A Quantum-inspired Optimization Heuristic for the Multiple Sequence Alignment Problem in Bio-computing

Konstantinos Giannakis, Christos Papalitsas
Georgia Theocharopoulou, Sofia Fanarioti
and Theodore Andronikos
Ionian University
Department of Informatics
Corfu, Greece
Email: {kgiann, c14papa, zeta.theo, sofiafanar, andronikos}@ionio.gr

Abstract—**Data related to biology are characterized by large volume and requirements for enormous computational power. Biological sequences, either of proteins or DNA/RNA segments, can be large and usually need massive computations in order to discover relations and study particular properties. Aligning sequences is of great importance for various practical reasons. Multiple sequence alignment studies the problem of aligning several strings resulting in a complete alignment, a problem for which several different approaches exist. In this work, a novel heuristic method to progressively solve this problem is proposed using elements of quantum-inspired optimization. The proposed algorithm is described in detail and evaluated through simulations against other aligning methods. The experimental results seem promising for providing a good initial alignment, especially for the case of large sets of sequences.**

Index Terms—**multiple sequence alignment, optimization, quantum-inspired, TSP, bio-inspired, bio-computing, bioinformatics**

I. Introduction

The successful alignment of biological sequences is of great importance, especially in the era of Big Data and the rise of machine learning, where tremendous computation capabilities are available for demanding tasks. In particular, the *multiple sequence alignment* or MSA, in short, considers the problem of aligning a set of sequences containing proteins or nucleotides [1], [2], [3]. Unlike the simple pairwise alignment, MSA tries to tackle the problem of aligning more than two sequences at the same time. In general, a sequence alignment attempts to solve the problem of arranging sequences of DNA, RNA, or proteins in order to study properties and relationships among themselves.

Such alignments are quite useful in many fields due to the fact that a successful alignment can reveal interesting properties about the underlying set of sequences. In particular, having such an alignment, specific areas of variability or conservation of distinct proteins can be observed and further analyzed. Moreover, structural and functional properties regarding the set of proteins or nucleotides arise, which make easier for one to determine the relationship among the sequences, offering better and more faithful phylogenetic trees [1]. Interestingly enough, literature has identified other uses of sequence alignment methods besides their biological one. For example, they are widely used in information-related fields, i.e., for computing the edit distance between strings. Such approaches are then used in fields like natural language processing, information retrieval, finances, comparative linguistics, etc. [4], [5].

Since the sets of sequences are usually large and multitudinous, combined with the fact that the MSA problem has been proven to be $\mathbf{NP}$-complete with SP-score [6], [7] and $\mathbf{NP}$-hard for most metrics [8], make the problem intractable for a human approach. Therefore, efficient and smart algorithms are needed in order to provide good alignments within acceptable time frames [9], at least for an initial version of the aligned sequence set.

These algorithms fall into two major categories, depending on the fraction of the sequences they try to align (the entire length, or just particular regions), which are *global* and *local* alignments. In this work, the main objective is global alignment. Usually, dynamic programming approaches are examined in order to find best alignments, but since the problem's hardness forbids the scalability of such methods, heuristic and/or probabilistic methods have been designed for larger instances of the problem (for smaller ones exact solutions can be efficiently implemented).

MSA solutions can be categorized into 4 categories: progressive methods, iterative methods, motif-based methods, and hybrid methods [10]. In the first class of algorithms (in which the proposed algorithm belongs), the most similar sequences are aligned first and then the rest of them are aligned in a decreasing order according to the similarity level [11]. This sort of alignment can be driven by a tree-based structure that resembles the sequences' relatedness. The iterative algorithms try to repeatedly improve the accuracy of the initial alignments, the algorithms based on motif take advantage of aligning short parts of the sequences searching for indicative patterns, whereas hybrid approaches combine

978-1-7281-4959-2/19/$31.00 ©2019 IEEE

the aforementioned ones, often by combining them with other computer science approaches like genetic algorithms, Markov processes, etc.

It is important to mention that alignments are scored using well-known scoring functions, like the sum-of-pairs metric, the weighted sum-of-pairs, etc. There are also interesting approaches regarding the implementation of simulated annealing [12], [13], quantum annealing techniques [14] (like the use of the D-WAVE systems that have already been used for the MSA problem [15]) or quantum genetic algorithms [16]. In general, quantum and quantum-inspired techniques try to take advantage of the superiority of quantum computing (for particular tasks), especially for optimization problems, both on actual quantum computing architectures, as well as simulated environments within classical platforms.

