Journal of Health Informatics & Management

Editorial

A SCITECHNOL JOURNAL

Quasi-Optimal Multiple Sequence Alignments

Quoc-Nam Tran* **and Mike Wallinga**

Abstract

Finding an optimal multiple sequence alignment (MSA) of three or more nucleic acid or amino acid sequences is a fundamental problem of bioinformatics with a large number of publications and citations over the last 30 years. Given a set of sequences, an optimal MSA identifies homologous characters, which have common ancestry. The resulting MSA is used for many downstream applications in medical and health informatics such as constructing phylogenetic trees, finding protein families, predicting secondary and tertiary structure of new sequences, and demonstrating the homology between new sequences and existing families.

Unfortunately, techniques that work well for pairwise alignment often become too computationally expensive when they are applied to multiple sequence alignment due the extremely large size of the search space. In fact, it is common for multiple sequence alignment problems to become computationally intractable. This is because multiple sequence alignment is a combinatorial problem, and as the number or size of the sequences in the problem set increases, the computational time required performing an alignment increases exponentially. That is, for n sequences of length l, computing the optimal alignment exactly carries a computational complexity of $O(ln)$. Thus, dynamic programming techniques such as the Needleman-Wunsch algorithm are guaranteed to produce optimal solutions to multiple sequence alignment problems, but are generally impractical for all but the smallest examples. In fact, multiple sequence alignment algorithms using the sum-of-pair heuristic is NP-complete. As a result, most currently-employed multiple sequence alignment algorithms are based on heuristics and must settle for providing a quasi-optimal alignment.

In this editorial article, we will summarize the previous works on MSA especially some recent computational methods for quasi-optimal multiple sequence alignments. We will also discuss some possible approaches for future works.

The Sequence Alignment Problem

Sequence alignment is the process of arranging primary sequences of DNA, RNA, or protein to identify regions of similarity in order to discover functional, structural, or evolutionary relationships between the sequences. These discoveries can result in the construction of phylogenetic trees, the discovery of new protein families, the prediction of secondary or tertiary structures of new sequences, and the demonstration of homology between existing families and the newly discovered sequences.

***Corresponding author:** Quoc-Nam Tran, Department of Computer Science, The University of South Dakota, USA, Tel: 605-677-5388; E-mail: QuocNam.Tran@usd.edu

Received: January 08, 2021 **Accepted:** January 23, 2021 **Published:** January 28, 2021

The goal of sequence alignment is to find an optimum match between the sequences being investigated. This is actually a case of text manipulation, as these sequences are represented as strings over a given alphabet. For example, a DNA sequence is represented as a string drawing from an alphabet of four characters (A, C, G and T) representing the four nucleotides. Similarly, a protein sequence is represented in the same way, but drawing from an alphabet of 20 different symbols, each representing a unique amino acid. The goal of alignment, then, is to arrange these strings so that they are vertically aligned in the optimal way to highlight similarities and differences. Blank spaces, called gaps, are inserted into the strings at strategic locations so that all of the sequences are extended to the same length and the symbols in each string match vertically with the corresponding symbols in the other strings as often as possible. The optimum match, then, is defined as the largest number of symbols from one sequence that can be matched with those of another sequence while allowing for all possible gaps [1].

A similarity measure is needed to quantify how well the sequences match in a given alignment, and these scores are compared to determine an optimal alignment. Any similarity measure must account for changes in the sequences that are due to insertion, deletion, or mutation through evolutionary processes. This is usually done by the insertion of gaps within the sequences, presuming the presence of a gap leads to a higher number of symbol matches and thus a higher similarity score. To offset the higher score obtained through the insertion of gaps, and to prevent the introduction of an excessive number of gaps in a sequence, the similarity measure must also introduce a gap insertion penalty. Typically, two penalty values are used: one for introducing a new gap into a sequence, and one for extending an existing gap. The scoring also relies on a substitution matrix, typically from the PAM or BLOSUM families, such as PAM250 or BLOSUM62, which assigns a score to each possible amino acid substitution, with higher values assigned to symbol mutations that are more likely to naturally occur. A primary example of a metric used to evaluate the quality of the alignment is the sum of pairs score (SP). Given a set of n sequences, the sum-of-pairs score is the sum of all of the corresponding pairwise alignment costs from the chosen matrix. Since a naive sum-of-pairs approach treats all pairwise alignments equally, even those are redundant or highly correlated, a weighted-sum-of-pair score is commonly used [2].

