A Quantum Evolutionary Algorithm for Effective Multiple Sequence Alignment

Souham Meshoul, Abdessalem Layeb, and Mohamed Batouche

Computer Science Department, PRAI Group, LIRE Laboratory,
University Mentouri of Constantine, 25000 Algeria
{meshoul, layeb, batouche}@wissal.dz

Abstract. This paper describes a novel approach to deal with multiple sequence alignment (MSA). MSA is an essential task in bioinformatics which is at the heart of denser and more complex tasks in biological sequence analysis. MSA problem still attracts researcher's attention despite the significant research effort spent to solve it. We propose in this paper a quantum evolutionary algorithm to improve solutions given by CLUSTALX package. The contribution consists in defining an appropriate representation scheme that allows applying successfully on MSA problem some quantum computing principles like qubit representation and superposition of states. This representation scheme is embedded within an evolutionary algorithm leading to an efficient hybrid framework which achieves better balance between exploration and exploitation capabilities of the search process. Experiments on a wide range of data sets have shown the effectiveness of the proposed framework and its ability to improve by many orders of magnitude the CLUSTALX's solutions.

1 Introduction

Bioinformatics is an interdisciplinary field which borrows techniques from many disciplines like biology, computer science and applied mathematics. It aims to model, to analyze, to compare and to simulate biological information including sequences, structures, functions and phylogeny [1]. Computer science takes a prominent part in bioinformatics as it is extensively used to address several information processing problems in many areas of computational molecular biology. One of such areas is sequence analysis where sequence alignment is the most common task. Sequence alignment is generally needed for sequence comparison purposes in an attempt to discover evolutionary, structural or functional relationships within a set of DNA, protein or biological macromolecule sequences. Practically, it is performed by studying sequences similarity. This is motivated by the fact that proteins or genes that have similar sequences are likely to perform the same function. Sequence alignment may concern two sequences or more. The latter case is known as Multiple Sequence Alignment (MSA). A comprehensive review of the related principle, uses and methods can be found in [2]. The areas where MSA is needed range from detecting homology between sequences and known protein/gene, to phylogenetic trees

C. Bento, A. Cardoso, and G. Dias (Eds.): EPIA 2005, LNAI 3808, pp. 260 – 271, 2005.

computation going through predicting secondary and tertiary structures of proteins, and identifying motifs preserved by evolution. MSA methods can be viewed as different combinations of choices for the following components: the set of sequences, the objective function and the optimization scheme. Indeed, aligning a set of sequences appropriately chosen is generally performed by optimizing a defined objective function. This latter is the mathematical tool used to measure the extent to which two or more sequences are similar. Hence, the definition of an adequate objective function is basically a biological problem and represents in its own an active research area.

The value of the objective function reflects the biological relevance of an alignment and indicates the structural or the evolutionary relation that exists among the aligned sequences. Sequence alignment allows mismatches and insertion/deletion which represent biological mutations. The weighted sum of pairs (WSP) [3] and COFFEE function [4] are notable examples of objective functions that are most commonly used for sequence alignment.

By another hand, optimizing an objective function is typically a computational problem. It consists in searching for the best alignment of sequences which maximizes a similarity measure or minimizes a cost function. Several optimization methods have been proposed in the literature to tackle this problem [2]. They can be loosely divided into the following classes depending on the search strategy used to perform the optimization task: progressive methods, exact methods and iterative methods.

In progressive methods, alignment is achieved by starting with the most related sequences and then gradually adding sequences one by one according to a pre-established order. Speed and simplicity are the main aspects of these methods. Most of them make use of dynamic programming. Clustalw is the example of the most widely used package which relies on this principle. The main drawback with the progressive methods can be explained by their greedy nature which can lead to sub-optimal solutions. To align all the sequences simultaneously, exact methods have been developed as a generalization of the Needlman and Wunsch algorithm [5]. The detection of the best alignment from the number of all potential alignments is at the heart of these methods. Their main shortcoming is their complexity which becomes even more critical with the increase of the number of sequences. To face the huge dimensionality of the combinatorial search space, iterative methods seem to be an interesting alternative. The key idea is to produce initial alignment and to refine them through a series of modification steps called iterations. The refinement can be performed in a deterministic way or stochastic one. During the last decade, several stochastic methods have been proposed to control the computational burden of the optimization process.

At first attempt to solve MSA, simulated annealing (SA) has been used [6] because it offers a sophisticated way to enhance the quality of a given alignment. It proposes to move from the current alignment to one of its neighbors, accepting with a certain probability to move also when the quality of the new solution is worst than previous ones. Other stochastic heuristic have been investigated like Genetic algorithm [7], Tabu Search [8] and Evolutionary algorithms [9]. Evolutionary algorithms adapt