Contribution. The present work proposes a novel algorithm that tries to solve the multiple sequence alignment problem. The basic idea is to use a traveling salesman tour (calculated using a quantum-inspired heuristic) to determine the order in which the sequences are progressively aligned. Similarly to other approaches, we take advantage of an alternative computational scheme, a quantum-inspired perturbation function. The main contribution lies in the proposal of an approach for the problem of multiple sequence alignment (MSA) based on a quantum-inspired VNS-based heuristic. The described progressive methodology starts with the calculation of the pairwise distances of all the sequences resulting in a distance matrix that acts as the input of the heuristic-based algorithm. Then, a simple guide tree is constructed according to the solution of the heuristic and then the procedure of MSA based on that tree takes place. The proposed solution is benchmarked on actual datasets derived from [17]. Simulation results show that the proposed solution is capable of aligning larger sequences (around 20-600 sequences of length 90-2100 each) with better sum-of-pairs score. It is, thus, believed that the present algorithm can quickly and efficiently provide an initial solution with good score.

The paper is structured as follows: after the introduction in Section I, Section II contains the discussion of the relevant literature. Next, Section III contains the necessary notation and definitions needed for the main part of the proposed algorithm, which is described in Section IV and its analysis is given in Section V. Simulation results that reveal an enhancement for larger sets of sequences are presented in Section VI, whereas Section VII contains the conclusion and some suggestions for future work.

II. Related works

The MSA problem has attracted a plethora of solutions that fall into different categories depending on the functionality of the algorithm, the number of iterations, etc. In this section, emphasis is given on works similar to our approach, whereas for others, not so closely-related works only a reference is given. A thorough work about the MSA problem, along with algorithms and related issues can be found in [2]. Sequence alignments (both the pairwise and multiple) are useful in various subfields of bioinformatics [1], [3], as well as in other fields (e.g., [4], [5]). The most well-known algorithm for solving the global alignment of two sequences is the NeedlemanWunsch dynamic programming algorithm [18], whereas the SmithWaterman algorithm is the prevailing one when local alignment of two sequences is needed [19]. The complexity for the NeedlemanWunsch is $O(n^2)$, where n is the maximum length of the two sequences [18].

The most widely-used platforms for professional sequence alignment are MUSCLE [20], Multalin [21], CLUSTAL W [22] (various CLUSTAL versions exist, the latest on is the CLUSTAL Omega [23]). Multalin is a progressive approach based on the unweighted pair group method using arithmetic averages (UPGMA) [24], [10], where the last alignments are along a guided tree. This approach has been used in the simulation part of this paper for the evaluation of the proposed algorithm. CLUSTAL W is another progressive approach that is based upon the neighbor-joining (NJ) algorithm. Both UPGMA and the neighbor-joining (NJ) algorithm require in principle the construction of a complete distance matrix of the sequences [10], [11]. MUSCLE is, also, another popular choice for aligning sequences [20].

Regarding the optimization part of this work, quantum-inspired techniques are usually conventional algorithms that utilize principles and ideas from quantum computing, exploiting the fact that quantum algorithms can vastly outperform their classical counterparts. Quantum computing was first proposed by Feynman, who realized that it is impossible to efficiently simulate an actual quantum phenomenon using a classical computer. The reader is referred to the textbook of Nielsen and Chuang for an in-depth understanding of quantum computation and information [25].

D-WAVE, an actual quantum computer based on quantum annealing, was used for a biological problem in [14], while in [26] pattern-matching on genomic sequences using quantum algorithms was thoroughly discussed. Quantum algorithms have also been deployed to solve the sequence comparison problem. In particular, Hollenberg discussed the use of efficient quantum search algorithms on comparing protein sequences [27]. A work that had a similar rationale with the one described here can be found in [28], where a quantum-inspired genetic algorithm is conceived in order to solve the MSA problem.

The use of concepts from the well-known traveling salesman problem for the MSA problem is not a new thing. The authors in [29] use an evolutionary algorithm trying to find the most appropriate order of sequences in order to progressively align in the next steps. Similarly, the construction of the evolution tree was modeled as a TSP problem in [30].

III. Definitions and formalism

In this Section, the necessary formalism pertaining to the MSA and TSP problems is presented. First, let $\Sigma = \{s_1, s_2, \ldots, s_m\}$ be a finite alphabet. A sequence is a string over Σ. We say that two sequences S_1' and S_2' are corresponding alignments of two given sequences S_1 and S_2 if for every

i, S'_i can be obtained from S_i by removing gaps (denoted by $-$). Thus, every character of sequence S'_1 corresponds to a character of sequence S'_2. This definition can easily been extended to include more sequences.

