3-Way Alignment can Improve Multiple Sequence Alignment of Highly Diverged Sequences

Mahbubeh Askari Rad

A thesis submitted

in partial fulfilment of the requirements for the

Master's degree in Biology

Supervised by

Dr. Xuhua Xia

Department of Biology
Faculty of Science
University of Ottawa

Abstract

The standard approach for constructing a phylogenetic tree from a set of sequences consists of two key stages. First, Multiple Sequence Alignment (MSA) of the sequences is computed. The aligned data is then used to reconstruct the phylogenetic tree. The accuracy of the final tree heavily relies on the quality of MSA. Consequently, obtaining an accurate MSA is critical for improving the accuracy of the phylogenetic information. The accuracy of MSA decreases with sequence divergence. Numerous studies have demonstrated the difficulty in obtaining good MSA for reconstructing deep phylogeny, e.g. arthropod phylogeny. The accuracy of MSA is universally recognized as the limiting factor for tracing evolutionary history back in time.

In this thesis, we assess the effect of guide tree on the accuracy of Multiple Sequence Alignment and subsequent phylogenetic tree and apply 3-way alignment to generate more accurate guide trees. We evaluate the performance of MAFFT on simulated dataset with different level of divergency including highly diverged sequences.

Our results show that 3-way alignment outperforms the MAFFT's default method in generating more accurate guide trees specially for highly diverged sequences. Moreover, MSAs generated by using 3-way alignment guide trees lead to more accurate subsequent phylogenetic trees compared to MSAs generated by MAFFT default method. In addition, we showed the significant role of a good guide tree in improving final phylogenetic trees. Finally, our results emphasis on the significantly low performance of MAFFT in aligning highly diverged sequences.

The thesis has been published in the journal Algorithms:

Askari Rad, M.; Kruglikov, A.; Xia, X. Three-way Alignment Improves Multiple Sequence Alignment of Highly Diverged Sequences. *Algorithms* **2024**, *17*, *205*.
https://doi.org/10.3390/a17050205

Acknowledgement

I express my sincere respect and gratitude to my supervisor Dr. Xuhua Xia for his invaluable support, cooperation, and inspiration. His feedback and guidance helped me to progress and guided me towards the right path.

I thank my advisory committee Dr. Marcel Turcotte, Dr. Stephane Aris-Brosou and Dr. Arvind Mer for their guidance in developing this thesis. I thank University of Ottawa and NSERC for their generous funding.

Table of Contents

1 Introduction 1

1.1 Sequence Alignment 1

1.2 Evolutionary significance 2

1.3 Problem Statement 3

2 Multiple Sequence Alignment 7

2.1 Scoring system 7

2.1.1 Scoring matrix 8

2.1.2 Gap penalty 9

2.2 Scoring Profile 11

2.2.1 Sum of pair score 11

2.2.2 Column score 12

2.3 Dynamic programming approaches 12

2.4 Progressive Alignment 14

2.5 Iterative refinement 16

3 Background 17

3.1 Progressive alignment techniques 17

3.1.1 CLUSTAL-W 17

3.1.2 T-COFFEE 18

3.1.3 MAFFT 18

3.1.4 MUSCLE 19

3.1.5 PROBCONS 20

3.1.6 PROGALIGN 20

3.1.7 Non-progressive alignment methods 20

3.2 The role of guide tree in progressive approach 21

3.3 Summary 22

4 Methodology 24

4.1 Carrillo-Lipman algorithm for three sequences 24

4.2 3-Way Alignment 27

4.3 Formulation for 3-way alignment using affine gap penalty 30

4.4 Trace back procedure 33

4.5 Reducing time and space complexity 34

4.6 Algorithm 35

4.7 Simulated dataset 37

4.8 Measuring distance matrix and constructing guide tree. 39
4.9 Aligning sequences with MAFFT 39
4.10 Comparing the accuracy of phylogenetic trees 40
5 Result 41
5.1 The 3-way alignment tends generate guide trees closer to the true tree than other approaches....41
5.2 The 3-way alignment leads to more accurate phylogenetic results than other approaches....43
5.3 Accuracy of the guide tree affects the accuracy of the final tree from MSA 46
5.4 The sum-of-pair score may not be a good criterion for choosing the best MSA....48
5.5 Performance of 3-way alignment on Benchmark dataset 52
6 Discussion 53
6.1 Is the true tree the best guide tree for the progressive multiple sequence alignment? 53
6.2 How to obtain the best guide tree? 54
6.3 Is sum-of-pair score or its derivative a good criterion for choosing the best MSA? 56
7 Conclusion....58

Figure 1: A guide tree of 5 seqeunces..... 15

Figure 2: An example of 3-way alignment with constant gap penalty 27

Figure 3: Flowchart of simulation..... 36

Figure 4: The 16-taxa trees used for simulating sequences.. 38

Figure 5: 8-taxa trees used for simulating sequences. 38

Figure 6: The relationship between RFd of guide trees and final phylogenetic trees for 8-taxa trees..... 47

Figure 7: The relationship between RFd of guide trees and final phylogenetic tree for 16-taxa trees..... 48

Figure 8: Conflict between SP-score and phylogenetic tree for sequences with length 500 50

Figure 9: Conflict between SP-score and phylogenetic tree for sequences with length 1500 51

Table 1: The result of comparing guide trees generated by 3-way alignment and L-INS-i for seqeunces generated on 8-taxa tree 41

Table 2: Quality of guide trees generated by three MAFFT options (FFT-NS-1, FFT-NS-2, L-INS-i) and by the 3-way alignment (3-WAY) for seqeunces simulated on 16-taxa 43

Table 3: result of comparing the reconstructed phylogenetic trees by PhyML using MSA geberated by L-INS-i and MAFFT using 3-way alignment 44

Table 4: : result of comparing the reconstructed phylogenetic trees by PhyML using MSA generated by FFT-NS-1, FFT-NS-2, L-INS-i, MAFFT using 3-way alignment 45

1 Introduction

1.1 Sequence Alignment

Aligning sequences refers to arrangement of two or more biological sequences (such as protein, DNA or RNA sequences) in the manner of maximizing similarity between them. The aim of sequence alignment is to determine homologous sites among sequences, facilitating a deeper understanding of their structural, functional characteristics and evolutionary relationships [1,2]. An optimal alignment provides insights into conserved regions, mutations, insertion, and deletion, thereby aiding in understanding the evolutionary history and functional implications of the sequences.

The existing alignment methods are primarily designed for either Pairwise Sequence Alignment (PSA) or Multiple Sequence Alignment (MSA). Pairwise sequence alignment specifically focuses on aligning two sequences which is computationally feasible and efficient even for long sequences [3]. On the other hand, multiple sequence alignment involves three or more sequences introducing a higher level of computational complexity, and it requires more computational time and memory resources compared to PSA [2].

There exist two primary alignment categories: global alignment, which, encompasses the full length of sequences in an end-to-end manner, and local alignment, which focuses on identifying the regions of sequences with the highest similarity. Local alignment is especially useful when sequences have evolved differently but retain certain conserved

domains or motifs [4]. Focussing on these conserved regions, local alignment can provide insights into shared functional or structural elements, even in sequences that might appear largely dissimilar at first.

1.2 Evolutionary significance

Inferring phylogenetic relationship among various species from molecular data, such as protein, DNA and RNA, is a fundamental problem in evolutionary biology. Multiple Sequence Alignment is a prerequisite step in phylogenetic reconstruction, and the accuracy of MSA is a bottleneck in deep phylogenetic reconstruction. Errors in sequence alignment can lead to significant inaccuracies in the molecular evolution [5]. There are two primary categories for reconstructing phylogenetic tree. The first and most widely used method is to use Multiple Sequence Alignment (MSA) as input and subsequently reconstruct the phylogenetic tree. In this approach, MSA is a critical step and the accuracy of the final tree is highly dependent on the quality of the MSA, and a poor sequence alignment can lead to inaccurate phylogenetic inference [6–8]. The second category of phylogenetic reconstruction approaches infers both the MSA and the tree structure simultaneously. This group of methods are less popular than the first one because they are typically limited to a smaller number of taxa and tend to be less accurate.

1.3 Problem Statement

Current sequence alignment methods and tools work well on the closely related sequences and align these sequences with high accuracy [9]. However, dealing with highly diverged sequences, the performance of these algorithm decreases significantly, and alignment of distantly related sequences is still a challenge [10–12]. Virtually everyone interested in deep phylogeny is looking for methods that can improve MSA. Some have incorporated secondary structure to guide the sequence alignment [13–16], while others explored post-alignment refinement [5,12]. The improvement of MSA with these approaches remain limited.

Highly diverged sequences have experienced higher number of insertion and deletions compared to closely related sequences. Another challenge in aligning diverged sequences is substitution saturation. Substitution saturation of molecular sequences, such as DNA, occurs when so many mutations have accumulated over time and it becomes difficult or impossible to trace the sequence's evolutionary history [17].

Substitution saturation introduce significant challenges to MSA by obscuring homologous positions in sequences, making it difficult to accurately identify evolutionary relationships. This loss of evolutionary signal leads to a higher risk of misalignment, as it becomes challenging to distinguish between similarities due to common ancestry and those resulting from multiple mutations [18]. Consequently, this

can lead to errors in phylogenetic trees, reflecting inaccurately on the evolutionary history of the species involved.

Another factor that makes alignment of distantly related sequence more challenging compared to closely related sequences is that in diverged sequences gaps and mismatches dominate matches. Therefore, different values for these two dominant events leads to different answers. Scoring scheme including gap penalty and match-mismatch matrix (also named scoring matrix) has a vital role in sequence alignment, and different scoring schemes may result in different alignments. In closely related sequences, matches are dominant and frequently occur in the alignment. Since all different scoring matrices assign high value to matches, using different scoring matrix approximately result in correct solution. This fact is also true about gap penalty, and a wide range of gap penalty may result in true solution for alignment of closely related sequences. Most of available MSA algorithms use one specific gap penalty and scoring matrix for aligning all sequences regardless of their similarity and evolutionary distance. Considering these challenges in alignment of distantly related sequences, alignment methods need to increase their accuracy.

Progressive approach is the most popular algorithm implemented in many MSA tools. The method involves constructing a guide tree to determine the order of sequence alignment and starting with the most similar pairs and progressively adding in the more divergent sequences [19].

Current MSA methods and tools perform well on aligning closely related sequences [9]. However, the performance of these methods decreases with sequence divergence [10–12]. Virtually everyone interested in deep phylogeny is looking for ways to improve MSA. Some have incorporated secondary structure to guide the sequence alignment [13–16], while others explored post-alignment refinement [5,12]. The improvement of MSA with these approaches remains limited.

A guide tree is a crucial component in the progressive alignment, and its accuracy affects the accuracy of output alignment [20]. A few studies have shown that an inaccurate guide tree could be a major source of error in progressive sequence alignment [21,22]. Therefore, different strategies for improving guide trees have been proposed [20]. Two criteria have been used to evaluate the effect of guide trees on the accuracy of MSA generated by MAFFT and ClustalW: 1) the sum-of-pair score (SPS) excluding shared gaps in pairwise comparisons, and 2) the accuracy of phylogenetic reconstruction [23,24]. The result indicates that the final SPS is little affected by the initial guide tree, but better guide trees significantly improve the accuracy of the reconstructed phylogenies.

Constructing an accurate guide tree is difficult for highly diverged sequences. The construction of the guide tree is usually based on pairwise comparisons, and the quality of these initial comparisons is crucial because errors at this stage can be propagated during the entire alignment process. A few studies have shown that an inaccurate guide tree could be a major source of error in progressive sequence alignment [21,22]. Aligning

three sequences by dynamic programming is expected to improve the alignment and, in particular, the estimated distances used to build the initial guide tree [25–28].

In this study, we aim to improve the accuracy of guide tree specially for highly diverged sequences by using 3-way alignment. 3-way alignment helps us to calculate the evolutionary distance between sequences more accurately compared to pairwise sequence alignment. We assess the performance of MAFFT [29] which is the most popular MSA tool using our generated guide trees based on 3-way alignment.

2 Multiple Sequence Alignment

Multiple Sequence Alignment (MSA) can be defined as an optimization problem that the optimal solution is one with the highest score based on a scoring system. In this section, we explain the important components of MSA, the scoring system and dynamic programing approach for aligning sequences.

2.1 Scoring system

In sequence alignment, a scoring system is employed to quantify the quality of alignments and to help find the best possible alignment of sequences. This system assigns scores to different types of matches, mismatches, and gaps in the aligned sequences. For example, a high score is typically given to identical or similar residues to promote matches, while mismatches and gaps are penalized with lower or negative scores. The choice of scoring system can greatly influence the outcome of alignment. These scores facilitate objective and quantitative comparisons between different possible alignments, allowing algorithms to choose the alignment with the highest overall score, which is interpreted as the most biologically relevant or likely alignment under the given scoring scheme.

Scoring system contains two components of scoring matrix and gap penalty. In the following sections, more details are provided for them.

2.1.1 Scoring matrix

Scoring matrix assigns scores to pair of sequence elements, such as amino acids or nucleotides, based on their likelihood of substitution over time. In sequence alignment, a scoring matrix is essential as it is used to align sequences and provides a quantitative measure to evaluate the quality of an alignment. By assigning scores to matches and mismatches, the matrix helps in determining the most biologically or evolutionarily likely alignment between two sequences.

BLOSUM [30] and PAM [31] are two popular scoring matrices. The BLOSUM (Blocks Substitution Matrix) series of matrices are generated based on empirical data derived from observed substitutions in blocks of local alignments of protein sequences. Initially, blocks of aligned sequences are collected, each block consists of sequences with a specific level of sequence identity. Each BLOSUM matrix is designed to comparing sequences with a certain level of similarity. For example, BLOSUM62 is based on blocks where sequences are at most 62% identical, and this matrix is used for aligning sequences with moderate level of identity [30]. Within these blocks, the frequency of each amino acid substitution is counted. These frequencies are then used to calculate odds ratios of observed frequency of substitutions to the expected frequency under random conditions. These ratios are subsequently converted into log-odds scores usually scaled and rounded for practical use. The scores are then arranged into a matrix, with each cell representing a score for substituting one amino acid with another one [32].

