Applied Soft Computing 41 (2016) 157–168

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Hybrid multiobjective artificial bee colony for multiple sequence alignment

Álvaro Rubio-Largo[a,*], Miguel A. Vega-Rodríguez[b], David L. González-Álvarez[b]

[a] NOVA Information Management School, University Nova of Lisbon, Portugal
[b] Department of Computer and Communications Technologies, University of Extremadura, Spain

ARTICLE INFO

Article history:
Received 28 July 2015
Received in revised form
11 December 2015
Accepted 21 December 2015
Available online 7 January 2016

Keywords:
Artificial bee colony
Multiobjective optimization
Multiple sequence alignment
Evolutionary computation
Swarm intelligence
Bioinformatics

ABSTRACT

In the bioinformatics community, it is really important to find an accurate and simultaneous alignment among diverse biological sequences which are assumed to have an evolutionary relationship. From the alignment, the sequences homology is inferred and the shared evolutionary origins among the sequences are extracted by using phylogenetic analysis. This problem is known as the multiple sequence alignment (MSA) problem. In the literature, several approaches have been proposed to solve the MSA problem, such as progressive alignments methods, consistency-based algorithms, or genetic algorithms (GAs). In this work, we propose a Hybrid Multiobjective Evolutionary Algorithm based on the behaviour of honey bees for solving the MSA problem, the hybrid multiobjective artificial bee colony (HMOABC) algorithm. HMOABC considers two objective functions with the aim of preserving the quality and consistency of the alignment: the weighted sum-of-pairs function with affine gap penalties (WSP) and the number of totally conserved (TC) columns score. In order to assess the accuracy of HMOABC, we have used the BAliBASE benchmark (version 3.0), which according to the developers presents more challenging test cases representing the real problems encountered when aligning large sets of complex sequences. Our multiobjective approach has been compared with 13 well-known methods in bioinformatics field and with other 6 evolutionary algorithms published in the literature.

1. Introduction

Any living species is represented by its biological sequence and; therefore, an accurate alignment among several biological sequences is critical for finding an evolutionary relationship among different species [1,2]. This problem is known in the literature as the multiple sequence alignment (MSA) problem [3]. However, MSAs not only allow us to infer phylogenetic relationships among living species, but also they can provide biological facts about proteins – most conserved regions are biologically significant [4]. Furthermore, an accurate MSA is highly valuable in the formulation and test hypotheses about protein 3-D structure and function, that is to say, it helps us to detect which regions of a gene are susceptible to mutation and which can have one residue replaced by another without changing the function.

The natural formulation of the MSA problem, in computational terms, is to define a model of sequence evolution that assigns probabilities to all possible elementary sequence edits and then to seek an optimal directed graph in which edges represents edits and terminal nodes represents the observed sequences [5]. Unfortunately, in biologically realistic models it is not possible to determining an optimal directed graph; therefore, we need to turn to approximate heuristics. A well-known heuristic is to optimize the sum of alignment score (SP score) between each pair of sequence.

The MSA problem may be defined as an NP-hard optimization problem [6] which can be solved by using dynamic programming with a time and space complexity of $O(k2^kL^k)$ [7] when aligning k sequences of length L. Although the use of dynamic programming guarantees mathematically optimal alignments, the problem space increases significantly with the number of sequences and with the length. In order to overcome this drawback, several heuristics have been proposed in the literature. We can classify them into two main categories: *progressive* and *iterative* alignments.

Progressive alignment is the most widely used technique for multiple sequence alignment in the literature. It basically starts aligning the closest evolutionary sequences and after that, continues with the more distant ones until all the sequences are aligned. This method presents the advantage of being simple and very fast; however, a certain level of accuracy is not guaranteed. In this way, we can highlight that the main disadvantage of this method is that it can be trapped in suboptimal alignments. Among the main multiple sequence aligners published in the literature that make use of progressive alignment are Clustal W [8], or Clustal Ω [9], Tree-based Consistency Objective Function For alignment Evaluation (T-Coffee) [10], PRANK [11], Fast Statistical Alignment (FSA) [12], or Kalign [13].

