907,985	286,197,750
	illumina_4	680	1,908,134	286,220,100

Baselines Methods

I compared the proposed method with three existing k-mer-based alignment-free methods, CVTree [56, 37], d_2^S [47], and skmer [42]. I used k values in the range from 8 to 31 as suggested by *CVTree*, d_2^S, and *skmer* proponents.

Evaluation Metric

I used the *Clustel Omega* tool [44], followed by the *dnadist* tool in the *PHYLIP* package [15], on aligned sequences from MSA to calculate distance matrices. For each distance matrix, either from MSA or from alignment-free methods, I used the *neighbor* tool in the *PHYLIP* package to construct a phylogenetic tree by the neighbor-joining method [41].

I used the popular Robinson–Foulds distance (RF) [40] for the evaluation. The RF value was calculated by counting the internal nodes that appear in one tree but not in the others. Let $N = (V, E)$ be a given phylogenetic tree. For any two nodes u, $v \in V$, v is a descendant of u if v is reachable from u in N. For any $v \in V$, define the cluster of v (denoted by $C(v)$) as the set of all leaf nodes that are descendants of v. The cluster collection of N is the multiset $C(N) = \{C(v)|v \in V\}$. The RF distance between two phylogenetic trees N_1 and N_2 is:

$$d_{RF}(N_1, N_2) = (|C(N_1) - C(N_2)| + |C(N_2) - C(N_1)|)/2$$

.

A small RF value between two trees means the shapes of the trees are similar. The values for RF range from zero, meaning the two trees are the same, to $2(n-3)$ where n is the number of leaf nodes.

Because MSA is limited by the size of the genome, only the 29 *mammalian mtDNA* dataset is capable of using the tree from MSA as the benchmark tree. The benchmark tree for 29 *Escherichia/Shigella* is the tree studied by the research [57, 51] and 18 *Drosophila* genomes tree is from the phylogenetic tree database *Open Tree of life* [21, 42, 43]. In the implementation, I also used the *USEARCH* tool [11] to search for the alignment pair of any NGS short reads.

To evaluate the consistency of the distance measurement, I utilized the *coefficient of variation* [12]. The coefficient of variation can be calculated by the ratio of the standard deviation σ to the mean μ as follows:

$$CV = \frac{\sigma}{\mu} * 100$$

.

5.3.2 The accuracy on phylogenetic tree reconstruction

I first estimated phylogenetic trees for 29 *mammalian mtDNA* sequences and 29 *Escherichia/Shigella* genome datasets. For each dataset, I simulated eight NGS short-read sets: four of *454* error

model and another four of *Illumina*. For all three alignment-free methods, I set different values of the k parameter to show the effect of this parameter on the phylogenetic tree result.

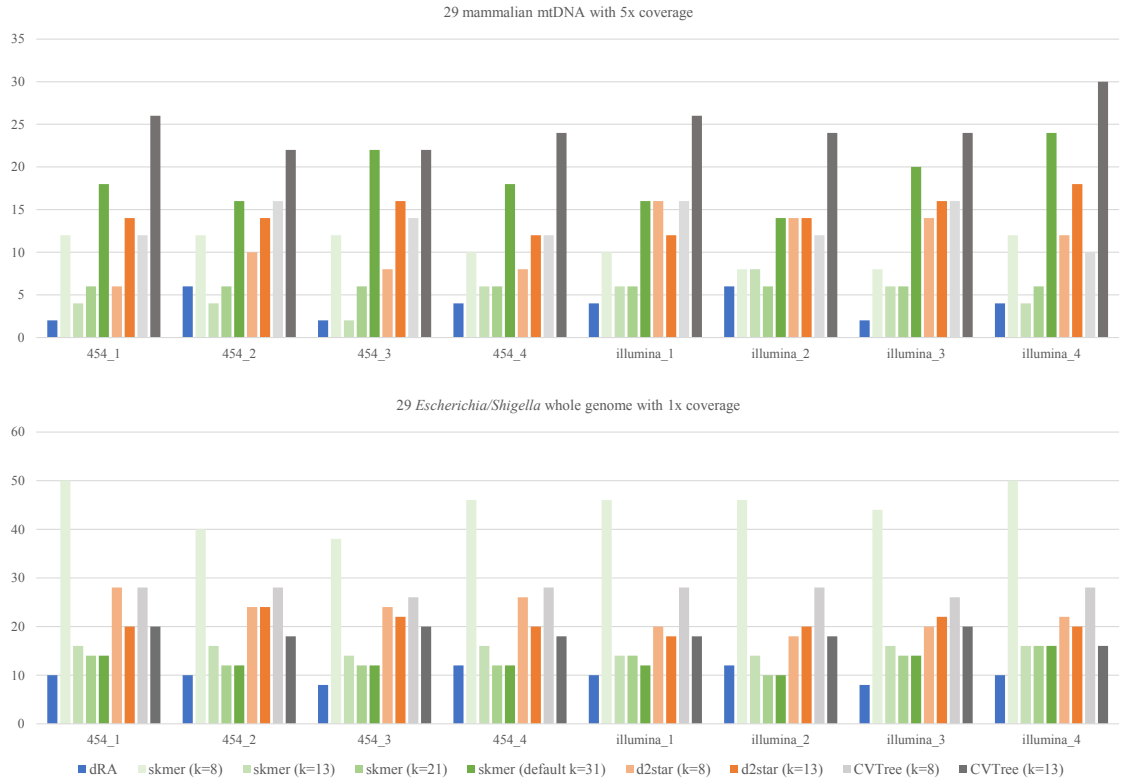


Figure 5.7: The RF distance between benchmark tree and phylogenetic trees reconstructed from the distance matrix estimated by the approach (d^{RA}), shown as the blue bar, and others k-mer based alignment-free methods

The results in Fig. 5.7 show that d^{RA} provides a beneficial distance measurement, which leads to accurate phylogeny reconstruction in both datasets. The RF distance between phylogenetic trees reconstructed from d^{RA} is the closest to the benchmark tree in most of the NGS short-read sets compared with other methods.

While the k parameter adjustment is required in *CVTree*, d_2^S, and *skmer* to provide the best phylogenetic tree results, d^{RA} does not require such adjustment to provide an accurate result, as shown in Table 5.3. According to Fig. 5.7, the optimal k for *skmer* in *mammalian mtDNA* sequences dataset is around 13, whereas k=31 provided the best result on the *Escherichia/Shigella* dataset. The phylogenetic tree results by *CVTree* and d_2^S were also affected by the k parameter. In practice, not many NGS sets have benchmark trees, thus adjusting the k parameter to provide the most accurate tree in further analysis is an ambiguous

process. d^{RA} is a k-free approach, so it can be applied even on NGS sets without benchmark trees.

Table 5.3: Average of RF distance between benchmark tree and phylogenetic trees of all simulated NGS short-read sets

	mammalian mtDNA (5x)	*Escherichia/Shigella* (1x)
d^{RA}	3.75	10
$Skmer(k=8)$	10.5	45
$Skmer(k=13)$	5	15.25
$Skmer(k=21)$	6	13
$Skmer(k=31)$	18.5	12.75
$d_2^S(k=8)$	11	22.75
$d_2^S(k=13)$	14.5	20.75
$CVTree(k=8)$	13.5	27.5
$CVTree(k=13)$	24.75	18.5

Because *Drosophila* has a much larger genome size than *Escherichia/Shigella*, the dataset that includes bacteria data, researchers usually manage to obtain low coverage data of the genome samples by using the NGS process. Therefore, I conducted experiments on 18 *Drosophila* datasets with 0.1x coverage to evaluate the accuracy of my proposed method on low coverage data. As shown in Fig. 5.8, d^{RA} provided a better phylogenetic tree for *Drosophila* in comparison with most of the other baseline approaches. Although *skmer* could also obtain low distances, as found in the approach, it required the k parameter to be tuned to achieve such results. I also observed that *CVTree* and d_2^S could not be used accurately to reconstruct the phylogenetic tree with this low coverage.

I then evaluated the effect of short-read length on the accuracy of d^{RA}. Fig 5.9 summarizes the accuracy with respect to short-read length. According to Fig. 5.9, d^{RA} does not provide a result on shorter reads (50 bp and 100 bp) that are as accurate as those of the longer reads. Given that the shorter reads contain less information on the alignment between them, the distance calculated from d^{RA} could be less accurate. However, d^{RA} still outperforms *skmer* with respect to accuracy on phylogenetic tree reconstruction.

