Parallel genetic algorithm for performance-driven sequence alignment

L. A. Anbarasu
Scientific & Engineering Group
C - DAC
Pune 411 007

V. Sundararajan
Scientific & Engineering Group
C - DAC
Pune 411 007

P. Narayanasamy
School of Computer Science & Eng.
Anna University
Chennai 600 025

Abstract

The simultaneous alignment of three or more nucleotide or amino acid is among the most important tools for analyzing biological sequences. Multiple alignments are used to find characteristic motifs and conserved regions in protein families; to help demonstrarte homology between new sequences and existing families; to improve the prediction of secondary and tertiary structure of new sequences; and an essential pre-requisite to phylogenetic reconstruction. The fact that the multiple sequence alignment problem is of high complexity has led to the development of different algorithms. These algorithms fall into two categories namely the greedy ones that rely on pairwise alignment and those that attempt to align all the sequences simultaneously.

It is a standard practice to use the dynamic programming method to align a pair of sequences. To find an optimal alignment for a sequence of length m and n, the dynamic programming method requires $O(mn)$ time and $O(mn)$ space. The complexity of this method grows to $O(m^n)$ when applied to m sequences of length n. Therefore, all the methods capable of handling larger problems in practical time scales make use of progressive alignment. In progressive alignment, sequences are aligned in an order imposed by some estimated phylogenetic tree. It first aligns the most closely related sequences and then gradually adds the more distant ones. Some of the most widely used multiple sequence alignment packages are based on this algorithm. They have the advantage of being fast and simple as well as reasonably sensitive. Their main drawback is the 'local minimum' problem that stems from the greedy nature of the algorithm. This means that alignments formed early in the process are constructed without the knowledge of most of the available data, and therefore, may easily freeze in a mistake that cannot be corrected later. It is to avoid this pitfall that the second method has been designed.

The second type of method uses information from all the input sequences at the same time. Finding the best alignment from the number of possible alignments by this method is very hard and for the 'sum - of - pairs', this problem is shown to be NP – Complete. Other global alignment techniques using sum - of - pairs function involve the use of stochastic heuristics such as simulated annealing and genetic algorithms.

There are two main advantages of using these stochastic optimization methods. First of all, they do not have any strong restrictions on the number of sequences to align or the length of those sequences. Secondly, they are very flexible in optimizing any objective function. Genetic algorithm is one of the well known stochastic search methods that is capable of finding near optimal alignment from totally unaligned sequences. Implicit parallelism is an added advantage of genetic algorithms and could be exploited to get both speed up and better quality in convergence.

To develop automatic computational methods, multiple sequence alignment is formulated as a combinatorial optimization problem. Although numerous methods based on various principles have been proposed, computational biologists are still seeking better methods in terms of accuracy and computational speed.

This research work is aimed at developing a new technique for efficient multiple sequence alignment. The new method is based on parallel genetic algorithm that runs on a distributed network of workstations. The algorithm searches for an alignment among the isolated populations evolving independently by optimizing weighted sum-of-pairs objective function, which measures the alignment quality. Using adaptive operator fitness technique each isolated population evolves independently with different behavior and explores the solution space effectively. The parallel approach is implemented on the PARAM 10000, a parallel computer with a cluster of workstations and is shown to consistently perform better than the sequential genetic algorithm. The algorithm yields alignments that are qualitatively better than an alternative method, Clustal W. An extensive investigation of the algorithm's parameters further confirms the better performance.

Computing with Membranes: P Systems with DNA-worms

Alfonso R. Patón
Fac. de Informática
Campus de Montegancedo
28660 Madrid, Spain
arpaton@fi.upm.es
+34 91 3367436

Abstract

We introduce a variant of P systems with worms inspired in the DNA computing area. P systems with worms work with multisets of strings-objects processed by splitting, mutation, replication and recombination. The new model is simpler (we eliminate the replication operation) and more realistic (the recombination operation is changed by the simpler one of suffix-prefix or head-tail concatenation developed in the DNA computing framework). In this work we prove that these new P systems can generate all recursively enumerable sets of numbers.

1 INTRODUCTION AND RESULTS

P systems are a class of distributed parallel computing models inspired from the way the alive cells process chemical compounds, energy, and information. In short, in the *regions* delimited by a *membrane structure* are placed *multisets* of *objects*, which evolve according to *evolution rules* associated with the regions; the objects can also be *communicated* from a region to another one, according to certain *target indications*.

It is possible to consider objects described by symbols or by strings. We combine here the two classes of P systems: we work with multisets of string-objects and we consider the result of a computation as the number of strings in a given output membrane. In [CPP00] it is proved that each r.e. set of natural numbers can be compute by a P system with worms.

The operations of our new model are: suffix-prefix concatenation, splitting and mutation. Let V be an alphabet and $M \subseteq V^* \times V^*$ a finite set of pairs of strings. For $w_1, w_2 \in V^*$ we define:

1. *Suffix-prefix concatenation.* Consider a finite set $M \subseteq V^* \times V^*$ of pairs suffix-prefix and two strings $w_1, w_2 \in V^+$. We write $(w_1, w_2) \vdash_M w_1 w_2$ if $w_1 = x_1 x_2 x_3$, $w_2 = y_1 y_2 y_3$ and $(x_3, y_1) \in M$, for some $x_1, x_2, x_3, y_1, y_2, y_3 \in V^*$. (*head-tail* or *suffix-prefix* conditions are imposed on the concatenated strings).

2. *Splitting.* If $a \in V$ and $u_1, u_2 \in V^+$, then $r : a \to u_1|u_2$ is called a *splitting rule*. For strings $w_1, w_2, w_3 \in V^+$ we write $w_1 \Longrightarrow_r (w_2, w_3)$ (and we say that w_1 is splitted with respect to rule r) if $w_1 = x_1 a x_2$, $w_2 = x_1 u_1$, $w_3 = u_2 x_2$, for some $x_1, x_2 \in V^*$.

3. *Mutation.* A *mutation rule* is a context-free rewriting rule, $a \to u$, over V. For strings $w_1, w_2 \in V^+$ we write $w_1 \Longrightarrow_r w_2$ if $w_1 = x_1 a x_2$, $w_2 = x_1 u x_2$, for some $x_1, x_2 \in V^*$.

In the proof of the computational completeness of our new P system with DNA-worm we use the following results. For a family FL of languages, we denote by lFL the family of length sets of languages in FL, that is, $lFL = \{length(L) \mid L \in FL\}$, for $length(L) = \{|x| \mid x \in L\}$ and let MAT_{ac} the family of languages generated by matrix grammars with appearance checking. We make use of the equality $lRE = lMAT_{ac}$. Our proof does not provide a bound on the number of membranes; this remains as an *open problem*.

Acknowledgments. Research supported by the TUCS (Turku Centre for Computer Science) and by the Politechnical University of Madrid.

References

[CPP00] Juan Castellanos, Gh. Păun, and Alfonso R. Patón. Computing with membranes: P systems with worm-objects. In *IEEE 7th. Intern. Conf. on String Processing and Information Retrieval, SPIRE'2000*, pages 64–74, La Coruña, September 2000.