Let m denote the length of sequences S'_1 and S'_2 (a particular aligned version of S_1 and S_2). The total *cost* for this alignment can be expressed as $\sum_{i=1}^{m} d(S'_1(i), S'_2(i))$, where d is the chosen *score scheme* over the alphabet $\Sigma \cup \{-\}$ and $S'_j(i)$ is the *i*th character of sequence S'_j. A standard score scheme is the one where matches are scored 1 and mismatches 0.

$$d(S'_1(i), S'_2(i)) = \begin{cases} 1, & \text{if match} \\ 0, & \text{if mismatch} \end{cases} \tag{1}$$

An optimal alignment is the one with the maximum score among every possible alignment. As already stated, the above definitions can be easily extended to include more sequences. We denote by n the number of sequences that have to be aligned. Each sequence is allowed to have different length m_i, but we add gaps (denoted by $-$) in order to make them all of the same length m. Then, the matrix of size $n \times m$ is called an alignment of the n sequences. In the case of many sequences, the scoring scheme has to be properly adapted. There are different scoring schemes that can be used to define the alignment's cost for many sequences. The most widely used is the *sum of pairs* method [6], which is also used in this work to compare the produced alignments against other approaches. The sum of pairs method uses the sum of the costs of aligning the n sequences in pairs (resulting in $\binom{n}{2}$ total pairs).

Formally, let $S_1, S_2, \ldots, S_n$ define the n sequences that are about to be aligned. Then, the *sum of pairs* score is

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} cost(S_i, S_j). \tag{2}$$

Note that the sum of pairs score can be slightly altered if one prefers to "punish" mismatches, e.g., by giving -1 to any mismatch. Below, we provide some simple examples of aligned sequences as depicted using a matrix-like scheme that is the dominant representation of aligned sequences. For example, if we have the sequences $S_1 = ACTTAA$ and $S_2 = CAGTAC$, then below we have a possible alignment of them:

$$\begin{array}{cccccccc} \text{A} & \text{C} & \text{T} & \text{-} & \text{T} & \text{A} & \text{A} & \text{-} \\ \text{-} & \text{C} & \text{A} & \text{G} & \text{T} & \text{A} & \text{-} & \text{C} \end{array}$$

The score for this alignment is 3, since we have 3 matches. If you add an extra sequence $S_3 = CGGACA$, then a possible multiple alignment of the three sequences would be the following one:

$$\begin{array}{ccccccccc} \text{A} & \text{C} & \text{T} & \text{-} & \text{T} & \text{A} & \text{A} & \text{-} & \text{-} \\ \text{-} & \text{C} & \text{A} & \text{G} & \text{T} & \text{A} & \text{-} & \text{C} & \text{-} \\ \text{-} & \text{C} & \text{G} & \text{G} & \text{-} & \text{A} & \text{-} & \text{C} & \text{A} \end{array}$$

In this case, the sum of pairs score of the above alignment is 9, since we have 9 pairwise matches.

The choice of an appropriate scoring function has to be carefully considered. It should reflect any biological or statistical association that has been identified in the past regarding the relation among the families of proteins (or nucleotides), as well as appropriate gap penalties (e.g., by giving a -1 score for the insertion of a gap). Gap penalties can also be associated with the evolutionary relationship among the sequences. It is important to note that in this work, we only use the most generic scoring functions and gap penalty. Some well-known scoring matrices are the PAM, BLOSUM, etc.

As it is noted later in this work, the proposed algorithm models the similarity of sequences as a TSP instance, which is usually represented using a graph. Thus, we aim to tackle a minimization problem at this stage of the methodology. Note that, although the actual MSA problem is a maximization problem in terms of a score function, in this approach sequences' distances are used as input, thus the less their distance the larger their similarity. The underlying structure is a complete graph $G = (V, A)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $A = \{(v_i, v_j) : v_i, v_j \in V \text{ and } v_i \neq v_j\}$ is the set of the edges. For the particular problem, the graph is assumed to be undirected. Each edge is associated with a weight w_{ij} which stands for the distance between the two vertices. For the proposed algorithm, this distance is the similarity distance among sequences, where each sequence is modeled as a vertex of the graph. This similarity matrix is symmetrical, i.e., $w_{ij} = w_{ji}$, thus the resulting TSP instance is also symmetric.

IV. Our approach

In this Section, we describe the proposed algorithm for solving the MSA problem. The main algorithm is presented as pseudocode which reveals the rationale behind its functionality. First, the quantum-inspired part of the heuristic (called qGVNS [31], [32]) is presented and then the description of the complete algorithm follows. The quantum-inspired part of the algorithm is based on the work in [32].

In order to improve the computational time, it is acceptable to lose some of the solution's quality by adopting heuristic and metaheuristic approaches [33], [34]. Heuristics are fast approximation computational methods divided into construction and improvement heuristics. Construction heuristics are used to build feasible initial solutions and improvement heuristics are applied to achieve better solutions.

Similar to the original GVNS [34], qGVNS consists of a VND local search, a diversification procedure and a neighborhood change step. Here, the pipe-VND is used during the improvement phase. When the pipe-VND's improvement phase of pipe-VND takes place, two classic local search strategies are deployed: the relocate and the 2-opt strategy. In relocate, the solutions are obtained by moving a node to a different position of the current route, whereas in the 2-opt strategy the solutions are obtained by breaking two distinct edges and consequently reconnecting them in a different order. VNS and

GVNS share similarities with the way living organisms try to adapt in a new area around their habitat [35].

