World Academy of Science, Engineering and Technology
International Journal of Biomedical and Biological Engineering
Vol:3, No:10, 2009

Multi-Agent Systems Applied in the Modeling and Simulation of Biological Problems: A Case Study in Protein Folding

Pedro Pablo González Pérez, Hiram I. Beltrán, Arturo Rojo-Domínguez and Máximo Eduardo Sánchez Gutiérrez

***Abstract*—**Multi-agent system approach has proven to be an effective and appropriate abstraction level to construct whole models of a diversity of biological problems, integrating aspects which can be found both in "micro" and "macro" approaches when modeling this type of phenomena. Taking into account these considerations, this paper presents the important computational characteristics to be gathered into a novel bioinformatics framework built upon a multi-agent architecture. The version of the tool presented herein allows studying and exploring complex problems belonging principally to structural biology, such as protein folding. The bioinformatics framework is used as a virtual laboratory to explore a minimalist model of protein folding as a test case. In order to show the laboratory concept of the platform as well as its flexibility and adaptability, we studied the folding of two particular sequences, one of 45-mer and another of 64-mer, both described by an HP model (only hydrophobic and polar residues) and coarse grained 2D-square lattice. According to the discussion section of this piece of work, these two sequences were chosen as breaking points towards the platform, in order to determine the tools to be created or improved in such a way to overcome the needs of a particular computation and analysis of a given tough sequence. The backwards philosophy herein is that the continuous studying of sequences provides itself important points to be added into the platform, to any time improve its efficiency, as is demonstrated herein.

***Keywords*—**multi-agent systems, blackboard-based agent architecture, bioinformatics framework, virtual laboratory, protein folding.

Dr. Hiram Isaac Beltrán* (corresponding author), Departamento de Ciencias Naturales, DCNI, Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Pedro Antonio de los Santos 84. Col. San Miguel Chapultepec. Miguel Hidalgo 11850. México, D.F. e-mail: hbeltran@correo.cua.uam.mx. Tel. +52 (55) 26363853.

Dr. Arturo Rojo-Domínguez, Departamento de Ciencias Naturales, DCNI, Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Pedro Antonio de los Santos 84. Col. San Miguel Chapultepec. Miguel Hidalgo 11850. México, D.F. e-mail: arojo@correo.cua.uam.mx.

Máximo Eduardo Sánchez, Departamento de Matemáticas Aplicadas y Sistemas, DCNI, Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Pedro Antonio de los Santos 84. Col. San Miguel Chapultepec. Miguel Hidalgo 11850. México, D.F.

Dr. Pedro Pablo González* (corresponding author), Departamento de Matemáticas Aplicadas y Sistemas, DCNI, Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Pedro Antonio de los Santos 84. Col. San Miguel Chapultepec. Miguel Hidalgo 11850. México, D.F. Tel. +52 (55) 26363805. e-mail: pgonzalez@correo.cua.uam.mx.

I. Introduction

THE notion of agent has been described in several ways in literature [1-7] with different acceptance according to the research field where it has been considered – from distributed artificial intelligence (DAI) to software engineering and concurrent/distributed systems, from social, psychological and economic sciences to computer supported cooperative work (CSCW), among others.

In its most general conception, we can say that an agent is a *description* of an entity that *interacts* on its environment [8]. Hereinafter, a multi-agent system (MAS) is an ensemble of interacting agents. Interaction among agents can be realized by direct communication, according to some agent communication language, or indirectly, by exploiting some environmental resources acting as communication/ coordination intermediate abstractions. These abstractions range from a simple communication channel to sharing information spaces, such as blackboards, that are useful for agents to synchronize their tasks. In particular, the latter, are examples of a coordination artifact, i.e. mediating abstractions and providing specific coordination functionalities [9]. So, a MAS can also be characterized as a task-oriented entity, where collective goals are achieved by means of the interaction of the individual agents that are a part of the system. Coordination is the fundamental systemic dimension, ensuring that the interaction among the individual entities is fruitful, so as to effectively achieve the collective objectives.

