biomolecules

MDPI

Review

Developments in Algorithms for Sequence Alignment: A Review

Jiannan Chao [1], Furong Tang [2,3] and Lei Xu [3,*]

[1] Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China; jn.chao@outlook.com
[2] Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou 324003, China; furong.tang@hotmail.com
[3] School of Electronic and Communication Engineering, Shenzhen Polytechnic, Shenzhen 518055, China
* Correspondence: csleixu@szpt.edu.cn

Abstract: The continuous development of sequencing technologies has enabled researchers to obtain large amounts of biological sequence data, and this has resulted in increasing demands for software that can perform sequence alignment fast and accurately. A number of algorithms and tools for sequence alignment have been designed to meet the various needs of biologists. Here, the ideas that prevail in the research of sequence alignment and some quality estimation methods for multiple sequence alignment tools are summarized.

Keywords: heuristic alignment algorithms; alignment scoring; multiple sequence alignment; alignment refinement; alignment quality estimation

Citation: Chao, J.; Tang, F.; Xu, L. Developments in Algorithms for Sequence Alignment: A Review. *Biomolecules* **2022**, *12*, 546. https://doi.org/10.3390/biom12040546

Academic Editor: Lukasz Kurgan

Received: 11 February 2022
Accepted: 31 March 2022
Published: 6 April 2022

1. Introduction

The developments in sequencing technologies have enabled unprecedentedly fast sequencing speeds and large-scale sequencing capabilities. The increasing number of sequences are challenging the automated sequence analysis procedures [1,2]. Sequence alignment is one of the basic tasks in the processing of biological sequences, and the accuracy of alignment affects the subsequent analyses [3]. Phylogenetics, comparative genomics, and protein structure and function prediction all depend on sequence alignment to look for conserved regions [4–7].

Sequence alignment software usually inserts gaps between the nucleotides or amino acid residues in the sequences, so that as many similar sites as possible can be aligned. Finally, a character matrix with the same number of columns and rows that correspond to the number of the sequences is obtained. In this review, the pairwise sequence alignment algorithms and the corresponding scoring system, heuristic algorithms for multiple sequence alignment and their defects, and quality estimation methods used to test multiple sequence alignment software are reviewed. There have been several reviews for multiple sequence alignment, such as Refs [8,9]. In order to be distinct from the previous work, this review will try to present a general overview of the algorithms that prevail in this field and cover the work of the last several years.

2. Pairwise Sequence Alignment

Pairwise sequence alignment is the basis of multiple sequence alignment and mainly divided into local alignment and global alignment. The former is to find and align the similar local region, and the latter is end-to-end alignment. A commonly used global alignment algorithm is the Needleman–Wunsch algorithm [10], which has become the basic algorithm that is used in many types of multiple sequence alignment software. The algorithm usually consists of two steps: one is calculating the states of the dynamic programming matrix; and the other is tracking back from the final state to the initial state of the dynamic programming matrix to obtain the solution of alignment. Time and space

complexity of pairwise sequence alignment algorithms based on dynamic programming is $O(l_1 l_2)$, where l_1 and l_2 are the lengths of the two sequences to be aligned. Such overheads are acceptable for short sequences but not for sequences with more than several thousand sites. As a space-saving strategy of the dynamic programming algorithm, the Hirschberg algorithm [11] is able to complete alignment by the space complexity of $O(l)$ without any sacrifice of quality.

An optimal solution for the pairwise sequence alignment of very long sequences is usually impossible to find in practice. Heuristic algorithms can be used to reduce the time and space cost incurred by dynamic programming. For this, the most widely applied method is to limit the state transition and conduct the alignment in a smaller search space. Although heuristic algorithms do not guarantee that there will be no poor results, ideal alignments can be achieved in many types of software because the sequences to be aligned are usually quite similar.

In addition, the hidden Markov model (HMM) is also widely utilized in sequence alignment tools, such as HHalign [12], which can perform high accurate profile HMM alignment. In terms of sequence alignment, an HMM is a statistical model that describes probability distribution over biological sequences. According to the three problems that are interesting when using HMMs [13], the adoption of HMMs in sequence alignment has three corresponding issues: the scoring problem, the alignment problem, and the training problem [14]. The first two problems are about how likely a given HMM could generate a sequence and how the HMM could produce the corresponding alignment, and the third problem is about how to build the structure and estimate the parameters of the HMM based on given sequences, which could be either aligned or unaligned. The scoring parameters are implicitly trained, and thus, the alignment methods based on HMMs do not rely on explicit scoring systems, which makes them independent of the empirical scoring methods described in Section 2.3. HMM-based sequence alignment methods are gaining popularity nowadays. COVID-Align [15], one of the efforts to combat the COVID-19 pandemic, is an online alignment tool based on a profile HMM [16] estimated using HMMER [17] and about 2500 high-quality SARS-CoV-2 genomes. Additionally, MAGUS + eHMMs [18], an alignment tool for fragmentary sequences, chooses eHMM rather than MAFFT-addfragments [19], which adopts the Smith–Waterman algorithm [20], to add sequences to the backbone alignment because of its better performance in terms of alignment accuracy.

2.1. Divide and Conquer

One of the heuristic methods is based on divide and conquer. In such methods, homologous segments (or seeds) are found and used as the "anchors" for the alignment. Each anchor point can divide the dynamic programming matrix into four sub-matrices located at the four corners. Backtracking always goes toward the upper left direction, and these anchors are regarded as the waypoints that the optimal path must pass; therefore, the sub-matrices located at the lower left and upper right are useless and naturally disregarded. When more anchor points distributed throughout the sequences are found, the scale of the dynamic programming matrix can be greatly reduced, thereby reducing the time and space complexity (Figure 1A).

