Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on NSGA-II

Francisco Ortuno, Javier P. Florido, Jose M. Urquiza, Hector Pomares, Alberto Prieto and Ignacio Rojas

Department of Computer Architecture and Computer Technology
University of Granada, Granada (Spain)

Email: fortuno@atc.ugr.es, jpflorido@ugr.es, jurquiza@atc.ugr.es, hector@atc.ugr.es, aprieto@atc.ugr.es and irojas@atc.ugr.es

Abstract—Multiple sequence alignment (MSA) is one of the most studied approach in Bioinformatics to carry out other outstanding tasks like structural predictions, biological function analysis or next-generation sequencing. However, MSA algorithms do not achieve consistent results in all cases, as alignments become difficult when sequences have low similarity. In other words, each algorithm is focused in specific features of sequences and their results depend on them. For this reason, each approach could align better those sections of sequences that include such features, obtaining partially optimal solutions. In this work, a multiobjective evolutionary algorithm based on NSGA-II will be implemented in order to assemble previously aligned sequences, trying to avoid suboptimal alignments.

Index Terms—multiple sequence alignment (MSA), multiobjective, evolutionary algorithms, NSGA-II, optimization.

I. INTRODUCTION

Multiple sequence alignment (MSA) has become a widely used task of the current Molecular Biology. This technique is useful in order to compare new sequences with those well-known ones, extracting their shared information and their significant differences. MSA methods have traditionally been essential to analyze biological sequences and to design applications in structure modeling, functional prediction, phylogenetic analysis and sequence database searching [1]. Currently, these approaches are also used in applications, for example, to compare protein structures [2], to predict protein mutations and interactions [3] [4] or to reconstruct phylogenetic trees [5].

MSA methodologies are also relevant due to new experimental techniques like high-throughput experiments or next-generation sequencing. These technologies provide a large amount of data that has to be analyzed, processed and assessed. Current computational strategies are then required to extract biological meanings from such information. MSA methodologies also take advantage of this heterogeneous data provided by recent biological progress in functional, structural and genomic researches to obtain more accurate alignments in a reasonable time [6]. With this environment, MSA represents one of the more powerful and useful procedure of analysis in Bioinformatics.

First tools in MSA were designed to align only two sequences (pairwise alignment), using dynamic programming (DP) procedure. These alignments are usually evaluated by scoring matrices as Needleman-Wunsch [7] or Smith-Waterman [8]. Although DP finds the optimal alignment, it assesses all possible alignments. Therefore, this strategy cannot be applied for multiple alignments, as computational cost is unapproachable with higher number of sequences. For this reason, other approaches like progressive algorithms or consistency-based methods were developed to produce multiple alignments.

Progressive algorithms obtain multiple sequence alignments progressively assembling previously built pairwise alignments through a clustering method. Some representative programs of this procedure are ClustalW [9], Muscle [10] and Kalign [11]. These methods achieve high accuracy when sequences have high similarity but they could produce suboptimal alignments if sequences are more distant or some noise is introduced in previous alignments. These drawbacks are usually reduced in current strategies, working in two steps: first, alignments are obtained by the normal procedure; and second, the clustering method is rebuilt from the final result in order to optimize the alignment.

These errors can also be removed by consistency-based strategies as T-Coffee [12] or, more currently, MSAProbs [13]. Consistency-based approaches draw on iterative techniques applying, for example, refinement strategies [14] or hidden Markov models [15] to realign subsequences, improving the final alignment. Evolutionary algorithms have also been applied in these techniques to accurately align sequences [16].

Finally, as shown in this previous study of different strategies [17], other methodologies including more heterogeneous information can be applied, e.g. Promals [18] or 3D-Coffee [19]. Features like domains, structures or homologies are retrieved from several databases to enrich the alignment information. However, the consumed time to align sequences with this kind of algorithms is excessive and improvements only are relevant in specific cases.

Taking all this into account, a new multiobjective evolutionary algorithm will be proposed to optimize some sequences previously aligned by these other methodologies. Evolutionary approaches have already been widely used in multiple sequence alignment tools like SAGA [20] or AlineaGA [21]. In this case, a NSGA-II approach is implemented. This algorithm defines a novel representation of alignments that is useful...
to design efficient mutation and crossover operators. Since several independent evaluations to estimate the accuracy of an alignment are usually applied [22], we propose a multiobjective approach including three of them: SP-score, total columns and number of gaps. This algorithm will then be trained and tested with sets of sequences provided by the benchmark BAliBASE [23].