Due to its constitution a MAS can be exploited both as i) an engineering paradigm, for designing and programming complex software systems; as well as ii) an analytical tool, for modeling and simulating existing complex systems in order to study their systemic behavior. In particular, multi-agent-based models are often used for the simulation of systemic and social phenomena [10, 11]. Recently, however, their effectiveness has been remarked also beyond social simulation, in domains where traditional techniques are typically adopted [12], such as parallel and distributed discrete event system simulation, object oriented simulation, and dynamic micro simulation.

Generally speaking, simulations and models, based on the MAS paradigm, integrate aspects that can be found both in "micro" and "macro" approaches. When mimicking micro approaches, a MAS models the behavior of specific

individuals or components; this is in contrast to mimicking macro approaches that are typically based on mathematical models, where the characteristics of a population are averaged together and the model attempts to simulate changes in the averaged characteristics of the whole population and not of the individual components. Thus, in macro approaches the set of individuals is viewed as a structure that can be characterized by a number of variables, whereas in micro approaches the structure emerges directly from the interactions between the individuals. Last decade Parunak et al. [13] compared these approaches and pointed-out that " ... *agent-based modeling is most appropriate for domains characterized by a high degree of localization and distribution and dominated by discrete decisions. Equation-based modeling is most naturally applied in systems that can be modeled centrally, and in which the dynamics are dominated by physic laws rather that information processing* ...".

Taking into account the previous statements, in this work we promote a conceptual framework for modeling biological systems heavily based on heterogeneous behavior, complex nature, localization, distribution and interaction of the components of the system itself, so the agent-based approach seems to adequately fit to these requirements.

II. Multi-agent systems for modeling biological systems

MAS have proven to be an effective and appropriate abstraction level to construct whole models of a diversity of biological systems [14-17]. If an agent represents an individual active component of the system, the overall MAS including the communication/interaction abstractions captures the overall set of the biological components including also the structures involved in their interaction (i.e. cellular compartments, intracellular messengers, etc.) leading to a thorough mimic of the system under study. Communication/interaction components, such as blackboard [18], can be adopted to model the various interacting patterns found in biological processes, such as mathematical, physical, chemical and biological information exchanges, and very importantly the mixtures that can be drawn from them.

The notion of agent group or society can be adopted for scaling with complexity, providing a way to decompose recursively a MAS according to the coordination task(s) happening inside. A coarse grained model of society can be defined as a set of agents plus the resources - including the communication/interaction components - involved in the same social task characterizing the society. The adoption of the society notion makes it possible to identify the level of granularity of this coarse point of view, since different levels of description of the same system have different resolutions, being the lower the hierarchy, the bigger the resolution. At the first level of resolution there is an individual agent, at a more hierarchical level of resolution there is a society of agents, at the third level there should be a society of societies of agents, etc. and the vice-versa gives the zoom of the entire system. Such a recursive decomposition is very effective when modeling biological systems, which naturally involve different levels of description - from organs to cells, down to chemical or physical meaning aggregations -, each one characterized by different kinds of interaction/coordination processes and emerging phenomena.

Finally, MAS paradigm can be effective for devising a methodology for covering the whole simulation engineering spectrum, from design to development, execution and runtime (dynamic) control. Critical points of biological systems - concerning structures, activities, interactions - can be captured directly by abstractions that *are kept alive from design to runtime*, supported by suitable infrastructures. The simulation then can be framed as an *online experiment*, where the scientist or the user can observe and dynamically interact with the system and its environment, both by changing its structure for instance - by introducing new agents representing biological components or removing existing ones -, and by viewing the whole system – through global biological processes - by acting on the communication/interaction components.

III. The Protein Folding Problem

The folding of a given protein is the natural process by which these macromolecules pass-through to define its characteristic and functional tridimensional structure. It has been stipulated that each protein is found, at first principles, as a polypeptidic chain, traduced from an mRNA sequence as a linear chain of amino acids. Based on this premise, each polypeptide lacks of characteristic or functional three-dimensional structure. Nonetheless each amino acid queued in the polypeptidic chain contains particular chemical characteristics, due to its position in the linear sequence and inherent to its own physicochemical nature, being e.g. hydrophobic, hydrophilic, or electrically charged. The supramolecular self-assembly of a single amino acid, related to its neighbors and the exposed media and repeated self-consistency to each unit, should produce a well defined three-dimensional shape. This latter is the folded state of the protein, better known as the native state. The folding mechanism is not clearly understood yet and it represents a fundamental problem in Mother Nature.