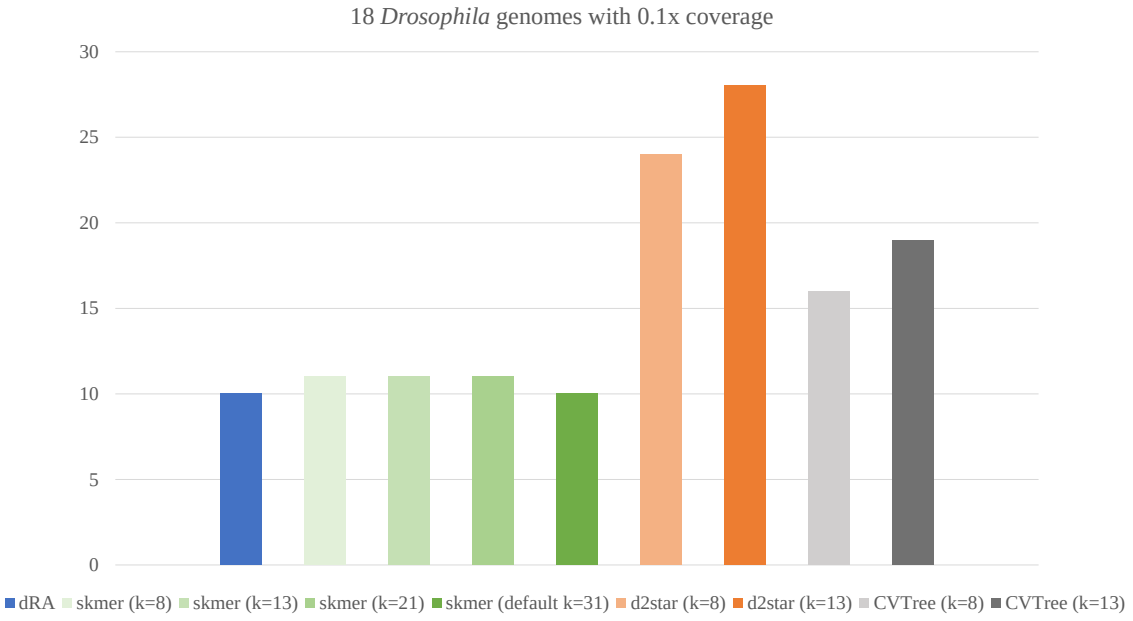


Figure 5.8: The average RF distance between benchmark tree and phylogenetic trees reconstructed from the distance matrix estimated using the approach (d^{RA}), shown as the blue bar

d^{RA} evaluates the distance between a pair of NGS sets according to the alignment pair of NGS short reads. I conducted experiments (as shown in Fig. 5.10) to examine whether d^{RA} can be used to measure accurate distances with different query sizes. Instead of using all short reads in each set as the query to search for its alignment pair in the other sets, I randomly chose a specific number of short reads as queries. For the 18 *Drosophila* datasets with 0.1x coverage, with the data size of 700 MB, I randomly sampled NGS short reads from each set with an overall size of 50, 70, 100, 350, and 700 MB (all short reads) as a query. The results, summarized in Fig. 5.10, indicate that d^{RA} can provide tree results close to the benchmark even with low query sizes. According to the statement from [24] the "Big data basically focuses on quality data rather than having very large irrelevant data so that better results and conclusions", this could provide the general concept to the d^{RA} not using all short reads as the queries analogy the d^{NS}. In the case of d^{RA}, the necessary information provides in the form of the weight of each alignment pair. When we consider the bimodal distribution of all alignment pairs according to their similarity (Fig. 5.5), the random sample of those alignment pairs still provides the same distribution. The other reasons are that the weight of the corresponding pairs is much higher than non-corresponding pairs which analogical to assigning more importance to relevant than irrelevant data. Although the random sample

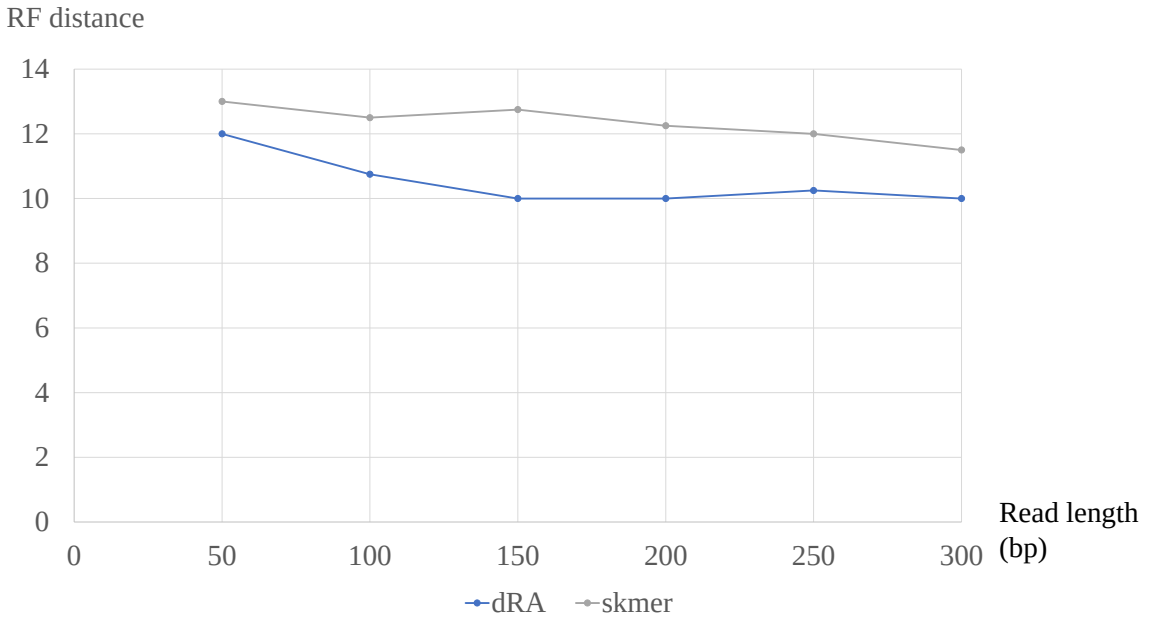


Figure 5.9: The RF distance between phylogenetic tree constructed by d^{RA} and the benchmark tree w.r.t. short-read length on the *Escherichia/Shigella* dataset

might not provide exactly the same distribution, but with this weight, the contribution of irrelevant data could be eliminated.

Moreover, as shown in Fig. 5.8, on all three datasets, the phylogenetic trees of d^{RA} are the closest to the benchmark, regardless of the size and coverage of the datasets. Although the *skmer* can also provide the same results in some cases, I noted that *skmer* is not effective on small datasets such as *mammalian mtDNA* and *Escherichia/Shigella*. Because *skmer* measures the distance based on the k-mer occurrences, small datasets do not provide enough k-mer information for *skmer* to measure accurate distance. *CVTree* and d_2^S also have the same problem. This shows how d^{RA} is a general and effective approach that can be used on any dataset and can be a better option than the other methods.

I also compared the true edit distance and estimated distance given by the following equation:

$$D'(A,B) = \sum_{(a,b)\in P(A,B)} d(a,b) * w_{s(a,b)}$$

where $w_{s(a,b)}$ is the weight in Eq. 5.2.4.

In this experiment, I used simulated sequences from the *E. coli O157* entire genome sequence. The simulated sequences were set with normalized edit distances equal to 0, 0.01, 0.05, 0.1, 0.15, and 0.2 compared with *E. coli O157* sequence as "true edit distance."

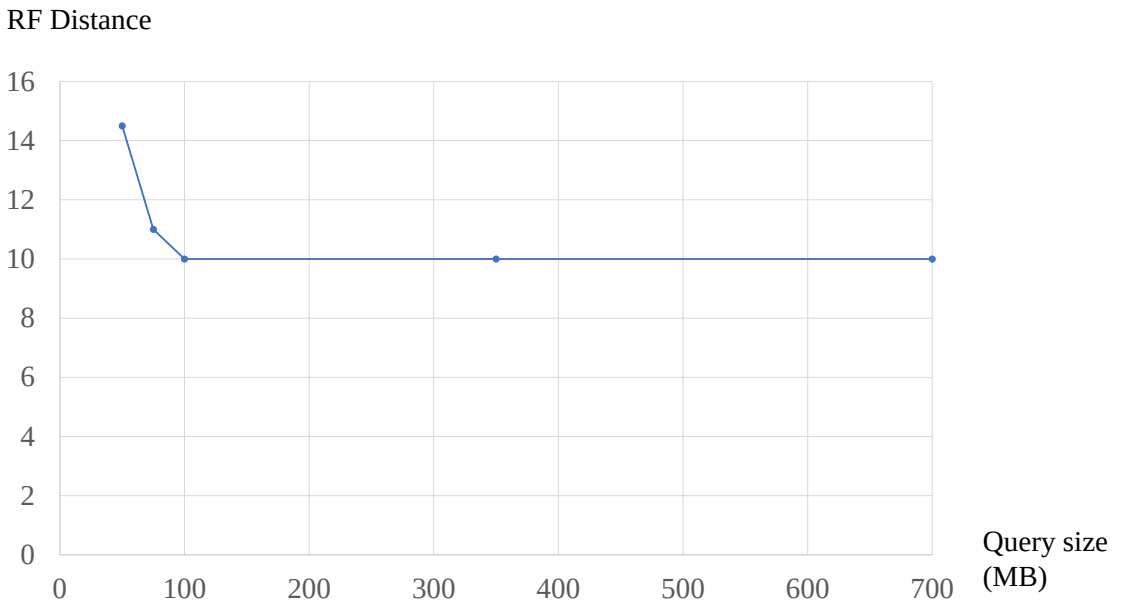


Figure 5.10: The RF distance between phylogenetic tree constructed by d^{RA} with varied query size and the benchmark tree on 18 *Drosophila* dataset

For each simulated sequence and also the *E. coli O157* sequence, I generated the corresponding NGS set with varied coverage 0.25x, 0.5x, 1x, 2x, and 4x. Because the distance calculated from d^{RA} already includes the evolutionary distance model in the calculation (the term $-\frac{3}{4}ln\left(1-\frac{4}{3}d(a,b)\right)$ in Eq.5.2.2), this evolutionary distance model is applied to each alignment pair, not the entire genome sequence. To evaluate the accuracy of the method with true edit distance, I considered using just $d(a,b)$ in Eq.5.2.2 instead of evolutionary distance model term.

The results, shown in Fig. 5.11, are the estimated distances from d^{RA} between simulated sequences and *E. coli O157* with different true edit distance and coverage. The x-axis is the true edit distance between simulated sequences and *E. coli O157* sequence. According to Fig. 5.11, d^{RA} can be used to estimate accurate distances between NGS sets of simulated sequence and the *E. coli O157* sequence. However, the coverage affects the estimated distance calculated by d^{RA}.