II. MATERIALS AND METHODS

A. Input dataset: BAliBASE

In order to validate the proposed multiobjective algorithm, a set of input sequences has to be defined. Currently, a huge amount of datasets and techniques have been designed to standardize the comparison of sequence alignment results, for example, OXBench [24], HOMSTRAD [25] or Prefab [10]. In our case, the well-known BAliBASE dataset (v3.0) [23] has been selected.

This dataset consists of several protein sequences that can be easily aligned by standard alignment algorithms. The complete dataset includes 218 sets of sequences that have manually been extracted from different databases. Sequences are organized according to some specific features (similarity, structure in PDB database [26] or families) in the next subsets: (i) equidistant PDB sequences sharing <20% of identity between any pair of sequences (RV11 and RV12); (ii) PDB orphan sequences from families with >40% of identity and, at least, one known 3D-structure (RV20); (iii) subfamilies sharing >40% of identity but <20% with other subfamilies (RV30); (iv) sequences with >20% of identity but including N/C terminal extensions (RV40); and (v) sequences with internal insertions associated with other different sequences (RV50). BAliBASE also defines two scores (SP-score and TC-score) to evaluate the obtained alignments against a set of handmade optimal alignments (gold standard).

B. Alignment approaches

As described before, current MSA algorithms apply techniques like progressive strategies or consistency-based methods to align multiple sequences. In this paper, eight representative MSA tools have been selected to obtain the alignments that will be optimized.

Among progressive algorithms, ClustalW [9], Muscle [10], Kalign [11], Mafft [27] and RetAlign [28] have been selected. ClustalW designs a clustering tree algorithm to find the multiple alignment by means of distance scores and a gap weighting scheme obtained from previous pairwise alignments. Muscle develops a three-stage strategy (based on a tree dependent partitioning approach) to refine alignments and to align faster. Kalign employs the Wu-Manber string-matching algorithm to improve the measurement of distances of a classical progressive approach. Mafft reduces the computational cost using the Fast Fourier Transform (FFT) to rapidly identify common homologies. Finally, RetAlign uses a progressive corner cutting algorithm to identify optimal and suboptimal alignments in a network.

Three more sophisticated algorithms based on consistency are also included in the optimization. T-Coffee [12] obtains the final alignment storing in a library the number of times that each pair of residues is aligned in previous pairwise alignments. FSA (Fast Statistical Alignment) [15] compares pairwise sequences by a statistical analysis framework. It also estimates the insertion and deletion processes through a pair of hidden Markov models (HMM). Finally, ProbCons [14] also includes hidden Markov models to optimize the classical scoring schemes. It applies a bi-phasic penalty procedure in order to penalize gaps and mismatches in alignments.

A summary of these methodologies is shown in Table I. The eight programs were applied in order to align the 218 sets of sequences proposed by BAliBASE. All of them were executed with their default configuration and parameters.

C. Multiobjective Evolutionary Algorithm

In this paper, a new evolutionary approach to optimize alignments from other multiple sequence algorithms is proposed. This system is designed as a multiobjective algorithm including three different evaluations related to the similarity and the gaps in the alignment. Among the different approaches in multiobjective optimization, the Non Dominated Sorting Genetic Algorithm (NSGA-II) scheme [29] has been chosen. NSGA-II is a classical but recognized method that produces efficient solutions. This algorithm identifies the optimal Pareto front by means of the application of a non-dominated sorting strategy. Then, NSGA-II defines a complete procedure of selection using the dominance concept. Besides, chromosomes can also be selected by the crowding distance to include more diversity in the following generations.

Formally, the optimization problem is formulated as follows. Alignments obtained from the eight previous methodologies are included in the vector \( A = \{a_1, a_2, ..., a_8\} \). The population in the \( i^{th} \) generation is defined as \( P_i \). \( M \) and \( C \) include alignments which are produced by mutation and crossover respectively. Other configurable parameters are also defined to complete the algorithm: the number of chromosomes in the population \( (n) \), the number of generations \( (g) \) or the percentage of mutation \( (p_m) \) and crossover \( (p_c) \). Such parameters are previously tuned by an empirical procedure. Finally, the \( F_f \) set returns alignments derived from the last optimal Pareto front. Following, the complete pseudo-code scheme of the optimization procedure is shown:

### Table I

<table>
<thead>
<tr>
<th>METHOD</th>
<th>VERSION</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClustalW [9]</td>
<td>2.0.10</td>
<td>Progressive</td>
</tr>
<tr>
<td>MUSCLE [10]</td>
<td>3.8.31</td>
<td>Progressive</td>
</tr>
<tr>
<td>Kalign [11]</td>
<td>2.0.8</td>
<td>Progressive</td>
</tr>
<tr>
<td>Mafft [27]</td>
<td>6.85</td>
<td>Progressive</td>
</tr>
<tr>
<td>RetAlign [28]</td>
<td>1.0</td>
<td>Progressive</td>
</tr>
<tr>
<td>TCOFFEE [12]</td>
<td>8.97</td>
<td>Consistency-based</td>
</tr>
<tr>
<td>ProbCons [14]</td>
<td>1.12</td>
<td>Consistency-based</td>
</tr>
<tr>
<td>FSA [15]</td>
<td>1.15.5</td>
<td>Consistency-based</td>
</tr>
</tbody>
</table>
This implementation of the NSGA-II approach was taken from the Global Optimization toolbox of @Matlab (version R2010b), designing our own implementations of operators and fitness functions:

1) Codification and Initialization: Previous evolutionary algorithms in multiple sequence alignment [20] [21] codified alignments using the classical representation: the standard alphabet for aminoacids and the "-" symbol for gaps (Figure 1(a)). However, a novel codification is defined for this work. To efficiently design our crossover operator, alignments must be codified in order to identify exact positions where they can be crossed. Consequently, alignments are represented as a matrix including two conditions: (i) aminoacids are codified by their positions in the sequence where they belong; (ii) gaps are codified by the position of the last aminoacid in the sequence but with a negative value. An example of the proposed codification is shown in Figure 1(b). That representation produces significant improvements in the management of alignments and it avoids possible mistakes in the subsequent crossover performance. The relevance of using such representation is described in the definition of the crossover operator.

On the other hand, the initial population is firstly formed by alignments from the eight selected methodologies. The population is then completed to the final length (100 chromosomes) using the mutation operator defined below.

2) Operators: The two standard operators have been defined in this system: mutation and crossover. Since the order in sequences cannot be altered, some modifications must be introduced to both operators. In the case of mutation, only gaps are mutated in order to maintain the positions of the aminoacids. The mutation operator is then defined as the shift of a randomly chosen set of closed gaps. These gaps are selected from one sequence and they are shifted to another position in the same sequence. Two important aspects are introduced with this definition: first, new variants not taken into account until now can be introduced; and second, columns only containing gaps will be removed, reducing the number of gaps in some cases (gaps are usually overused in alignment methods [30]). A specific example of the mutation operation is shown in the Figure 2.

The crossover definition is the most important issue in this procedure in order to obtain better solutions than previous alignments. This operation is essential in our problem, as alignment methodologies can be accurate in some sectors of the sequences but suboptimal in others. The main condition
Fig. 3. Example of the crossover operation using the novel representation. Identical columns are identified in both representations and blocks between two of them are crossed, creating two new solutions. The information in sequences is not altered with this procedure.

is to maintain the complete sequences, but crossing in some positions does not fulfill such condition. Consequently, positions where two chromosomes can be crossed are restricted. Thus, the codification described before permits to easily calculate these suitable positions. The system must identify those columns that are identical in both codified matrices. In other words, if two pair of identical columns are randomly chosen and blocks between them are crossed, we can assure that the sequences are not altered in the two new chromosomes. The complete procedure of crossover is graphically explained in Figure 3.

3) Evaluation: The evaluation of solutions is usually done through the fitness function. Since this approach is designed as a multiobjective algorithm, several fitness functions have to be defined. Specifically, this system includes three different scores to evaluate each alignment: PAM250 score, percentage of non-gaps and percentage of completely aligned columns. The first one is a typical SP-score calculated using the well-known scoring matrix PAM250. This matrix includes an evaluation for each two possible aligned aminoacids according to their similarity and penalizations of gaps. The total score is then calculated as the accumulated scores of every pair of aligned aminoacids. The second evaluation is calculated as the number of aminoacids in sequences regarding the total size of the alignment. That is, the percentage of aminoacids according to the number of gaps is measured. Thus, the algorithm tries to reduce the number of gaps, which are usually overused in some alignment methodologies [30]. Finally, the third evaluation consists of the number of columns that are completely aligned with exactly the same aminoacid. Complete columns are not usually favored in methodologies using progressive strategies [22], although they are considered in some alignment evaluations [23]. Besides, sections with complete columns in alignments indicate more conserved regions in sequences. These three fitness functions must be maximized in order to optimize the alignment results.