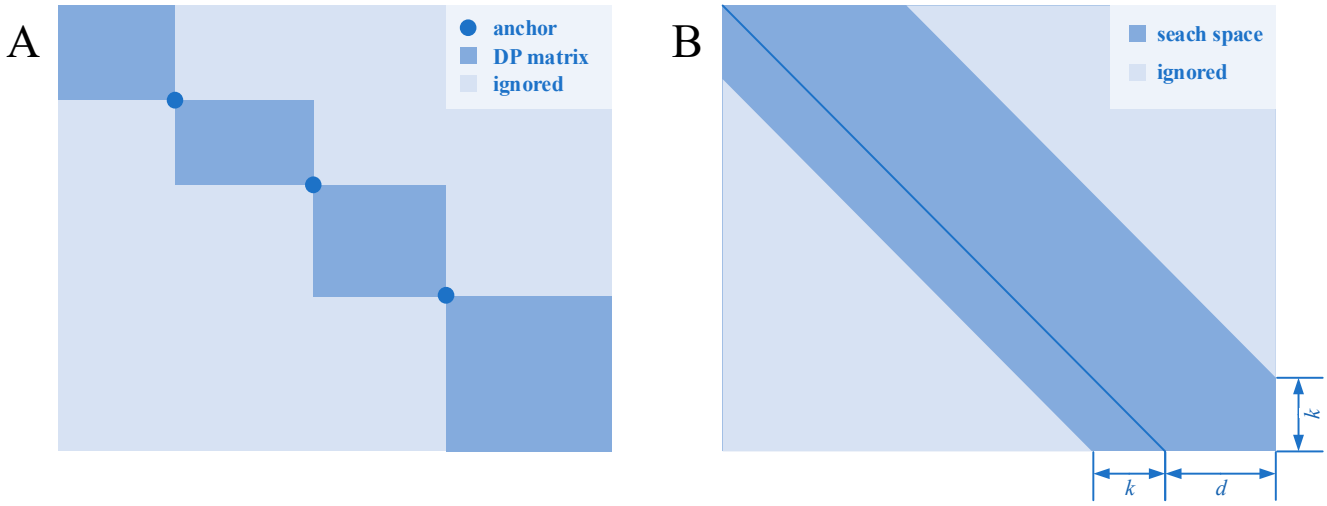

Figure 1. Two heuristic algorithms for pairwise sequence alignment. (**A**) A dynamic programming matrix, which is separated by several anchors, which is certain to be in the optimal path. (**B**) A shape-based bounded dynamic programming matrix in which the light blue block is calculation-free because these states are thought to be less likely to be in the optimal path.

Many alignment algorithms are designed to find such seeds to accelerate alignment. FASTA [21,22] and BLAST [23] are two of the classic ones. They both search for common substrings in a way similar to the Rabin–Karp algorithm [24]. Moreover, some other software, such as MUMmer [25,26], uses data structures, such as suffix tree [27], suffix array [28], or FM-index [29], to find homologous segments. Recently, Minimap2 [30], which also adopts this strategy, is gaining popularity and is used as the basis of ViralMSA [31], which can perform multiple sequence alignment of viral genomes with the help of a reference sequence and is linearly scalable with the number of sequences. MAFFT [32] utilizes fast Fourier transform (FFT) to accelerate the calculation of the correlation between two sequences, which is an indicator of homologous regions of the sequences. These homologous regions can then act as anchors in pairwise sequence alignment. Additionally, it is concluded in Ref [32] that the accuracy of FFT-based methods is almost unaffected by this heuristic approach.

The concept of divide and conquer is also adopted in some multiple sequence alignment software to scale to large datasets, such as FAME [33] and FMAlign [34], which vertically divide the sequences using common seeds among the sequences, and MAGUS [35] and Super5 [36], which divide the sequences horizontally into subsets that are small enough to be aligned fast and accurately.

2.2. Bounded Dynamic Programming

Bounded dynamic programming [37] is based on a heuristic idea: if two sequences have close similarity, then the number of gaps inserted in the sequences during alignment will be relatively small. Therefore, the possible backtracking paths will be close to the diagonal of the dynamic programming matrix for similar sequences. The possible interval can be seen as a strip with certain width parallel to the diagonal (Figure 1B). The states located in the strip are calculated, while the others are ignored. The width of the strip reflects the trade-off between the alignment accuracy and time consumption: a wide strip means more states needed to be calculated, whereas a narrow strip means that more states could be ignored, which will, however, increase the possibility of missing the optimal path.

Several methods are available for determining the strip range. A basic idea is to use the shape-based division, but this does not fully consider the biological significance and is rarely used. A simple improvement of this method is to set a threshold to filter the states that could be ignored. If the score of one state in the dynamic programming plus the score for the transition from this state to the final state is greater than the threshold,

then transitions from this state are allowed. However, this approach requires transition scores to be estimated and a threshold to be set. The transition scores can be conservatively estimated by its upper bound, and the threshold can be generally determined using the iterative method proposed in Ref [38].

2.3. Scoring System of Pairwise Sequence Alignment

The most critical factor for the quality of a pairwise sequence alignment is the scoring system. It is the basis of the sequence alignment, including multiple sequence alignment, because it determines the direction of the alignment and reflects its quality. Most types of the sequence alignment software aim to obtain good alignment by defining an explicit or implicit objective function for scoring and improving their ability to achieve high score by adjusting alignment strategy. The higher score an alignment can achieve, the higher we think its accuracy will be in the corresponding scoring system. As an example, a model and the corresponding scoring system for pairwise alignment of nucleotide sequences containing frameshifts and stop codons comprise the main feature of MACSE, a multiple sequence alignment tool that is specific to coding sequences and takes into account frameshifts and stop codons [39].

Generally, the score of a pairwise sequence alignment is the sum of the scores of all aligned pairs. For alignment of two protein sequences, for example, each pair of aligned sites is scored depending on whether a gap is involved, or, if no gaps are involved, whether the two aligned residues are matched or mismatched. When a gap is involved, a gap penalty, which is usually a negative score, is given. Additionally, the score for matched and mismatched amino acid residues is generally determined using a substitution matrix.

The most basic substitution matrix is based on whether the two residues are matched or not. More complex matrices also take into consideration the attributes of nucleotides or amino acid residues. For example, the score of conversion–transversion matrix reflects the difference between conversion and transversion frequency in natural mutations [40]. Protein sequences contain more types of residues than nucleic acid sequences, and therefore, the substitutions and the frequency involved are more complicated. Early amino acid substitution matrix is based on the properties and the codons of amino acids [41]. More recent amino acid substitution matrices rely on the analysis of the substitution frequency of a large number of homologous sequences [41] and aim to reflect the natural probability of substitutions among amino acids at certain evolutionary distances by giving conservative substitutions higher scores. Typical amino acid substitution matrices are percent accepted mutations (PAM) [41,42] and BLOcks SUbstitution matrix (BLOSUM) [43]. Although these matrices are widely adopted, there are also matrices designed for specific protein domain, such as GPCRtm, which is summarized from the transmembrane segments of the G-protein-coupled receptor (GPCR) rhodopsin family for the reason that it is not optimal to align sequences with marked compositional biases using the general-purpose matrices [44].

In addition to substitution matrices, gap penalty is also an important part of the scoring system. A simple rule is to assign a fixed negative score when a nucleotide or amino acid residue aligns with a gap. However, this scoring method has some intrinsic limitations, mainly because insertions and deletions (indels) are small-probability events, especially in nucleic acid sequences where indels can cause frameshifts and disrupt all subsequent codons. Once a gap is inserted in an alignment, adjacent gaps are more likely to occur compared with gaps inserted at a distance from the first gap. Therefore, almost all sequence alignment algorithms now use the gap penalty rule based on the number of the gaps successively inserted, and the most typical one is the affine penalty. No optimal solution is universally applicable for the gap penalty, which is referred to as a "black art" requiring constant trial of errors [45].