Fig. 5.12 shows that the length of the short reads affects the distance calculation of d^{RA}. In this experiment, I also compared the simulated sequences which are set to have the normalized edit distance equal to 0, 0.01, 0.05, 0.1, 0.15, and 0.2 compared with *E.coli O157* sequence as "true edit distance". Fig. 5.12 shows the distance result of the calculating distance between NGS sets of simulated sequence and *E.coli O157* sequence with different short reads length. With the length of the short reads of 50bp, d^{RA} tends to calculate the

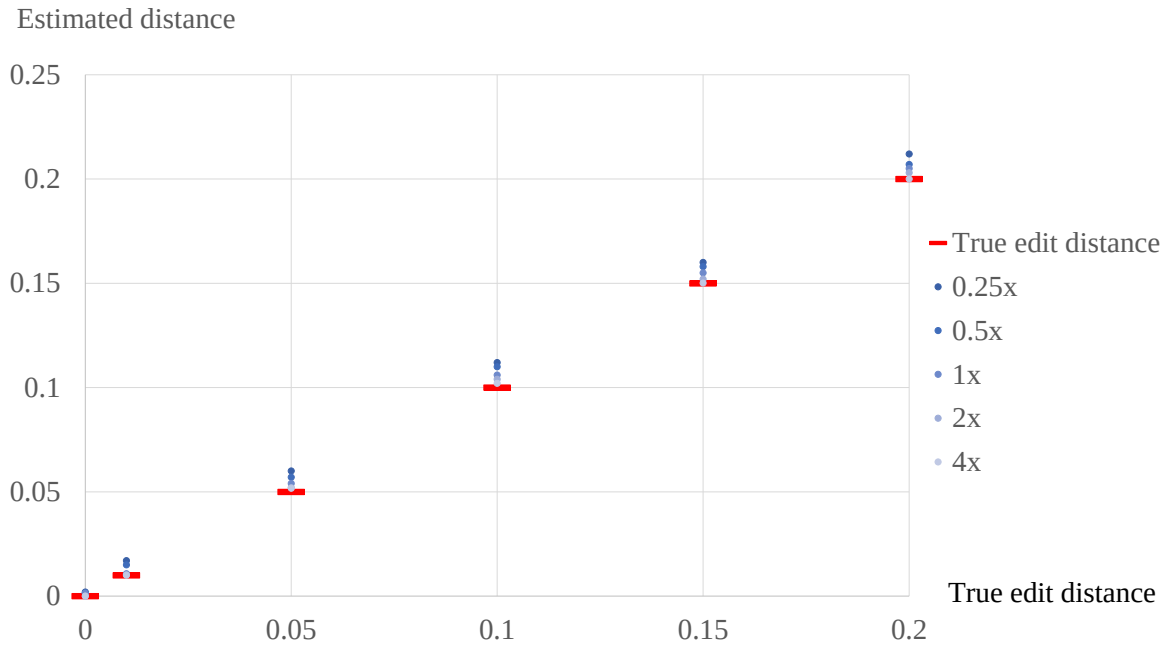


Figure 5.11: Comparison of distance calculated by d^{RA} and true edit distance

distance lower than the true edit distance while the longer length provides the distance close to true edit distance. Although the estimated distances are less accurate when reads length is short, the phylogenetic trees constructed from those results still provide the good tree, as shown in Fig. 5.9.

The proposed method uses the unit cost edit distance to measure the distance d^{RA} among short read sets. However, the costs may affect the resultant trees. I evaluated the accuracy of resultant trees with several costs. For the three datasets, Table 5.4 shows the average RF distance between phylogenetic tree and trees constructed by the proposed method d^{RA} and the state-of-the-art *Skmer* with the best parameter k. I examined the accuracy for the costs of the unit cost (1,1,1), hamming distance (1,0,0), (2,1,1), and (1,2,2) where first (resp. second and third) component stands for the cost of substitution (resp. addition and deletion).

As I can see in Table 5.4, the edit cost affects the accuracy, especially for the data containing diverse species like *mammalian mtDNA*. However, the proposed method still constructs a better tree than the state-of-the-art *Skmer* with the best k. Therefore I use the unit cost edit distance for measuring the distance of d^{RA}.

Figure 5.13 shows an example of a phylogenetic tree result for 29 *mammalian mtDNA* dataset. The tree that was reconstructed from the distance matrix calculated by d^{RA} is almost the same as the benchmark tree. According to the dataset, I can categorize the input species into four groups: *Primates*, *Ferunguletes*, *Rodents*, and *Outgroup*. d^{RA} was able to separate the 29 species into these four groups effectively. The only difference to the benchmark tree

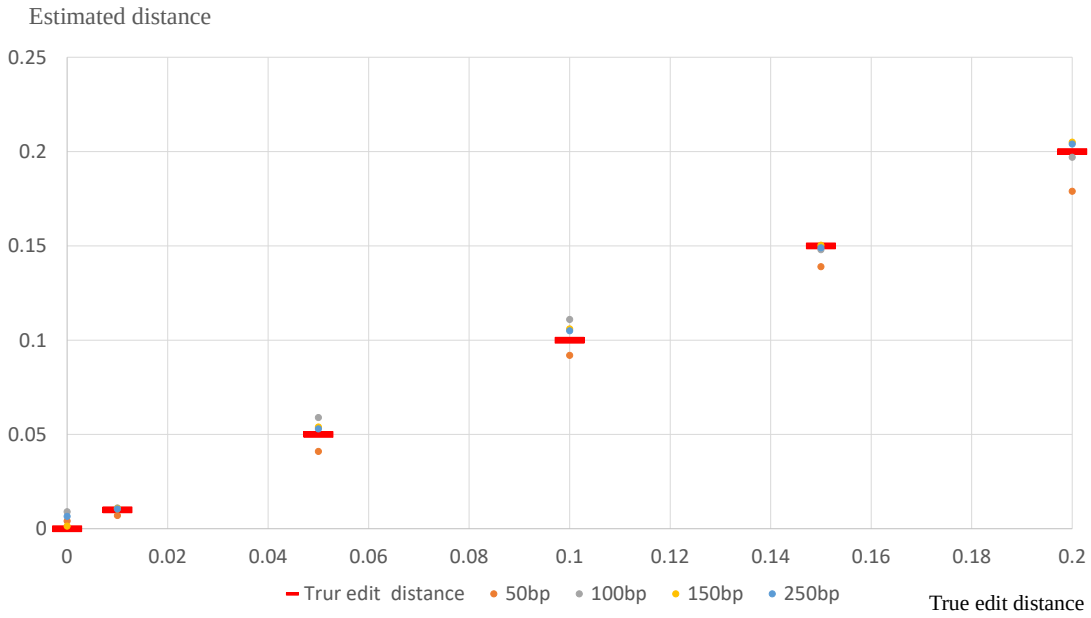


Figure 5.12: The comparison of distance calculated by d^{RA} and true edit distance w.r.t. short read length

is the branch between cat and dog. In the d^{RA} tree, cat and dog are in a group apart from two seal species. However, in the benchmark tree, the cat is branched out from dogs and seals. In this result, the distance measurement from d^{RA} between the cat and the group of dogs and seals is not high enough to distinguish them.

Figure 5.14 shows the comparison of phylogenetic tree of 29 *Escherichia/Shigella* between d^{RA} the benchmark tree, d^{RA} and skmer with k=8. The tree that reconstructed from the distance matrix calculated by d^{RA} is similar to the benchmark tree. The species in this dataset can be categorized into two main groups, Escherichia and *Shigella*. d^{RA} can separate the 29 species into these two groups as the benchmark tree. On the other hand, the tree from skmer is different from the benchmark. Although skmer can provides a similar result to d^{RA} with k=31, the result tree would be different when a k parameter is changed. Without a benchmark tree, identifying the optimal phylogenetic tree from a different k parameter is not a trivial task. For the 18 *Drosophila* dataset, Figure 5.15 also show the tree that reconstructed from the distance matrix calculated by d^{RA} is similar to benchmark tree.

5.3.3 Distance consistency for pair-wise distance

For any dataset, the distance measurement between NGS sets should be almost the same every time, regardless of different NGS short reads. The consistency exposes the difference

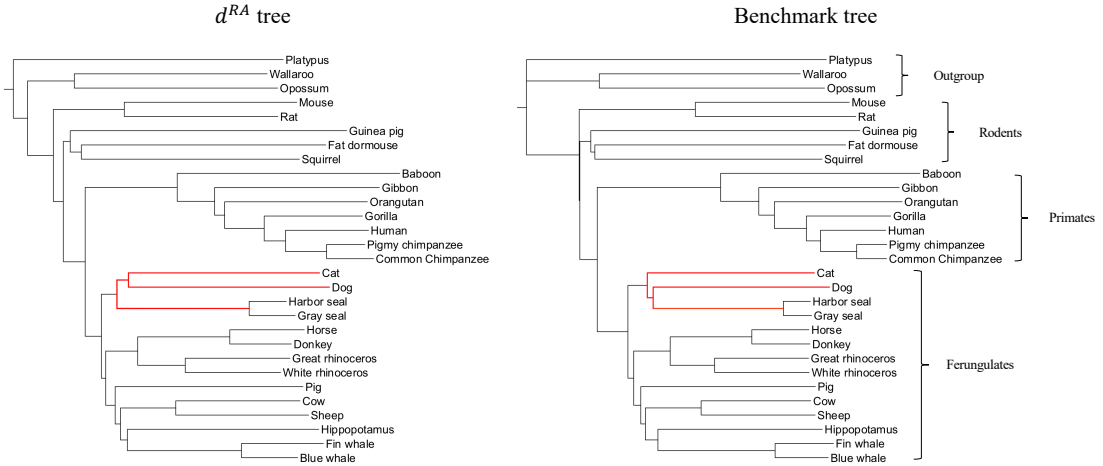


Figure 5.13: The comparison of phylogeny tree of 29 *mammalian mtDNA* between d^{RA} tree (left) and the benchmark tree (Right)