The primary disadvantage of similarity measures like weighted-sum-of-pairs is the use of general substitution matrices. These matrices generally have been formed via statistical analysis of a large number of sample alignments, but may not be adapted to the specific set of sequences being aligned for a given problem. An alternative is to use a Hidden Markov Model approach, in which sequences are used to generate statistical models to create operational sequences of gap insertions and deletions. Unlike the standard substitution matrices, the model can be developed and trained based on the characteristics of the sequences to be aligned. Given a trained model, the sequences of interest are aligned to the model in succession, producing a multiple sequence alignment [3]. Unfortunately, there isn't a known deterministic algorithm that can successfully guarantee an optimally trained Hidden Markov Model within a reasonable time limit. Some algorithms, such as the forward-backward algorithm (also known as

the BW algorithm) by Baum and Welch, use statistical approximations to determine a suitable Hidden Markov Model. Some stochastic approaches have been tried, but generally only for smaller Hidden Markov Models (10 states or less) [4].

Progressive Methods for Multiple Sequence Alignment

The most common heuristic method today is the progressive alignment technique. Progressive alignment requires initial guesses about the relationships between sequences in the set, and uses those guesses to build a guide tree to represent those relationships. The most closely related pairs of sequences are aligned using traditional dynamic programming methods, following the guide tree to start with the most similar pair and working towards the least similar pair. At each step, two sequences are aligned, or one sequence is aligned to an existing alignment. In the latter case, any gaps that were introduced in earlier alignments are kept when a new sequence is added to the group. These groups of pairwise alignments are iteratively aligned together, resulting in the final multiple sequence alignment.

Phylogenetic trees such as a guide tree are usually produced using a similarity (or difference) matrix, so the "initial guesses" required prior to building the guide tree are actually used to construct such a matrix. The matrix classifies the sequences according to their differences, which is assumed to be a proxy for the evolutionary distance between them. The two key features of any tree are the branching order (called the topology) and the branch lengths, which should be proportional to the evolutionary distances between species. The trees that account for today's extant species using the smallest number of historical genetic events are considered the best, so any tree-building algorithm will favor scores of high similarity (and thus, low difference). Many researchers have contributed to this area such as [5,6]. Feng and Doolittle's progressive approach became the most popular technique for carrying out multiple sequence alignment, and was used as the basis of the Clustal software package by Desmond Higgins and Paul Sharp. The software has remarkably evolved in the last 30 years with Clustal Omega as the current version of the software package [7].

A major drawback to the progressive approach is the local minimum problem. It is due to the "greedy" nature of the alignment strategy. As the most similar pairs of sequences are aligned together early in the algorithm, there is no guarantee that those pairwise alignments will place gaps in the optimal positions for the multiple sequence alignment. They will be optimal for that particular pairwise alignment, and since the best matches are being aligned first, they are assumed to be of a sufficient quality and correctness. Even so, some misalignments will occur, especially for more divergent sequences. Unfortunately, when such a misalignment occurs early on, by the nature of the algorithm, it will never be corrected later. In many cases, these misalignment errors will compound through multiple iterations. Thus, there is no guarantee that the global optimum solution for the set of sequences will be found by the progressive alignment approach. This local minimum problem was addressed by using Coffee, a novel objective function that measures the degree of consistency between a multiple sequence alignment and a library of pairwise alignments of the same sequences. The objective function is a global measure for evaluating an entire alignment, with a higher objective function score indicating a more biologically sound and relevant alignment [3]. The library of pairwise alignments must be built before the objective function can be used. The library is specific to a given set of sequences, so a new one must be made for each desired multiple sequence alignment. Generally speaking, given N sequences to be aligned, the library will contain at least $(N^2-N)/2$ pairwise alignments, one for each of the possible pairings. In reality, there is no limit to the amount of redundancy that can be included in the library, so more pairwise alignments can be added as desired. Any appropriate method can be used to generate the pairwise alignments, and the amount of time required to produce the library is dependent upon the method used and increases quadratically with the number of sequences [3].