3. Multiple Sequence Alignment

Optimal alignment results can be obtained under a certain scoring system with the multi-dimensional dynamic programming by extending the classic Needleman–Wunsch

algorithm. However, the time and space requirements grow exponentially with the number of sequences, which makes it impossible to complete an alignment in an acceptable time or space using this method even for a small sequence data set. To overcome this problem, some heuristic algorithm frameworks that seek quasi-optimal alignment solutions within reasonable time and space have been developed. There are also important algorithms that cannot be categorized as one of the algorithm frameworks described below, such as MAHDS [46], which is partially based on tandem repeat search algorithms [47,48] and is able to produce statistically significant alignments from highly divergent nucleotide sequences, while the other methods could hardly cope with these kinds of data sets.

3.1. Star Alignment Strategy

The star alignment strategy [49] is a heuristic method that aligns multiple sequences by the consistencies among pairwise alignments of each of the sequences to be aligned and a center sequence, which could be either generated or selected from the input (Figure 2A). It is based on the assumption that given sequence $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and pairwise alignments $A(\boldsymbol{a}, \boldsymbol{c})$, $A(\boldsymbol{b}, \boldsymbol{c})$, if there is $\boldsymbol{a}_i \sim \boldsymbol{c}_k$ in $A(\boldsymbol{a}, \boldsymbol{c})$ and $\boldsymbol{b}_j \sim \boldsymbol{c}_k$ in $A(\boldsymbol{b}, \boldsymbol{c})$, then $\boldsymbol{a}_i$ should be aligned with $\boldsymbol{b}_j$, where $\boldsymbol{c}$ is regarded as the center sequence, and $\boldsymbol{a}_i \sim \boldsymbol{b}_j$ means that the ith residue of $\boldsymbol{a}$ is aligned with the jth residue of $\boldsymbol{b}$.

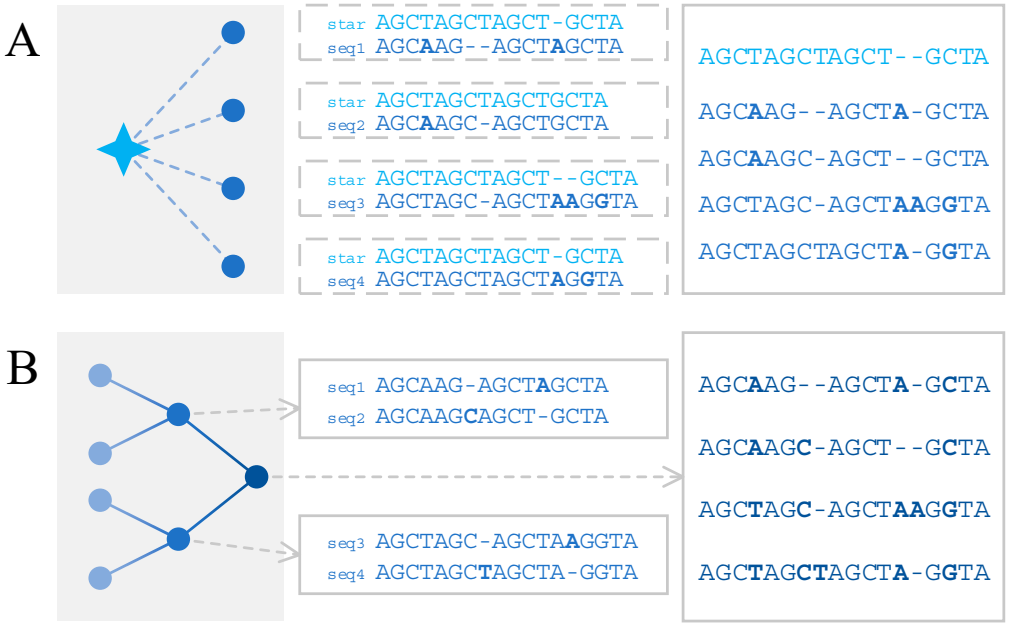

Figure 2. Two strategies for multiple sequence alignment. (**A**) Star alignment strategy performs multiple sequence alignment based on the consistencies among pairwise alignments of the sequences pending alignment and a center sequence, which is in light blue. (**B**) Progressive alignment strategy performs multiple sequence alignment along a pre-built guide tree, each of whose internal nodes represents an alignment of two sequences, one sequence with one profile, or two profiles.

The star alignment algorithm uses the star topology to simulate the relationship between sequences, and the selection of the center sequence determines the accuracy of the alignment to some extent. A brute-force choice is the sequence that has the shortest average distance from the other ones, but the time required is proportional to the square of the number of the sequences. Therefore, it is usually difficult to apply this scheme to data sets with large number of sequences. Generally, the longer the sequence is, the more likely the information lost by other sequences due to mutations will be retained. On the basis of this assumption, the clustering software CD-HIT [50] selects the longest sequence in each possible cluster as the cluster center. HAlign [51], which adopts star alignment strategy to perform multiple sequence alignment, also selects the longest sequence as the center.

This system has a limited ability to simulate the relationship between sequences, so it is less effective for data sets with complex relationships; but good alignment results can often be obtained for data sets with high similarities. Moreover, thanks to its extremely low time complexity, it can handle tasks that are almost impossible to be completed by other algorithms within a finite time, so that an approximate solution can be quickly obtained. The pairwise sequence alignments involved in the star alignment strategy are independent of each other, making it easy to parallelize programs based on the star alignment strategy, for example MASC [52] and I-CSA-M [53], which is able to utilize multiple mainframes simultaneously.

3.2. Progressive Alignment Strategy

The progressive alignment strategy [54] is another heuristic method, which performs multiple sequence alignment based on a pre-built guide tree (Figure 2B). Each leaf node in the guide tree represents a sequence, and each internal node represents an alignment of the sequences corresponding to its descendant leaf nodes (also known as profile). Two sequences (two leaf nodes), a sequence and a profile (a leaf node and an internal node), or two profiles (two internal nodes), are sequentially aligned along the paths from the leaf to the root. When the alignment proceeds with the root node, the multiple sequence alignment result is obtained.