PAM (Point Accepted Mutation) matrices are generated for protein sequence alignment by analyzing closely related proteins, typically sharing over 85% sequence identity [31]. The process involves collecting and aligning these homologous sequences, and then identify single amino acid substitutions that occurred during evolution. These substitutions are counted to calculate the probabilities of each amino acid being replaced by another. PAM matrices such as PAM1, are normalized to represent mutation probabilities over specific evolutionary intervals. These probabilities are then transformed into log-odds scores [18].

2.1.2 Gap penalty

A gap penalty is a value assigned in sequence alignment algorithms to penalize the gaps (insertion/deletion) in the alignment of two sequences. The purpose of the gap penalty is to discourage the creation of gaps unless they are biologically or evolutionary justified. Dealing with insertion and deletion that create gap in alignment is more challenging compared to substitution. Alignment methods usually do not differentiate between insertion and deletion and use indel for referring both. The mostly used gap penalty function is affine gap penalty (AGP) which is a linear function of constant values for gap opening (GO) penalty and gap extension (GE) penalty, in the form of GO+GE*L (L is the length of gap) [33]. The advantage of AGP is simplicity to use, but it requires the values of GO and GE to be constant in alignment process. A few other gap penalty models have

been proposed for improving the alignment quality, such as non-local gap penalty[34], logarithmic affine gap penalty[35] and "long indel" model[36].

Although, evidence shows that some residues are preferred in gap regions, common gap penalty models consider a uniform gap penalty for entire length of sequence and different residues. Dealing with this issue some methods like CLUSTAL W and MAFFT define a residue- and position-specific gap penalty.

Some algorithms use structural information of proteins in placement of gaps[37,38]. However, these methods are not practical in all cases of alignment because the structure of many proteins is not known yet.

Since there is little theoretical basis for the form of gap penalty, the optimal placement of gaps is determined empirically. For the first time, [39] proposed a variable gap penalty based on protein secondary structure instead of using a constant gap penalty for entire sequence. It considered a higher penalty for gaps within helical regions. Furthermore, by pairwise structural comparison of proteins with known 3D structures, [40] showed gaps are rare within secondary structural elements α-helices and β-strands, and it occurs often between them,. Another work [41] performed a more comprehensive study on FSSB database and showed gaps occur more frequently in residues with small side chain, in contrast to residues with hydrophobic side chains.

Gap penalty function plays a significant role to achieve the correct alignment, especially for highly diverged sequences.

2.2 Scoring Profile

In order to evaluate the accuracy and significance of aligning two or more sequences, it is necessary to assign a score to the resulting profile. The score is used to evaluate the quality of the alignment, allowing the comparison of different alignments of the same sequences, and determining which alignment is better. For this purpose, many scoring methods have been developed, each with its own advantages and disadvantages, which will be explored in the subsequent discussion.

2.2.1 Sum of pair score

The sum-of-pair score (SP score) can be applied to both PSA and MSA for measuring the score of alignment. In MSA, it easily defines as the summation of all pairwise alignments. For N sequences we have $N(N-2)/2$ pair of sequences and SPS defines as the following [29]:

$$SP = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} S(s_i, s_j)$$

where $S(s_i, s_j)$ is the score of aligning the sequences s_i and s_j. These scores are calculated based on a scoring matrix and gap penalty values.

2.2.2 Column score

Columns Score is another criterion for comparing two alignments, which defined as the proportion of perfectly aligned columns to the number of columns [29].

$$Column\ score = \frac{\#\ of\ perfectly\ aligned\ columns}{\#\ of\ columns}$$

The column score thus provides the percentage of how many columns are perfectly aligned in a MSA. This score is a measure of alignment's overall quality, with higher values indicating better alignment.

2.3 Dynamic programming approaches

The most common method for global alignment of two sequences (Pairwise Sequence Alignment or PSA) is Needleman-Wunsch algorithm[42] which is a dynamic programing (DP) approach and produce one of equally optimal solutions based on given scoring scheme (scoring matrix and gap-penalty). Dynamic programing is an optimization and problem-solving technique widely used in bioinformatics, which divides a complex problem into simpler subproblems [43].

The problem of achieving optimal solution for PSA is feasible in quadratic runtime and memory, which can be reduced to be a linear in space [44]. The method of using DP for sequence alignment can be easily extended to MSA. Unfortunately, solving Multiple Sequence Alignment (MSA) using multiple dimensional DP is practically infeasible even for small number of sequences due to time and space complexity [45]. As the number of

sequences grows, both runtime and space requirement for this extension increase exponentially. Typically, to determine the optimal alignment of n sequences with length l, it takes time proportional to $\mathcal{O}(l^n)$. Therefore, the complexity of aligning multiple sequences using SP-score is NP-complete, making it impractical for larger tasks within a reasonable timeframe. To address this, several algorithms have been proposed to either reduce the search space or to approximate the best possible alignment.

Carrillo-Lipman algorithm reduces the search area of n-dimensional DP by focusing on a subset of n-dimensional matrix that contains the optimal alignment path [46]. This method is based on the principle of pairwise alignment, and first calculates the pairwise alignments between all sequence pairs to define an upper bound. Finally, it just considers cells of the matrix that justify all bounds.

Another category uses divide-and-conquer method to narrow down the search space to find the alignment in acceptable time [47]. The idea involves breaking down the entire MSA into many smaller alignments (segments) that can be aligned independently. These small alignments are then combined to form the optimal alignment. However, this approach introduces another challenge: determining where to segment. This issue is also NP-complete, making the strategy not completely practical [47].

While the last three algorithms provide optimal solution for MSA, they are quite slow. To address this, instead of seeking the optimal solution, we can construct an approximate

alignment step by step. The center-star algorithm is one of this techniques identifying a sequence that minimizes the mutations compared to all other sequences [48].

Most of popular MSA methods use progressive approach proposed in [19]. While there are other techniques that utilize hidden Markov models [49], genetic algorithms [50] or simulated annealing [51], they are not related to what we discuss here.

2.4 Progressive Alignment

Progressive alignment is core idea of many MSA methods e.g., CLUSTAL-W [26], DIALIGN [52], T-COFFEE [25], MUSCLE [53], MAFFT [29], PROBCONS [54] and Probalign [55].

The main idea of progressive alignment is to reduce the complexity of multiple sequence alignment by breaking it down into a series of pairwise and profile alignment. Most of heuristic methods use "progressive approach" proposed by [19]. The method involves constructing a guide tree to determine the order of aligning sequences and starting with the most similar pairs, then progressively adding in the more divergent sequences.

The guide tree is usually constructed using distance-based methods like UPGMA and Neighbor Joining. Therefore, the first step is measuring the evolutionary distance between sequences and construct distance matrix. There are two common ways for measuring distance of sequences. First one is aligning all pair of sequences and measure evolutionary distance between each pair. The second way is much faster and does not

need aligned sequences. It uses k-tuple similarity of each pair of sequences and convert this value to distance between them.

In the progressive step, starting from leaves of guide tree to root, sequences are aligned along the tree by pairwise or profile alignment. Figure2 shows a guide tree for five sequences. Two sequences seq1 and seq2 should be aligned by doing a PSA, and seq4 and seq5. These pairwise alignments give aligned sequences for internal node a and b respectively. Next, seq3 should be aligned with the aligned sequences at node a by doing profile alignment, gives the alignment of node c. Finally, two alignments of node c and b should be done resulting in the complete alignment of five sequences.

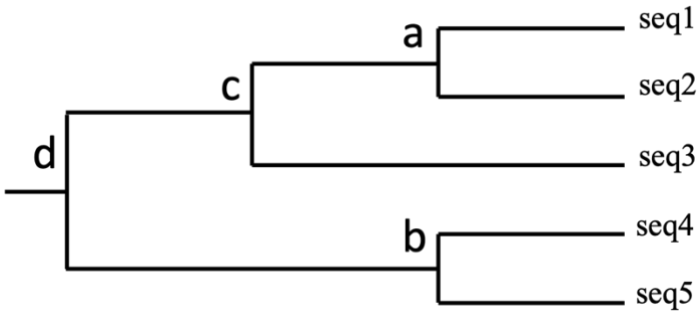

Figure 1: A guide tree of 5 seqeunces

Although progressive alignment facilitates the alignment process, it has two serious problems. One limitation of progressive alignment is that once two sequences are aligned, it is not subject to realignment during the process. This can be a drawback because the ability to repeatedly realign sequences could enhance the overall quality of alignment. However, adding refinement steps to progressive approach introduce a new challenge because new criterion should be defined for termination the refinement [26]. This concept

of repeated sequence realignment is known as iterative refinement, which is explored further in the next section.

2.5 Iterative refinement

An effective strategy to improve the alignment is to iteratively rearrange the output of progressive step, to correct the errors generated in early steps of alignment [53]. The process can stop when no improvement occurred in objective function. This strategy is used in MUSCLE, MAFFT and PROBCONS by partitioning the tree and re-aligning profiles of the two sub-trees. At each step of refinement, the new alignment is chosen if it has higher SP score.

3 Background

3.1 Progressive alignment techniques

Progressive alignment is core idea of many MSA methods e.g., CLUSTAL-W, DIALIGN, T-COFFEE, MUSCLE, MAFFT, PROBCONS and Probalign. In this chapter, we discuss sum of popular methods in more details.

3.1.1 CLUSTAL-W

ClustalW [26] is one of well-known tools in bioinformatics for conducting multiple sequence alignment. The algorithm starts with performing pairwise sequence alignment (PSA) for all possible sequence pairs within a dataset to compute distance matrix based on these PSA. ClustalW then employs the Neighbor-Joining (NJ) method to construct a guide tree, a phylogenetic tree that determine the order of aligning sequences, starting with the most closely related sequences, and progressively adding more distantly related ones.

In refining the alignment process, ClustalW introduce some modifications to improve the use of scoring matrix and gap penalty in the traditional progressive alignment [26]. It adapts the initial gap penalty values as a function of sequence similarity, evolutionary distance, and sequence length which leads to more biologically relevant alignment. Furthermore, it manipulated initial gap penalty in a residue- and position specific manner, offering a more complex approach to gap insertion. Additionally, ClustalW

calculates a weight for each sequence based on the guide tree, which is used as scale in profile alignments. These methodological modifications enable ClustalW to generate MSAs that are not only computationally efficient but also of high quality, reflecting the true phylogenetic relationship among sequences [56].

3.1.2 T-COFFEE

T-COFFEE [25] improved the progressive approach by introducing a new scoring system rather than traditional use of scoring matrices. T-COFFEE uses a progressive approach similar to CLUSTALW but assigns a score to each residue pair of each pair of sequences. This score is computed by gathering information from all other sequences from comparing all triplets of sequences, integrating information from various sequence alignments to evaluate the global context of each residue pair [25]. This kind of pair residue scoring is known as consistency-based alignment, and it allows for a more accurate prediction of residue homology. T-COFFEE uses these scores in pairwise alignment and profile alignment instead of using conventional scoring matrix values.

3.1.3 MAFFT

MAFFT is a multiple sequence alignment method that uses Fast Fourier Transform (FFT) to rapidly identify the homologous regions between two sequences [29]. Once these homologues regions are identified, MAFFT employs a dynamic programing algorithm to arrange these homologous regions. This approach takes less time compared to regular

dynamic programing over complete sequences because it focuses on aligning the most relevant parts of the sequences, rather than applying the computationally intensive process to the entire sequence length. MAFFT extend this process for aligning groups of sequences (profile alignment). MAFFT also modified the scoring system including gap-penalty and scoring matrix.

3.1.4 MUSCLE

The alignment process in MUSCLE consists of three main stages: the draft progressive, improved progressive and refinement step [53]. Initially, MUSCLE generates a rough alignment quickly by using a progressive algorithm that builds a tree using UPGMA based on a k-mer distance between each pair of sequences and then aligns sequences from leaves to the root of the guide tree. In the next step, the algorithm uses the MSA achieved from stage one to measure a more accurate distance matrix of sequences using Kimura distance and then constructs a new guide tree based on this new distance matrix. Afterwards, MUSCLE aligns sequences progressively according to new guide tree. The final refinement stage iteratively improves the alignment by dividing sequences into two sets and realigning them, keeping new alignment if it has higher SP score compare the previous one.

3.1.5 PROBCONS

Another category of alignment approaches uses Hidden Markov Model (HMM) to make probabilistic interpretation of alignment. PROBCONS is HMM-based method computing the maximal-expected-accuracy alignment instead of maximum SP score [57]. The expected accuracy of an alignment is calculated as summation over posterior probabilities of all residue pairs [58]. PROBCONS computes these probabilities using HMM for pairwise sequence alignment. The HMM parameters are learned using unsupervised learning on the BAliBASE 2.0 benchmark dataset.

3.1.6 PROGALIGN

Probalign is another HMM-based method that estimates the posterior probabilities from partition function described in [58]. This method calculates the probabilities of all possible alignments and uses these probabilities to determine the most likely alignment for a given set of sequences. Probalign uses the probability consistency transformation described by PROBCRONS, to iteratively refine the alignment probabilities by considering the alignment of sequence pairs across the entire dataset. The integration of these posterior probabilities with the consistency-based framework of PROBCRONS allows Probalign to produce highly accurate sequence alignments.

3.1.7 Non-progressive alignment methods

DIALIGN [52] and picXAA [59] are two non-progressive alignment methods. Instead of relying on a global alignment of entire sequence, DIALIGN focuses on identifying and

aligning small parts of sequences, known as segments of local homology, without doing a global alignment across all sequences. It uses a weighted sum-of-pair scoring system that allows for combination of these segments into multiple alignment.

picXAA represents a non-progressive approach to MSA that try to maximize expected accuracy. This method is different from the traditional progressive alignment techniques because it does not conduct alignment in hierarchical process based on the guide tree [59]. Instead, picXAA begins the alignment process by identifying pair of residues that have the highest probability of homology. Once these high-confidence pairs are aligned, the method aligns residues with lower probabilities. This approach focuses on most reliable parts of alignment and avoide propagating errors.

3.2 The role of guide tree in progressive approach

Guide tree is a fundamental component in determining the accuracy of output MSA generated by progressive methods. The construction of the guide tree is based on pairwise comparisons, and the quality of these initial comparisons is crucial because errors at this stage can be propagated as alignment process progresses. Several studies have shown that an inaccurate guide tree could be a major source of error in progressive sequence alignment [21,22], therefore different strategies for improving guide tree construction have been proposed [20]. For example, the influence of guide trees on the accuracy of Multiple Sequence Alignments (MSAs) generated by MAFFT and ClustalW

was assessed using two criteria: the sum-of-pair score (SPS) and phylogenetic reconstruction [23]. The result showed that guide tree has little effect on SPS-based MSAs but using better guide trees significantly improved the accuracy of phylogenies when compared to the default guide trees.