The Levinthal's paradox (proposed by Cyrus Levinthal in 1969) [19], stipulates that if a protein was sequentially folded, sampling the space of all degrees of freedom or possible conformations, it will take an astonishing amount of time to reach the right three-dimensional structure even if all the checked conformations were sampled and analyzed in an extremely fast ratio (nano- or pico-seconds time scale). The main clue in this problem is that the real folding time of proteins occurs much faster than it can be conceived. Therefore, Levinthal proposed that this stochastic conformational analysis is not the taking place mechanism of the real folding. Besides, it is a completely driven process directed from the physicochemical conditions of the carried biosynthesis and the biochemical pathways that suffer these molecules a posteriori.

The inherent variability and very high complexity of protein folding makes it a benchmark test case per excellence for high throughput computational tools. Even though remarkable

effort has been done in this line, the needed for understanding the protein folding problem and not to say the functionality of those biomolecules is not yet reached [4]. Many different models, approaches and criteria have been proposed until today regarding protein folding [20-22], ranging from the coarse–grained minimalistic models to the more detailed atomistic models.

The protein folding problem has often been tackled using the coarse-grained minimalistic model, e.g. lattice bead models [23-25]. This is due to the increasingly realistic nature of the current coarse-grained models, which can be used to describe, in a quantitative manner, the detailed folding mechanism of specific proteins [26].

The following section describes an original bioinformatics framework for studying and exploring protein folding problem, obtained by integrating lattice bead models [27, 28], evolutionary algorithms and other computational techniques into a MAS architecture.

IV. THE BIOINFORMATICS FRAMEWORK

A. The blackboard-based agent architecture

The generic bioinformatics framework architecture can be seen in Fig. 1, where its main components are the *agents* and the *blackboard*. As can be seen in this figure, the architecture provides three types of agents: *model agents*, *algorithm agents* and *interface agents*. According to this MAS architecture, the interaction among agents is realized by indirect communication, through a communication/coordination abstraction, represented in this figure by the *blackboard*. Fig. 2 shows the semantics assigned to the different blackboard levels and types of agents when protein folding is modeled. Finally, the architectural pattern of the bioinformatics framework is represented in Fig. 3. As can be seen in Fig. 3, the *Blackboard* contains all structural representations for a linear protein sequence provided as input (e.g. *LinearSequence*, *2DSequence* and *3DSequence*), and it is the structure through which the interaction among agents takes place.

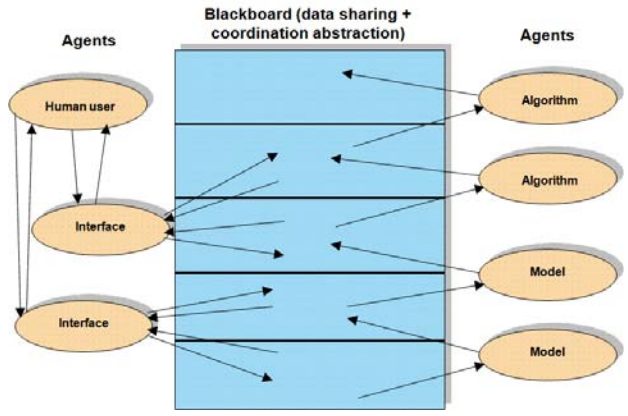


Fig. 1 General overview of Evolution architecture. Interaction among agents is realized indirectly through a blackboard, which represents the communication/coordination abstraction

B. The Blackboard

In Evolution architecture, the blackboard means both data sharing and interaction/coordination artifacts. On the blackboard levels are recorded the solution elements needed for the resolution of the current problem. Such solution elements correspond to i) initial conditions, ii) partial results and iii) final results, all involved in the protein folding prediction process. Through this mapping, the blackboard structure allows for modeling of the protein folding process through its different abstraction levels. As we can see in Fig. 2, the blackboard has been broken down into following five abstraction levels: 1) amino acid sequence, 2) HP sequence, 3) conformational space, 4) algorithm workspace, and 5) plausible conformations.