The tree topology is used in the progressive alignment algorithm to simulate the relationships between sequences, and therefore, the alignment quality depends on the quality of the guide tree. Several established hierarchical clustering algorithms for the construction of guide trees are available. Among them, unweighted pair group method with arithmetic mean (UPGMA) [55] has been the most widely used. However, because UPGMA depends on the distance matrix of the sequences, the $O(n^2)$ time and space complexity is still a problem, where n represents the number of the sequences to be aligned. To avoid the calculation of the distance matrix, some multiple sequence alignment software adopts heuristic methods, such as PartTree [56] used in MAFFT [32] and mBed [57] used in ClustalO [58] and Kalign 3 [59]. On the other hand, there are kinds of string distances for the estimation of similarities between two sequences. Several of them, such as edit distance and Damerau–Levenshtein distance, have a time complexity of $O(l^2)$, which is too time consuming. Therefore, the string similarity based on k-mers [42] has become popular because of its lower time complexity. In addition, it is worth noting that simple chained guide trees, even in a random order, might help to produce more accurate alignment of protein families of a large number of sequences for structure-based benchmarks [60] and reduce the time consumption of the guide tree construction. The effect of chained guide trees has been discussed in Refs [61–63].

In addition, PRANK [64] and PAGAN [65] implemented a phylogeny-aware progressive alignment algorithm, which takes into consideration the difference between the insertions and deletions and makes them especially suitable for producing alignments for evolutionary analyses, at a cost of poor performance on structural benchmarks. ProPIP [66,67] goes further by the adoption of an explicit mathematical model describing the evolution of indels.

Compared with the star alignment, the tree topology used in the progressive alignment fits far more closely with the natural relationships among sequences; therefore, progressive alignment can often process more complex data [68] and has become the core of algorithms used in most multiple sequence alignment software. Both star alignment and progressive alignment try to transform a multiple sequence alignment problem to approximately n pairwise alignments. However, the pairwise alignments in the latter involve profile-to-profile alignment, which requires additional computation. Furthermore, the selection of the center sequence in star alignment could be very simple, while the construction of a guide tree in progressive alignment requires a hierarchical clustering, which is complicated and thus takes much more time. These make progressive alignment more time consuming than star alignment, especially when the number of sequences is over tens of thousands.

4. Defects of Heuristic Algorithms and Countermeasures

The star alignment and the progressive alignment strategies have two main defects. The "once a gap, always a gap" cardinal rule [69] is one of the problems; once a wrong gap is inserted in the alignment, it will continue to exist and affect all the subsequent alignment. Although this rule causes the gap error to spread, it does not create a baseless error, and the root of the error lies in another defect, i.e., the local optimum trap. Local optima usually arise from the greedy principle. Specifically, the optimum selected for a particular sub-problem is not necessarily optimal for the overall problem, and this will adversely affect the final alignment. For these two heuristic methods, every pairwise sequence alignment process, regardless of whether a profile is involved, aims to find a local optimal solution, and therefore, the multiple sequence alignment, which is pieced together based on these local optima, will inevitably have the possibility of falling into a local optimum trap. The star alignment is more affected by this defect because of its limited system ability compared with progressive alignment.

Two methods can be applied to handle these problems [70]: one is consistency objective function, which aims to reduce the probability of introducing errors in the alignment stage, and the other is iterative refinement, which aims to remove any errors in the final alignment by post-processing.

4.1. Consistency Objective Function

Consistency objective function provides another scheme for scoring two aligned residues. For example, consider three sequences, $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, for which the pairwise sequence alignments are $A(\boldsymbol{a}, \boldsymbol{b})$, $A(\boldsymbol{a}, \boldsymbol{c})$, and $A(\boldsymbol{b}, \boldsymbol{c})$. If there is $\boldsymbol{a}_i \sim \boldsymbol{b}_j$ in $A(\boldsymbol{a}, \boldsymbol{b})$ and $\boldsymbol{b}_j \sim \boldsymbol{c}_k$ in $A(\boldsymbol{b}, \boldsymbol{c})$, then the score of $\boldsymbol{a}_i \sim \boldsymbol{c}_k$ should be increased in the multiple sequence alignment of these three sequences. Match scores of every possible pair are re-estimated in this way. Gap penalties are already considered in the pairwise sequence alignment. The matching score of the two residues obtained by re-estimation with the consistency objective function is the score after the gaps have been penalized. Therefore, gap penalties need not be considered in the subsequent multiple sequence alignment process. It takes $O(n^2l^2)$ time to obtain all the possible pairwise alignments and $O(n^3l)$ time to re-estimate the match scores of all the pairs indirectly aligned [71], which is impractical for large data sets. The consistency objective function was proposed by Notredame et al. and has already been implemented in multiple sequence alignment software T-Coffee [71,72]. ProbCons [73] goes further by the adoption of HMM, a more formal probabilistic framework. The concept of consistency is also utilized to combine several alignments into a better alignment, which is implemented in M-Coffee [74].

4.2. Iterative Refinement

The iterative refinement method [54,75] processes the results of multiple sequence alignment to remove errors caused by the local minimum trap and the "once a gap, always a gap" rule. There are several ways to perform the iterative refinement, two of which will be introduced in this section.

Progressive alignment relies on the guide tree, but the heuristic distance estimation or hierarchical clustering do not necessarily produce the optimal tree for alignment. Therefore, in some iterative refinement algorithms, the completed alignment results are used to recalculate the sequence distance matrix to construct a more solid guide tree, which can be used to improve the alignment performance in an additional round of alignments. This refinement method is seldom used as the core of iterative refinement because of its excessive time overhead. Nevertheless, SATé [76,77] uses this iterative technique to meet the challenge of highly accurate alignment estimation on data sets with hundreds of sequences.

Currently, an iterative refinement method based on division and re-alignment is widely used. In this method, the results of the old completed alignment are divided horizontally into two parts, and the columns that contain only gaps are deleted to form a valid alignment

for each part. Then, the two parts are re-aligned to get new completed alignment. Finally, the old and new completed alignments are compared, and the better one is selected as the basis for the next iteration. The iteration will continue until it meets a certain condition, which is usually provided as an argument for users to determine, depending on how much time they can afford to improve the accuracy of the alignment. A commonly used horizontal segmentation method is to cut an edge of the guide tree to divide all sequences into two parts, which has been shown to produce the best trade-off results between time consumption and accuracy [78,79]. Moreover, FAMSA divides an alignment based on whether a gap is present in a certain column and is claimed to be able to improve the alignment quality for data sets up to 1000 sequences [80]. There are also methods that vertically divide a completed alignment, such as SpliVert [81].

A reliable scoring method is needed to quickly and effectively compare the two completed alignments. The SP (sum of pairs) [82,83] score is commonly used. SP sums the scores of all-against-all pairwise alignments of all the sequences (the scores of the pairwise sequence alignments are calculated as described in Section 2.3). Generally, the alignment of two gaps is ignored, so the score will be 0. A brute-force way to calculate the SP score of an alignment is costly, but some efficient algorithms are available, such as Ref [84], which is based on the pre-computation of the gap interval information of each input sequence.