GLProbs [24] is another algorithm trying to improve the MSA by constructing accurate guide tree. This method introduces a new measure to estimate the similarity between sequences and aligns each sequence pairs either globally or locally based on their level of similarity. This simple idea results in more accurate distance matrix, which is computed based on the posterior probabilities of local or global aligned sequences. GLProbs was compared to most of popular algorithms such as MAFFT, MUSCLE and picXAA and has a higher accuracy [24].

3.3 Summary

Many algorithms have been proposed for sequence alignment which try to improve the speed and accuracy of MSA. MAFFT and MUSCLE are two mostly used algorithms in many studies. However, these algorithms although can align a large number of sequences in just some minutes, they could not achieve acceptable accuracy for aligning highly diverged sequences.

In this study, we use 3-way alignment to calculate distance matrix of sequences and construct the guide tree. An important source of error in MSA is caused by a wrong guide

tree, and this error even cannot be corrected by iterative refinement step. Therefore, by generating an accurate guide tree we can avoid many errors in MSA and even avoid doing an intense iterative refinement.

4 Methodology

4.1 Carrillo-Lipman algorithm for three sequences

Carrillo-Lipman algorithm was proposed to narrow down the search space within N-dimensional dynamic programing for optimal MSA. The idea behind it is to combine initial MSA with information from each pairwise alignment to define lower bounds for the two-dimensional projection of the optimal path. Consequently, this enables us to focus solely on the cells within the N-dimensional lattice that satisfy these bounds.

In this section, we restate the Carrillo-Lipman equations for three sequences [46]. Suppose we have three sequences s_1, s_2 and s_3. The optimal alignment for these three sequences has the highest score based on the SPS criterion. Therefore, any other alignment has a lower score, leading to the following inequality:

$$S(\gamma^*) - S(\gamma^e) \geq 0, \tag{1}$$

where $\gamma^*$and γ^e are the optimal and an arbitrary alignment respectively. The SPS of a 3-way alignment is:

$$S(\gamma^*) = S(\gamma_{12}) + S(\gamma_{13}) + S(\gamma_{23}), \tag{2}$$

$$S(\gamma_{12}) + S(\gamma_{13}) + S(\gamma_{23}) - S(\gamma^e) \geq 0 \quad , \tag{3}$$

where the γ_{12} is the pairwise alignment of s_1 and s_2, γ_{13} is the pairwise alignment of s_1 and s_3, and γ_{23} is the pairwise alignment of s_2 and s_3. In other words, any of these

pairwise alignment can be considered as the projection of γ^* on the three surfaces of 3D-DP. Based on equation (3), we can write three inequalities for each projection of γ^*:

$$\begin{aligned} S(\gamma_{12}) &\geq S(\gamma^e) - \big(S(\gamma_{13}) + S(\gamma_{23})\big) \\ S(\gamma_{13}) &\geq S(\gamma^e) - \big(S(\gamma_{12}) + S(\gamma_{23})\big) \\ S(\gamma_{23}) &\geq S(\gamma^e) - \big(S(\gamma_{12}) + S(\gamma_{13})\big) \end{aligned} \tag{4}$$

For each pair of sequences, we can find the optimal alignment which has the highest SPS, so we can write:

$$\begin{aligned} S(\gamma_{12}^*) &\geq S(\gamma_{12}) \\ S(\gamma_{13}^*) &\geq S(\gamma_{13}) \\ S(\gamma_{23}^*) &\geq S(\gamma_{23}) \end{aligned} \tag{5}$$

where γ_{12}^*, γ_{12}^* and γ_{12}^* are the optimal pairwise alignment. Using these three inequalities, we can rewrite the equation (4) as the following inequalities:

$$\begin{aligned} S(\gamma_{12}) &\geq S(\gamma^e) - \big(S(\gamma_{13}^*) + S(\gamma_{23}^*)\big) \\ S(\gamma_{13}) &\geq S(\gamma^e) - \big(S(\gamma_{12}^*) + S(\gamma_{23}^*)\big) \\ S(\gamma_{23}) &\geq S(\gamma^e) - \big(S(\gamma_{12}^*) + S(\gamma_{13}^*)\big) \end{aligned} \tag{6}$$

The Carrillo-Lipman defines the three boundaries based on the above inequalities:

$$\begin{aligned} L_{12} &= S(\gamma^e) - \big(S(\gamma_{13}^*) + S(\gamma_{23}^*)\big) \\ L_{13} &= S(\gamma^e) - \big(S(\gamma_{12}^*) + S(\gamma_{23}^*)\big) \\ L_{23} &= S(\gamma^e) - \big(S(\gamma_{12}^*) + S(\gamma_{13}^*)\big) \end{aligned} \tag{7}$$

L_{12} is the lower bound for the score of pairwise alignment of S_1 and S_2, L_{13} is the lower bound for the score of pairwise alignment of S_1 and S_3, and L_{23} is the lower bound for the

score of pairwise alignment of S_2 and S_3. In other words, L_{12}, L_{13} and L_{23} are the lower bounds for the measure of the projection of any 3-dimensional optimal path into the planes determined by each pair of sequences. Then, when looking for γ^*we need only consider those paths in the cubic that their pairwise alignment satisfies the related inequality.

Same as Carrillo-Lipman algorithm, we call the set of paths that their projection on the plane (S_1, S_2), (S_1, S_3) and (S_2, S_3) satisfies the inequality (7), X_{12}, X_{13} and X_{23} respectively. Thus, the paths in the set

$$X = X_{12} \cap X_{13} \cap X_{23}, \tag{8}$$

are the only possible candidates to be an optimal path. To consider only paths in X means having to apply the dynamic programming procedure to find γ^* only in subregion Y of the cubic. Let Y_{12}, Y_{13} and Y_{23} be the set of points that their projection on each plain satisfies the related bound. Therefore, the set:

$$Y = Y_{12} \cap Y_{13} \cap Y_{23}, \tag{9}$$

This theory proves that it is unnecessary to apply the dynamic programming method to the entire cubic, but it suffices to consider just subregion Y. For each pair of sequences, we use 2D dynamic programing to find the PSA score to calculate $\gamma_{12}^*, \gamma_{13}^*$ and γ_{23}^*, as required for calculating the lower bounds, applying the Carrillo-Lipman algorithm on all possible triplets. It is noteworthy that the performance of this method heavily relies on

the initial alignment γ^e used for identifying lower bounds. To significantly reduce the search area, this alignment should closely approximate the optimal path. The time and space saved by this method is more for highly similar sequences than for highly diverged sequences.

4.2 3-Way Alignment

Three dimensional dynamic programing (3D_DP), formulated by Gotoh [60], represents an extension of the Needleman-Wunch algorithm proposed for pairwise sequence alignment. Gotoh contributed to developing 3D-DP for affine gap penalty. This approach for 3D-DP increased time and space requirement to cubic complexity. 3D-DP for aligning three simple sequences with constant gap penalty showed in Figure2.

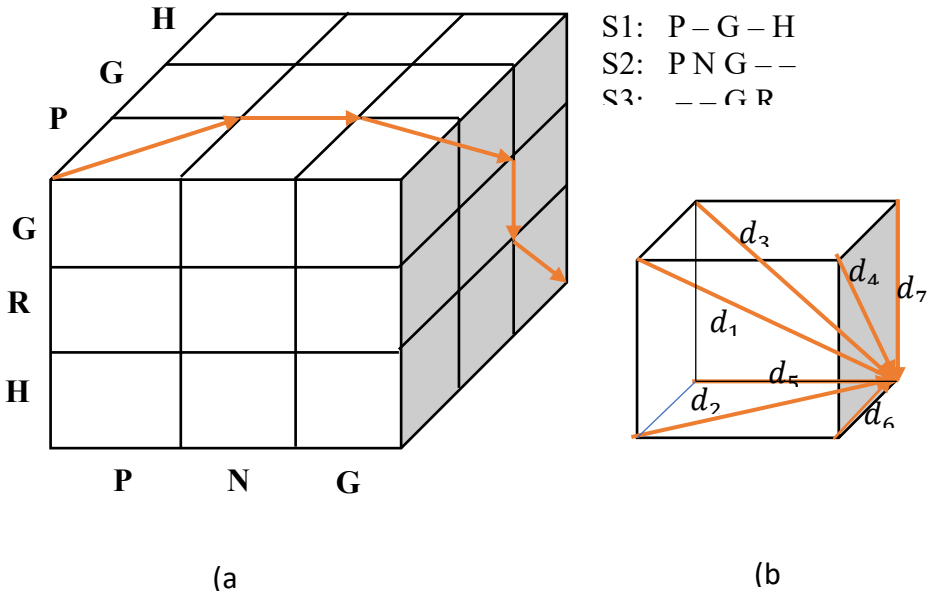

Figure 2: An example of 3-way alignment with constant gap penalty

Figure2 (a) shows the path corresponding the alignment of S1, S2 and S3, starting from beginning vertex of the cubic and ending to the other vertex.

3-way alignment uses a three-dimensional matrix (a cubic) where each dimension corresponds to one of three sequences. Each cell in this 3D matrix corresponds to a specific combination of positions in the three sequences. For example, a position (i, j, k) represents the alignment state considering the i-th position of first sequence, the j-th position of the second sequence, and k-th position of the third sequence.

Just like in 2D dynamic programming, scoring involves assigning values for matches, mismatches and gaps. This can include scoring a match between all three sequences, a match between two sequences and a gap in the third, and other combinations. The scoring system becomes more complex as it needs to account for a larger variety of possibilities in sequence alignment.

In the alignment with dynamic programing the process typically involves deciding the direction to proceed from a given cell in the alignment matrix. In aligning two sequences, the choice of direction is dedicated by whether to insert a gap in one of the sequences, both, and to align the residues directly. This leads to three basic directions for progression from any given cell: diagonal (no gap), horizontal (gap in the first sequence), or vertical (gap in the second sequence).

However, the scenario becomes more complex in three-dimensional dynamic programming (3D-DP) for aligning three sequences. Here at each position in the cubic,

there are seven potential directions to progress. These directions are determined by the different combinations of gaps that might occur across the three sequences. Each direction corresponds to one of the seven possible gap combinations: no gap in any sequence, a gap in one of the sequences (three possibilities), or a gap in two out of three sequences (three possibilities).

In Figure2 (b) moving toward d_1 means aligning three residues, d_2, d_3 and d_4 mean putting a gap in one sequence and align residues in two other sequences, and d_5, d_6 and d_7 mean putting gap in two sequences. Therefore, we need to calculate the score for all seven possible direction and move forward one with the highest score at each position.

When incorporating an affine gap penalty function, which adds different cost for opening a gap and a separate cost for gap extension, the representation and computation become even more complicated. The standard cubic model, which suffices for constant gap penalty in 3-way alignment, is inadequate. Instead, the process necessitates a more sophisticated model involving multiple cubes: seven for the forward process of filling the matrix with scores and another set of seven for the traceback process, which reconstructs the optimal alignment path from the completed matrix. We need seven cubes to distinguish between gap opening and gap extension due to different scores. For instance, opening a gap in one sequence while extending a gap in another involves a different cost structure compared to extending gaps in all sequences or opening a new gap while other gaps are being extended.

4.3 Formulation for 3-way alignment using affine gap penalty

Let A, B and C represent three sequences and their lengths are denoted by $n, m,$ and l, respectively. For three residues of A_i, B_j and C_k at position (i, j, k), there are seven possible alignment configurations. $M(i, j, k)$ represents the best score when three residues are aligned. $I_{xy}(i, j, k)$, $I_{xz}(i, j, k)$ and $I_{yz}(i, j, k)$ are the scores of introducing one gap in C_k, B_j and A_i respectively. Similarly, $I_x(i, j, k)$, $I_y(i, j, k)$ and $I_z(i, j, k)$ represent the scores for aligning a residue in A_i, B_j and C_k while introducing gaps in the other two sequences.