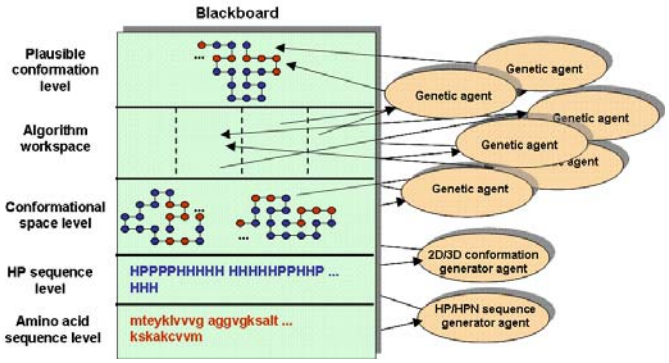


Fig. 2 Semantics assigned to the different blackboard levels and types of agents in Evolution when modeling protein folding

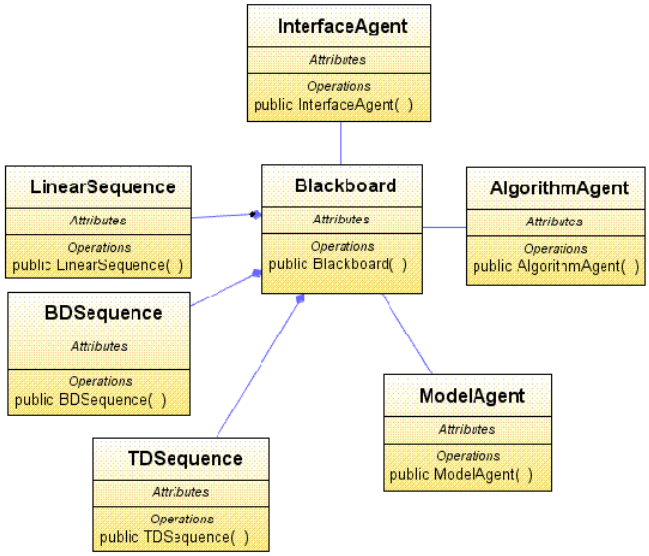


Fig. 3 The architectural pattern of the bioinformatics framework. The major entities of Evolution architecture are represented as classes in this diagram

The blackboard, as a communication/coordination abstraction, also provides a continuous trace of all interactions occurring among the agents. This trace can be seen as a topological map distributed among the blackboard levels, and its solution elements reflect the different protein conformational states that characterize the protein folding pathway in a given time.

The characteristics of the solution elements recorded on the blackboard levels are described below.

- **Amino acid sequence level**. The solution elements recorded on this blackboard level correspond to amino acid sequences introduced directly by the user or taken from a sequence file. An amino acid sequence is represented as a chain of "n" characters belonging to alphabet {A, … , Y} \ {B, J, O, U} (e.g. *ALCNCNRIIIPHMCWKKCGKK* (http://www.pdb.org/pdb/explore/quickPDB.do?structureId=1TER), *DAEFRHDSGYEVHHQK* (http://www.pdb.org/pdb/explore/quickPDB.do?structureId=2BP4).

- **HP (HPN) sequence level**. On this blackboard level are recorded HP (HPN) sequences, which can be either generated by HP (HPN) *model generator agent* (see Fig. 2) or introduced directly by the user. An HP sequence is represented as a chain of "n" characters belonging to a same length two-element alphabet {H, P}, e.g. HHPPPPPHHHHHPHPHPPPHPP (HP traduced from ALCNCNRIIIPHMCWKKCGKK), where "n" is the same for both, original and HP traduced sequences. If the alphabet would increase to HPN, hence the coding should be given by, also same length, HPN traduced sequence, etc.

- **Initial Conformational space level**. This blackboard level corresponds to the initial conformational space created when the HP model, being discrete or continuous, is executed on 2D triangular, 2D square, 3D diamond or 3D cubic lattices (this action is carried out by the *2D/3D conformation generator agent*, in Fig. 2). The solution elements on this level are 2D/3D conformations characterized by three main parameters: energy of the conformation, radius of gyration and maximal diameter. Only the energy is optimized at this level of abstraction.