Unlike classic GVNS, the used quantum-inspired heuristic utilizes a modified diversification phase which successfully resolves local minima traps within a VNS procedure. The perturbation is achieved by exploiting quantum computation techniques. In particular, a simulated quantum register generates a complex n-dimensional unit vector, during each shaking call. Note that a quantum register is actually the quantum analogue of a classical register. The dimension n of the complex unit vector is greater than, or equal, to the dimension of the problem. The complex n-dimensional vector is fed as input in the algorithm and a real n-dimensional vector (whose components are real numbers in $[0, 1]$) is its outcome. The i-th component of the real vector is equal to the modulus squared of the i-th component of the complex vector.

During this heuristic, each node of the current solution is in one to one correspondence with the components of the real n-dimensional vector. Having this mapping, between vector components and nodes, the sorting of the components of the real vector will introduce an identical ordering among the solution nodes. Therefore, the ordered route that is generated after this particular shaking move will result in guiding the algorithm to another search space. Note that the Nearest Neighbor heuristic is used in order to produce an initial feasible solution. From an algorithmic perspective, the procedure is summarized in the next pseudocode fragment in Algorithm 1 from [31].

Algorithm 1: Pseudocode of GVNS routine

```
Data: an initial solution
Result: an optimized solution
Initialization of the feasibility distance matrix
begin
    $X \leftarrow$ Nearest Neighbor heuristic;
    repeat
        $X' \leftarrow$ Quantum-Perturbation($X$)
        $X'' \leftarrow$ pipeVND($X'$)
        if $X''$ is better than $X'$ then
            $X \leftarrow X''$
        end
    until optimal solution is found or time limit is met;
end
```

The main contribution of this paper, i.e., the proposal of a MSA algorithm, is presented in Algorithm 2. In order to achieve a more generic approach, no *a priori* weight concerning the relatedness of the sequences is used, since they are assumed to be unknown. Although this knowledge is considered to be advantageous, the proposed approach aims to cover the general case. It is also important to mention that since it is a heuristic approach, it is difficult to assert the procedure's formal complexity.

Sol declares a matrix containing an approximation of the shortest Hamiltonian of $Dist$. It is calculated through qGVNS of Algorithm 1, which is a heuristic algorithm and that is why we have an approximation. The algorithm starts with the calculation of the distance matrix among the sequences. This is a trivial part for the most MSA algorithms, although heuristics that provide approximation solutions exist for the cases of many sequences, in order to lower the computation effort. Such methods could be incorporated in future extensions of the proposed algorithm. The choices for a score matrix and a gap penalty are irrelevant to the main algorithm; they can be separately chosen. This is an advantage of the proposed algorithm, since it enhances its wide applicability and generic form.

Algorithm 2: The proposed algorithm for MSA

```
Data: a set of sequences
Result: a set of aligned sequences
$Dist_{i,j}$: The pairwise distance matrix of the sequences
begin
    $i \leftarrow$ the i-th sequence
    $j \leftarrow$ the j-th sequence
    repeat
        $Dist_{i,j} \leftarrow$ Calculate pairwise distance of i-th and
          j-th sequences
    until all pairs are examined;
end
$Sol \leftarrow$ Approximation of the shortest Hamiltonian of
  $Dist$ ▷ Derived using qGVNS of Algorithm 1
Score_matrix ▷ Choose a score matrix of your choice
Gap_penalty ▷ Choose the value of gap penalty
Align sequences following the $Sol$ matrix as guide
```

V. Analysis

The proposed algorithm requires the calculation of pairwise distances of the sequences, which is a standard procedure for most aligning methods that deal with an acceptable number of sequences. In case this number is tremendously scaled-up (for hundreds of thousand or even millions of sequences), particular heuristics can be used. This calculation is a demanding task that requires exhaustive comparisons. We chose to calculate the complete distance matrix in order to have a fair comparison with other methods that use a similar approach.

This calculation has complexity $\mathcal{O}(m(n^2 - n))$, where m is the maximum length of a sequence and n the number of sequences. euristics and information theoretic approaches could be considered in order to lower the complexity in case m and n get high enough (hundreds of thousand). The calculation of the shortest Hamiltonian path of the distance matrix is a known **NP**-hard problem and at first sight seems like the proposed algorithm simply moves the "hard" computational part in another routine hiding the algorithm's overall overhead. This is not true, though, since the GVNS-based heuristic we used here only approximates the problems solution in a specific time frame that is chosen according to the accuracy level the user prefers.