4) Selection: The selection procedure is clearly defined by the NSGA-II algorithm [29]. First, the total population for each generation, including parent alignments and new alignments created by operators, is classified in different Pareto fronts in order to obtain a non-dominated sorting. Then, the best non-dominated Pareto fronts are progressively included in the population of the next generation. Finally, when the new population includes the required number of chromosomes, the remaining Pareto fronts are discarded. A special case of this selection is the last considered front, as it is possible that only some solutions in such front are necessary to complete the next generation. In this case, NSGA-II proposes to include those solutions that are situated in little explored areas or, in other words, distant solutions. Thus, solutions in the last front are selected according to their distances to other solutions. This distance is defined by NSGA-II as crowding distance. For each solution \( i \), the crowding distance is measured as shown in Equation 1.

\[
d_i = \sum_{m=1}^{N} \frac{f_{i+1}^m - f_{i-1}^m}{f_{max}^m - f_{min}^m}
\]

where \( N \) is the number of objective (three in our algorithm); \( f_{i+1}^m \) and \( f_{i-1}^m \) are the nearest previous and posterior neighbours in the objective \( m \); and \( f_{max}^m \) and \( f_{min}^m \) are the maximum and minimum values of the objective \( m \), respectively.
TABLE II
TUNED PARAMETERS IN ORDER TO OPTIMIZE THE 218 ALIGNMENT PROBLEMS

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>100</td>
</tr>
<tr>
<td>Number of generations</td>
<td>500</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.8</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.2</td>
</tr>
<tr>
<td>Repetitions of each problem</td>
<td>10</td>
</tr>
</tbody>
</table>

D. Performance Assessment

The proposed evolutionary algorithm is defined as a stochastic procedure, because different solutions are returned when the method is applied to the same problem several times. Several solutions of the same problem must then be obtained in order to statistically evaluate its performance. Some proposals are also found in the bibliography to carry out the assessment of multiobjective stochastic optimizers [31]: hypervolume, dominance rankings, attainment function method, etc. These quality indicators aim to reduce Pareto front solutions to an unique value which is easier to compare.

In this sense, this work applies the hypervolume indicator. Hypervolume solution was proposed by Zitzler et al. [32] in order to measure the portion of the objective space which is dominated by a Pareto front according to a bounding point. This option is the most suitable for our purpose. Hypervolume returns a more realistic and accurate value than dominance ranking, but without the computational complexity of other indicators as the attainment function solution.

However, this is not the only strategy taken into account in the evaluation of this algorithm. Since a stochastic system is performed, the relevance of obtained results must be statistically validated. These approaches are usually analyzed using non-parametric tests [33]. In this work, the classical non-parametric statistical test called Kruskal-Wallis [34] is used. The Kruskal-Wallis test returns a p-value that indicates if two independent variables are significantly different. Thus, the evolutionary optimizer is run several times with the same problem and hypervolume is measured for each solution. Consequently, final results can be statistically compared against alignments of previous methodologies.

III. RESULTS

The 218 problems proposed by BAliBASE have previously been aligned with the eight selected methodologies. As described above, the initial population for each problem was formed by the eight previous alignments and new alignments provided by the mutation operator. The optimization was repeated 10 times for each problem. So, a total of 2180 Pareto fronts were obtained (10 solutions by 218 problems). Previously, the NSGA-II approach was also run several times with different configurations in order to find the optimal parameters. Such parameters have been heuristically tuned to achieve the most efficient results (Table II). Nevertheless, we have empirically proved that the final results are not excessively modified when other suboptimal parameters are set.

A typical solution from one of the problem (BB12021) is shown in Figure 4. This obtained alignment is provided by partial alignments of other methods like RetAlign, TCOFFEE or Kalign and shifting some groups of gaps. The three fitness results for the same problem are shown in Table III. This table shows how the five optimized results improve the eight previous alignments in, at least, one evaluation.

However, hypervolume is a more suitable measure for comparing results against previous alignments. Thus, boxplots in Figure 5 represent the hypervolume results for six different problems, defining the vector (2, 2, 2) as the bounding point. The three objectives were previously normalized to the range [0, 1] in order to give them the same relevance. These hypervolume results clearly show smaller values in
output alignments have to be maximized, better results entail smaller hypervolume values (the higher objectives, the less covered space in hypervolume).