- **Algorithm workspace level**. On this blackboard level are recorded all the states visited by two processes, the heuristic search and the optimization, for searching the optimal and suboptimal conformations, providing the full landscape of the conformational space. Therefore, each iteration corresponds to a conformational space, and the whole level represents the evolutionary story of the protein folding mechanism. These iterations are created by the coordinated actions of the *genetic agents* (see Fig. 2).

- **Plausible conformation level**. The solution elements on this blackboard level correspond to the optimal and suboptimal conformations found, again, by the two processes, the heuristic search and the optimization. Thus, the conformations created on this level represent the best solutions found by the *genetic agents* during their bottom-up and top-down processes through the last three blackboard levels.

C. The agents

As can be seen in Fig. 2, the architecture of bioinformatics framework for the protein folding defines three types of agents: 1) *HP (HPN) sequence generator agents*, 2) *2D/3D conformation generator agents* and 3) *Genetic agents*.

- **HP (HPN) sequence generator agents**. This type of agent implements the HP model [29, 30] and the extended HPN model. The main task of the agent is the translation of an amino acid sequence, starting from an alphabet of 20 amino acids, to a reduced alphabet, composed by 2 (HP) or 3 (HPN) symbols. According to HP model, the 20 amino acids are classified as either Hydrophobic (H) or Polar (P), whereas when using the extended HPN model a given amino acid can also be classified as Neutral (N). As a result of the agent task, a protein is then modeled as a chain made up of beads, where each bead represents one of the symbols H, P or N, depending on the used model.

- **2D/3D conformation generator agents**. An agent of this type has the ability to generate a 2D/3D starting conformational space from a target linear sequence implementing a wide range of lattice bead models [27, 28]. The agent operates on a linear sequence of beads (H, P or N), implementing the HP (HPN) model on 2D, triangular and square lattices, or 3D, diamond and cubic lattices. When the conformational space has been created, the agent calculates the energy of each conformation thus generated. The energy of a conformation is given by the interaction between beads that are topological neighbors in the 2D/3D structure, named through space interactions, but not neighbors in the sequence of bead that make up the chain, through chain interactions, this can be simply viewed in Fig. 4 b. Besides, in Fig. 5 a schematic representation of a *2D/3D conformation generator agent* is shown. Finally, in Fig. 6 can be seen the agent designed as a class for further implementation.

- **Genetic agents**. The coordinated work of these agents, through the *blackboard*, can be issued itself as a simple genetic algorithm. Among all these agents is carried out the heuristic search and optimization processes on the conformational space for searching the optimal and suboptimal conformations. The *genetic agents* encapsulate a wide variety of genetic operators and techniques needed to find good-enough solutions to protein folding. Some of them are selection, reproduction, crossover, mutation and elitism.

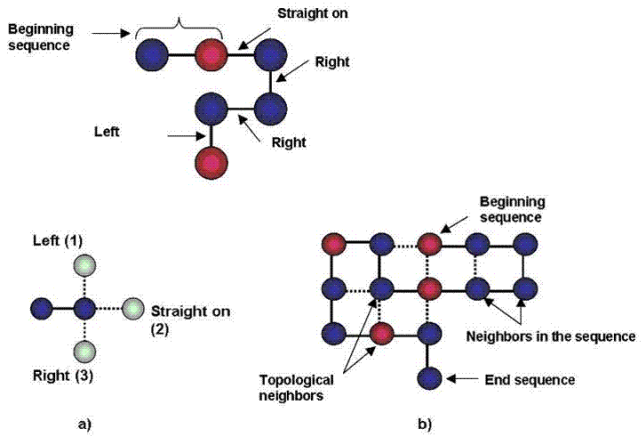


Fig. 4 A 2D square lattice. The H and P beads are constrained to lie upon a square lattice. a) The local coordinate system. b) Topological neighbors and neighbors in the sequence