Then the actual alignment process of the sequences is straightforward and follows the standard methods. The only difference is that there is no need to have complex guide trees that are used as phylogenetic elements. In particular, a simple guide matrix is used as guide tree that is a simple full binary tree where each node has either 2 or 0 children. This is depicted in Fig. 1.

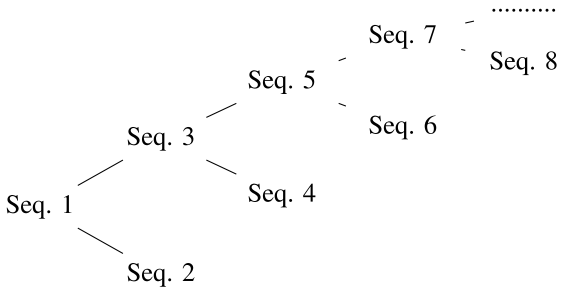


Fig. 1: A depiction of the tree produced after calculating the shortest path of the distance matrix. It is just an indicative part of the tree, since the actual tree continues until every sequence is added. Here, the numbering of sequences is arbitrary.

VI. Simulation results

After the analysis of the proposed method, in this section we present the experimental results. The proposed solution is benchmarked on 33 real sequences retrieved from [17]. Samples from different kinds of organisms were chosen. In particular, sequences of various lengths coding proteins and RNA of *archaea*, *invertebrates*, *plants*, and *plasmids* were used. A simulation program in *MatLab* was developed, using the *GONNET* scoring matrix for every scenario.

Simulation results demonstrate that the proposed solution is capable of better sum-of-pairs score, especially for larger sequences, both for the case of negative mismatches and simple matches. The results are illustrated in Fig. 2. The results are split into three parts for each of the two scoring schemes. Also, sequences are depicted in decreasing order of score in order to enhance their readability.

For Figs. 2a-2c, the score was measured using the sum of pairs where each match is scored with 1, whereas any mismatch is scored with -1 (this explains the negative total score for most sequences). The higher the score, the better was the alignment. Similarly, for Figs. 2d-2f, the score was measured using the sum of pairs without negative score for mismatches (a match was still rewarded with 1). Again, the higher the score, the better the alignment. The numerical results for these experiments are provided in the Appendix of the paper.

The proposed algorithm was compared against two well-known algorithms. The first one is the UPGMA algorithm [24], [10], and the other one is the single neighbor algorithm (its simplest implementation, extensions and modifications of the single neighbor approach could be considered in a future work). Both of them operate under the progressive alignment scheme (like our algorithm) and they use the full distance matrix, again, similarly to our algorithm. For these reasons, these particular algorithms were chosen for the evaluation part, since all of them aim at producing an initial alignment which could be further optimized in later stages using appropriate methodologies. Finally, all the algorithms use the same method for pairwise alignment.