The 3-way alignment algorithm was formulated by Gotoh [60] for affine gap penalty. By convention, the criterion for choosing the best alignment among all possible ones is equivalent to maximum parsimony, i.e., the alignment with the smallest alignment cost incurred by indels and mismatches is the best alignment. Expressed alternatively, the best alignment is the one with the highest alignment score as a function of matches and mismatches, as well as gap open and gap extension penalties. With three sequences, an aligned site with two residues in the first two sequences and a gap in the third sequence is interpreted as having a single change (a deletion in the third sequence), with u_D representing the deletion cost. Similarly, an aligned site with a single residue in sequence 1 and a gap in the two other sequences is also interpreted as a single change, i.e., a single insertion in sequence 1, with u_I representing this insertion cost. Gotoh [60] used $u_D = u_I = u$ in his alignment algorithm, with the implicit assumption of insertions and deletions occur equally frequently. This was also adopted by Huang [61]. However,

Kruspe and Stadler [27] treated u_D and u_I differently. We defined equations (10)-(16) in a similar way to those in [27] with a slight modification to facilitate the implementation of the Carrillo-Lipman algorithm:

$$M(i,j,k) = \left[\max \begin{Bmatrix} M(i-1,j-1,k-1) \\ I_{xy}(i-1,j-1,k-1) \\ I_{xz}(i-1,j-1,k-1) \\ I_{yz}(i-1,j-1,k-1) \\ I_x(i-1,j-1,k-1) \\ I_y(i-1,j-1,k-1) \\ I_z(i-1,j-1,k-1) \end{Bmatrix} \right] + \; S(A_i, B_j, C_k), \qquad (10)$$

$$I_{xy}(i,j,k) = \left[\max \begin{Bmatrix} M(i-1,j-1,k) - 2GO \\ I_{xz}(i-1,j-1,k) - 2GO \\ I_{yz}(i-1,j-1,k) - 2GO \\ I_z(i-1,j-1,k) - 2GO \\ I_{xy}(i-1,j-1,k) - 2GE \\ I_x(i-1,j-1,k) - 2GE \\ I_y(i-1,j-1,k) - 2GE \end{Bmatrix} \right] + \; S(A_i, B_j), \qquad (11)$$

$$I_{xz}(i,j,k) = \left[max \begin{Bmatrix} M(i-1,j,k-1) - 2GO \\ I_{xy}(i-1,j,k-1) - 2GO \\ I_{yz}(i-1,j,k-1) - 2GO \\ I_y(i-1,j,k-1) - 2GO \\ I_{xz}(i-1,j,k-1) - 2GE \\ I_x(i-1,j,k-1) - 2GE \\ I_z(i-1,j,k-1) - 2GE \end{Bmatrix} \right] + S(A_i, C_k), \qquad (12)$$

$$I_{yz}(i,j,k) = \left[max \begin{Bmatrix} M(i,j-1,k-1) - 2GO \\ I_{xy}(i,j-1,k-1) - 2GO \\ I_{xz}(i,j-1,k-1) - 2GO \\ I_x(i,j-1,k-1) - 2GO \\ I_{yz}(i,j-1,k-1) - 2GE \\ I_y(i,j-1,k-1) - 2GE \\ I_z(i,j-1,k-1) - 2GE \end{Bmatrix} \right] + S(B_j, C_k), \qquad (13)$$

$$I_x(i,j,k) = max \begin{cases} M(i-1,j,k) - 2GO \\ I_{yz}(i-1,j,k) - 2GO. \\ I_{xy}(i-1,j,k) - GO - GE \\ I_{xz}(i-1,j,k) - GO - GE, \\ I_y(i-1,j,k) - GO - GE \\ I_z(i-1,j,k) - GO - GE \\ I_x(i-1,j,k) - 2GE \end{cases} \quad (14)$$

$$I_y(i,j,k) = max \begin{cases} M(i,j-1,k) - 2GO \\ I_{xz}(i,j-1,k) - 2GO \\ I_{xy}(i,j-1,k) - GO - GE \\ I_{yz}(i,j-1,k) - GO - GE, \\ I_x(i,j-1,k) - GO - GE \\ I_z(i,j-1,k) - GO - GE \\ I_y(i,j-1,k) - 2GE \end{cases} \quad (15)$$

$$I_z(i,j,k) = max \begin{cases} M(i,j,k-1) - 2GO \\ I_{xy}(i,j,k-1) - 2GO \\ I_{xz}(i,j,k-1) - GO - GE \\ I_{yz}(i,j,k-1) - GO - GE, \\ I_x(i,j,k-1) - GO - GE \\ I_y(i,j,k-1) - GO - GE \\ I_z(i,j,k-1) - 2GE \end{cases} \quad (16)$$

In the formulae above, GO and GE are gap open and gap extension penalties and $S(\alpha, \beta)$ denotes the score of aligning two residues, determined by a scoring matrix such as PAM or BLOSUM. The score of aligning three residues is the sum-of-pair score (SPS), i.e., $S(A_i, B_j, C_k) = S(A_i, B_j) + S(A_i, C_k) + S(B_j, C_k)$.

The specification in equations (10)-(16) carry some benefits in the context of the Carrillo-Lipman method described before. Since we are searching for an optimal 3-way alignment satisfying Gotoh's equations, we need to estimate the γ^e which is an arbitrary alignment of three sequences A, B and C. In our implementation, we estimated γ^e by progressive

alignment and used it in Carrillo-Lipman equations. Therefore, we used 2GO for (I_{xy}, I_{xy}, I_{xz}) to be consistent with calculation of SPS. An aligned site with two residues in two sequences and a gap in the third sequence is counted as two indel events in SPS (i.e., an indel between sequence 1 and sequence 3 and an indel between sequence 2 and sequence 3). Similarly, an aligned site with a residue in sequence 1 and a gap in sequence 2 and sequence 3 is also counted as two indel events is SPS. By using 2GO in (I_{xy}, I_{xy}, I_{xz}) we can estimate γ^e based on progressive alignment and use it in Carrillo-Lipman equations. Equations (10)-(16) do not conflict with Gotoh's equations.

Similar to the PSA with the affine gap penalty function, we need to establish seven traceback matrices to reconstruct the optimal alignment once the scoring matrices are completed. The values within these matrices are determined during the forward procedure and are used in the subsequent traceback procedure.

4.4 Trace back procedure

We have seven traceback cubes $B_M, B_{xy}, B_{xz}, B_{yz}, B_x, B_y$, and B_z. Each cell in trace back cubes is assig a number among 0 to 6 based on which score is highest. For example, if the $M(i-1, j-1, k-1)$ result in highest score of $M(i, j, k)$, we put 0 in the corresponding cell of B_M. Similarly, if the highest value obtained by $I_{xy}(i-1, j-1, k-1)$, $I_{xz}(i-1, j-1, k-1)$, $I_{yz}(i-1, j-1, k-1)$, $I_x(i-1, j-1, k-1)$, $I_y(i-1, j-1, k-1)$ or $I_z(i-1, j-1, k-1)$ we put 2, 3, 4, 5, 6 or 7 in B_M respectively.

The same process should be done at each cell for all seven traceback matrices until full all cells in the cubes. Traceback starts from the end corner cell of traceback matrix that its scoring matrix has the highest score in the end corner cell. For example, if M has the highest score at that cell, we start trackbacking from B_M. When we are at the B_M, the next cell in the process is $(i-1, j-1, k-1)$. Therefore, we align A_N, B_M and $C_K$and move to the cell $(N-1, M-1, K-1)$ of one of the traceback matrices depending on the value of $B_M(N, M, K)$. If $B_M(N, M, K)$ is 0, the next cell is $B_M(N-1, M-1, K-1)$. If $B_M(N, M, K)$ is 1, the next cell is $B_{XY}(N-1, M-1, K-1)$, and so on. Suppose the value of $B_M(N, M, K)$ and we move to the $B_{XZ}(N-1, M-1, K-1)$, we should align A_{N-1} and C_{K-1} and put a gap in second sequence at this position.

Generally, when moving to $B_M$at any position (i, j, k), we should align $A_i, B_j$and C_k. When moving to B_{XY}, we should align A_i and B_j and put a gap in third sequence. Similarly, for B_{XZ} and B_{YZ} we put gap in sequence Y and X respectively and align the residue of other two sequences. When moving to B_X, B_Y and B_Z, we should put gap in two sequences and align with A_i, A_j and A_k respectively. We follow this process until reach the beginning cell in one of the traceback matrices.

4.5 Reducing time and space complexity

Carrillo-Lipman algorithm reduce the searching area of dynamic programming, which reduce time complexity. However, we cannot determine the exact time complexity of

the algorithm because it depends on the level of identity of sequences and how much we are successful in finding a good estimation for γ^e near the optimal path.

We also reduce the required memory. Instead of defining three dimensional matrices and just filling some cells that satisfy the Carrillo-Lipman criteria, we just define a list and save the values of cells in that list. Each row of the list refers to a candidate cell, and we put values of all forward scoring matrices and traceback matrices in that row. Therefore, instead of fourteen matrices we have just one list. In the case of memory reduction, it depends on the number of candidates of the cells selected by Carrillo-Lipman bounds.

4.6 Algorithm

In this study we apply 3-way alignment to all possible triplets of a given set of sequence to generate distance matrix based on them and reconstruct guide tree. Then, we compare the performance of MAFFT using our guide trees and its default guide tree on simulated datasets. The process is summarized in the following steps (The flowchart is provided in Figure 3).

- Simulating different datasets with IQ-TREE
- Applying 3-way alignment on all possible triplets of each data
- Calculating evolutionary distance between all pair of sequences using aligned triplets
- Constructing guide tree by distance-based algorithm

- Inputting guide trees to MAFFT and aligning sequences (MSA + 3-way alignment)
- Aligning sequences by MAFFT and its default guide tree (MSA)
- Using PhyML to construct the phylogenetic tree for both MSA and MSA + 3-way alignment
- Measuring RF distance between reconstructed phylogenetic trees and original one
- Comparing RF distance of PhyML trees of both strategies

In the following sections, we provide more details on mentioned steps.

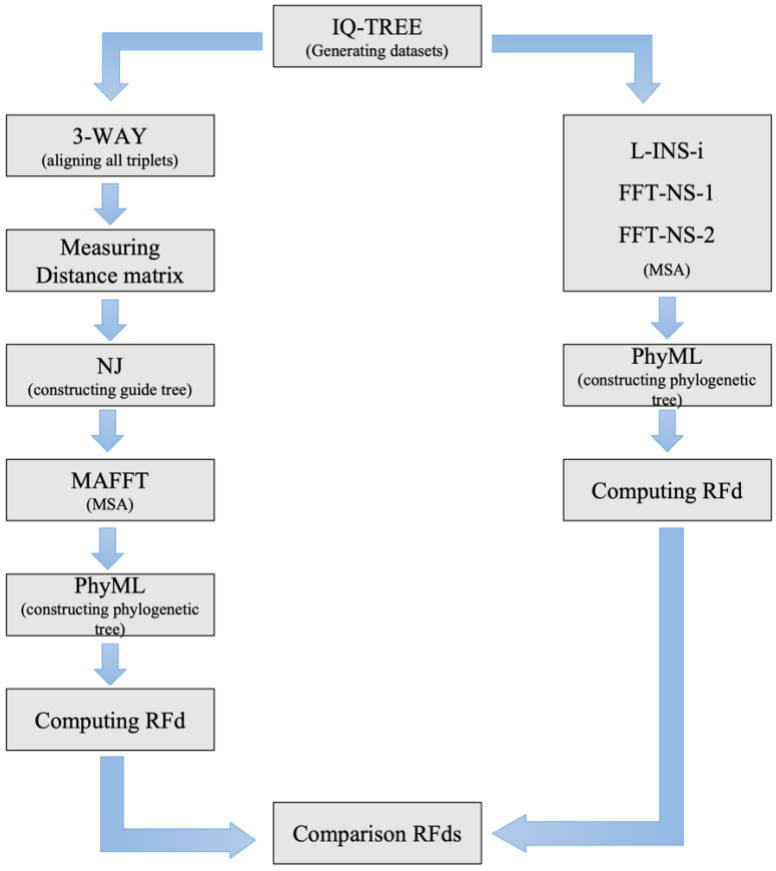

Figure 3: Flowchart of simulation

4.7 Simulated dataset

We generated our amino acid sequence datasets based on symmetric and asymmetric trees with 16-taxa (Figure 4) and 8-taxa (Figure 5). For 16-taxa tree, two different sets of branch lengths were used to generate sequences with different levels of divergency, and 50 datasets have been generated for each tree. We used the Alisim tool, provided by IQ-TREE [62], to produce aligned sequences with an average length of 500 for each tree. The JTT (Jones-Taylor-Thornton) substitution model [63] was used for all datasets. There are two types of amino acid substitution models. The first type is based on counting empirical substitutions from a large number of aligned protein sequences, with the hope of the resulting substitution model will be a one-hat-fits-all. The second type is derived from the maximum likelihood method based on a specific set of protein sequences (e.g., vertebrate mitochondrial proteins). They all specify the transition probabilities between amino acids given a branch length in a tree.

An insertion/deletion rate of 0.05 was used for both 16-taxa and 8-taxa trees. POW (power law distribution) was used as insertion/deletion size with a=2, and power=100. Therefore, we have four datasets based on 16-taxa trees: 1) half-symmetric tree with branch scale 0.3, 2) symmetric tree with branch scale 0.6, 3) half-asymmetric tree with branch scale 0.15 and 4) asymmetric tree with branch scale 0.3. We have two datasets based on 8-taxa trees: 1) symmetric tree with branch scale 0.8, and 2) asymmetric tree with branch scale 0.4. The

simulated data and C source code implementing the 3-way alignment are included in the supplemental file.

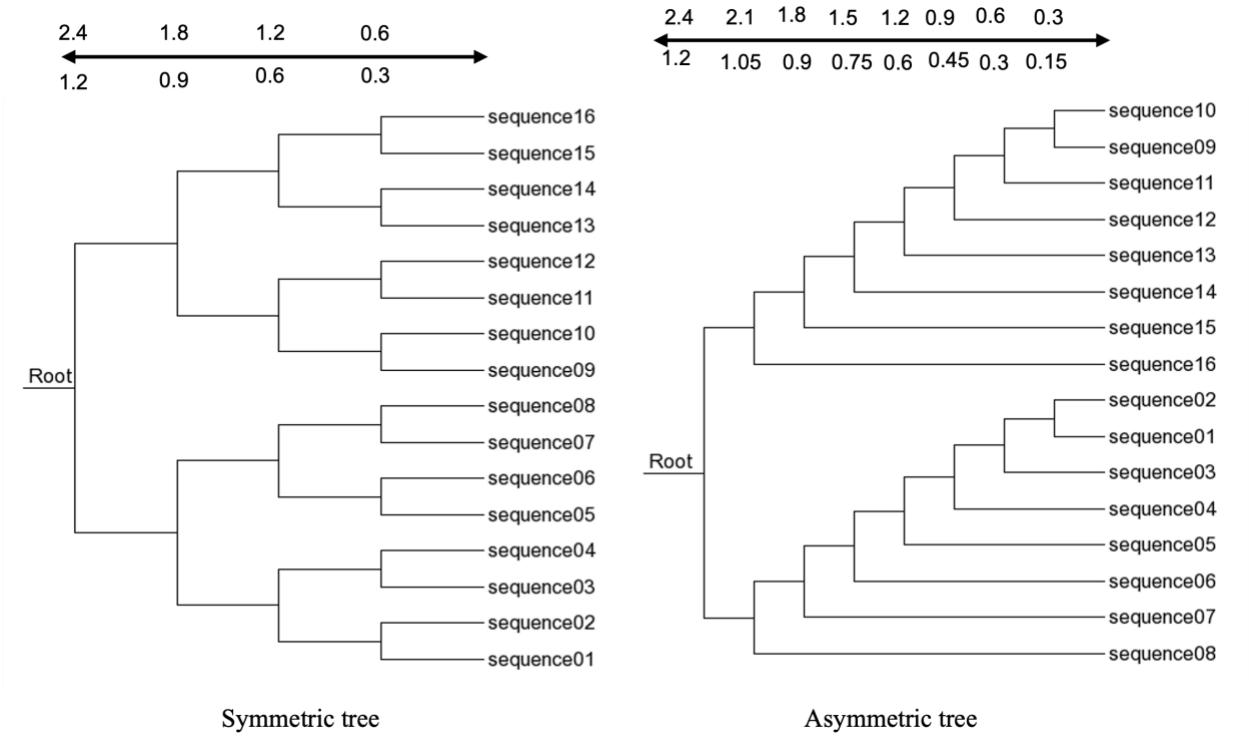

Figure 4: The 16-taxa trees used for simulating sequences. The branch length from the leaf to each internal nodes are indicated by the scale above the tree. Trees referred to as symmetric and asymmetric trees use top numbers of the scale. Trees referred to as half symmetric and half asymmetric trees use bottom numbers of the scale.