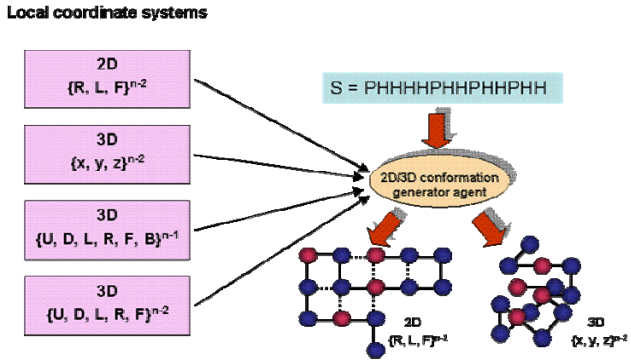


Fig. 5 A schematic representation of a 2D/3D conformation generator agent

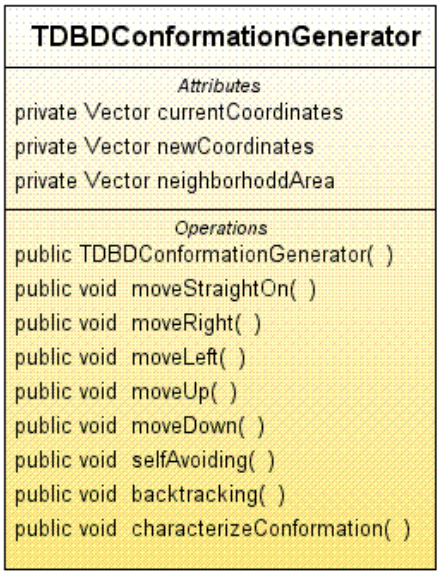


Fig. 6. The 2D/3D conformation generator agent designed as a class

D. Evolution behavior

Fig. 7 shows the diagram of activities involving the main functionalities provided by Evolution to the user. As depicted in this figure, the user can execute a wide range of required tasks when an experiment is carried on, which include the following tasks:

1. Generation of the 2D/3D initial conformational space from an amino acid sequence or a bead HP sequence, provided as input.
2. Visualization of the 2D/3D initial conformational space and the interaction with a particular conformation. This interaction allows the user to discover more details of the conformation through the zoom, rotation and translation of it.
3. Prediction of optimal and suboptimal conformations according to a predefined criterion of optimization, e.g. the lower energy conformations possible.
4. Exploration of the different iterations involved in the entire protein folding process.
5. Change the characteristics of a particular visited iteration by this artificial protein folding, recording it and used as seed to reset the whole optimization process from this state.

The *2D/3D conformation generator agents* carry on task 1. Task 3 is reached during the interaction and coordination of the *genetic agents* through the *blackboard*. Finally, tasks 2, 4 and 5 are supported by the *interface agents* defined as abstract entities in the Evolution generic architecture (see Fig. 1) and later implemented for their specific purpose when modeling protein folding.

E. The graphic user interfaces

One of the most remarkable characteristics of Evolution is the vast number of graphical user interfaces (GUI) and graphical tools provided for the user-system interaction. Evolution offers to the user a wide variety of 2D/3D charts (e.g. diagrams, graphics and tables) to interpret the partial and final results obtained during experiment execution. Among the available 2D/3D charts are the following: 2D/3D protein conformational spaces, evolution charts (i.e. worst, best and averaged fitness evolution), generation charts (i.e. conformation fitness, parent/offspring fitness), tracking of radius of gyration and maximal distance through the generations, statistical measures related to protein conformation evolution, etc. But the bioinformatics framework is not limited to these, since new graphic outputs of indexes can be effortless added, according to the problem to be explored. In Evolution, all GUI and graphical tools have been implemented as *interface agents*. Thus, during the experiment execution, a GUI is an agent that provides personalized assistance to users with their tasks.

First of all, in Fig. 8, 9, 10 and 11 are described some of the major Evolution GUI snapshots. Fig. 8 shows the main GUI through which the experiment can be started, recovered, continued and saved. Later on, in Fig. 9 the type of 2D/3D lattice is selected and the number of conformations to generate is specified. These actions take place after the HP bead

sequence (or amino acid sequence) has been provided as input. In Fig. 10 can be appreciated two optimal conformations reached for a 27-mer hydrophobic chain provided as input. All shown conformations were produced by the coordinated work of genetic agents during the heuristic search and optimization processes. Finally, in Fig. 11 is depicted a generation evolution chart which shows the worst, best and averaged fitness values through the different iterations produced during the heuristic search and optimization processes.