The *iterative* alignment techniques make use of one method to produce an initial alignment (such a progressive method) and then refine this initial alignment by performing diverse iterations until a given stopping criterion. The main idea behind this technique is therefore to consider the initial alignment as suboptimal and then refine it until no further improvements can be achieved. In the literature we find several approaches that takes the advantage of performing an iterative refinement in order to obtain more accurate alignments, among the main ones are MUltiple Sequence Comparison by Log-Expectation (MUSCLE) [5], Multiple Alignment using Fast Fourier Transform (MAFFT) [14], PROBabilistic CONSistency-based multiple sequence alignment (ProbCons) [15], MSAProbs [16], or MUMMALS [17]. Genetic algorithms and evolutionary computation have also been considered for solving the multiple sequence alignment problem, we find diverse genetic

* Corresponding author. Tel.: +34 927257000.
E-mail addresses: arl@unex.es (Á. Rubio-Largo), mavega@unex.es (M.A. Vega-Rodríguez), dlga@unex.es (D.L. González-Álvarez).

http://dx.doi.org/10.1016/j.asoc.2015.12.034
1568-4946/© 2015 Elsevier B.V. All rights reserved.

algorithms (GA) in the literature: Sequence Alignment by Genetic Algorithm (SAGA) [18], Multiple Sequence Alignment Genetic Algorithm (MSA-GA) [19], Rubber Band Technique Genetic Algorithm (RBT-GA) [20], Vertical Decomposition Genetic Algorithm (VDGA) [21], Genetic Algorithm for Multiple Sequence Alignment using Progressive Alignment Method (GAPAM) [22], Multiobjective Optimizer for Sequence Alignments based on Structural Evaluations (MO-SAStrE) [23]. In addition, we find other single-objective approaches based on swarm intelligence, such as Artificial Bee Colony (ABC) [24,25], Ant Colony Optimization algorithm (ACO) [26,27], or Immune Artificial System Algorithm (IMSA) [28].

In the last years some efforts were done on incorporating structural information for obtaining more accurate alignments. Basically, these methods use Protein Data Bank (PDB) structures as template in order to guide the alignment of a given set of unaligned sequences using structure-based sequence alignment methods, two examples of structural-based methods are 3D-COFFEE [29] and MO-SAStrE. The main drawback of these methods is the limited availability of PDB structures.

One of the main contributions of this work is to use multiobjective evolutionary computation to solve the MSA problem. In the literature, we find evolutionary approaches that optimize the sum-of-pairs function (SAGA [18], MSA-GA [19], RBT-GA [20], VDGA [21], GAPAM [22], ABC [24,25], ACO [27], or IMSA [28]) or the column score (RBT-GA [20], ACO [26]). In [30], a multiobjective evolutionary algorithm was implemented with the aim of assembling previously aligned sequences, trying to optimize jointly the sum-of-pairs function and the column score.

In this work, we also optimize at the same time two of the most widely-used objective functions in the literature: the weighted sum-of-pairs function with affine gap penalties (WSP) and the number of totally conserved (TC) columns score. Therefore, each objective function focuses on either preserving the quality of the alignment and consistency; respectively.

In addition, we apply a well-known swarm intelligence approach, the Artificial Bee Colony (ABC) algorithm [31]; but adapted to handle multiobjective problems, we refer to it as MOABC. The ABC algorithm was developed by D. Karaboga, inspired by the foraging and dance of honey bee colonies [31]. The swarm algorithms, such as ABC, have been successfully applied to solve real-world problems in different domains, such as the design and manufacturing problem [32], selection of cutting parameters in machining operations [33], the structural damage detection problem [34], image segmentation problems [35], image classification [36], the abnormal brain detection [37], or in the path planning problem [38].

As we have mentioned, several Genetic Algorithms (GAs) have been proposed in the literature for solving the MSA problem (SAGA [18], MSA-GA [19], RBT-GA [20], VDGA [21], GAPAM [22], or MO-SAStrE [23]). Whereas GAs take the information from 2–3 parents to generate a new solution; the algorithms based on swarm intelligence produce new individuals taking into account information not only from their parents, but also from the rest of the population. The effectiveness and goodness of the ABC against traditional GAs has been widely studied in the literature [39,40].

In the ABC algorithm, we find three types of bees: employed bees, onlooker bees, and scout bees. In the canonical ABC, an employed bee becomes scout if it reaches a certain number of iterations with no improvements, which means that this bee is replaced by a new random solution. In our proposal, when an employed bee becomes scout, its stagnated solution (alignment) will be processed by the fast and accurate Kalign [13], avoiding the stagnation of the algorithm and promoting the diversity of the population as a result. In this way, the multiobjective ABC algorithm proposed in this paper was hybridized with the progressive, fast, and accurate Kalign to boost the accuracy and effectiveness of the algorithm, we refer to it as hybrid multiobjective artificial bee colony (HMOABC). In [41], a hybrid multi-objective artificial bee colony is proposed for burdening optimization of copper strip production. The main difference between the approach proposed in [41] and ours relies on the use of a deterministic heuristics (Kalign) in the scout phase of the ABC algorithm.

The remainder of this paper is organized as follows. Section 2 describes the multiple sequence alignment problem. A detailed description of HMOABC is presented in Section 3. Section 4 is devoted to analysis of the experiments carried out and also a comparison with other approaches published in the literature. Finally, Section 5 summarizes the conclusions of the paper and discusses possible lines of future work.