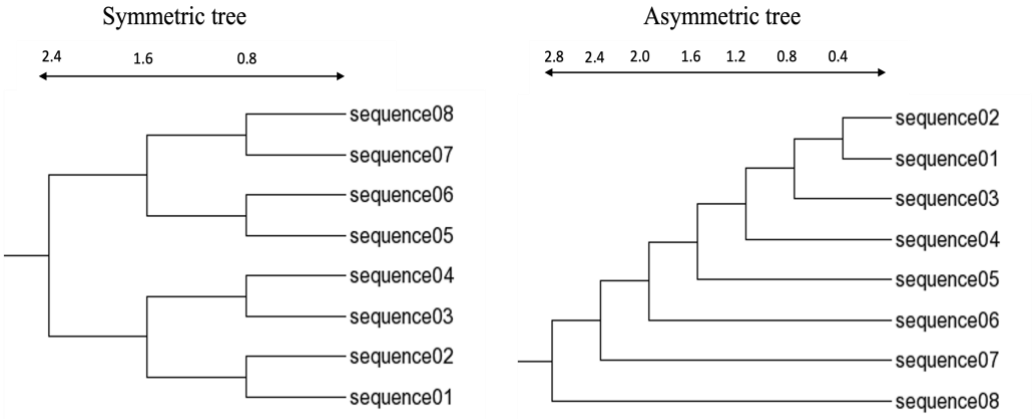

Figure 5: 8-taxa trees used for simulating sequences of high divergence. The scale indicates the branch lengths from the leaf to internal nodes.

4.8 Measuring distance matrix and constructing guide tree.

For each dataset, we aligned all possible triplets (56 and 560 triplets for 8-taxa and 16-taxa topologies respectively) by Carrillo-Lipman algorithm with the BLOSUM62 matrix, a gap-open penalty of 10 and a gap-extension penalty of 2. Each pair of sequences exists in $(n-2)$ triplets. Therefore, to measure the distance between two sequences, we calculated the average over their distances in all those $(n-2)$ triplets. The final distance matrix contains the average distance of each sequence pair. We used the JTT model to measure evolutionary distances between each pair of sequences in the aligned triplets. The resulting distance matrices were then used as input to the NJ algorithm in the PHYLIP package to construct guide trees that would later be used in progressive multiple alignment.

4.9 Aligning sequences with MAFFT

We compared the performance of MAFFT with the MAFFT-generated guide trees against the 3-way alignment guide trees. In this study, we assess the performance of three different algorithms of MAFFT: FFT-NS-1, FFT-NS-2 and L-INS-i (which is the most accurate option in MAFFT). We used MAFFT default except for specifying FFT-NS-1, FFT-NS-2, or L-INS-i. The FFT-NS-1 option measures distances based on the sharing of k-tuples between sequences (where k is typically 6). A guide tree is then reconstructed using UPGMA to guide the subsequent multiple alignment. FFT-NS-2 reconstructs a new

guide tree using the alignment generated by FFT-NS-1, and re-align sequences based on the new tree. We expect FFT-NS-2 to generate more accurate alignment compared to the FFT-NS-1 because of the recomputing of the guide tree. L-INS-i uses local alignments with the Smith-Waterman algorithm to generate distance matrix instead of the k-tuple method. Moreover, it uses a new objective function combining the weighted sum-of-pair score (WSP) and COFFEE-like score which measure the consistency between MSA and PSA [29].

4.10 Comparing the accuracy of phylogenetic trees

MSAs generated in the previous step were used to construct phylogenetic trees using PhyML [64], with the option of simultaneously optimizing tree topology, branch length and rates. These PhyML trees were then compared with the true tree for both 16-taxa and 8-taxa trees shown in Figure 4 and Figure 5 , through calculation of Robinson-Foulds distances (RFd) [65]. RFd between the true tree and the reconstructed tree is taken as a proxy for phylogenetic accuracy, where RFd = 0 means that the two trees share the same topology and larger RFd values are associated with inaccuracies. RFd values between trees were computed using the APE package [66] in R. Note that RFd measures only the topological difference between trees, but not differences in branch lengths. Thus, a reconstructed tree would be considered as identical to the true tree when RFd = 0, even if the two trees differ in branch lengths.

5 Result

5.1 The 3-way alignment tends generate guide trees closer to the true tree than other approaches

We first evaluated the two guide trees, one generated from the MAFFT L-INS-i option and the other by our 3-way alignment (3-WAY in Table 1) by comparing them with the true tree (i.e., the tree used for sequence simulation). With the symmetric tree, both approaches recovered some true trees, but the 3-way alignment approach recovered slightly more true trees (Table 1). Similarly, RFd is greater for L-NS-i than for the 3-way alignment approach. These results are consistent with our hypothesis that the 3-way alignment approach would produce better guide trees. However, these differences are small and not statistically significant given our sample size of 50 sets of simulated sequences (two-tailed paired-sample t-test, t = 1.1881, DF = 49 and p = 0.2405, Table 1). Given the effect size, a sample of 140, respectively, would be needed get a p value below 0.05.

Table 1: The result of comparing guide trees generated by 3-way alignment and L-INS-i method based on simulated amino acid sequences on 8-taxa symmetric and asymmetric trees.

	Symmetric tree			**Asymmetric tree**		
Guide tree	N_{true}(1)	RFd(2)	SE_{RFd} (3)	N_{true}(1)	RFd(2)	SE_{RFd}(3)
L-INS-i	27	0.92	0.1424	0	5.12	0.1993
3-WAY	31	0.76	0.1387	0	4.84	0.2067

(1) N_{true}: the number of correctly reconstructed trees (RFd = 0) by a method
(2) RFd: mean RFd from 50 simulated sets of sequences.
(3) SE_{RFd}: standard error of RFd.

With the asymmetric tree, neither the L-NS-i approach nor the 3-way alignment results in a guide tree that is identical to the true tree (Table 1). However, the difference in RFd, similar to the results with symmetric trees, is in the expected direction, i.e., being smaller for the 3-way alignment than for the L-INS-i approach. However, the difference between the two groups is not statistically significant given the sample size of 50 for each group ($t = 1.1586$, $DF = 49$, $p = 0.2522$).

Table 2 presents the result of comparisons of guide trees for 16-taxa trees. There are four different 16-taxa trees, half-symmetric, symmetric, half-asymmetric and asymmetric trees, represented as H-S tree, S tree, H-AS tree and AS tree, respectively in Table 2. We compared the guide trees from four different approaches, 3-way alignment (3-WAY in Table 2), and the three MAFFT options (FFT-NS-1, FFT-NS-2 and L-INS-i), based on the simulated sequences. The guide trees reconstructed from k-tuple similarities (FFT-NS-1 and FFT-NS-2, with $k = 6$) are apparently much worse than those reconstructed from pairwise alignment (L-INS-i) or 3-way alignment (Table 2). However, just as in Table 1, there is no significant difference between the last two approaches. The L-INS-i approach actually performed slightly better than the 3-way alignment approach with the H-AS Tree, recovering more true trees (41 versus 39) and having a smaller mean RFd (0.36 versus 0.52) than the 3-way alignment approach (Table 2), although the difference is not statistically significant. The only difference reaching borderline significance involves the asymmetric tree (Table 2). The 3-way alignment appears to produce a better guide tree,

with an RFd nearly significantly smaller than that from the L-INS-i option (paired sample t-test, t = 1.8448, DF = 49, p = 0.0711).

Table 2: Quality of guide trees generated by three MAFFT options (FFT-NS-1, FFT-NS-2, L-INS-i) and by the 3-way alignment (3-WAY), based on simulated amino acid sequences the 16-taxa trees including Half-symmetric (H-S tree), Symmetric (S tree), Half-asymmetric (H-AS tree), and asymmetric (AS tree) trees. Other column labels are the same as in Table 1

	H-S tree			**S Tree**			**H-AS Tree**			**AS Tree**		
Guide tree	N_{true}	RFd	SE_{RFd}	N_{true}	RFd	SE_{RFd}	N_{true}	RFd	SE_{RFd}	N_{true}	RFd	SE_{RFd}
FFT-NS-1	35	1.08	0.2693	0	9.24	0.5107	15	2.52	0.3514	0	15.32	0.5817
FFT-NS-2	34	1.28	0.2956	0	7.24	0.4495	29	1.16	0.2497	0	13	0.4891
L-INS-i	50	0	0	39	0.6	0.1429	41	0.36	0.1098	0	6.48	0.2325
3-WAY	50	0	0	40	0.4	0.1143	39	0.52	0.1491	3	5.8	0.3886

5.2 The 3-way alignment leads to more accurate phylogenetic results than other approaches

How will the difference in the guide tree affect the final phylogenetic reconstruction? We obtained MSA from each of the three types of guide trees: 1) the true tree used for sequence simulation, 2) the guide tree reconstructed by the L-INS-i approach, and 3) the guide tree from the 3-way alignment (3-WAY in Table 3). These MSAs are then used to reconstruct phylogenies by PhyML. We expect the MSAs obtained with the true tree as the guide tree to recover true trees but are interested in whether the 3-way alignment approach will outperform the L-INS-i approach.

Using the true tree as the guide tree apparently increases the chance of the true tree being recovered through the aligned sequences, which is true for both the 8-taxa symmetric and

asymmetric trees (Table 3). With the symmetric tree, the 3-way alignment approach outperformed the L-NS-i approach, recovering more true trees and having a smaller mean RFd (Table 3). However, the difference is not statistically significant given the sample size of 50 for each group (two-tailed paired-sample t-test, t = 1.4289, DF = 49, p = 0.1594).

Table 3: Phylogenetic accuracy from different guide trees. Sequences were simulated for the 8-taxa symmetric and asymmetric trees. MSAs were generated 1) with the true tree (True Tree), 2) from the L-INS-i option (L-INS-i), and 3) from the 3-wau alignment (3-WAY). Phylogenetic reconstruction was done with PhyML. Other column headings are the same as in Table 2.

	Symmetric tree			**Asymmetric tree**		
Guide tree	N_{true}	RFd	SE_{RFd}	N_{true}	RFd	SE_{RFd}
True Tree	50	0	0	21	1.52	0.2180
L-INS-i	29	0.92	0.1637	0	6	0.2356
3-WAY	34	0.68	0.1469	0	5	0.2231

With the asymmetric tree, none of the 50 MSAs from the L-INS-i approach recovered a true tree, neither did the 3-way alignment approach (Table 3). However, RFd is smaller for the 3-way alignment approach (mean RFd = 5) than the L-NS-i approach (RFd = 6). The difference is statistically significant based on a paired-sample t-test (t = 3.6293, DF = 49, p = 0.0007). This difference between the L-INS-i and the 3-way alignment approach is also consistent with the results in Table 1.

We also performed the same comparison of phylogenetic results from the 16-taxa trees (Table 4). We compared the accuracy of reconstructed phylogenetic tree from the FFT-NS-1, FFT-NS-2, L-INS-i and 3-way alignment methods. The results are similar to those in Table 2, i.e., the guide trees reconstructed from 6-tuple similarities (FFT-NS-1 and FFT-

NS-2) are worse than those reconstructed from pairwise alignment (L-INS-i) or 3-way alignment (Table 4). When the true tree was used as the guide tree, the resulting MSA recovered the true tree except in the case of the asymmetric tree (AS tree in Table 4). Thus, the true tree is indeed the best guide tree, although there are controversies on this seemingly self-evident statement as we discuss later.

Table 4: : result of comparing the reconstructed phylogenetic trees by PhyML using MSA generated by FFT-NS-1, FFT-NS-2, L-INS-i, MAFFT using 3-way alignment guide trees, and true tree as input to MAFFT for 16-taxa trees including Half-Symmetric, Symmetric, Half-Asymmetric and Asymmetric trees. Column headings are the same as in Table 2.

	H-S tree			**S tree**			**H-AS tree**			**AS tree**		
Guide tree	N_{true}	RFd	SE_{RFd}	N_{true}	RFd	SE_{RFd}	N_{true}	RFd	SE_{RFd}	N_{true}	RFd	SE_{RFd}
True tree	50	0	0	50	0	0	50	0	0	18	2.40	0.3182
FFT-NS-1	45	0.12	0.0679	11	3.28	0.3865	40	0.44	0.1314	0	11.16	0.4203
FFT-NS-2	47	0.20	0.0857	13	3.28	0.3908	43	0.32	0.1193	0	10.52	0.4345
L-INS-i	50	0	0	45	0.20	0.0857	44	0.28	0.1144	0	6.60	0.3758
3-Way	50	0	0	48	0.12	0.0887	46	0.16	0.0755	1	6.36	0.3114

For the half-symmetric tree (H-S tree) and half-asymmetric tree (H-AS tree), because of reduced sequence divergence, the true tree was recovered from most of the data sets. Even the FFT-NS-1 and the FFT-NS-2 approaches perform well, recovering 90% and 94% of the true trees, respectively in the H-S tree case and 80% and 81% in the H-AS case (Table 4).

For the symmetric tree (S tree in Table 4), the FFT-NS-1 and the FFT-NS-2 approaches recovered few true trees, but the L-INS-i and the 3-way approaches recovered most of the true trees (Table 4). RFd is slightly smaller for the 3-way alignment approach than for the

L-INS-i approach, but the difference is not significant (paired-sample t-test, t=0.7035, DF = 49, p = 0.2425). For the asymmetric tree (AS Tree in Table 4), both the L-INS-i and the 3-way alignment approach recovered few true trees. RFd is slightly smaller for the 3-way alignment approach, but the difference is not significant (paired-sample t-test, t = 0.4928, DF = 49, p = 0.6244).