Another important kind of multiple sequence alignment methodology is stochastic and mainly based on evolutionary algorithms, which are intrinsically iterative. It starts with a set of possible alignments and performs genetic operators on them to yield a new generation of alignments, which act as the initial alignments of the next iteration. This kind of algorithm has become a promising alternative field of multiple sequence alignment research [8]. Two of the methods that adopt evolutionary algorithms are MO-SAStrE [85] and its parallel version M2Align [86], which aim to optimize multiple objective functions simultaneously (while the classic methodologies optimize only one objective function). Recent research has shown the ability of genetic algorithms to combine and improve the quality of several quasi-optimal alignments [87] (while traditional methods, such as SA-GA [88], tend to start from random alignments).

5. Quality Estimation of Multiple Sequence Alignment Software

Multiple sequence alignment software uses different algorithms, and therefore, the alignment quality can vary. To score the alignment algorithms and to provide references for the design and development of sequence alignment software, different quality estimation methods have been proposed. Several kinds of classical quality estimation methods [89], as well as their pros and cons based on the desirable properties proposed in Ref [90], are discussed in this section and summarized in Table 1.

5.1. Estimation Based on Reference Alignment

Structured benchmarks for protein alignment software include BAliBASE [91], which is one of the most commonly used benchmarks. These benchmarks provide fixed test sets and reference alignments that have been manually or automatically refined based on the three-dimensional structure of proteins, with the assumption that amino acid residues corresponding to the same position in the three-dimensional structure should be aligned. Although BAliBASE and other similar benchmarks that provide fixed test sets and reference alignments are widely used, if they are not regularly updated, the developers of multiple sequence alignment software may tend to optimize their software only on limited data sets, resulting in the "high in score but low in ability" phenomenon.

Another method scores multiple sequence alignment software using generated data sets. These methods simulate the evolution of sequences (i.e., substitution or indel of residues) using a probabilistic model and generate the evolved sequences and the reference alignment based on the evolution. Because the generated mutations are completely determined by the evolution model, the accuracy of such benchmarks is restricted by the degree to which the adopted model represents the natural evolution. In some conditions, the

probabilistic model can even bias the estimation if it is similar to the sequence relationship model the tested software adopts, which also poses a challenge to the establishment and selection of the probabilistic model.

These two types of benchmarks provide preset reference alignment against which the performance of newly developed software can be tested. Therefore, a scoring method is needed to measure the degree to which the alignment of the tested software is close to the reference. Two commonly used scoring methods are the sum of pairs (SP) and total column or true column (TC) scores [92], which estimate the similarity of two alignments by counting the number of common pairs and common columns in the two alignments. The Friedman rank test [93,94] and the Wilcoxon signed-rank test [32,80] could be used for alignment accuracy discrimination by reporting a *p*-value, which indicates the likelihood that the performance difference between different methods is due to chance.

Table 1. Quality estimation methods of multiple sequence alignment software.

	Structural Benchmark	Simulated Sequences	Commonality-Based
Scalability	Low	High	High
Pre-Built Alignment	Yes	Yes	No
Scoring Methods	Sum of pair score and true column score	Sum of pair score and true column score	Multiple overlap score and head-or-tail score
Dependency	Protein structure	Probabilistic model	/
Test Sets	Fixed	Configurable	Not limited
Drawbacks	Limited to the diversity of benchmarks	Adopted model may have defects	Tested software can make common mistakes
Examples	BAliBASE [91] Pfam [95] HOMSTRAD [96]	ROSE [97] INDELible [98] Dawg [99]	MUMSA [100]

5.2. Estimation Based on the Commonality among Alignments by Different Software

Another kind of quality estimation method does not rely on reference alignment but the commonality among the alignments obtained by different multiple sequence alignment strategy. It is based on the idea that if several different pieces of software consistently align two residues, then, most likely, these two sites are correctly aligned. However, an important defect of this method is that if different pieces of software consistently but wrongly align two residues, then this error will be deemed correct in the scoring. Unlike the estimation methods based on reference alignment, the commonality-based methods use some special scoring methods, such as the multiple overlap score [100] and the head-or-tail (HoT) score [101]. The multiple overlap score considers that every aligned pair in an alignment could get a higher score if the pair appears in more alignments. The HoT score assumes that a good sequence alignment tool should not have the assumption about the direction of sequences, which means that it should produce two consistent alignments based on sequences in the original and the reversed order.

6. Conclusions

The explosive growth of biological sequence data has brought unprecedented challenges to the automated biological sequence analysis software. Sequence alignment, as the basis of sequence analysis, restricts subsequent analysis in applicability and accuracy. In this review, pairwise sequence alignment and its scoring system, main algorithms for multiple sequence alignment, as well as their advantages and disadvantages, and the quality estimation methods for multiple sequence alignment software, are presented and discussed. The collation of this information (Supplementary Table S1) might help the researchers of sequence alignment and the users of sequence alignment tools.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/biom12040546/s1, Table S1: The overview of tools and methods mentioned in the review.