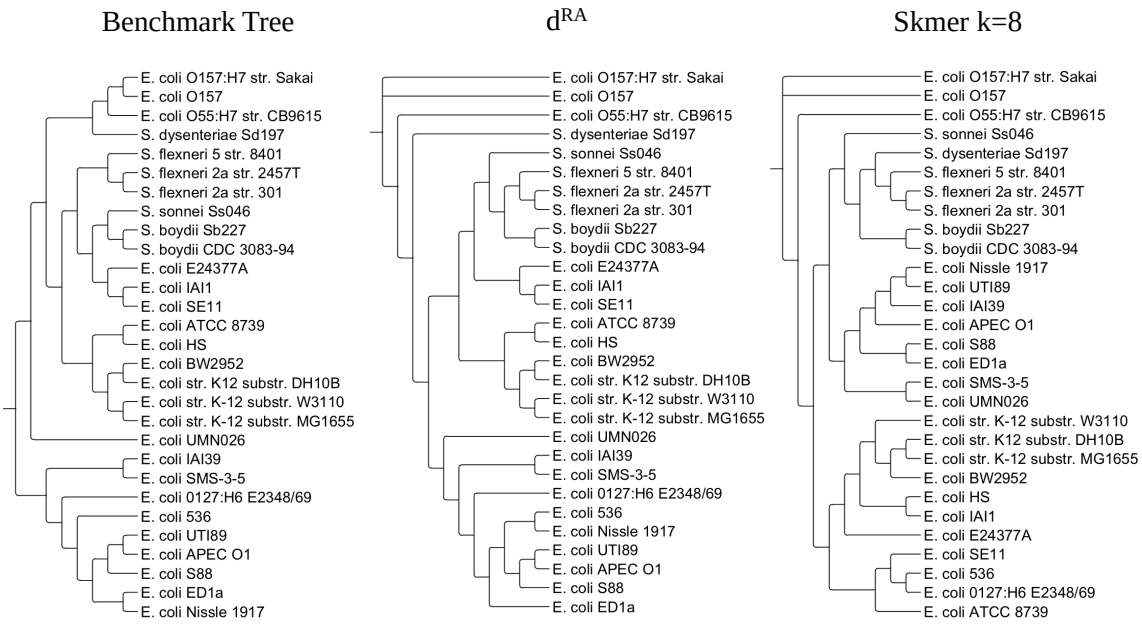


Figure 5.14: The comparison of phylogeney tree of 29 *Escherichia/Shigella* between d^{RA} tree (center), the benchmark tree (left) and skmer with k=8 (right)

Table 5.4: Average RF distance between benchmark tree and phylogenetic trees constructed from NGS short read sets w.r.t. the edit distance cost

	mammalian mtDNA (5x)	*Escherichia /Shigella* (1x)	*Drosophila* (0.1x)
d^{RA} : Uniform (1,1,1)	3.75	10	10
d^{RA} : Hamming (1,0,0)	6	11	10
d^{RA} : (2,1,1)	10.25	11.5	10.25
d^{RA} : (1,2,2)	8	11.25	10
$Skmer(k = 13)$	5	15.25	11
$Skmer(k = 31)$	18.5	12.75	10

in distances among multiple NGS sets in the same dataset. Even though the accuracy of the phylogenetic tree reconstruction is an important aspect of evaluating the methods, without consistency, the accuracy is not convincing. Therefore, I also conducted experiments to evaluate the consistency of the distance measurement. I used the coefficient of variation to evaluate the consistency of the methods. Fig. 5.16 presents a heatmap of the coefficient of variation for each element in the distance matrices calculated from multiple NGS sets in the *Escherichia/Shigella* dataset. In the figure, d^{RA} is compared with the *skmer* (k=31) because it provided RF distance results similar to those of d^{RA} in the accuracy evaluation shown in Table 5.3. According to Fig. 5.16, d^{RA} provides a lower coefficient of variation in most of the elements in the distance measurement while *skmer* (k=31) reveals a very high coefficient of variation of distance between some pairs in the *Escherichia/Shigella* dataset despite the good RF distance results.

I evaluated the difference of pairwise distances computed by the distance matrices in the method d^{RA}, *CVTree*, d_2^S, and *skmer* using different simulated NGS sets of each dataset. Table 5.5 shows the average coefficient of variation values of all pairs in the distance matrices. d^{RA} provided a relatively low value of the coefficient of variation compared with the other methods. Thus, it can estimate the distances with not much difference between NGS sets. In this respect, although *CVTree* can calculate the most consistent result, it provided the worst accuracy. When considering the accuracy along with the consistency, d^{RA} reveals the effectiveness of distance measurement with NGS short reads data. d^{RA} provides the closest

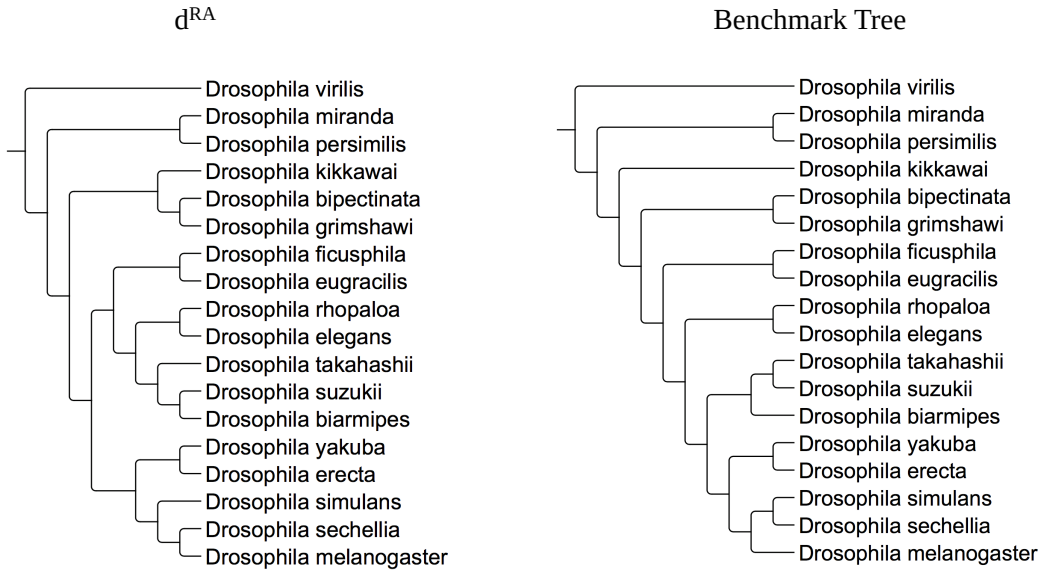


Figure 5.15: The comparison of phylogeney tree of 18 *Drosophila* between d^{RA} tree (left), the benchmark tree (right)

phylogenetic tree to the benchmark while maintaining consistency with a low coefficient of variation value compared with the other methods.

5.3.4 Efficiency evaluation

I compared the runtime of the proposed method with the others using all three datasets with different sizes. The 29 *mammalian mtDNA* with 5x coverage, 29 *Escherichia/Shigella* with 1x coverage, and 18 *Drosophila* with 0.1x coverage have data sizes of 5, 300, and 700 MB, respectively. The experiments are conducted by using Intel Core i7-4980HQ 2.8 GHz processor which includes four independent processors with 16 GB DDR3L SDRAM.

For d^{RA}, I ran experiments with a query size of 100 MB for the *Drosophila* dataset. The runtime results are shown in Table 5.6. I observed that d^{RA} could calculate the distance between NGS sets as fast as the alignment-free approaches, although it is based on the alignment among short reads. In k-mer-based methods, the computational time is varied by k parameter. The bigger the k value, the longer the time required for the distance calculations. d_2^S showed a huge difference between k=8 and k=13, as did *CVTree*. With the k-free approach, d^{RA} does not require additional calculations to tune the k parameter; thus it provides an accurate phylogenetic tree within reasonable processing time.

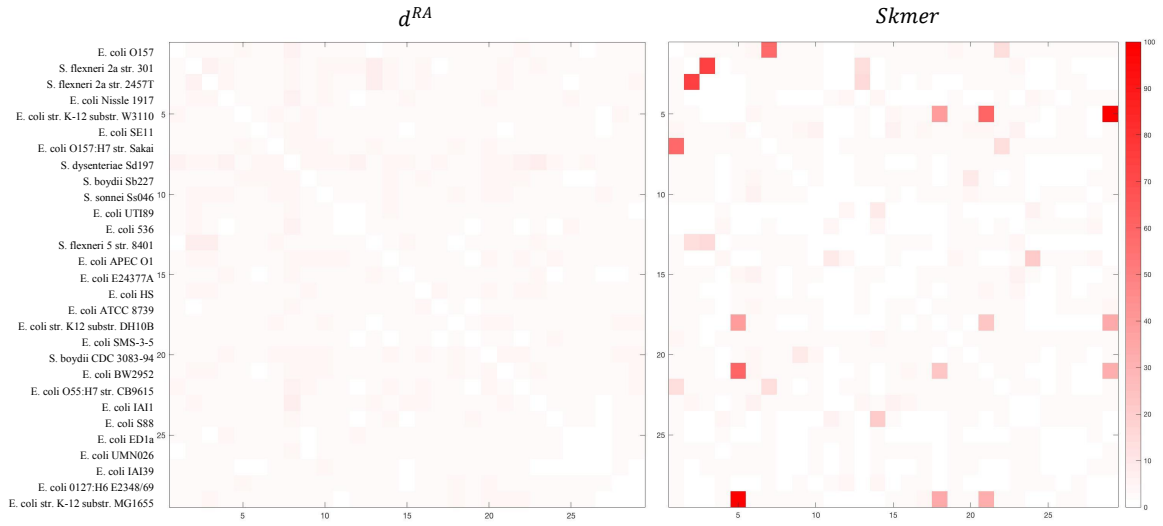


Figure 5.16: A heatmap showing the value of the coefficient of variation for each pair-wise distance on multiple NGS sets of the *Escherichia/Shigella* dataset. Red refers to a high coefficient of variation and white is low

In some cases, d^{RA} runs slower than the other methods. However, in such cases, d^{RA} offers much better phylogenetic tree results. Therefore, it is a worthy trade-off between efficiency and effectiveness. For instance, although d_{RA} is three times slower than *skmer* with k=8 on the *Escherichia/Shigella* dataset, the resultant phylogenetic tree result obtained by *skmer* is more different to the benchmark than d^{RA} for 4 times.