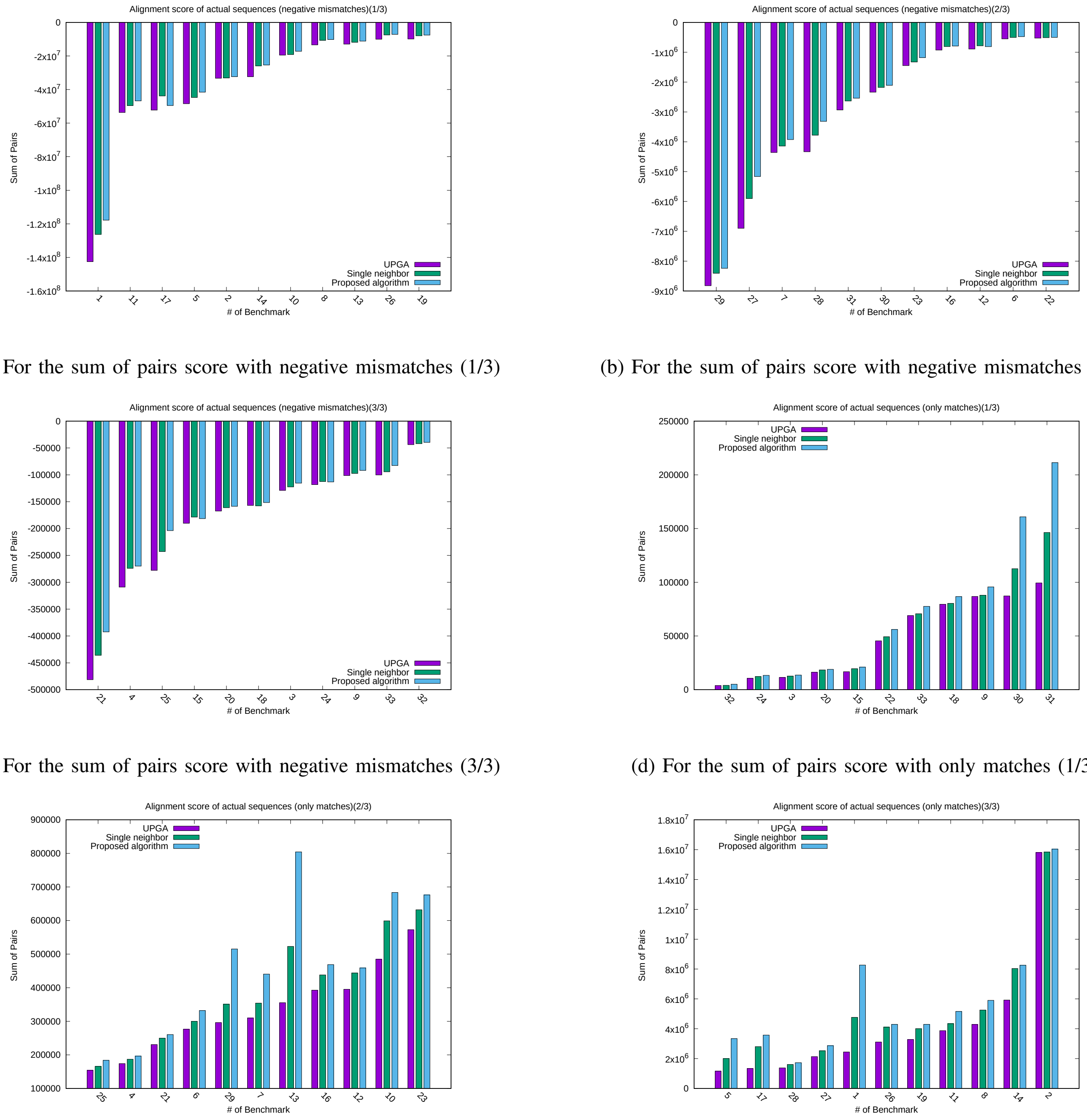
The experimental results demonstrate that the proposed algorithm outperforms the other two, especially when larger sets of sequences are used (since for relatively smaller sequences the proposed heuristic would not yield any considerable improvement compared to the other approaches). The increase in the performance is notable in many occasions, but there are also instances that the difference is marginal. These observations show that the proposed algorithm is a promising solution, especially for cases where no prior knowledge on the relation among the sequences is known and a quick alignment is needed, with as few as possible mistakes (as measured through the chosen scoring function).

VII. Conclusion and future work

Data related to biology and genomics are characterized by large volume and require enormous computational power. Biological sequences are usually large and demand massive computing resources in order for someone to be able to correlate them and further study particular properties. Multiple sequence alignment (or MSA) tackles the problem of aligning several strings resulting in a complete alignment.

There are many different approaches to solve the MSA problem and plenty of scoring methods to evaluate the alignment results. In this work, a novel method to progressively solve the multiple sequence alignment problem is proposed using elements of quantum-inspired optimization. The proposed algorithm is described in detail and then evaluated through simulations on actual sets of sequences from [17]. These results seem promising, revealing that the proposed algorithm is capable of providing alignments with good scores, especially when the sets of sequences are large. The algorithm's output can be used as an initial alignment with acceptable score that could be further optimized in later stages.

Future extensions could be considered in combination with other bio-inspired approaches, especially those belonging to the field of evolutionary computation. Also, potential enhancements that would cover particular cases taking into consideration prior knowledge on the relation of the sequences that need to be aligned could be considered. Finally, adjustments and heuristics regarding the computation of the distance matrix, sometimes used by other approaches, could also be useful, especially for larger sets of sequences. In any case, the work described here acts as a more general approach that covers the most usual cases and guarantees efficient results.



(a) For the sum of pairs score with negative mismatches (1/3)

(b) For the sum of pairs score with negative mismatches (2/3)

(c) For the sum of pairs score with negative mismatches (3/3)

(d) For the sum of pairs score with only matches (1/3)

(e) For the sum of pairs score with negative mismatches (2/3)

(f) For the sum of pairs score with negative mismatches (3/3)

Fig. 2: Simulation results on actual sets of sequences retrieved from [17]. Sequences are ordered according to their size.

Acknowledgment

This research is funded in the context of the project "Investigating alternative computational methods and their use in computational problems related to optimization and game theory, (MIS 5007500) under the call for proposals "Supporting researchers with an emphasis on young researchers (EDULL34). The project is co-financed by Greece and the European Union (European Social Fund - ESF) by the Operational Programme Human Resources Development, Education and Lifelong Learning 2014-2020.

References

[1] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[2] K. Nguyen, X. Guo, and Y. Pan, *Multiple Biological Sequence Alignment: Scoring Functions, Algorithms and Evaluation*. John Wiley & Sons, 2016.

[3] D. A. Morrison, "Multiple sequence alignment is not a solved problem," *arXiv preprint arXiv:1808.07717*, 2018.