Author Contributions: Writing—original draft preparation, J.C.; writing—review and editing J.C., F.T.; visualization, J.C.; supervision, L.X.; project administration, L.X.; funding acquisition, F.T., L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62001311, 61902259), the Special Science Foundation of Quzhou (2021D004), and the Startup Postdoc Foundation of Shenzhen Polytechnic (6021330003K), Fellowship of China Post-doctoral Science Foundation (2021M700702).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: We acknowledge the help from the other members of our group: Juntao Chen, Yanming Wei and Yixiao Zhai for providing critical opinions during the preparation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zou, Q.; Lin, G.; Jiang, X.; Liu, X.; Zeng, X. Sequence clustering in bioinformatics: An empirical study. *Brief. Bioinform.* **2018**, *21*, 1–10. [CrossRef] [PubMed]
2. Lewin, H.A.; Robinson, G.E.; Kress, W.J.; Baker, W.J.; Coddington, J.; Crandall, K.A.; Durbin, R.; Edwards, S.V.; Forest, F.; Gilbert, M.T.P.; et al. Earth BioGenome Project: Sequencing life for the future of life. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 4325–4333. [CrossRef] [PubMed]
3. Wong, K.M.; Suchard, M.A.; Huelsenbeck, J.P. Alignment Uncertainty and Genomic Analysis. *Science* **2008**, *319*, 473–476. [CrossRef] [PubMed]
4. Phillips, A.; Janies, D.; Wheeler, W. Multiple Sequence Alignment in Phylogenetic Analysis. *Mol. Phylogenet. Evol.* **2000**, *16*, 317–330. [CrossRef]
5. Rost, B.; Sander, C. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins Struct. Funct. Bioinform.* **1994**, *19*, 55–72. [CrossRef]
6. Fukuda, H.; Tomii, K. DeepECA: An end-to-end learning framework for protein contact prediction from a multiple sequence alignment. *BMC Bioinform.* **2020**, *21*, 10. [CrossRef]
7. Hu, G.; Feng, J.; Xiang, X.; Wang, J.; Salojärvi, J.; Liu, C.; Wu, Z.; Zhang, J.; Liang, X.; Jiang, Z.; et al. Two divergent haplotypes from a highly heterozygous lychee genome suggest independent domestication events for early and late-maturing cultivars. *Nat. Genet.* **2022**, *54*, 73–83. [CrossRef]
8. Chowdhury, B.; Garai, G. A review on multiple sequence alignment from the perspective of genetic algorithm. *Genomics* **2017**, *109*, 419–431. [CrossRef]
9. Chatzou, M.; Magis, C.; Chang, J.-M.; Kemena, C.; Bussotti, G.; Erb, I.; Notredame, C. Multiple sequence alignment modeling: Methods and applications. *Brief. Bioinform.* **2016**, *17*, 1009–1023. [CrossRef]
10. Needleman, S.B.; Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **1970**, *48*, 443–453. [CrossRef]
11. Hirschberg, D.S. A linear space algorithm for computing maximal common subsequences. *Commun. ACM* **1975**, *18*, 341–343. [CrossRef]
12. Söding, J. Protein homology detection by HMM-HMM comparison. *Bioinformatics* **2005**, *21*, 951–960. [CrossRef] [PubMed]
13. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [CrossRef]
14. Eddy, S.R. Hidden Markov models. *Curr. Opin. Struct. Biol.* **1996**, *6*, 361–365. [CrossRef]
15. Lemoine, F.; Blassel, L.; Voznica, J.; Gascuel, O. COVID-Align: Accurate online alignment of hCoV-19 genomes using a profile HMM. *Bioinformatics* **2021**, *37*, 1761–1762. [CrossRef]
16. Durbin, R.; Eddy, S.R.; Krogh, A.; Mitchison, G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*; Cambridge University Press: Cambridge, UK, 1998.
17. Eddy, S.R. Profile hidden Markov models. *Bioinformatics* **1998**, *14*, 755–763. [CrossRef]
18. Shen, C.; Zaharias, P.; Warnow, T. MAGUS+eHMMs: Improved multiple sequence alignment accuracy for fragmentary sequences. *Bioinformatics* **2022**, *38*, 918–924. [CrossRef]
19. Katoh, K.; Frith, M. Adding unaligned sequences into an existing alignment using MAFFT and LAST. *Bioinformatics* **2012**, *28*, 3144–3146. [CrossRef]
20. Smith, T.F.; Waterman, M.S. Identification of common molecular subsequences. *J. Mol. Biol.* **1981**, *147*, 195–197. [CrossRef]

21. Lipman, D.J.; Pearson, W.R. Rapid and Sensitive Protein Similarity Searches. *Science* **1985**, *227*, 1435–1441. [CrossRef]
22. Pearson, W.R.; Lipman, D.J. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA* **1988**, *85*, 2444–2448. [CrossRef] [PubMed]
23. Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. Basic local alignment search tool. *J. Mol. Biol.* **1990**, *215*, 403–410. [CrossRef]
24. Karp, R.M.; Rabin, M.O. Efficient randomized pattern-matching algorithms. *IBM J. Res. Dev.* **1987**, *31*, 249–260. [CrossRef]
25. Delcher, A.L.; Kasif, S.; Fleischmann, R.D.; Peterson, J.; White, O.; Salzberg, S.L. Alignment of whole genomes. *Nucleic Acids Res.* **1999**, *27*, 2369–2376. [CrossRef]
26. Marçais, G.; Delcher, A.L.; Phillippy, A.; Coston, R.; Salzberg, S.; Zimin, A. MUMmer4: A fast and versatile genome alignment system. *PLOS Comput. Biol.* **2018**, *14*, e1005944. [CrossRef]
27. Weiner, P. Linear pattern matching algorithms. In Proceedings of the 14th Annual Symposium on Switching and Automata Theory (Swat 1973), Iowa City, IA, USA, 15–17 October 1973; pp. 1–11.
28. Manber, U.; Myers, G. Suffix Arrays: A New Method for On-Line String Searches. *SIAM J. Comput.* **1993**, *22*, 935–948. [CrossRef]
29. Ferragina, P.; Manzini, G. Opportunistic data structures with applications. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, USA, 12–14 November 2000; pp. 390–398.
30. Li, H. Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics* **2018**, *34*, 3094–3100. [CrossRef]
31. Moshiri, N. ViralMSA: Massively scalable reference-guided multiple sequence alignment of viral genomes. *Bioinformatics* **2021**, *37*, 714–716. [CrossRef]
32. Kazutaka, K.; Misakwa, K.; Kei-ichi, K.; Miyata, T. MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.* **2002**, *30*, 3059–3066. [CrossRef]
33. Naznooshsadat, E.; Elham, P.; Ali, S.-Z.; Etminan, N.; Parvinnia, E.; Sharifi-Zarchi, A. FAME: Fast and memory efficient multiple sequences alignment tool through compatible chain of roots. *Bioinformatics* **2020**, *36*, 3662–3668. [CrossRef]
34. Liu, H.; Zou, Q.; Xu, Y. A novel fast multiple nucleotide sequence alignment method based on FM-index. *Brief. Bioinform.* **2022**, *23*, bbab519. [CrossRef] [PubMed]
35. Smirnov, V.; Warnow, T. MAGUS: Multiple sequence Alignment using Graph clUStering. *Bioinformatics* **2021**, *37*, 1666–1672. [CrossRef] [PubMed]
36. Edgar, R.C. MUSCLE v5 enables improved estimates of phylogenetic tree confidence by ensemble bootstrapping. *bioRxiv* **2021**. [CrossRef]
37. Spouge, J.L. Speeding up Dynamic Programming Algorithms for Finding Optimal Lattice Paths. *SIAM J. Appl. Math.* **1989**, *49*, 1552–1566. [CrossRef]
38. Korf, R.E. Depth-first iterative-deepening: An optimal admissible tree search. *Artif. Intell.* **1985**, *27*, 97–109. [CrossRef]
39. Ranwez, V.; Harispe, S.; Delsuc, F.; Douzery, E.J.P. MACSE: Multiple Alignment of Coding SEquences Accounting for Frameshifts and Stop Codons. *PLoS ONE* **2011**, *6*, e22594. [CrossRef] [PubMed]
40. Li, W.-H.; Wu, C.I.; Luo, C.C. A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution considering the relative likelihood of nucleotide and codon changes. *Mol. Biol. Evol.* **1985**, *2*, 150–174. [CrossRef]
41. Schwartz, R.M.; Dayhoff, M.O. Matrices for Detecting Distant Relationships. In *Atlas of Protein Sequences*; National Biomedical Research Foundation: Washington, DC, USA, 1978; pp. 353–359.
42. Jones, D.T.; Taylor, W.R.; Thornton, J.M. The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.* **1992**, *8*, 275–282. [CrossRef]
43. Henikoff, S.; Henikoff, J.G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* **1992**, *89*, 10915–10919. [CrossRef]
44. Ríos, S.; Fernandez, M.F.; Caltabiano, G.; Campillo, M.; Pardo, L.; Gonzalez, A. GPCRtm: An amino acid substitution matrix for the transmembrane region of class A G Protein-Coupled Receptors. *BMC Bioinform.* **2015**, *16*, 206. [CrossRef]
45. Vingron, M.; Waterman, M.S. Sequence alignment and penalty choice: Review of concepts, case studies and implications. *J. Mol. Biol.* **1994**, *235*, 1–12. [CrossRef]
46. Korotkov, E.V.; Suvorova, Y.M.; Kostenko, D.O.; Korotkova, M.A. Multiple alignment of promoter sequences from the *Arabidopsis thaliana* L. Genome. *Genes* **2021**, *12*, 135. [CrossRef] [PubMed]
47. Pugacheva, V.; Korotkov, A.; Korotkov, E. Search of latent periodicity in amino acid sequences by means of genetic algorithm and dynamic programming. *Stat. Appl. Genet. Mol. Biol.* **2016**, *15*, 381–400. [CrossRef] [PubMed]
48. Korotkov, E.; Korotkova, M.A. Search for regions with periodicity using the random position weight matrices in the C. elegans genome. *Int. J. Data Min. Bioinform.* **2017**, *18*, 331–354. [CrossRef]
49. Zou, Q.; Guo, M.Z.; Wang, X.K.; Zhang, T.T. An algorithm for DNA multiple sequence alignment based on center star method and keyword tree. *Tien Tzu Hsueh Pao/Acta Electron. Sin.* **2009**, *37*, 1746–1750.
50. Li, W.; Jaroszewski, L.; Godzik, A. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics* **2001**, *17*, 282–283. [CrossRef]
51. Zou, Q.; Hu, Q.; Guo, M.; Wang, G. HAlign: Fast multiple similar DNA/RNA sequence alignment based on the centre star strategy. *Bioinformatics* **2015**, *31*, 2475–2481. [CrossRef]