Once the library is built, evaluation of a given alignment is performed using the Coffee objective function. Each pair of aligned residues (either two residues aligned with each other or a residue aligned with a gap) in the input alignment is compared to the contents of the library. The residues are identified by their position in the sequence, and the overall consistency score is equal to the number of pairs of residues found in the multiple alignments that are also present in the library, divided by the total number of pairs in the multiple sequence alignment, which will produce a consistency score between 0 and 1. This simplistic scoring scheme was improved by adding weighting. In the final Coffee objective function, each pairwise alignment in the library was weighted according to the percent identity between the two aligned sequences. This ensured that the alignment of a given sequence was most influenced by its closest relatives, and that the most closely related pairs of sequences were correctly aligned in the final multiple alignments [3].

Among progressive alignment techniques, T-Coffee's two distinctive features are its use of heterogeneous data sources from its pairwise alignment library, and its optimization method. The objective function is based on a standard progressive strategy similar to what is used in ClustalW, but integrates information from the library during each step. The progressive alignment uses a dynamic programming algorithm, but sets the gap-opening and gap-extension penalties to zero (just as gap penalties were not needed in the original genetic-algorithm-based Coffee calculation). Also as in the original Coffee, the weights from the library are used in place of the weights from a standard substitution matrix, which reduces the "greedy" effect by utilizing information that was specially generated for the current set of sequences, not only the position-dependent weighting present in the original Coffee scheme, but also the context-aware weighting produced from the library extension process described above.

Once the dynamic programming algorithm has completed an alignment step, Feng and Doolittle's "once a gap always a gap" principle is maintained; once a gap is introduced in the progressive alignment, it is never removed. The key difference here is that the placement of those gaps is better informed by data customized for this specific alignment, resulting in fewer misplaced gaps earlier in the process [8]. When tested using the BAliBase database of multiple sequence alignments, T-Coffee produced the highest average accuracy among four leading software tools (including ClustalW, Prrp, and Dialign), especially on more divergent test cases. The increased accuracy came at the expense of computational cost and running time; even when given a previously generated pairwise library, T-Coffee ran about two times slower than ClustalW [8].

Recent Developments for Quasi-Optimal Multiple Sequence Alignments

Since multiple sequence alignment can be viewed as an optimization problem with the goal of maximizing the scoring function, it comes as no surprise that stochastic optimization and swarm intelligence techniques have emerged as a prevailing option for

improving the computational cost of MSA. The two major advantages of using stochastic methods are a lower degree of complexity and greater flexibility in the objective function used for scoring, while a major disadvantage is that, by their nature, they do not guarantee optimality [3].

Several stochastic techniques have been employed with success, but drawbacks still exist. For example, simulated annealing was pursued as an alternative method but has since been given consideration only as an alignment improver. It proved to be too slow to converge and was too often trapped by local optima. Evolutionary algorithms such as genetic algorithms (GA) were explored, resulting in techniques such as Notredame and Higgins' SAGA (Sequence Alignment by Genetic Algorithm). Genetic algorithms have proven to be a good alternative for finding optimal solutions for a small number of sequences, but still experience exponential growth in computational time as the number of sequences increases. Novel combinations of techniques and objective functions provide reasons for optimism. For example, a recent research project modified the objective function used by the GA-based tool MSA-GA. MSA-GA normally uses a weighted sum-of-pair (WSP) objective function, but weighted sum-of-pairs is known to have limitations when dealing with sequences with regions of low similarity. In 2014, Amorim, Zafalon, Neves, Pinto, Valencio, and Machado replaced the weighted sum-of-pair objective function with Notredame's Coffee with promising results. The Coffee-based implementation outperformed the WSP-based implementation in 81% of test cases with low similarity [9].