[4] A. Abbott and A. Tsay, “Sequence analysis and optimal matching methods in sociology: Review and prospect,” *Sociological methods & research*, vol. 29, no. 1, pp. 3–33, 2000.

[5] J. Prokić, M. Wieling, and J. Nerbonne, “Multiple sequence alignments in linguistics,” in *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education*. Association for Computational Linguistics, 2009, pp. 18–25.

[6] L. Wang and T. Jiang, “On the complexity of multiple sequence alignment,” *Journal of computational biology*, vol. 1, no. 4, pp. 337–348, 1994.

[7] I. Elias, “Settling the intractability of multiple alignment,” *Journal of Computational Biology*, vol. 13, no. 7, pp. 1323–1339, 2006.

[8] W. Just, “Computational complexity of multiple sequence alignment with sp-score,” *Journal of computational biology*, vol. 8, no. 6, pp. 615–623, 2001.

[9] S. Shehab, S. Abdulah, and A. E. Keshk, “A survey of the state-of-the-art parallel multiple sequence alignment algorithms on multicore systems,” *arXiv preprint arXiv:1805.12223*, 2018.

[10] T. J. Wheeler and J. D. Kececioglu, “Multiple alignment by aligning alignments,” *Bioinformatics*, vol. 23, no. 13, pp. i559–i568, 2007.

[11] G. Blackshields, F. Sievers, W. Shi, A. Wilm, and D. G. Higgins, “Sequence embedding for fast construction of guide trees for multiple sequence alignment,” *Algorithms for Molecular Biology*, vol. 5, no. 1, p. 21, 2010.

[12] J. Kim, S. Pramanik, and M. J. Chung, “Multiple sequence alignment using simulated annealing,” *Bioinformatics*, vol. 10, no. 4, pp. 419–426, 1994.

[13] S. Lindgreen, P. P. Gardner, and A. Krogh, “MASTR: multiple alignment and structure prediction of non-coding rnas using simulated annealing,” *Bioinformatics*, vol. 23, no. 24, pp. 3304–3311, 2007.

[14] A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rose, and A. Aspuru-Guzik, “Finding low-energy conformations of lattice protein models by quantum annealing,” *Scientific reports*, vol. 2, p. 571, 2012.

[15] “D-Wave initiates open quantum software environment,” https://www.dwavesys.com/press-releases/d-wave-initiates-open-quantum-software-environment, accessed: 2019-04-09.

[16] L. Abdesslem, M. Soham, and B. Mohamed, “Multiple sequence alignment by quantum genetic algorithm,” in — *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*. IEEE, 2006, p. 360.

[17] “NCBI RefSeq project, national center for biotechnology information,” ftp://ftp.ncbi.nlm.nih.gov/refseq/release/, accessed: 2019-04-09.

[18] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[19] T. F. Smith and M. S. Waterman, “Comparison of biosequences,” *Advances in applied mathematics*, vol. 2, no. 4, pp. 482–489, 1981.

[20] R. C. Edgar, “MUSCLE: multiple sequence alignment with high accuracy and high throughput,” *Nucleic acids research*, vol. 32, no. 5, pp. 1792–1797, 2004.

[21] F. Corpet, “Multiple sequence alignment with hierarchical clustering,” *Nucleic acids research*, vol. 16, no. 22, pp. 10 881–10 890, 1988.

[22] J. D. Thompson, D. G. Higgins, and T. J. Gibson, “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice,” *Nucleic acids research*, vol. 22, no. 22, pp. 4673–4680, 1994.

[23] F. Sievers and D. G. Higgins, “Clustal omega for making accurate alignments of many protein sequences,” *Protein Science*, vol. 27, no. 1, pp. 135–145, 2018.

[24] T. Stefan Van Dongen and B. Winnepenninckx, “Multiple UPGMA and neighbor-joining trees and the performance of some computer packages,” *Mol. Biol. Evol*, vol. 13, no. 2, pp. 309–313, 1996.

[25] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[26] A. Sarkar, “Quantum algorithms: for pattern-matching in genomic sequences,” 2018.

[27] L. C. Hollenberg, “Fast quantum search algorithms in protein sequence comparisons: Quantum bioinformatics,” *Physical Review E*, vol. 62, no. 5, p. 7532, 2000.

[28] H.-W. Huo, V. Stojkovic, and Q.-L. Xie, “A quantum-inspired genetic algorithm based on probabilistic coding for multiple sequence alignment,” *Journal of bioinformatics and computational biology*, vol. 8, no. 01, pp. 59–75, 2010.

[29] J. D. B. Tapia, Y. J. T. Valdivia, J. H. C. Humari, and P. Arequipa, “Optimizing multiple sequence alignments using traveling salesman problem and order-based evolutionary algorithms,” *Proceedings of CIBB*, vol. 2, p. 1, 2012.