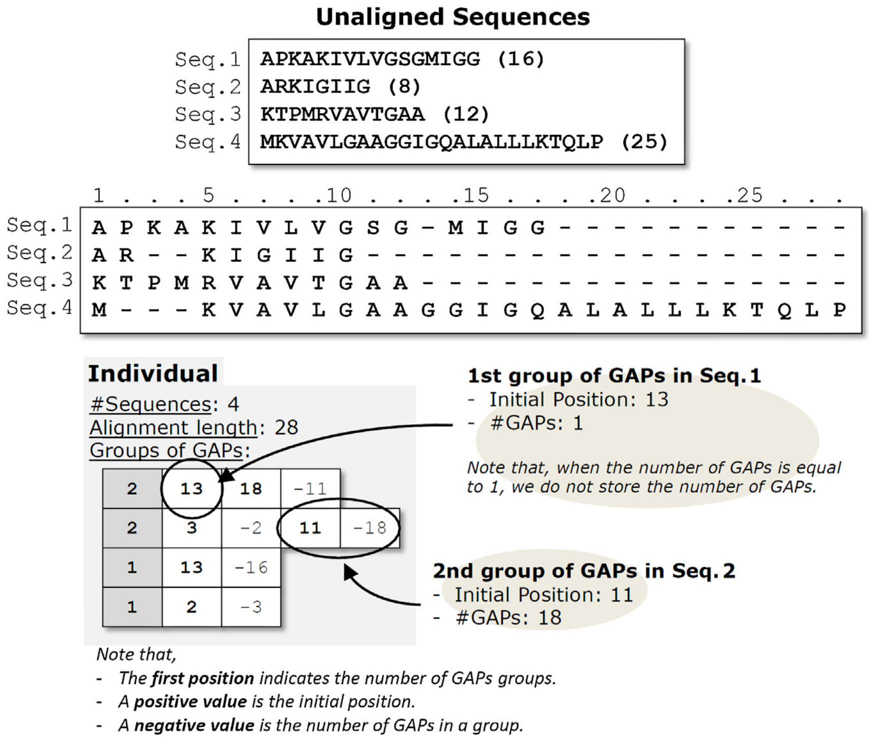
2. Multiple sequence alignment

Multiple sequence alignment (MSA) is simply an alignment of more than two sequences and is considered as an NP-hard optimization problem [42]. The MSA problem can be defined as follows:

Given a set of sequences S: $\{s_1, s_2, \ldots, s_k\}$ of lengths $|s_1|, |s_2|, \ldots, |s_k|$ defined over an alphabet Σ, for example $\Sigma_{DNA} = \{A, C, G, T\}$ or $\Sigma_{protein} = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$.

A multiple sequence alignment of S is defined as S': $\{s'_1, s'_2, \ldots, s'_k\}$, where the length of the all the k sequences is exactly the same. Note that, S' is defined over the same alphabet as S (Σ) with an additional gap symbol $(-)$; so, S' is defined over the alphabet $\Sigma \cup \{-\}$. The gap symbol refers to *indels*, that is to say, the insertion or deletion of bases in the unaligned sequences.

In this way, a multiple alignment is obtained by adding gaps to the sequences of S so that their lengths become the same. It can be seen as a matrix representation where the rows are sequences and


Unaligned Sequences

```
Seq.1  APKAKIVLVGSGMIGG (16)
Seq.2  ARKIGIIG (8)
Seq.3  KTPMRVAVTGAA (12)
Seq.4  MKVAVLGAAGGIGQALALLLKTQLP (25)
```

```
       1 . . . 5 . . . .10 . . . .15 . . . .20 . . . .25 . . .
Seq.1  A P K A K I V L V G S G - M I G G - - - - - - - - - - -
Seq.2  A R - - K I G I I G - - - - - - - - - - - - - - - - - -
Seq.3  K T P M R V A V T G A A - - - - - - - - - - - - - - - -
Seq.4  M - - - K V A V L G A A G G I G Q A L A L L L K T Q L P
```

Individual

#Sequences: 4

Alignment length: 28

Groups of GAPs:

2	13	18	-11	
2	3	-2	11	-18
1	13	-16		
1	2	-3		

1st group of GAPs in Seq.1
- Initial Position: 13
- #GAPs: 1

Note that, when the number of GAPs is equal to 1, we do not store the number of GAPs.

2nd group of GAPs in Seq.2
- Initial Position: 11
- #GAPs: 18

Note that,
- *The **first position** indicates the number of GAPs groups.*
- *A **positive value** is the initial position.*
- *A **negative value** is the number of GAPs in a group.*

Fig. 1. Representation of an individual.