Various machine learning and regression techniques were experimented to determine if they could predict the alignment quality of several alignment tools, using 216 sequence sets from BaliBase as the benchmark. Features from well-known biological databases were extracted and used to supplement and enrich the sequence information. Four different regression techniques had been studied: regression trees, bootstrap aggregation trees, least-squares support vector machines (LS-SVM) and Gaussian processes. These techniques were used to estimate each alignment's quality, with the alignment's Baliscore value used as the benchmark. The most popular currently used alignment evaluation systems, including PAM, BLOSUM, RBLOSUM, and STRIKE, were also referenced for comparison purposes. The normalized mutual information feature selection (NMIFS) procedure was used to determine which of the twenty-two selected biological features were most relevant for each model and thus worthy of inclusion. The regression techniques were able to predict the quality of the alignments with a high correlation against the Baliscore values (R>0.9), while STRIKE had a slightly worse correlation (R=0.714) and PAM250, BLOSUM62, and GONNET had a far worse correlation with Baliscore (R< 0.21). The Gaussian and LS-SVM techniques performed the best of the four regression techniques studied, with the Gaussian processes being slightly better overall. It has been suggested that, in addition to using supplementary biological features and multiple scoring methods for alignment evaluation, these regression models could be used in the design and optimization of MSA tools, perhaps in the design of objective functions. Another suggestion is that traditional alignment quality scoring methods may not use enough information to provide realistic evaluations of alignments, to the detriment of the software tools that rely on them [10].

In addition to the stochastic methods, particle swarm optimization-based techniques for multiple sequence alignment have been utilized with positive results. However, the standard particle swarm algorithm must be modified in a few key ways to successfully adapt it to sequence alignment. First, problem-specific operators should be designed to achieve better results. Second, experimentation on parameters is often needed to obtain the most appropriate range of values. Third, problem-specific domain knowledge must be incorporated to reduce randomness and the computational time required [11].

Many techniques employ a particle swarm in conjunction with another method, or in addition to the existing software tool, in order to improve the latter's results. One of the first such techniques, published in 2003, achieved better protein sequence alignments by using a combination of particle swarm optimization and an evolutionary algorithm to train Hidden Markov Models (HMMs) [4]. The general approach to using Hidden Markov Models to perform multiple sequence alignment, apart from using a particle swarm, is as follows: the set of states in the HMM is divided into three groups (match, insert, and delete), and the model moves between states using directed transitions with associated probabilities. The Hidden Markov Model uses a nondeterministic walk to generate a path of visited states and a sequence of emitted observables. The sequence of observables represents an unaligned sequence, and the goal is find a path that yields the best alignment. The most probable state sequence path for each sequence is determined. Each match or insert state in the path emits the next symbol in the sequence, while a delete state emits a gap. Once this process has been completed for all of the sequences, they are aligned according to their common match or delete states in shared positions. Before the Hidden Markov Model can be used, the transition and emission probabilities must be determined or trained. No exact method for determining the probabilities has been discovered; one of the most well-known and widely-used approaches is the Baum-Welch (BW) method [4].

Particle swarm optimization (PSO) has been used in multiple sequence alignment research in other ways beyond training Hidden Markov Models such as using a particle swarm to improve an alignment originally obtained *via* ClustalX. In this approach, each particle in the swarm represented a different candidate alignment, with each particle's coordinates representing a set of vectors specifying the positions of the gaps in each sequence. ClustalX produced an initial alignment used to seed the swarm, and other particles were derived from this seed by applying a mutation operator similar to that of a genetic algorithm. The size of the swarm was set by the user. The allowed length of a candidate alignment was given as a range, with the minimum value equal to the longest sequence to be aligned, and the maximum value defaulting to twice the length of the longest sequence. The sum-of-pairs similarity score was used as the objective function to be optimized, and the particle best and global best values were the best similarity scores obtained [12,13].

Reviewing the prior research on particle swarm optimization for multiple sequence alignment led to a few key takeaways and conclusions. First, particle swarm optimization has proven to be versatile. It has been used as a trainer, to improve the results of an alignment obtained from another method, and to produce an alignment by itself. In all cases, the results have been positive and promising, but open research questions and opportunities for further improvement remain plentiful.