Fig. 5.17 shows how the runtime increases with respect to the data size in comparison with the other methods. Most of the methods showed a linear d_{RA} growth according to the data size. However, the k parameter significantly affects the runtime of *skmer*, d_2^S, and *CVTree*. For *skmer*, lower k requires a larger number of k-mers to be considered in the distance calculation. On the other hand, larger k results in a larger dimension k-mer profile for d_2^S and *CVTree*. This result also shows the advantage of the k-parameter-free method.

The complexity of the method is $O(nl + qnl + qi)$ where n is the total number of the short reads, q is the number of queries, and l is the length of short reads. At the first step of alignment pair searching, it requires $O(nl)$ to do indexing for every n short read with length l and searching for every q query to find the short read with minimum distance from the query is required $O(qnl)$. $O(qi)$ is complexity in the Gaussian mixture model steps using the EM algorithm, and i is the number of iteration. The last step is the pairwise distance calculation, which uses $O(q)$ times.

Table 5.5: The average coefficient of variation

	mammalian mtDNA (5x)	*Escherichia /Shigella* (1x)	*Drosophila* (0.1x)
d^{RA}	3.62	2.56	2.74
$Skmer(k=8)$	6.03	11.88	4.65
$Skmer(k=13)$	4.11	5.01	3.54
$Skmer(k=21)$	3.38	3.51	2.98
$Skmer(k=31)$	2.03	3.26	1.76
$d_2^S(k=8)$	24.60	55.74	2.35
$d_2^S(k=13)$	4.83	18.89	1.75
$CVTree(k=8)$	1.59	3.58	1.3
$CVTree(k=13)$	1.21	1.39	0.82

5.3.5 Comparison between d^{RA} and d^{NS}

In this section, I compared the accuracy between d^{RA}, d^{NS}, and the state-of-art method: *skmer* with the simulated datasets. The previous experiments have been done on real-world datasets with their benchmark trees. Since there is no ground truth tree for any given dataset, hence the evaluation result might not fully show the efficiency of the method by using the benchmark tree used by other researches. To provide the concrete evidence for the efficiency of d^{RA} and d^{NS}, I also conducted the experiments on the simulated dataset with the specific tree as the benchmark tree as the ground truth for the evaluation.

Simulated datasets

I simulated three datasets, which each consist of 30 species with a specific phylogeny tree as shown in Fig. 5.18 as the ground truth tree. The difference between the three datasets is the branch length of the tree, which represents the evolutionary distance between nodes. The branch length between every node in the tree is set to 0.001, 0.01, and 0.1. To simulate the NGS dataset, I use a tool called *Seq-Gen* [38]. The sequences in each dataset are simulated from the template sequence to provide the input phylogeny tree. Then use *ART* to simulate 8 NGS set from each sequence in the dataset with 1x coverage.

Algorithm 2: Algorithm for d^{RA} to calculate pair-wise distance

Input : Set of NGS short reads of m species $S = \{R_1, \ldots, R_m\}$ and $R_i = \{r_{i,1}, \ldots, r_{i,o}\}$, the total number of reads is $n = m * o$
Output : A Distance matrix denoted by D with $m \times m$ dimensions;
Initialize $USEARCH^1$ index I;
foreach $r_{i,j} \in R_i$ **do**
  foreach $R_i \in S$ **do**
    I.insert($r_{i,j}$)
  end
end
$Q \subset$ all NGS short reads as query;
Initialize a set A of alignment pairs;
foreach $q_i \in Q$ **do**
  foreach $R_j \in S$ **do**
    res=searchPair(q_i,I,R_j);
    A.push([q_i,res]);
  end
end
Estimate the parameters of bimodal Gaussian mixture model M;
foreach $A_i \in A$ **do**
  $prob$=$Prob(A_i, 1|M)$ is probability of A_i be in group 1;
  D=updateDistance(A_i,D,$prob$);
end

Evaluation metrics

Since the datasets are simulated, we have the information about the branch length of the benchmark tree. In this section, I also use the additional evaluation metric called *Branch-Score distance* (Bs) [27]. While the RF-distance compares the trees by considering just topology, the Branch-Score distance also considers the branch length. Consider the set of all possible splits for N species $(B_1, B_2, ..., B_N)$. Each tree can be represented by such an array, in which $B_i = 0$ if the split is not found in the tree, and the length of the branch if the split is found. The Branch-Score distance between two trees $(B_1, B_2, ..., B_N)$ and $(B'_1, B'_2, ..., B'_N)$ is denoted by

$$Bs = \sum_{i=1}^{N} (B_i - B'_i)^2$$

Table 5.6: The runtime of each method for all three datasets (seconds)

	d^{RA}	$Skmer(k=8)$	$Skmer(k=31)$	$d_2^S(k=8)$	$d_2^S(k=13)$	$CVTree(k=8)$	$CVTree(k=13)$
mammalian(5x:5MB)	5	7	8	41	4812	3	3
Escherichia/Shigella(1x:300MB)	78	26	42	87	5439	30	528
Drosophila(0.1x:700MB)	147	1514	87	100	4153	63	812

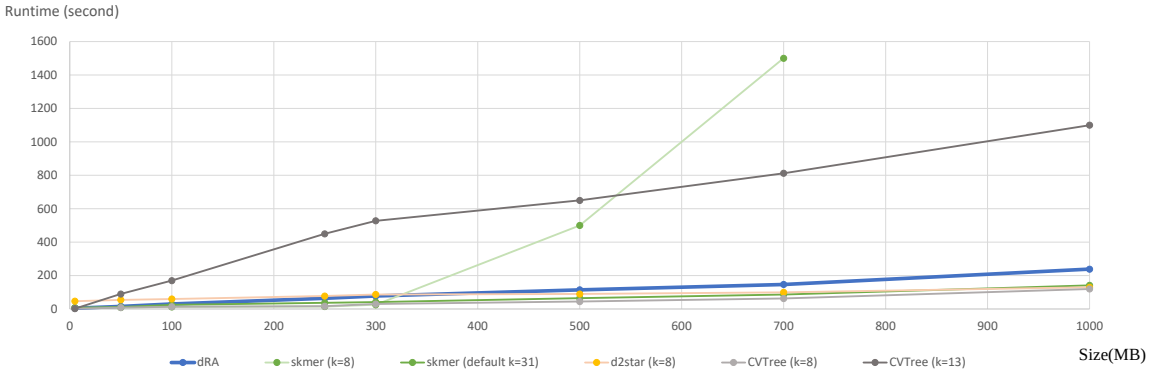


Figure 5.17: The runtime of each method w.r.t. data size

Experimental Result

The result is Fig. 5.19 shows the average RF-distance between the phylogenetic tree constructed from each method and the benchmark tree. d^{RA} can construct the phylogenetic tree, which perfectly the same to the benchmark in the datasets of the sequences with branch length 0.01 and 0.1. However, for the dataset with branch length 0.1, d^{RA} performs worse with some mistake. On the other hand, d^{NS} manages to perform best on the dataset with branch length 0.1. This result shows that d^{RA} is suitable for the dataset of closely related species (short branch length) while d^{NS} is suited for a dataset of diverse species (long branch length). Fig. 5.19 also show the efficiency of both d^{RA} and d^{NS} which out-perform the state-of-art method *skmer*. The parameter k affects the accuracy of *skmer* severely, as shown in Fig. 5.19. In addition, the alignment-based method: *Clustal Omega* is also used this experiment. Since the coverage of the NGS data is equal to 1x. The assembly process might cause the problem

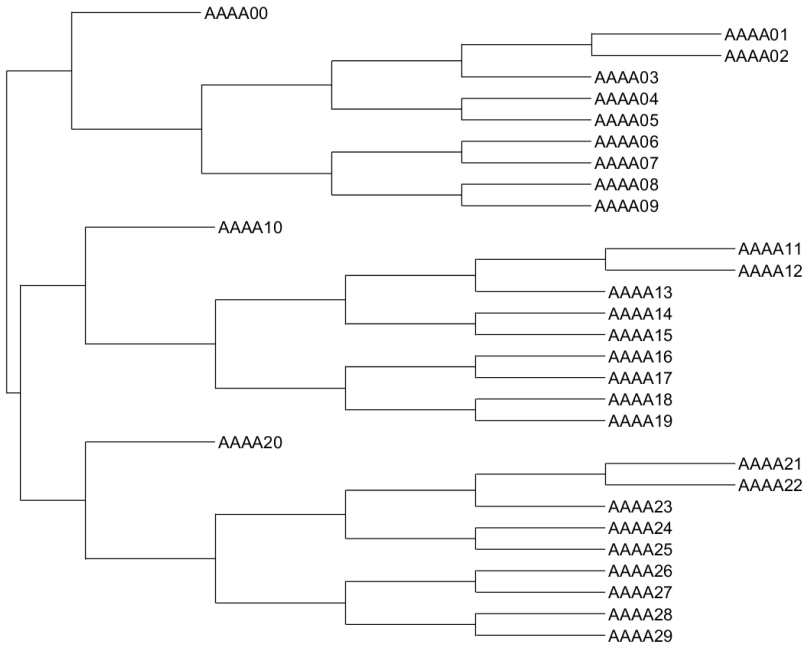


Figure 5.18: Benchmark tree for the simulated datasets

of the alignment-based method. Hence the *Clustal Omega* do not always provide accurate results as shown in Fig. 5.19

In addition, Table 5.7 shows the Branch-Score distance between the phylogenetic tree constructed from each method and the benchmark tree. d^{RA} provide the best score for datasets of the sequences with branch length 0.01 and 0.1 which indicate that the branch length and topology of phylogenetic trees from d^{RA} are closest to the benchmark trees. As same as the RF distance result, d^{NS} provides the best Branch-Score distance for the dataset with branch length 0.1, which represent the dataset of diverse species.