5.3 Accuracy of the guide tree affects the accuracy of the final tree from MSA

We evaluated the hypothesis that the quality of guide trees directly influences the phylogenetic accuracy by directly examining the association in RFd between the guide tree and the final phylogenetic reconstruction from PhyML. For the 8-taxa tree, we combined results from two simulations (symmetric and asymmetric trees) and from the two types of guide trees (the L-INS-i and 3-way alignment approaches), so that there 200 guide trees and 200 PhyML trees from the resulting MSA. There is a strong association in RFd between the guide tree and the PhyML-reconstructed final tree (Figure 6). When the guide tree has an identical topology as the true tree (RFd = 0 between the two), the resulting PhyML-reconstructed tree also tend to have the topology of the true tree; when the guide tree deviates much from the true tree, so does the resulting PhyML-reconstructed tree (Figure 6).

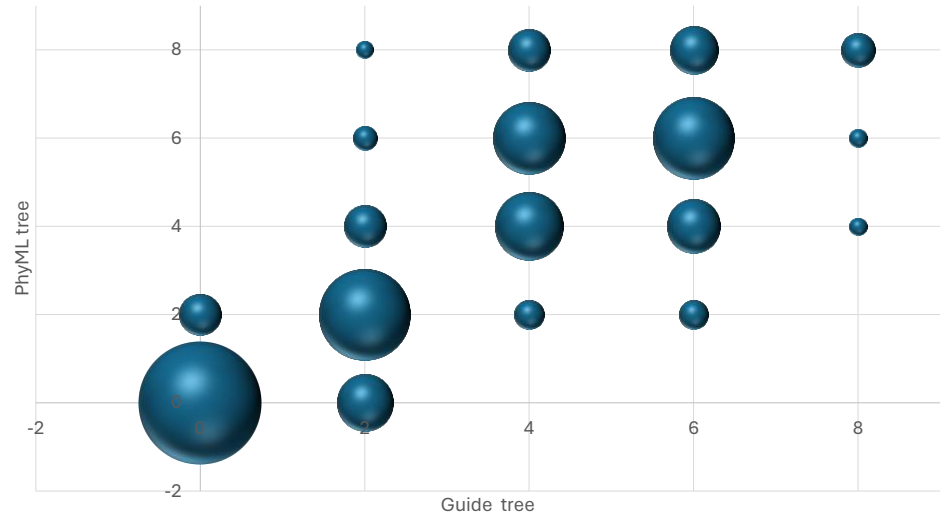

Figure 6: The relationship between RFd of a guide tree and RFd of the corresponding PhyML tree from the resulting MSA, based on the 8-taxa symmetric and asymmetric trees. The bubble plot was used because many points overlap each other. The relationship is highly significant ($n = 200$, $r = 0.83143$, $p < 0.0001$).

We have done the same for 16-taxa trees (Figure 7), including the fast but inaccurate FFT-NS-1 and FFT-NS-2 options in MAFFT in addition to the L-INS-i and the 3-way alignment approaches. For each of these approaches, we combined the results from four simulations (the symmetric and asymmetric trees and the half-symmetric and half-asymmetric trees). Thus, each sub-figure in Figure 7 includes 200 guide trees and 200 PhyML trees. It is clear that the two fast and inaccurate options (that generate guide trees from 6-tuple similarities) produced both poor guide trees (large RFd values) as well as the final PhyML trees from the resulting MSA (Figure 7A-B) relative to the L-INS-i approach that generated the guide tree from local pairwise alignment (Figure 7C) or to the 3-way alignment approach (Figure 7D). However, for all the four approaches, the guide tree

quality strongly affects the accuracy of the final PhyML tree (Figure 7). The relationship between the guide tree RFd and PhyML tree RFd are all highly significant ($p < 0.0001$).

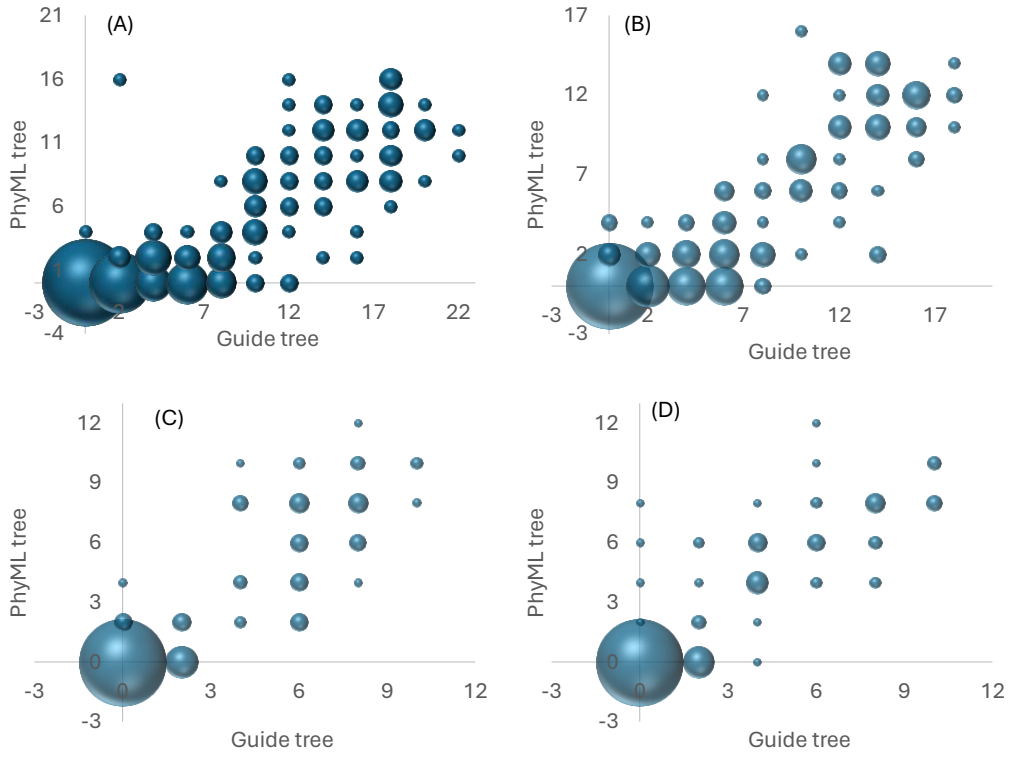

Figure 7: The relationship between RFd of guide trees and RFd of phylogenetic trees generated by PhyML, based on 16-taxa trees. (A) The FFT-NS-1 approach in which the guide tree was generated from 6-tuple similarities. (B). The FFT-NS-2 approach where the guide tree is recomputed from the first round of multiple sequence alignment. (C) The L-INS-i approach where the guide tree is from local pairwise alignment. (D) The 3-way alignment approach where the guide tree was described in the Materials and Methods section. The bubble plot was used because many points overlapped with each other.

5.4 The sum-of-pair score may not be a good criterion for choosing the best MSA

There are two criteria that can be used to evaluate the quality of an MSA. The first is phylogenetic accuracy, i.e., the MSA that results in the most accurate phylogenetic reconstruction is the best MSA. This criterion is conceptually fine but not computationally practical. Also, one generally can evaluate phylogenetic accuracy only for simulated sequences with a known true tree. The second criterion is the sum-of-pair score (SPS) or

its variations such as weighted SPS [1,67,68]. This weighted SPS is used in the default option in MUSCLE and the G-INS-i and L-INS-i options in MAFFT. The criterion is computationally practical and expected to be generally consistent with the first criterion. Our results in the previous section show that, when an MSA is generated with the true tree as a guide tree, this MSA tends to result in the most accurate phylogenetic reconstruction. It is therefore interesting to know if an MSA generated with the true tree as a guide tree also leads to the highest SPS.

We compared SPS from two types of MSAs, one generated using the true tree as the guide tree (the "trueTree" approach) and the other generated using the accurate L-INS-i option (the "L-INS-i" approach which creates the guide tree based on local pairwise alignment) in MAFFT. The input sequences are simulated with symmetric and asymmetric trees as before, with an average sequence length of 500 amino acids. Each simulated data set generated two MSAs, one from the trueTree approach and the other from the L-INS-i approach. The two MSAs were also used for phylogenetic reconstruction using PhyML. When the true tree was used as a guide tree, the final PhyML tree is closer to the true tree (smaller RFd) than that from the L-INS-i approach (Figure 8A and 8C). This difference is highly significant based on paired-sample t-test ($p < 0.0001$ for data in both Figure 8A and Figure 8C. Thus, when phylogenetic accuracy is used as a criterion, the MSA resulting from using the true tree as a guide tree is better than MSA from the L-INS-i approach.

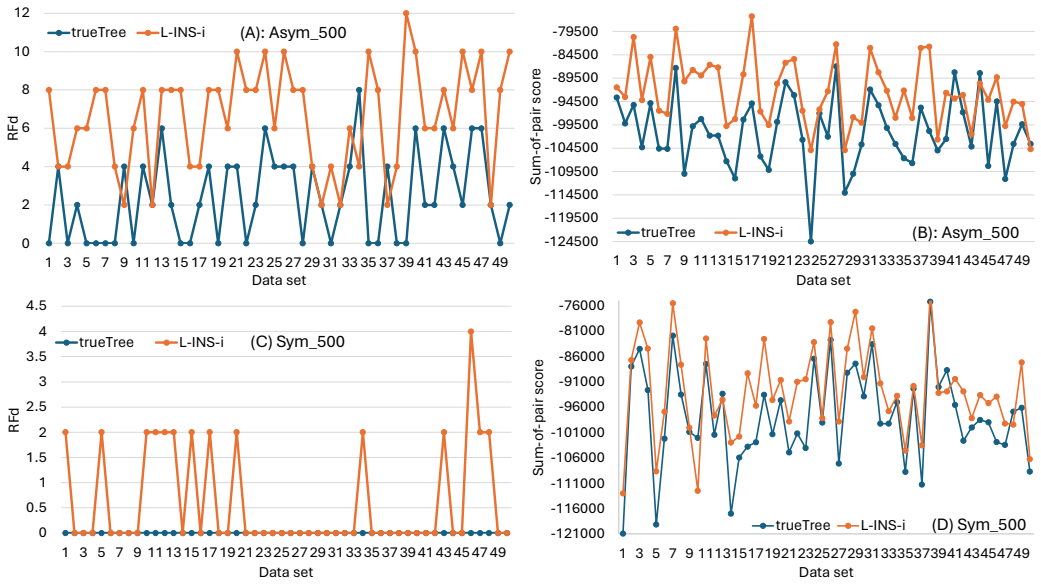

Figure 8: Conflict between two criteria (phylogenetic accuracy and sum-of-pair score) in choosing the best MSA. Sequences with average length of 500 are simulated with the 16-taxa symmetric and asymmetric trees. Two MSAs were produced from each set of sequences, one with the true tree as the guide tree (trueTree) and the other with the L-INS-i approach (L-INS-i) which generates a guide tree from local pairwise alignment. Sum-of-pair score was calculated for each MSA. PhyML was used for phylogenetic reconstruction for each MSA, and the Robinson-Foulds distance (RFd) was calculated between the true tree and the PhyML tree. The trueTree approach produced PhyML trees closer to the true tree than the L-INS-i approach for both the asymmetric tree (A) and symmetric tree (C). However, the L-INS-i approach produced MSAs with higher sum-of-pair scores than the trueTree approach, which is true for both the asymmetric tree (B) and symmetric tree (D).

Surprisingly, SPS is higher for the MSA from the L-INS-i than the MSA from using the true tree as the guide tree (Figure 8 B and 8C). This is consistent for both the asymmetric tree and the symmetric tree. The difference is highly significant based on paired-sample t-test ($p < 0.0001$). This creates a conflict in choosing the best MSA. With the criterion of phylogenetic accuracy as a criterion, then the MSA from the trueTree approach is better; with the SPS as the criterion, the MSA from the L-INS-i approach is better.

To further confirm the results in Figure 8, we simulated longer sequences with an average length of 1500 amino acids according to those symmetric and asymmetric trees. The same computation was repeated. For asymmetric trees (Figure 9A), the trueTree approach (MSA obtained with the true tree as a guide tree) generated PhyML trees more similar

than those generated from the L-INS-i approach. This difference in RFd between the trueTree and the L-INS-i approach is highly significant (paired-sample t-test, $p < 0.0001$). The longer sequence length allowed both the trueTree and the L-INS-i approach to recover all symmetric true trees (Figure 9C). Thus, the criterion of phylogenetic accuracy still favors the trueTree approach over the L-INS-i approach. The relevant scatter plots for dataset used in Figure 8 and Figure 9 are provided in Suppfile1.xlsx sheet "SP-score".

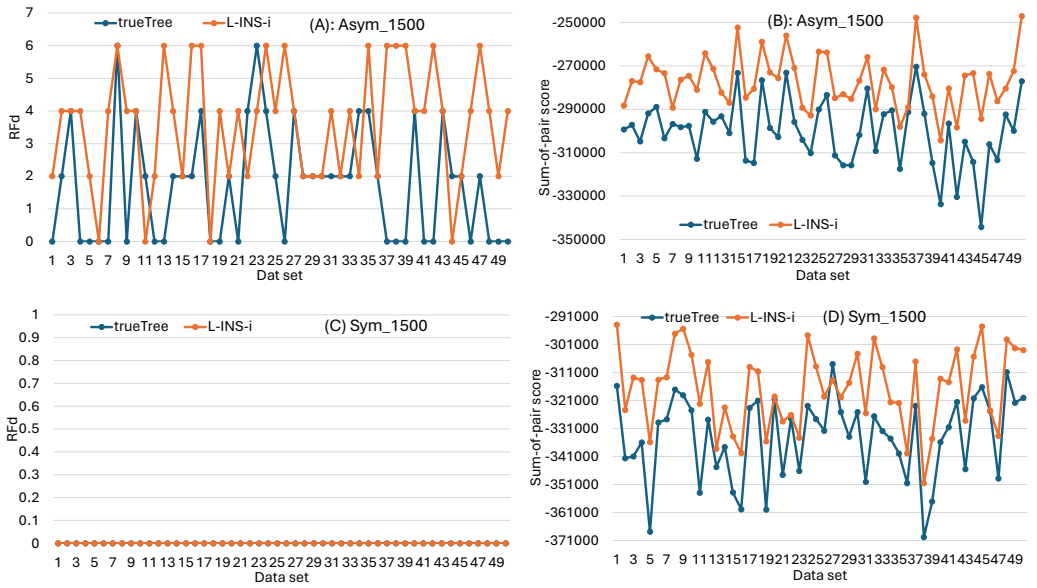

Figure 9: Conflict between two criteria (phylogenetic accuracy and sum-of-pair score) in choosing the best MSA. Sequences with average length of 1500 are simulated with the 16-taxa symmetric and asymmetric trees. Computations are same as Figure 8. The trueTree approach produced PhyML trees closer to the true tree than the L-INS-i approach for both the asymmetric tree (A) and both algorithms produced PhyML trees identical to true tree for symmetric tree (C). However, the L-INS-i approach produced MSAs with higher sum-of-pair scores than the trueTree approach, which is true for both the asymmetric tree (B) and symmetric tree (D).