52. Su, W.; Liao, X.; Lu, Y.; Zou, Q.; Peng, S. Multiple Sequence Alignment Based on a Suffix Tree and Center-Star Strategy: A Linear Method for Multiple Nucleotide Sequence Alignment on Spark Parallel Framework. *J. Comput. Biol.* **2017**, *24*, 1230–1242. [CrossRef]
53. Dong, G.; Fu, X.; Li, H.; Li, J. An accurate algorithm for multiple sequence alignment in MapReduce. *J. Comput. Methods Sci. Eng.* **2018**, *18*, 283–295. [CrossRef]
54. Barton, G.; Sternberg, M. A strategy for the rapid multiple alignment of protein sequences: Confidence levels from tertiary structure comparisons. *J. Mol. Biol.* **1987**, *198*, 327–337. [CrossRef]
55. Sokal, R.; Michener, C. A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull.* **1958**, *38*, 1409–1438.
56. Katoh, K.; Toh, H. PartTree: An algorithm to build an approximate tree from a large number of unaligned sequences. *Bioinformatics* **2007**, *23*, 372–374. [CrossRef] [PubMed]
57. Blackshields, G.; Sievers, F.; Shi, W.; Wilm, A.; Higgins, D.G. Sequence embedding for fast construction of guide trees for multiple sequence alignment. *Algorithms Mol. Biol.* **2010**, *5*, 21. [CrossRef] [PubMed]
58. Sievers, F.; Wilm, A.; Dineen, D.; Gibson, T.J.; Karplus, K.; Li, W.; Lopez, R.; McWilliam, H.; Remmert, M.; Söding, J.; et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* **2011**, *7*, 539. [CrossRef] [PubMed]
59. Lassmann, T. Kalign 3: Multiple sequence alignment of large datasets. *Bioinformatics* **2020**, *36*, 1928–1929. [CrossRef]
60. Boyce, K.; Sievers, F.; Higgins, D.G. Simple chained guide trees give high-quality protein multiple sequence alignments. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 10556–10561. [CrossRef]
61. Yamada, K.D.; Tomii, K.; Katoh, K. Application of the MAFFT sequence alignment program to large data—Reexamination of the usefulness of chained guide trees. *Bioinformatics* **2016**, *32*, 3246–3251. [CrossRef]
62. Tan, G.; Gil, M.; Löytynoja, A.P.; Goldman, N.; Dessimoz, C. Simple chained guide trees give poorer multiple sequence alignments than inferred trees in simulation and phylogenetic benchmarks. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, E99–E100. [CrossRef]
63. Boyce, K.; Sievers, F.; Higgins, D.G. Reply to Tan et al.: Differences between real and simulated proteins in multiple sequence alignments. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, E101. [CrossRef]
64. Löytynoja, A. Phylogeny-aware alignment with PRANK. *Methods Mol. Biol.* **2014**, *1079*, 155–170. [CrossRef]
65. Löytynoja, A.; Vilella, A.J.; Goldman, N. Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. *Bioinformatics* **2012**, *28*, 1684–1691. [CrossRef] [PubMed]
66. Maiolo, M.; Zhang, X.; Gil, M.; Anisimova, M. Progressive multiple sequence alignment with indel evolution. *BMC Bioinform.* **2018**, *19*, 331. [CrossRef] [PubMed]
67. Maiolo, M.; Gatti, L.; Frei, D.; Leidi, T.; Gil, M.; Anisimova, M. ProPIP: A tool for progressive multiple sequence alignment with Poisson Indel Process. *BMC Bioinform.* **2021**, *22*, 518. [CrossRef] [PubMed]
68. Zou, Q.; Shan, X.; Jiang, Y. A Novel Center Star Multiple Sequence Alignment Algorithm Based on Affine Gap Penalty and K-Band. *Phys. Procedia* **2012**, *33*, 322–327. [CrossRef]
69. Feng, D.-F.; Doolittle, R.F. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *J. Mol. Evol.* **1987**, *25*, 351–360. [CrossRef]
70. Gotoh, O. Heuristic Alignment Methods. *Methods Mol. Biol.* **2014**, *1079*, 29–43. [CrossRef]
71. Notredame, C.; Holm, L.; Higgins, D.G. COFFEE: An objective function for multiple sequence alignments. *Bioinformatics* **1998**, *14*, 407–422. [CrossRef]
72. Notredame, C.; Higgins, D.; Heringa, J. T-coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.* **2000**, *302*, 205–217. [CrossRef]
73. Do, C.B.; Mahabhashyam, M.S.; Brudno, M.; Batzoglou, S. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Res.* **2005**, *15*, 330–340. [CrossRef]
74. Wallace, I.M.; O'Sullivan, O.; Higgins, D.G.; Notredame, C. M-Coffee: Combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Res.* **2006**, *34*, 1692–1699. [CrossRef]
75. Berger, M.P.; Munson, P.J. A novel randomized iterative strategy for aligning multiple protein sequences. *Bioinformatics* **1991**, *7*, 479–484. [CrossRef] [PubMed]
76. Liu, K.; Raghavan, S.; Nelesen, S.; Linder, C.R.; Warnow, T. Rapid and Accurate Large-Scale Coestimation of Sequence Alignments and Phylogenetic Trees. *Science* **2009**, *324*, 1561–1564. [CrossRef] [PubMed]
77. Liu, K.; Warnow, T.J.; Holder, M.; Nelesen, S.M.; Yu, J.; Stamatakis, A.P.; Linder, C.R. SATé-II: Very Fast and Accurate Simultaneous Estimation of Multiple Sequence Alignments and Phylogenetic Trees. *Syst. Biol.* **2012**, *61*, 90. [CrossRef] [PubMed]
78. Hirosawa, M.; Totoki, Y.; Hoshida, M.; Ishikawa, M. Comprehensive study on iterative algorithms of multiple sequence alignment. *Bioinformatics* **1995**, *11*, 13–18. [CrossRef] [PubMed]
79. Gotoh, O. A weighting system and aigorithm for aligning many phylogenetically related sequences. *Bioinformatics* **1995**, *11*, 543–551. [CrossRef] [PubMed]
80. Deorowicz, S.; Debudaj-Grabysz, A.; Gudyś, A. FAMSA: Fast and accurate multiple sequence alignment of huge protein families. *Sci. Rep.* **2016**, *6*, 33964. [CrossRef]
81. Zhan, Q.; Fu, Y.; Jiang, Q.; Liu, B.; Peng, J.; Wang, Y. SpliVert: A Protein Multiple Sequence Alignment Refinement Method Based on Splitting-Splicing Vertically. *Protein Pept. Lett.* **2020**, *27*, 295–302. [CrossRef]
82. Altschul, S.F. Gap costs for multiple sequence alignment. *J. Theor. Biol.* **1989**, *138*, 297–309. [CrossRef]