Table 5.7: Branch-Score distance for each method with simulated datasets

Branch length	d^{RA}	$d^{NS}(k=8)$	$d^{NS}(k=13)$	$Skmer(k=13)$	$Skmer(k=31)$	$ClustalOmega$
0.001	**0.00596674**	0.0475123	0.0945824	0.0069331	0.0067413	0.0060657
0.01	**0.01415633**	0.0923781	0.1397226	0.0647134	0.0426146	0.031125
0.1	0.3533541	**0.1964865**	0.2765413	0.4613973	0.4178123	0.462866

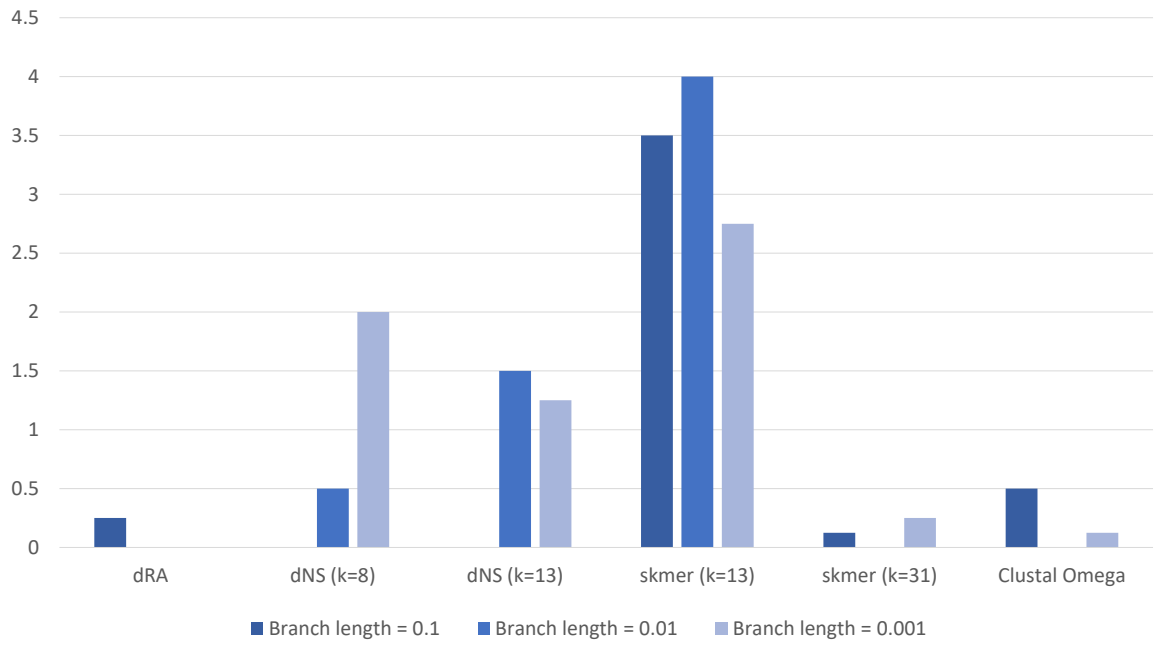


Figure 5.19: RF distance for each method with simulated datasets

5.4 Conclusion

In this chapter, I proposed the k-free approach d^{RA} for NGS data sequence comparison effectively to reconstruct accurate phylogenetic trees and measure the distance between reconstructed trees and benchmark trees. d^{RA} is a novel approach that lies between alignment-based and alignment-free approaches. The d^{RA} distance measurement is based on the collection of alignment between unassembled NGS short reads pairs. While taking advantage of the accuracy aspect of the alignment method, d^{RA} can be performed without an assembly process and can avoid the computational cost associated with assembling and aligning long sequences. The empirical results show that d^{RA} is capable of reconstructing accurate phylogenetic trees without the k parameter even with low coverage data. Although some results obtained at runtime are worse than some other alignment-free methods, there is a fair trade-off with respect to the accuracy without the ambiguous k parameter tuning in the practical use of the method. For the contribution of open software, the link https://github.com/Opalescence/Semi-Alignment-Free-phylogeny provides the GitHub page for the d^{RA} : An effective parameter-free comparison of NGS short reads for phylogeny reconstruction.

Chapter 6

Summary

This thesis presents novel approaches for NGS sequence comparison in the phylogeny reconstruction application.

I propose a novel approach for an alignment-free method d^{NS} that is focused on NGS short-read data and based on neighbor searching. Its main advantage is that it provides an accurate alignment-free sequence comparison method for reconstructing a phylogenetic tree more consistently than other k-mer-based alignment-free methods. Although it might lose significant information in the NGS data when ignoring the k-mer frequencies, the method can specify the distance between NGS sets with reasonable accuracy when using a sufficient number of queries. However, d^{NS} still lacks accuracy when calculating the distance between closely related species. d^{NS} is effective when applying to a dataset of diverse species with high coverage.

Then, I propose a novel sequence comparison approach, namely d^{RA}, which requires no k parameter while maintaining the accuracy of the result. By searching for the corresponded NGS short reads between each set and then calculating the distance from their alignment, this method allows us to calculate the distance with no dependency on k parameter and maintain the same accuracy as the alignment-based approach. Our method also has no requirement for assembly like alignment-free approaches. Because d^{RA} is a k-free approach, it can be applied even on NGS sets without benchmark trees, while other alignment-free approaches have difficulty to adjust the k parameter in such NGS sets. We utilize the Gaussian mixture model to improve the accuracy in the distance measurement of our approach. The experimental results show that d^{RA} can calculate accurate distance even in the dataset of closely related species. With a large dataset, d^{RA} can perform well on low coverage data. However, because d^{RA} is based on alignment between NGS short reads, d^{RA} might not be optimal for a very diverse dataset like the other alignment-based methods.

However, there's an opportunity and also challenges to improve in the future work of the assembly-free alignment-based approach. In the first step, I would consider improving the computational efficiency of the methods and made them more scalable to the larger dataset. One way to achieve this goal is to consider a more efficient search process suitable for the approach. To improve the accuracy of distance measurement, several statistic models could be applied to the approach. Since d^{RA} now considers only the alignment pairs between two NGS sets, hence only pairwise distance is calculated form this approach. One of the solutions as the plan to improve the efficiency of the approach is to calculate distance based-on multiple alignment pairs across all input NGS sets. With this idea framework, the distance from multiple alignment pairs should provide more accurate results compared with pairwise alignment pairs of d^{RA} as the multiple sequence alignment is more accurate than pairwise alignment. Although the idea seems promising, this method also introduces a lot of challenges. Searching for multiple short read alignment across all NGS sets is the first challenge. Unlike the search for the alignment pair of d^{RA}, the multiple short reads alignment pair of all NGS sets can not be simply defined as the same definition as alignment pair of d^{RA} since we need to consider multiple short reads at once. The second challenge is the computational efficiency of the method. We need to find the solution to find the multiple alignment pair within a short amount of time which is a big challenge for all bioinformatic fields of study as well.

In conclusion, two different assemble-free and alignment-free methods are proposed in this thesis. While d^{NS} is effective when applying to the dataset of diverse species with high coverage, d^{RA} is more optimal on the dataset of closely related species with low coverage. These two approaches are proposed to be additional methods and tools for researchers who interest and need to use the phylogeny reconstruction tools. The proposed methods aim to solve the current problem of current phylogeny reconstruction methods which allow researchers to analyze the phylogeny results for their application more efficiently. Since the main problem of using phylogeny reconstruction tools is k parameter tuning, the proposed methods fully contribute in this regard. Finally, I also think that in the future the research on the phylogeny reconstruction and sequence comparison would more focus toward the k parameter-free approach.

Publication list

Journals

- Chotnithi Phanucheep, Hong Van Le, and Atsuhiro Takasu, “An effective parameter-free comparison of NGS short reads for phylogeny reconstruction” Journal of Information Processing, Vol.27, pp. 730 - 741, October 2019.

International conferences

- Chotnithi, Phanucheep, and Atsuhiro Takasu, “Alignment-free Sequence Comparison based on NGS Short-reads Neighbor Search”, The Tenth International Conference on Information, Process, and Knowledge Management (eKNOW2018), pp. 122 - 127,, March 2018.
- Chotnithi, Phanucheep and Atsuhiro Takasu, “Fine-grained k-mer Feature Modification for Alignment-free NGS Sequence Comparison”, Research in Computational Molecular Biology (RECOMB 2018), Paris, France, April 2018. (Abstract)
- Chotnithi, Phanucheep and Atsuhiro Takasu,“Frequent Multi-Byte Character Subtring Extraction using a Succinct Data Structure” The ACM Symposium on Document Engineering (DocEng 2016), pp. 103 - 106, Vienna, Austria, September 2016. (Short Paper)

Domestic conferences

- Chotnithi, Phanucheep and Atsuhiro Takasu, "Frequent Multi-Byte Characters String Mining using Wavelet Tree-based Compressed Suffix Array", The 78th National Convention of IPSJ, Kanagawa, Japan, March 2016.