In contrast, SPS is higher for MSA from the L-INS-i approach than from the trueTree approach (Figure 9B-C), consistent with results in Figure 8. Thus, the SPS criterion tends to favor MSAs that do not generate the best tree. The conflict between the two criteria appears real.

5.5 Performance of 3-way alignment on Benchmark dataset

We performed a quick evaluation of the performance of the 3-way alignment approach by using the BAliBASE [69] benchmark datasets of protein sequences. We selected 60 highly diverged reference alignments including 1) the first 20 sets in in RV11 (BB110001-BB11020), 2) 20 randomly chosen sets in RV30, and 3) 20 arbitrarily chosen sets from RV12 (BB12002-BB12006, BB12009, BB12010, BB12012-BB12024). These MSAs were corroborated with other information such as protein structure and may be considered as the best approximation of the true alignment. From each of these 60 sets of protein sequences, we generated two additional alignments, one from MAFFT with the accurate L-INS-i option and the other from the 3-way alignment approach. These three MSAs are referred to as BAliBase, L-INS-i and 3-Way. From each alignment, a PhyML tree is built with the default LG model and the simultaneous optimization of tree topology, branch lengths, and rates. The three resulting trees were also designated BAliBase, L-INS-i and 3-Way, respectively. The BAliBase tree was taken as the best approximation of the true tree. The RFd vale was calculated between the BAliBase tree and the L-INS-i tree and between the BAliBase tree and the 3-Way tree. The results are similar to those with the simulated sequences. The mean RFd is 3.03333 between the BAliBase and the L-INS-i trees and 2.66667 between the BAliBase and 3-Way trees. The two are marginally significant based on a one-tailed paired-sample test ($t = 1.6638$, $DF = 59$, one-tailed $p = 0.0507$).

6 Discussion

In this research, we explore the potential of 3-way alignment in enhancing the accuracy of guide trees, which are crucial for multiple sequence alignment (MSA). Our primary focus is on MAFFT, one of the most widely used MSA tools, and how it can benefit from guide trees generated through 3-way alignment. The core hypothesis of our study is that 3-way alignment can produce guide trees of higher accuracy compared to traditional methods, subsequently leading to improved MSAs when used with MAFFT.

There are disagreements involving guide trees in progressive multiple sequence alignment. First, what is the best guide tree for the progressive multiple sequence alignment? Second, how to obtain the best guide tree? There are also disagreements on what criterion should be used in choosing the optimal MSA. If phylogenetic reconstruction is the ultimate goal, then phylogenetic accuracy obviously should be the ultimate criterion for choosing the best MSA. Given that this criterion cannot be practically used, does the SPS criterion serve as a good proxy? This study aims to address these questions, with a focus on highly diverged sequences which are hard to align.

6.1 Is the true tree the best guide tree for the progressive multiple sequence alignment?

One would tend to assume that the true tree should be the best guide tree. However, this assumption conflicts with the principle that multiple sequence alignment should start with the most similar sequences and progress towards less similar sequences (R. C.

Edgar, pers. comm.). This conflict is illustrated with the following true tree: ((S1:0.001, S2:0.1):0.001, (S3:0.001,S4:0.1):0.001); S1 and S3 are the most similar sequences, with a pairwise distance of only 0.003. They should therefore be aligned first following the principle stated above. However, the true tree would not allow S1 and S3 to be aligned first and would force S1 and S2 (or S3 and S4) to be aligned first. This is one of the reasons for widely used multiple sequence alignment programs such as MAFFT [29] and MUSCLE [53] to use a modified version of UPGMA to reconstruct the guide tree, because UPGMA will cluster S1 and S3 together. Such a guide tree ensures that S1 and S3 would be aligned first. Will such a guide tree and the resulting MSA cause phylogenetic distortion in the final reconstructed tree? Our results, especially those in Table 3, Figure 6 and Figure 7, suggest that, if the accuracy of the final phylogeny is taken as a criterion, the true tree indeed is the best guide tree. Version 5 of MUSCLE [70] includes an ensemble of trees for exploring the consequence of the resulting MSA on phylogenetic reconstruction. This would help phylogeneticists to appreciate the variation in guide trees and the variation in the resulting reconstructed phylogenies.

6.2 How to obtain the best guide tree?

If we agree that the true tree is the best guide tree, then how to obtain a guide that is the best approximation of this true tree? In this research, we explore the potential of 3-way alignment in improving the accuracy of the guide tree. Our results are consistent with the

hypothesis that 3-way alignment can produce better guide trees (exhibiting lower RFd with true tree) compared to guide trees from PSA or k-tuple approaches, leading to improved MSAs and the phylogenetic reconstruction based on the MSAs (Tables 1-4). Two lines of evidence were presented to support the conclusion that the guide tree from the 3-way alignment (3-WAY) is better than that generated from the most accurate option in MAFFT (L-INS-i, which creates the initial guide tree from local pairwise alignment). First, the guide tree from the 3-WAY approach is closer to the true tree than that from the L-INS-i approach. Second, when the MSA generated from 3-WAY and L-INS-i guide trees were fed to PhyML for phylogenetic reconstruction, the MSA from the 3-WAY guide tree produced PhyML trees closer to the true tree than that from the L-INS-i approach.

While guide trees based on k-tuple similarities in MAFFT are poor, the guide tree from the L-INS-i option in MAFFT is very good, and the 3-way alignment may be useful only in the most challenging cases with extremely diverged sequences. Sequences simulated from our half-symmetric and half-asymmetric trees are comparable in divergence to many real homologous amino acid sequences, yet MAFFT performed well with these sequences. Only with the highly diverged sequences simulated from the asymmetric trees did MAFFT experienced difficulties in generating quality MSA (Tables 1-4).

6.3 Is sum-of-pair score or its derivative a good criterion for choosing the best MSA?

The best MSA should produce the true tree, especially when the objective of sequence alignment is accurate phylogenetic reconstruction. However, phylogenetic accuracy cannot be used directly as a criterion because the true tree is unknown except in simulated sequences. One would hope that the sum-of-pair score (SPS) or its variations such as weighted SPS, which is computationally practical and widely used as a criterion for choosing the best MSA, would be equivalent to the criterion of phylogenetic accuracy. In other words, the MSA with the highest SPS is also the MSA that would result in the most accurate phylogeny. Our results (Figures 5-6) suggest that this is not the case. From each set of our simulated sequences, two MSAs were produced, one with the true tree as the guide tree (trueTree) and the other using the guide tree from the L-INS-i approach (L-INS-i). When these two MSAs were fed to PhyML for phylogenetic reconstruction, the MSA from the trueTree approach produced trees more similar to the true tree than that from the L-INS-i approach. However, the latter has significantly higher SPS than the former. Thus, the two criteria are inconsistent.

It is difficult to provide a specification of time complexity with the Carrillo-Lipman algorithm. This algorithm for the 3-way alignment includes three pair-wise alignments, followed by a simplified 3-way alignment that does not need to visit all cells in the cube. The time complexity for this last step is difficult to express because the time required for this step depends on the nature of the three sequences. If the three sequences are nearly

identical, then we have the best scenario and the time required for this step would be almost linear. If the three sequences are highly diverged and differ much in length (i.e., many indel events), then the time requirement for this step would be similar to the plain 3-way alignment by dynamic programming. Because we aim to improve sequence alignment of highly diverged sequences, the time saved from the Carrillo-Lipman algorithm is not substantial.

One time-saving protocol is to first identify regions of consistency among the three pairwise alignments in each 3-way alignment, using the approach proposed by Gotoh [71]. The regions of consistency do not need 3-way alignment. They can serve as anchors so that one only needs to do 3-way alignment for sequence segments between such anchors.

7 Conclusion

Multiple Sequence Alignment (MSA) is a crucial problem in bioinformatics and other related fields. Achieving an accurate MSA for a given set of sequences is still challenging and many studies have tried to propose a method to improve alignment because the accuracy of MSA affect the result of next analysis on the aligned sequences. For example, phylogenetic reconstruction is highly dependent on the input MSA. In this thesis, we assessed the effect of MSA on the accuracy of reconstructed phylogenetic trees. Since progressive alignment is one of the mostly used approaches for MSA, we focused on improving guide tree which is a crucial component of progressive algorithms. We showed the role of guide tree on output MSA by different simulated datasets.

Generally, guide tree is reconstructed based on a distance matrix calculated by pairwise comparison between sequences. Since distance-based algorithms like UPGMA and NJ are highly dependent on the input distance matrix, we tried to generate more accurate distance matrix using 3-way alignment instead of pairwise alignment. To improve the time complexity of 3-way alignment we applied Carrillo-Lipman algorithm which decreases the searching area of 3-dimensional dynamic programming.

We assessed the performance of three algorithms of MAFFT which is popular alignment tools. Our results showed that 3-way alignment has the potential to generate more accurate guide trees compared to the default guide trees generated by MAFFT. Moreover, we showed the effect of guide tree in the accuracy of progressive approach. This

comparison was critical in highlighting the varying degrees of accuracy among different guide tree construction techniques. Our analysis revealed a clear trend: more accurate guide trees, such as those derived from the true topology or constructed using the advanced 3-way alignment method, consistently resulted in MSAs that exhibited a stronger phylogenetic signal. The detailed results presented in this study serve as a foundation for the future research endeavors aimed at optimizing MSA techniques and improving the accuracy of phylogenetic analyses.

The overarching conclusion from our analysis is that 3-way alignment offers improvement in the construction of guide trees, which in turn leads to more accurate phylogenetic reconstructions, especially in cases involving complex and highly diverged sequences. This finding has significant implications for the field of bioinformatics and evolutionary biology, suggesting that incorporating 3-way alignment into phylogenetic analysis could lead to more precise understanding of evolutionary histories and relationships. The comprehensive data presented in this section not only supports this conclusion but also provides a detailed roadmap for future research and application in enhancing phylogenetic reconstruction methodologies.

Moreover, we undertook a detailed evaluation to understand the impact of the accuracy of guide trees on the quality of multiple sequence alignments and subsequent phylogenetic analysis. A critical part of this evaluation involved using the true topology

of each dataset as an input for MAFFT. This approach allowed us to establish a benchmark for assessing the effectiveness of various guide tree construction methods.

The RFd of trees generated by PhyML served as a key metric in our evaluation. By calculating the RFd for trees derived from MSAs based on different guide trees, we were able to quantitatively assess the impact of guide tree accuracy on phylogenetic reconstruction. The lower RFd values associated with MSAs derives from more accurate guide trees, particularly those aligning closely with the topology, highlighted the importance of precision in guide tree construction.

This analysis not only demonstrates the critical role of accurate guide trees in enhancing the quality of MSAs but also provides valuables insights into the potential improvements in phylogenetic signal when using advanced alignment strategies.

References

1. Edgar, R.C.; Batzoglou, S. Multiple Sequence Alignment. *Current Opinion in Structural Biology* **2006**, *16*, 368–373, doi:10.1016/j.sbi.2006.04.004.

2. Gotoh, O. Multiple Sequence Alignment: Algorithms and Applications. *Advances in Biophysics* **1999**, *36*, 159–206, doi:10.1016/S0065-227X(99)80007-0.

3. Haque, W.; Aravind, A.; Reddy, B. Pairwise Sequence Alignment Algorithms: A Survey. In Proceedings of the Proceedings of the 2009 conference on Information Science, Technology and Applications; Association for Computing Machinery: New York, NY, USA, March 20 2009; pp. 96–103.

4. Altschul, S.F.; Madden, T.L.; Schäffer, A.A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D.J. Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. *Nucleic Acids Research* **1997**, *25*, 3389–3402, doi:10.1093/nar/25.17.3389.

5. Xia, X. Post-Alignment Adjustment and Its Automation. *Genes* **2021**, *12*, 1809, doi:10.3390/genes12111809.

6. Hall, B.G. Comparison of the Accuracies of Several Phylogenetic Methods Using Protein and DNA Sequences. *Molecular Biology and Evolution* **2005**, *22*, 792–802, doi:10.1093/molbev/msi066.

7. Goldman, N. Effects of Sequence Alignment Procedures on Estimates of Phylogeny. *BioEssays* **1998**, *20*, 287–290, doi:10.1002/(SICI)1521-1878(199804)20:4<287::AID-BIES4>3.0.CO;2-N.

8. Morrison, D.A.; Ellis, J.T. Effects of Nucleotide Sequence Alignment on Phylogeny Estimation: A Case Study of 18S rDNAs of Apicomplexa. *Molecular biology and evolution* **1997**, *14*, 428–441.

9. Thompson, J.D.; Plewniak, F.; Poch, O. A Comprehensive Comparison of Multiple Sequence Alignment Programs. *Nucleic Acids Research* **1999**, *27*, 2682–2690, doi:10.1093/nar/27.13.2682.

10. Regier, J.C.; Shultz, J.W.; Zwick, A.; Hussey, A.; Ball, B.; Wetzer, R.; Martin, J.W.; Cunningham, C.W. Arthropod Relationships Revealed by Phylogenomic Analysis of Nuclear Protein-Coding Sequences. *Nature* **2010**, *463*, 1079–1083, doi:10.1038/nature08742.

11. Xia, X. PhyPA: Phylogenetic Method with Pairwise Sequence Alignment Outperforms Likelihood Methods in Phylogenetics Involving Highly Diverged Sequences. *Molecular Phylogenetics and Evolution* **2016**, *102*, 331–343, doi:10.1016/j.ympev.2016.07.001.