[30] C. Korostensky and G. H. Gonnet, “Using traveling salesman problem algorithms for evolutionary tree construction,” *Bioinformatics*, vol. 16, no. 7, pp. 619–627, 2000.

[31] C. Papalitsas, P. Karakostas, and K. Kastampolidou, “A quantum inspired GVNS: Some preliminary results,” in *GeNeDis 2016*, P. Vlamos, Ed. Cham: Springer International Publishing, 2017, pp. 281–289.

[32] C. Papalitsas, P. Karakostas, T. Andronikos, S. Sioutas, and K. Giannakis, “Combinatorial GVNS (general variable neighborhood search) optimization for dynamic garbage collection,” *Algorithms*, vol. 11, no. 4, p. 38, 2018.

[33] R. F. D. Silva and S. Urrutia, “A general VNS heuristic for the traveling salesman problem with time windows,” *Discrete Optimization*, vol. 7, no. 4, pp. 203–211, 2010.

[34] N. Mladenovic, R. Todosijevic, and D. Urosevic, “An efficient GVNS for solving traveling salesman problem with time windows,” *Electronic Notes in Discrete Mathematics*, vol. 39, p. 8390, 2012.

[35] M. B. B. Loranca, R. G. Velazquez, E. O. Benitez, D. P. Avendano, J. P. Rodriguez, and J. L. M. Flores, “Partitioning with variable neighborhood search: A bioinspired approach,” in *2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)*. IEEE, 2012, pp. 150–155.

Appendix

N	Size	UPGA	single	Proposed Alg.
1	898	-142523088	-126315814	-117792189
2	237	-33293391	-33101406	-32336818
3	21	-128976	-122415	-115459
4	21	-309000	-274034	-269871
5	536	-48422862	-44718577	-41554521
6	28	-552519	-503351	-477481
7	118	-4358519	-4141859	-3925419
8	135	-13384128	-10699662	-10203670
9	16	-101230	-97476	-91816
10	267	-19483390	-19146707	-17220682
11	503	-53667376	-49551353	-46743623
12	35	-892612	-783885	-810280
13	233	-12952986	-11877690	-11132666
14	207	-32364856	-25950595	-25393594
15	31	-190255	-178685	-181522
16	38	-928216	-808198	-793102
17	408	-52249519	-43821648	-49528355

N	NoS	UPGA	single	Proposed Alg.
18	18	-156976	-157776	-151573
19	117	-9869902	-7972916	-7558387
20	27	-167478	-161125	-158635
21	31	-481274	-435969	-392305
22	44	-525386	-509893	-501521
23	46	-1447160	-1327145	-1180051
24	24	-118125	-112477	-113219
25	24	-277897	-242824	-203750
26	126	-1E+07	-7494668	-7118485
27	100	-6897746	-5901467	-5161430
28	76	-4334603	-3778940	-3315710
29	195	-8822833	-8404489	-8239317
30	644	-2337085	-2177805	-2107528
31	124	-2933740	-2633563	-2538346
32	18	-43724	-41981	-39625
33	19	-100167	-94268	-82618

TABLE I: Numerical results regarding sets of sequences from [17]. The first column is the number of the benchmark, Size includes the size of each set. Score is calculated by giving 1 for matches and -1 for mismatches.

N	Size	UPGA	single	Proposed Alg.
1	898	2442464	4763508	8268588
2	237	15808398	15842811	16039622
3	21	11393	12702	13673
4	21	173756	186961	196307
5	536	1157632	2004663	3340256
6	28	276324	299969	331963
7	118	309752	354226	440475
8	135	4283585	5255672	5901962
9	16	86817	87957	95701
10	267	484669	599044	683589
11	503	3871356	4341563	5157986
12	35	395375	443729	458912
13	233	355247	522927	804438
14	207	5926077	8035926	8253775
15	31	16747	19574	21027
16	38	392727	438003	468757
17	408	1333885	2805038	3575982

N	NoS	UPGA	single	Proposed Alg.
18	18	79562	80393	86673
19	117	3280783	4005033	4293242
20	27	16283	18400	18947
21	31	230730	249768	260459
22	44	45488	49297	56119
23	46	572563	632060	676404
24	24	10611	12296	13339
25	24	153919	165578	183849
26	126	3113338	4108543	4294035
27	100	2133176	2524886	2866489
28	76	1378922	1605658	1726975
29	195	296065	350825	515435
30	110	87262	112610	160942
31	124	99393	146312	211388
32	18	3881	4073	5136
33	19	68984	70704	77594

TABLE II: Numerical results regarding sets of sequences from [17]. The first column is the number of the benchmark, Size includes the size of each set. Score is calculated by giving 1 for matches and 0 for mismatches.