Bibliography

[1] Atteson, K. (1999). The Performance of Neighbor-Joining Methods of Phylogenetic Reconstruction. *Algorithmica*, 25(2-3):251–278.

[2] Bilmes, J. A. et al. (1998). A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *International Computer Science Institute*, 4(510):126.

[3] Broder, A. Z., Charikar, M., Frieze, A. M., and Mitzenmacher, M. (2000). Min-Wise Independent Permutations. *Journal of Computer and System Sciences*, 60(3):630–659.

[4] Cao, Y., Janke, A., Waddell, P. J., Westerman, M., Takenaka, O., Murata, S., Okada, N., Pääbo, S., and Hasegawa, M. (1998). Conflict Among Individual Mitochondrial Proteins in Resolving the Phylogeny of Eutherian Orders. *Journal of Molecular Evolution*, 47(3):307–322.

[5] Chan, C. X. and Ragan, M. A. (2013). Next-Generation Phylogenomics. *Biology Direct*, 8(1):3.

[6] Chotnithi, P. and Takasu, A. (2018). Fine-grained k-mer feature modification for alignment-free NGS sequence comparison (poster). *Research in Computational Molecular Biology, Paris, France*.

[7] De Bruyn, A., Martin, D. P., and Lefeuvre, P. (2014). Phylogenetic Reconstruction Methods: An Overview. In *Molecular Plant Taxonomy*, pages 257–277. Springer.

[8] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

[9] Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

[10] Edgar, R. C. (2004). MUSCLE: Multiple Sequence Alignment with High Accuracy and High Throughput. *Nucleic Acids Research*, 32(5):1792–1797.

[11] Edgar, R. C. (2010). Search and Clustering Orders of Magnitude Faster than BLAST. *Bioinformatics*, 26(19):2460–2461.

[12] Everitt, B. (1998). The Cambridge Dictionary of Statistics Cambridge University Press. *Cambridge, UK*.

[13] Fan, H., Ives, A. R., Surget-Groba, Y., and Cannon, C. H. (2015). An Assembly and Alignment-free Method of Phylogeny Reconstruction from Next-generation Sequencing Data. *BMC Genomics*, 16(1):522.

[14] Felsenstein, J. (1981). Evolutionary trees from DNA Sequences: A Maximum Likelihood Approach. *Journal of Molecular Evolution*, 17(6):368–376.

[15] Felsenstein, J. (1993). *PHYLIP (Phylogeny Inference Package), Version 3.5 c*. Joseph Felsenstein.

[16] Fitch, W. M. (1971). Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Biology*, 20(4):406–416.

[17] Gionis, A., Indyk, P., Motwani, R., et al. (1999). Similarity Search in High Dimensions via Hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, volume 99, pages 518–529.

[18] Hartigan, J. A. (1973). Minimum Mutation Fits to a Given Tree. *Biometrics*, pages 53–65.

[19] Hasegawa, M., Kishino, H., and Yano, T.-a. (1985). Dating of the Human-ape Splitting by A Molecular Clock of Mitochondrial DNA. *Journal of Molecular Evolution*, 22(2):160–174.

[20] Higgins, D. G. and Sharp, P. M. (1988). CLUSTAL: A Package for Performing Multiple Sequence Alignment on a Microcomputer. *Gene*, 73(1):237–244.

[21] Hinchliff, C. E., Smith, S. A., Allman, J. F., Burleigh, J. G., Chaudhary, R., Coghill, L. M., Crandall, K. A., Deng, J., Drew, B. T., Gazis, R., Gude, K., Hibbett, D. S., Katz, L. A., Laughinghouse, H. D., McTavish, E. J., Midford, P. E., Owen, C. L., Ree, R. H., Rees, J. A., Soltis, D. E., Williams, T., and Cranston, K. A. (2015). Synthesis of Phylogeny and Taxonomy into A Comprehensive Tree of Life. *Proceedings of the National Academy of Sciences*, 112(41):12764–12769.

[22] Huang, W., Li, L., Myers, J. R., and Marth, G. T. (2011). ART: A Next-generation Sequencing Read Simulator. *Bioinformatics*, 28(4):593–594.

[23] Jukes, T. H., Cantor, C. R., et al. (1969). Evolution of Protein Molecules. *Mammalian Protein Metabolism*, 3(21):132.

[24] Katal, A., Wazid, M., and Goudar, R. (2013). Big data: Issues, Challenges, Cools and Good Practices. In *2013 Sixth International Conference on Contemporary Computing (IC3)*, pages 404–409. IEEE.

[25] Katoh, K., Misawa, K., Kuma, K.-i., and Miyata, T. (2002). MAFFT: A Novel Method for Rapid Multiple Sequence Alignment based on Fast Fourier Transform. *Nucleic acids research*, 30(14):3059–3066.

[26] Kimura, M. (1980). A Simple Method for Estimating Evolutionary Rates of Base Substitutions Through Comparative Studies of Nucleotide Sequences. *Journal of Molecular Evolution*, 16(2):111–120.

[27] Kuhner, M. K. and Felsenstein, J. (1994). A Simulation Comparison of Phylogeny Algorithms under Equal and Unequal Evolutionary Rates. *Molecular Biology and Evolution*, 11(3):459–468.

[28] Leimeister, C.-A., Boden, M., Horwege, S., Lindner, S., and Morgenstern, B. (2014). Fast Alignment-Free Sequence Comparison using Spaced-Word Frequencies. *Bioinformatics*, 30(14):1991–1999.

[29] Leimeister, C.-A. and Morgenstern, B. (2014). Kmacs: The K-Mismatch Average Common Substring Approach to Alignment-Free Sequence Comparison. *Bioinformatics*, 30(14):2000–2008.

[30] Metzker, M. L. (2010). Sequencing Technologies–the Next Generation. *Nature Reviews. Genetics*, 11(1):31.

[31] Miyazawa, S. (1995). A Reliable Sequence Alignment Method Based on Probabilities of Residue Correspondences. *Protein Engineering, Design and Selection*, 8(10):999–1009.

[32] Morrison, D. A. (1996). Phylogenetic Tree-building. *International Journal for Parasitology*, 26(6):589–617.

[33] Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment. *Journal of Molecular Biology*, 302(1):205–217.

[34] Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., and Phillippy, A. M. (2016). Mash: Fast Genome and Metagenome Distance Estimation using Minhash. *Genome Biology*, 17(1):132.

[35] Otu, H. H. and Sayood, K. (2003). A New Sequence Distance Measure for Phylogenetic Tree Construction. *Bioinformatics*, 19(16):2122–2130.

[36] Phillips, A., Janies, D., and Wheeler, W. (2000). Multiple Sequence Alignment in Phylogenetic Analysis. *Molecular Phylogenetics and Evolution*, 16(3):317–330.

[37] Qi, J., Luo, H., and Hao, B. (2004). CVTree: a Phylogenetic Tree Reconstruction Tool Based on Whole Genomes. *Nucleic Acids Research*, 32(suppl_2):W45–W47.

[38] Rambaut, A. and Grass, N. C. (1997). Seq-Gen: An Application for the Monte Carlo Simulation of DNA Sequence Evolution along Phylogenetic Trees. *Bioinformatics*, 13(3):235–238.

[39] Richter, D. C., Ott, F., Auch, A. F., Schmid, R., and Huson, D. H. (2011). MetaSim: A Sequencing Simulator for Genomics and Metagenomics. *Handbook of Molecular Microbial Ecology I: Metagenomics and Complementary Approaches*, pages 417–421.

[40] Robinson, D. F. and Foulds, L. R. (1981). Comparison of Phylogenetic Trees. *Mathematical Biosciences*, 53(1-2):131–147.

[41] Saitou, N. and Nei, M. (1987). The Neighbor-Joining Method: a New Method for Reconstructing Phylogenetic Trees. *Molecular Biology and Evolution*, 4(4):406–425.

[42] Sarmashghi, S., Bohmann, K., Gilbert, M. T. P., Bafna, V., and Mirarab, S. (2019). Skmer: Assembly-Free and Alignment-Free Sample Identification Using Genome Skims. *Genome Biology*, 20(1):34.

[43] Sessegolo, C., Burlet, N., and Haudry, A. (2016). Strong Phylogenetic Inertia on Genome Size and Transposable Element Content Among 26 Species of Flies. *Biology Letters*, 12(8).

[44] Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., et al. (2011). Fast, Scalable Generation of High-Quality Protein Multiple Sequence Alignments Using Clustal Omega. *Molecular Systems Biology*, 7(1):539.

[45] Sneath, P. H. (1957). The Application of Computers to Taxonomy. *Microbiology*, 17(1):201–226.

[46] Sokal, R. R. (1958). A Statistical Method for Evaluating Systematic Relationships. *University of Kansas Science Bulletin*, 28:1409–1438.

[47] Song, K., Ren, J., Zhai, Z., Liu, X., Deng, M., and Sun, F. (2013). Alignment-Free Sequence Comparison Based on Next-Generation Sequencing Reads. *Journal of Computational Biology*, 20(2):64–79.

[48] Stuart, G. W., Moffett, K., and Leader, J. J. (2002). A Comprehensive Vertebrate Phylogeny using Vector Representations of Protein Sequences from Whole Genomes. *Molecular Biology and Evolution*, 19(4):554–562.

[49] Tavaré, S. (1986). Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences. *Lectures on Mathematics in the Life Sciences*, 17(2):57–86.

[50] Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). 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):4673–4680.

[51] Tran, N. H. and Chen, X. (2014). Comparison of Next-generation Sequencing Samples using Compression-based Distances and its Application to Phylogenetic Reconstruction. *BMC Research Notes*, 7(1):320.