12. Noah, K.E.; Hao, J.; Li, L.; Sun, X.; Foley, B.; Yang, Q.; Xia, X. Major Revisions in Arthropod Phylogeny Through Improved Supermatrix, With Support for Two Possible

Waves of Land Invasion by Chelicerates. *Evol Bioinform Online* **2020**, *16*, 1176934320903735, doi:10.1177/1176934320903735.

13. Bellamy-Royds, A.B.; Turcotte, M. Can Clustal-Style Progressive Pairwise Alignment of Multiple Sequences Be Used in RNA Secondary Structure Prediction? *BMC Bioinformatics* **2007**, *8*, 190, doi:10.1186/1471-2105-8-190.

14. Masoumi, B.; Turcotte, M. Simultaneous Alignment and Structure Prediction of Three RNA Sequences. *International Journal of Bioinformatics Research and Applications* **2005**, *1*, 230–245, doi:10.1504/IJBRA.2005.007581.

15. Xia, X. Phylogenetic Relationship Among Horseshoe Crab Species: Effect of Substitution Models on Phylogenetic Analyses. *Systematic Biology* **2000**, *49*, 87–100, doi:10.1080/10635150050207401.

16. Xia, X.; Xie, Z.; Kjer, K.M. 18S Ribosomal RNA and Tetrapod Phylogeny. *Systematic Biology* **2003**, *52*, 283–295, doi:10.1080/10635150390196948.

17. Xia, X.; Xie, Z.; Salemi, M.; Chen, L.; Wang, Y. An Index of Substitution Saturation and Its Application. *Molecular Phylogenetics and Evolution* **2003**, *26*, 1–7, doi:10.1016/S1055-7903(02)00326-3.

18. Xia, X. *A Mathematical Primer of Molecular Phylogenetics*; CRC Press, 2020; ISBN 978-0-429-75931-4.

19. Feng, D.-F.; Doolittle, R.F. Progressive Sequence Alignment as a Prerequisite to Correct Phylogenetic Trees. **1987**, *25*, 351–360.

20. Zhan, Q.; Ye, Y.; Lam, T.-W.; Yiu, S.-M.; Wang, Y.; Ting, H.-F. Improving Multiple Sequence Alignment by Using Better Guide Trees. *BMC Bioinformatics* **2015**, *16*, S4, doi:10.1186/1471-2105-16-S5-S4.

21. Capella-Gutiérrez, S.; Gabaldón, T. Measuring Guide-Tree Dependency of Inferred Gaps in Progressive Aligners. *Bioinformatics* **2013**, *29*, 1011–1017, doi:10.1093/bioinformatics/btt095.

22. Penn, O.; Privman, E.; Landan, G.; Graur, D.; Pupko, T. An Alignment Confidence Score Capturing Robustness to Guide Tree Uncertainty. *Molecular Biology and Evolution* **2010**, *27*, 1759–1767, doi:10.1093/molbev/msq066.

23. Nelesen, S.; Liu, K.; Zhao, D.; Linder, C.R.; Warnow, T. The Effect of the Guide Tree on Multiple Sequence Alignments and Subsequent Phylogenetic Analyses. In *Biocomputing 2008*; WORLD SCIENTIFIC, 2007; pp. 25–36 ISBN 978-981-277-608-2.

24. Ye, Y.; Cheung, D.W.; Wang, Y.; Yiu, S.-M.; Zhan, Q.; Lam, T.-W.; Ting, H.-F. GLProbs: Aligning Multiple Sequences Adaptively. In Proceedings of the Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical

Informatics; Association for Computing Machinery: New York, NY, USA, September 22 2013; pp. 152–160.

25. Notredame, C.; Higgins, D.G.; Heringa, J. T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence alignment11Edited by J. Thornton. *Journal of Molecular Biology* **2000**, *302*, 205–217, doi:10.1006/jmbi.2000.4042.

26. Thompson, J.D.; Higgins, D.G.; Gibson, T.J. CLUSTAL W: Improving the Sensitivity of Progressive Multiple Sequence Alignment through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice. *Nucleic Acids Research* **1994**, *22*, 4673–4680, doi:10.1093/nar/22.22.4673.

27. Kruspe, M.; Stadler, P.F. Progressive Multiple Sequence Alignments from Triplets. *BMC Bioinformatics* **2007**, *8*, 254, doi:10.1186/1471-2105-8-254.

28. Chien, R.-T.; Liao, Y.-L.; Wang, C.-A.; Li, Y.-C.; Lu, Y.-C. Three-Dimensional Dynamic Programming Accelerator for Multiple Sequence Alignment. In Proceedings of the 2018 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC); October 2018; pp. 1–5.

29. Katoh, K.; Misawa, K.; Kuma, K.; Miyata, T. MAFFT: A Novel Method for Rapid Multiple Sequence Alignment Based on Fast Fourier Transform. *Nucleic Acids Research* **2002**, *30*, 3059–3066, doi:10.1093/nar/gkf436.

30. Henikoff, S.; Henikoff, J.G. Amino Acid Substitution Matrices from Protein Blocks. *Proceedings of the National Academy of Sciences* **1992**, *89*, 10915–10919, doi:10.1073/pnas.89.22.10915.

31. Dayhoff, M.; Schwartz, R.; Orcutt, B. A Model of Evolutionary Change in Proteins. *Atlas of protein sequence and structure* **1978**, *5*, 345–352.

32. Mount, D.W. Using BLOSUM in Sequence Alignments. *Cold Spring Harb Protoc* **2008**, *2008*, pdb.top39, doi:10.1101/pdb.top39.

33. Zachariah, M.A.; Crooks, G.E.; Holbrook, S.R.; Brenner, S.E. A Generalized Affine Gap Model Significantly Improves Protein Sequence Alignment Accuracy. *Proteins: Structure, Function, and Bioinformatics* **2005**, *58*, 329–338, doi:10.1002/prot.20299.

34. Taylor, W.R. A Non-Local Gap-Penalty for Profile Alignment. *Bulletin of Mathematical Biology* **1996**, *58*, 1–18, doi:10.1016/0092-8240(95)00303-7.

35. Cartwright, R.A. Logarithmic Gap Costs Decrease Alignment Accuracy. *BMC Bioinformatics* **2006**, *7*, 527, doi:10.1186/1471-2105-7-527.

36. Miklós, I.; Lunter, G.A.; Holmes, I. A "Long Indel" Model For Evolutionary Sequence Alignment. *Molecular Biology and Evolution* **2004**, *21*, 529–540, doi:10.1093/molbev/msh043.

37. Jennings, A.J.; Edge, C.M.; Sternberg, M.J.E. An Approach to Improving Multiple Alignments of Protein Sequences Using Predicted Secondary Structure. *Protein Engineering, Design and Selection* **2001**, *14*, 227–231, doi:10.1093/protein/14.4.227.

38. Shi, J.; Blundell, T.L.; Mizuguchi, K. FUGUE: Sequence-Structure Homology Recognition Using Environment-Specific Substitution Tables and Structure-Dependent Gap penalties11Edited by B. Honig. *Journal of Molecular Biology* **2001**, *310*, 243–257, doi:10.1006/jmbi.2001.4762.

39. Lesk, A.M.; Levitt, M.; Chothia, C. Alignment of the Amino Acid Sequences of Distantly Related Proteins Using Variable Gap Penalties. *Protein Eng Des Sel* **1986**, *1*, 77–78, doi:10.1093/protein/1.1.77.

40. Pascarella, S.; Argos, P. Analysis of Insertions/Deletions in Protein Structures. *Journal of Molecular Biology* **1992**, *224*, 461–471, doi:10.1016/0022-2836(92)91008-D.

41. Wrabl, J.O.; Grishin, N.V. Gaps in Structurally Similar Proteins: Towards Improvement of Multiple Sequence Alignment. *Proteins: Structure, Function, and Bioinformatics* **2004**, *54*, 71–87, doi:10.1002/prot.10508.

42. Needleman, S.B.; Wunsch, C.D. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of molecular biology* **1970**, *48*, 443–453.

43. Eddy, S.R. What Is Dynamic Programming? *Nat Biotechnol* **2004**, *22*, 909–910, doi:10.1038/nbt0704-909.

44. Hirschberg, D.S. A Linear Space Algorithm for Computing Maximal Common Subsequences. *Commun. ACM* **1975**, *18*, 341–343, doi:10.1145/360825.360861.

45. Smith, T.F.; Waterman, M.S. Identification of Common Molecular Subsequences. *J Mol Biol* **1981**, *147*, 195–197, doi:10.1016/0022-2836(81)90087-5.

46. Carrillo, H.; Lipman, D. The Multiple Sequence Alignment Problem in Biology. *SIAM J. Appl. Math.* **1988**, *48*, 1073–1082, doi:10.1137/0148063.

47. Stoye, J. Multiple Sequence Alignment with the Divide-and-Conquer Method. *Gene* **1998**, *211*, GC45–GC56, doi:10.1016/S0378-1119(98)00097-3.

48. Gusfield, D. Efficient Methods for Multiple Sequence Alignment with Guaranteed Error Bounds. *Bulletin of Mathematical Biology* **1993**, *55*, 141–154, doi:10.1016/S0092-8240(05)80066-7.

49. Lee, C.; Grasso, C.; Sharlow, M.F. Multiple Sequence Alignment Using Partial Order Graphs. *Bioinformatics* **2002**, *18*, 452–464, doi:10.1093/bioinformatics/18.3.452.

50. Notredame, C.; Higgins, D.G. SAGA: Sequence Alignment by Genetic Algorithm. *Nucleic Acids Research* **1996**, *24*, 1515–1524, doi:10.1093/nar/24.8.1515.

51. Kim, J.; Pramanik, S.; Chung, M.J. Multiple Sequence Alignment Using Simulated Annealing. *Bioinformatics* **1994**, *10*, 419–426, doi:10.1093/bioinformatics/10.4.419.

52. DIALIGN: Finding Local Similarities by Multiple Sequence Alignment. | Bioinformatics | Oxford Academic Available online: https://academic.oup.com/bioinformatics/article/14/3/290/224213?login=false (accessed on 7 November 2023).

53. Edgar, R.C. MUSCLE: Multiple Sequence Alignment with High Accuracy and High Throughput. *Nucleic Acids Research* **2004**, *32*, 1792–1797, doi:10.1093/nar/gkh340.

54. Do, C.B.; Mahabhashyam, M.S.P.; Brudno, M.; Batzoglou, S. ProbCons: Probabilistic Consistency-Based Multiple Sequence Alignment. *Genome Res.* **2005**, *15*, 330–340, doi:10.1101/gr.2821705.

55. Probalign: Multiple Sequence Alignment Using Partition Function Posterior Probabilities | Bioinformatics | Oxford Academic Available online: https://academic.oup.com/bioinformatics/article/22/22/2715/197384?login=false (accessed on 7 November 2023).

56. Clustal W and Clustal X Version 2.0 | Bioinformatics | Oxford Academic Available online: https://academic.oup.com/bioinformatics/article/23/21/2947/371686?login=false (accessed on 6 November 2023).

57. Durbin, R.; Eddy, S.R.; Krogh, A.; Mitchison, G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*; 1st ed.; Cambridge University Press, 1998; ISBN 978-0-521-62041-3.

58. Miyazawa, S. A Reliable Sequence Alignment Method Based on Probabilities of Residue Correspondences. *Protein Engineering, Design and Selection* **1995**, *8*, 999–1009, doi:10.1093/protein/8.10.999.

59. Sahraeian, S.M.E.; Yoon, B.-J. PicXAA: Greedy Probabilistic Construction of Maximum Expected Accuracy Alignment of Multiple Sequences. *Nucleic Acids Research* **2010**, *38*, 4917–4928, doi:10.1093/nar/gkq255.

60. Gotoh, O. Alignment of Three Biological Sequences with an Efficient Traceback Procedure. *Journal of Theoretical Biology* **1986**, *121*, 327–337, doi:10.1016/S0022-5193(86)80112-6.

61. Huang, X. Alignment of Three Sequences in Quadratic Space. *SIGAPP Appl. Comput. Rev.* **1993**, *1*, 7–11, doi:10.1145/381771.381773.

62. Ly-Trong, N.; Naser-Khdour, S.; Lanfear, R.; Minh, B.Q. AliSim: A Fast and Versatile Phylogenetic Sequence Simulator For the Genomic Era 2021, 2021.12.16.472905.

63. Jones, D.T.; Taylor, W.R.; Thornton, J.M. The Rapid Generation of Mutation Data Matrices from Protein Sequences. *Bioinformatics* **1992**, *8*, 275–282, doi:10.1093/bioinformatics/8.3.275.

64. Guindon, S.; Dufayard, J.-F.; Lefort, V.; Anisimova, M.; Hordijk, W.; Gascuel, O. New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. *Systematic Biology* **2010**, *59*, 307–321, doi:10.1093/sysbio/syq010.

65. Robinson, D.F.; Foulds, L.R. Comparison of Phylogenetic Trees. *Mathematical Biosciences* **1981**, *53*, 131–147, doi:10.1016/0025-5564(81)90043-2.

66. Paradis, E.; Claude, J.; Strimmer, K. APE: Analyses of Phylogenetics and Evolution in R Language. *Bioinformatics* **2004**, *20*, 289–290, doi:10.1093/bioinformatics/btg412.

67. Gotoh, O. A Weighting System and Aigorithm for Aligning Many Phylogenetically Related Sequences. *Bioinformatics* **1995**, *11*, 543–551, doi:10.1093/bioinformatics/11.5.543.

68. Altschul, S.F.; Carroll, R.J.; Lipman, D.J. Weights for Data Related by a Tree. *Journal of Molecular Biology* **1989**, *207*, 647–653, doi:10.1016/0022-2836(89)90234-9.

69. Thompson, J.D.; Koehl, P.; Ripp, R.; Poch, O. BAliBASE 3.0: Latest Developments of the Multiple Sequence Alignment Benchmark. *Proteins: Structure, Function, and Bioinformatics* **2005**, *61*, 127–136, doi:10.1002/prot.20527.

70. Edgar, R.C. Muscle5: High-Accuracy Alignment Ensembles Enable Unbiased Assessments of Sequence Homology and Phylogeny. *Nat Commun* **2022**, *13*, 6968, doi:10.1038/s41467-022-34630-w.

71. Gotoh, O. Consistency of Optimal Sequence Alignments. *Bulletin of Mathematical Biology* **1990**, *52*, 509–525, doi:10.1016/S0092-8240(05)80359-3.