83. Lipman, D.J.; Altschul, S.F.; Kececioglu, J.D. A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA* **1989**, *86*, 4412–4415. [CrossRef]
84. Ranwez, V. Two Simple and Efficient Algorithms to Compute the SP-Score Objective Function of a Multiple Sequence Alignment. *PLoS ONE* **2016**, *11*, e0160043. [CrossRef]
85. Ortuño, F.M.; Valenzuela, O.; Rojas, F.; Pomares, H.; Florido, J.P.; Urquiza, J.M.; Rojas, I. Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: Structural information, non-gaps percentage and totally conserved columns. *Bioinformatics* **2013**, *29*, 2112–2121. [CrossRef] [PubMed]
86. Vega, C.G.Z.; Nebro, A.J.; García-Nieto, J.; Aldana-Montes, J.F. M2Align: Parallel multiple sequence alignment with a multi-objective metaheuristic. *Bioinformatics* **2017**, *33*, 3011–3017. [CrossRef] [PubMed]
87. Narayan, B.; Jeevitesh, M. Evolutionary computation approach to enhance protein multiple sequence alignments. *Res. Sq.* **2022**. Available online: https://www.researchsquare.com/article/rs-1236304/v1 (accessed on 26 March 2022).
88. Notredame, C. SAGA: Sequence alignment by genetic algorithm. *Nucleic Acids Res.* **1996**, *24*, 1515–1524. [CrossRef]
89. Iantorno, S.; Gori, K.; Goldman, N.; Gil, M.; Dessimoz, C. Who Watches the Watchmen? An Appraisal of Benchmarks for Multiple Sequence Alignment. In *Multiple Sequence Alignment Methods*; Russell, D.J., Ed.; Humana Press: Totowa, NJ, USA, 2014; pp. 59–73.
90. Aniba, M.R.; Poch, O.; Thompson, J.D. Issues in bioinformatics benchmarking: The case study of multiple sequence alignment. *Nucleic Acids Res.* **2010**, *38*, 7353–7363. [CrossRef]
91. Thompson, J.D.; Koehl, P.; Ripp, R.; Poch, O. BAliBASE 3.0: Latest developments of the multiple sequence alignment benchmark. *Proteins Struct. Funct. Bioinform.* **2005**, *61*, 127–136. [CrossRef]
92. Thompson, J.D.; Plewniak, F.; Poch, O. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.* **1999**, *27*, 2682–2690. [CrossRef]
93. Edgar, R.C. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **2004**, *32*, 1792–1797. [CrossRef]
94. Roshan, U.; Livesay, D.R. Probalign: Multiple sequence alignment using partition function posterior probabilities. *Bioinformatics* **2006**, *22*, 2715–2721. [CrossRef]
95. Mistry, J.; Chuguransky, S.; Williams, L.; Qureshi, M.; Salazar, G.A.; Sonnhammer, E.L.L.; Tosatto, S.C.E.; Paladin, L.; Raj, S.; Richardson, L.J.; et al. Pfam: The protein families database in 2021. *Nucleic Acids Res.* **2021**, *49*, D412–D419. [CrossRef]
96. Mizuguchi, K.; Deane, C.; Blundell, T.L.; Overington, J. HOMSTRAD: A database of protein structure alignments for homologous families. *Protein Sci.* **1998**, *7*, 2469–2471. [CrossRef] [PubMed]
97. Stoye, J.; Evers, D.; Meyer, F. Generating benchmarks for multiple sequence alignments and phylogenetic reconstructions. *Proceedings. Int. Conf. Intell. Syst. Mol. Boil.* **1997**, *5*, 303–306.
98. Fletcher, W.; Yang, Z. INDELible: A Flexible Simulator of Biological Sequence Evolution. *Mol. Biol. Evol.* **2009**, *26*, 1879–1888. [CrossRef] [PubMed]
99. Cartwright, R.A. DNA assembly with gaps (Dawg): Simulating sequence evolution. *Bioinformatics* **2005**, *21*, iii31–iii38. [CrossRef]
100. Lassmann, T.; Sonnhammer, E.L.L. Automatic assessment of alignment quality. *Nucleic Acids Res.* **2005**, *33*, 7120–7128. [CrossRef]
101. Landan, G.; Graur, D. Heads or Tails: A Simple Reliability Check for Multiple Sequence Alignments. *Mol. Biol. Evol.* **2007**, *24*, 1380–1383. [CrossRef]