### IV. Discussion

According to this previous review of methodologies [17], alignment approaches are highly dependent on sequence features. It is then reasonable to suppose that each methodology holds specific abilities to perform in a better way some parts of the alignment. This observation suggests that several alignments obtained from different methodologies could be mixed to build a new possible solution (See Example in Figure 4). Since the best aligned blocks in sequences are joined in another solution, this new result then produces a more accurate alignment than previous ones.

Therefore, the proposed multiobjective algorithm builds an ensemble of several alignments to obtain an optimized solution. The algorithm also reduces the number of gaps in order to achieve more compacted and realistic results. Boxplots presented in Figure 5 show how hypervolume indicators of obtained alignments clearly present optimized values against the initial eight alignments. Extrapolating this to the total number of problems and the times that each problem is repeated, hypervolume values show strictly better output alignments in 215 problems $98.62\%$. Only in the case of the problem BB40011 the alignment cannot be optimized in any of the ten times. The other two unsuccessful problems, BB20008 and BB40041, improve previous alignments in some cases but, in general, hypervolume evaluations are worst. These failures are produced by some special features presented in such problems: all of them include higher number of sequences with highly variable lengths. That situation makes more difficult the ensemble. Besides, these problems belong to groups of BAliBASE (RV20 and RV40) where results already are accurate with previous strategies.

However, this evaluation is not enough. It is not only necessary to know if alignments have been improved, but also if the improvement is significant. The non-parametric Kruskal-Wallis test is applied to study that significance. Ten executions have been done for each problem to determinate the relevance of the 218 optimizations (we will extend to 30 repetitions in the future). P-values provided by this test will confirm if output hypervolumes are statistically different against input ones. The significant level used to reject the null hypothesis in the 218 problems and to validate that the improvement is relevant has been set to $\alpha < 0.01$.

Thus, the Kruskal-Wallis test shows that 212 of the remaining hypervolume indicators are significantly improved. In other words, the $97.24\%$ of problems are successfully optimized using the proposed evolutionary algorithm. P-value results also confirm that most of the problems are being optimized in all cases ($p$-values=$2\times10^{-4}$). On the other hand, just in a few problems, the optimization was not possible in only one of the ten cases ($p$-value=$2.5\times10^{-3}$). Finally, the three problems which do not return relevant improvements according to the Kruskal-Wallis test were: BB11026, BB40015 and BB40049. Solutions for the RV40 subset appear again in this case and they also include highly divergent sequence lengths. A summary of p-value results is shown in Table IV.

Since we are dealing with an optimization problem, the
required computational time cannot be compared against initial methodologies. In general, the optimization of a specific alignment will spend more time than each individual method. However, although the complexity of this optimization is still high, the proposed implementation permits us to parallelize some subtasks. Hence, the algorithm time could be reduced to similar values as the applied methodologies. The required computational time is also clearly lower than that obtained by other more sophisticated methodologies including additional information, such as 3DCoffee [19] and Promals [18]. Moreover, alignments proposed here can also be compared in terms of accuracy to those built by these methodologies.

Despite these specific drawbacks, obtained alignments suggest that the algorithm is properly working and alignments are usually optimized with significant improvements. Other statistical tests and new operators will be proposed in the future in order to increase the quality of the obtained alignments.

V. Conclusions and Future Work

Multiple sequence alignments currently are an open issue. Aligners have to be continually improved, as they are essential in the analysis of huge amount of data provided by next generation sequencing (NGS) and high throughput experiments. For this reason, efficient computational techniques are fundamental to reduce the cost of analyzing new information.

In this work, a complete system to optimize multiple sequence alignments was proposed. This algorithm was developed by means of the multiobjective approach NSGA-II. In this case, alignments previously obtained from eight methodologies (mainly progressive and consistency-based ones) were codified using a novel representation and own-designed crossover/mutation procedures. Obtained alignments were built as an ensemble of the best aligned parts from these other solutions in order to adjust sequences as accurate as possible. Results for this system showed that alignments could generally be improved without the application of more time-consuming aligners. The evaluation with hypervolume indicator and the Kruskal-Wallis test has confirmed that the algorithm achieves optimized alignments.

Some modifications are also being prepared to improve this system. Other intelligent operators will be implemented in the future. Additional features about sequences like homologies or domains will be included in order to know if the optimized alignments could be competitive against other more sophisticated technologies. Besides, a similar algorithm is being prepared to align complete sequences without previous alignments. Thus, the main goal will be to convert the current system in a new alignment methodology instead of only an optimizer.

ACKNOWLEDGMENT

This work has been partially supported by Junta de Andalucía Project P09-TIC-175476

REFERENCES