One of the primary advantages of using a particle swarm for function optimization is the ability to apply the same technique to many different target functions. In the case of multiple sequence alignment, previous research has focused on the same traditional

metrics, such as utilizing the standard sum-of-pairs scoring method as the objective function to be optimized by the swarm or using Notredame's original Coffee function. Some of the more recent research is exploring the alternative scoring methods for the objective function such as the Universal Partitioning Search objective function [14].

Another distinguishing characteristic of a particle swarm implementation is the metric used to determine the distance between a particle's current position and its personal best or global best position, and the application of that distance metric to the particle swarm's standard velocity and position updating formulas. When optimizing most mathematical functions standard Euclidean measure of distance suffices, but multiple sequence alignment may call for more novel approaches. Some researchers used a simple percentage of similarity measure to determine distance, and employed a crossover technique to move the particles while others attempted to keep the standard velocity and position updating formulas intact, making minor adjustments as necessary when the Euclidean calculations moved a given amino acid to an illegal position in the alignment. Recent research has also explored a novel approach of determining particle distance and movement, inspired by wavelet-based volume morphing techniques from the field of computer animation and image processing [14].

Our final note is on the need of using High Performance Computing (HPC) techniques for improving the performance of PSO-based algorithms for MSA. PSO-based algorithms are inherently parallelizable. But, due to the amount of data needed for the communication between particles, using the technologies from the next generation of GPU-CPU parallelism would be more viable. GPU-based computation offers the advantages of large-scale parallelism and implementation using industry-standard libraries and tools such as NVidia's CUDA. With significant changes to the traditional particle swarm algorithm, including replacing the velocity and distance calculations, and using a computationally complex objective scoring function, adapting the implementation for parallel execution using CUDA would undoubtedly be a challenge but would also yield impressive improvements in running time.

References

1. Needleman S, Wunsch C (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48: 443-453.
2. Altschul S, Erickson B (1986) Optimal sequence alignment using affine gap costs. Bull Math Biol 48: 603-616.
3. Notredame C, Holm L, Higgins D (1998) Coffee: An objective function for multiple sequence alignments. Bioinform 14: 407-422.
4. Rasmussen T, Krink T (2003) Improved hidden markov model training for multiple sequence alignment by a particle swarm optimization- evolutionary algorithm hybrid. Bio Syst 72: 5–17.
5. Feng D, Doolittle R (1987) Progressive sequence alignment as a pre-requisite to correct phylogenetic trees. J Mol Evol 25: 351-360.
6. Waterman M (1986) Multiple sequence alignment by consensus. Nucleic Acids Res 14: 9095–9102.
7. Sievers F, Higgins DG (2014) Accurate alignment of very large numbers of sequences. Methods Mol Biol 1079: 105-116.
8. Notredame C, Higgins DG, Heringa J (2000) T-coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol 302: 205-217.
9. Amorim A, Zafalon G, Neves L, Pinto A, Valencio C, et al. (2015) Improvements in the sensibility of MSA-GA tool using coffee objective function. J Phys Confer Series 574: 1-4.
10. Ortuno F, Valenzuela O, Prieto B, Saez-Lara M, Torres C, et al. (2015) Comparing different machine learning and mathematical regression models to evaluate multiple sequence alignments. Neurocomput 1-14.
11. Das S, Abraham A, Konar A (2008) Swarm intelligence algorithms in bioinformatics. Stud Comput Intel 94: 113-147.
12. Rodriguez P, Nino L, Alonso O (2007) Multiple sequence alignment using swarm intelligence. Int J Comput Intell Resh 3: 123–130.
13. Gao Y (2014) A multiple sequence alignment algorithm based on inertia weights particle swarm optimization. J Bionanosci 5: 400–404.
14. Tran QN, Wallinga M (2017) Ups: A new approach for multiple sequence alignment using morphing techniques. Tech Rep.

Author Affiliations Top

Department of Computer Science, The University of South Dakota, USA