[52] Ulitsky, I., Burstein, D., Tuller, T., and Chor, B. (2006). The Average Common Substring Approach to Phylogenomic Reconstruction. *Journal of Computational Biology*, 13(2):336–350.

[53] Vinga, S. and Almeida, J. (2003). Alignment-Free Sequence Comparison—a Review. *Bioinformatics*, 19(4):513–523.

[54] Wang, A. and Ash, G. J. (2015). Whole Genome Phylogeny of Bacillus by Feature Frequency Profiles (FFP). *Scientific Reports*, 5:13644.

[55] Waterman, M. S. (1995). *Introduction to Computational Biology: Maps, Sequences and Genomes*. CRC Press.

[56] Xu, Z. and Hao, B. (2009). CVTree Update: a Newly Designed Phylogenetic Study Platform Using Composition Vectors and Whole Genomes. *Nucleic Acids Research*, 37(suppl_2):W174–W178.

[57] Zhou, Z., Li, X., Liu, B., Beutin, L., Xu, J., Ren, Y., Feng, L., Lan, R., Reeves, P. R., and Wang, L. (2010). Derivation of Escherichia Coli O157: H7 from its O55: H7 Precursor. *PloS One*, 5(1):e8700.

Appendix A

Datasets

There are tree dataset used in this thesis, *mammalian mtDNA* sequences,*Escherichia/Shigella*, and 18 *Drosophila* genomes. All of the sequence for each species can be searched in GenBank by using accession numbers provided in the Table.

The GenBank database is designed to provide and encourage access within the scientific community to the most up-to-date and comprehensive DNA sequence information. Therefore, NCBI places no restrictions on the use or distribution of the GenBank data. However, some submitters may claim patent, copyright, or other intellectual property rights in all or a portion of the data they have submitted. NCBI is not in a position to assess the validity of such claims, and therefore cannot provide comment or unrestricted permission concerning the use, copying, or distribution of the information contained in GenBank.

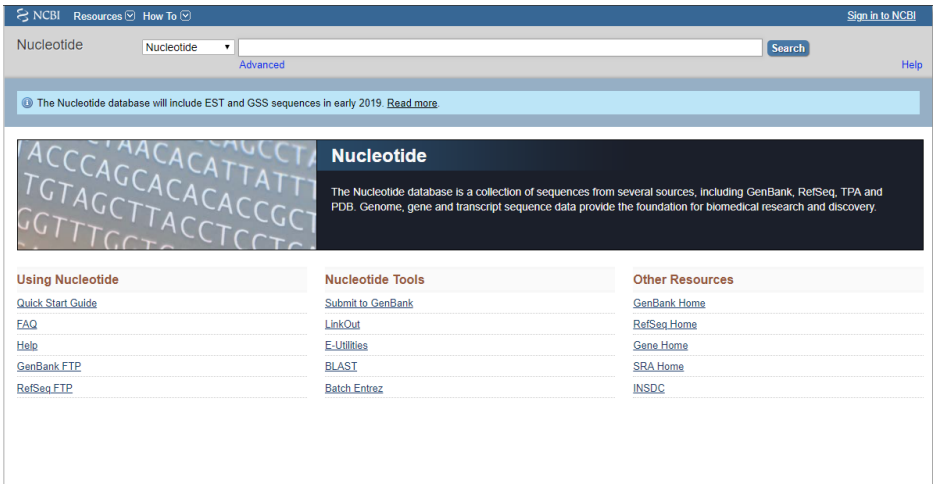


Figure A.1: GenBank database

Table A.1: GenBank accession numbers *mammalian mtDNA* sequences

Species	GenBank accession
E. coli O157	AE005174
S. flexneri 2a str. 301	AE005674
S. flexneri 2a str. 2457T	AE014073
(E. coli Nissle 1917	AE014075
E. coli str. K-12 substr. W3110	AP009048
E. coli SE11	AP009240
E. coli O157:H7 str. Sakai	BA000007
S. dysenteriae Sd197	CP000034
S. boydii Sb227	CP000036
S. sonnei Ss046	CP000038
E. coli UTI89	CP000243
E. coli 536	CP000247
S. flexneri 5 str. 8401	CP000266
E. coli APEC O1	CP000468
E. coli E24377A	CP000800
E. coli HS	CP000802
E. coli ATCC 8739	CP000946
E. coli str. K12 substr. DH10B	CP000948
E. coli SMS-3-5	CP000970
S. boydii CDC 3083-94	CP001063
E. coli BW2952	CP001396
E. coli O55:H7 str. CB9615	CP001846
E. coli IAI1	CU928160
E. coli S88	CU928161
E. coli ED1a	CU928162
E. coli UMN026	CU928163
E. coli IAI39	CU928164
E. coli 0127:H6 E2348/69	FM180568
E. coli str. K-12 substr. MG1655	U00096

Table A.2: GenBank accession numbers *Escherichia/Shigella* genomes

Species	GenBank accession
(Homo sapiens (Human)	V00662
Pan troglodytes (Common chimpanzee)	D38116
(Pan paniscus (Pigmy chimpanzee)	D38113
Gorilla gorilla (Gorilla)	D38114
Pongo pygmaeus (Orangutan)	D38115
Hylobates lar (Gibbon)	X99256
Papio hamadryas (Baboon)	Y18001
Equus caballus (Horse)	X79547
Ceratotherium simum (White rhinoceros)	Y07726
Phoca vitulina (Harbor seal)	X63726
Halichoerus grypus (Gray seal)	X72004
Felis catus (Cat)	U20753
Balenoptera physalus (Fin whale)	X61145
Balenoptera musculus (Blue whale)	X72204
Bos taurus (Cow)	V00654
Rattus norvegicus (Rat)	X14848
Mus musculus (Mouse)	V00711
Didelphis virginiana (Opossum)	Z29573
Macropus robustus (Wallaroo)	Y10524
Ornithorhyncus anatinus (Platypus)	X83427
Sciurus vulgaris (Squirrel)	AJ238588
Glis glis (Fat dormouse)	AJ001562
Cavia porcellus (Guinea pig)	AJ222767
Equus asinus (Donkey)	X97337
Rhinoceros unicornis (Indian rhinoceros)	X97336
Canis familiaris (Dog)	U96639
Ovis aries (Sheepsheep)	AF010406
Sus scrofa (Pig)	AJ002189
Hippopotamus amphibius (Hippopotamus)	AJ010957

Table A.3: GenBank accession numbers and URLs for *Drosophila* genomes

Species	GenBank assembly accession	URL
Drosophila biarmipes	GCA_000233415.2	http://www.insect-genome.com/data/genome_download/Drosophila_biarmipes/Drosophila_biarmipes_genomic.fasta.gz
Drosophila bipectinata	GCA_000236285.2	http://www.insect-genome.com/data/genome_download/Drosophila_bipectinata/Drosophila_bipectinata_genomic.fasta.gz
Drosophila elegans	GCA_000224195.2	http://www.insect-genome.com/data/genome_download/Drosophila_elegans/Drosophila_elegans_genomic.fasta.gz
Drosophila erecta	GCA_000005135.1	http://www.insect-genome.com/data/genome_download/Drosophila_erecta/Drosophila_erecta_genomic.fasta.gz
Drosophila eugracilis	GCA_000236325.2	http://www.insect-genome.com/data/genome_download/Drosophila_eugracilis/Drosophila_eugracilis_genomic.fasta.gz
Drosophila ficusphila	GCA_000220665.2	http://www.insect-genome.com/data/genome_download/Drosophila_ficusphila/Drosophila_ficusphila_genomic.fasta.gz
Drosophila grimshawi	GCA_000005155.1	http://www.insect-genome.com/data/genome_download/Drosophila_grimshawi/Drosophila_grimshawi_genomic.fasta.gz
Drosophila kikkawai	GCA_000224215.2	http://www.insect-genome.com/data/genome_download/Drosophila_kikkawai/Drosophila_kikkawai_genomic.fasta.gz
Drosophila melanogaster	GCA_000778455.1	http://www.insect-genome.com/data/genome_download/Drosophila_melanogaster/Drosophila_melanogaster_genomic.fasta.gz
Drosophila miranda	GCA_000269505.2	http://www.insect-genome.com/data/genome_download/Drosophila_miranda/Drosophila_miranda_genomic.fasta.gz
Drosophila persimilis	GCA_000005195.1	http://www.insect-genome.com/data/genome_download/Drosophila_persimilis/Drosophila_persimilis_genomic.fasta.gz
Drosophila rhopaloa	GCA_000236305.2	http://www.insect-genome.com/data/genome_download/Drosophila_rhopaloa/Drosophila_rhopaloa_genomic.fasta.gz
Drosophila sechellia	GCA_000005215.1	http://www.insect-genome.com/data/genome_download/Drosophila_sechellia/Drosophila_sechellia_genomic.fasta.gz
Drosophila simulans	GCA_000259055.1	http://www.insect-genome.com/data/genome_download/Drosophila_simulans/Drosophila_simulans_genomic.fasta.gz
Drosophila suzukii	GCA_000472105.1	http://www.insect-genome.com/data/genome_download/Drosophila_suzukii/Drosophila_suzukii_genomic.fasta.gz
Drosophila takahashii	GCA_000224235.2	http://www.insect-genome.com/data/genome_download/Drosophila_takahashii/Drosophila_takahashii_genomic.fasta.gz
Drosophila virilis	GCA_000005245.1	http://www.insect-genome.com/data/genome_download/Drosophila_virilis/Drosophila_virilis_genomic.fasta.gz
Drosophila yakuba	GCA_000005975.1	http://www.insect-genome.com/data/genome_download/Drosophila_yakuba/Drosophila_yakuba_genomic.fasta.gz