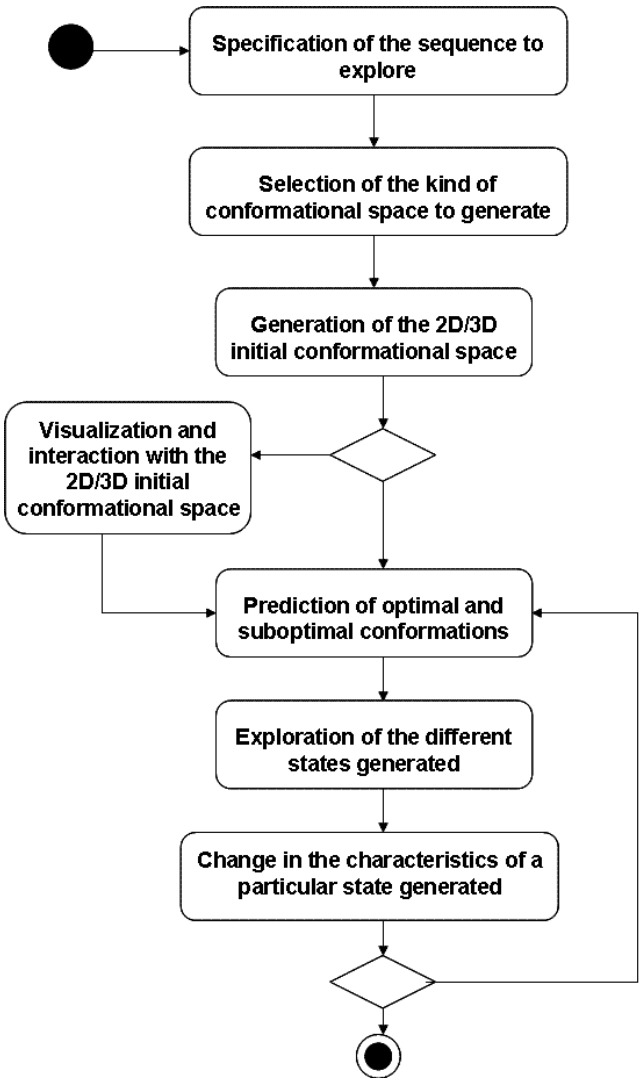


Fig. 7 Bioinformatics framework workflow. The activity diagram describes the sequence of the major activities carried on during a run

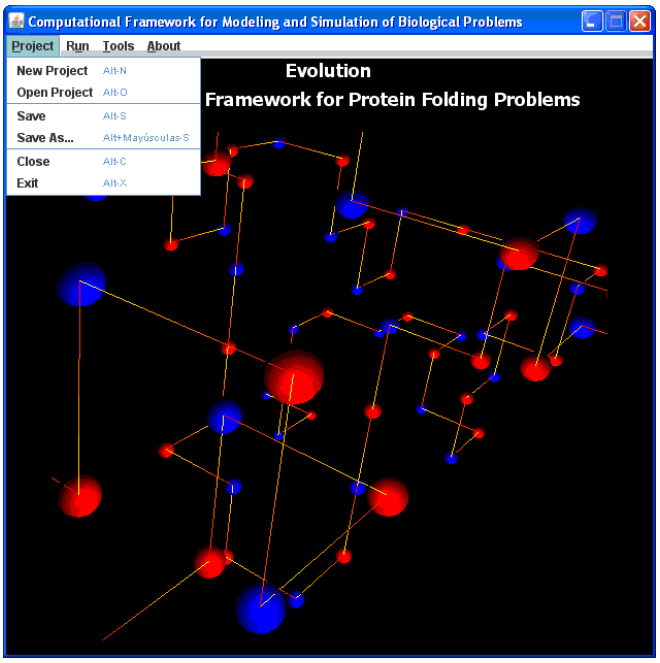


Fig. 8 The main Evolution GUI

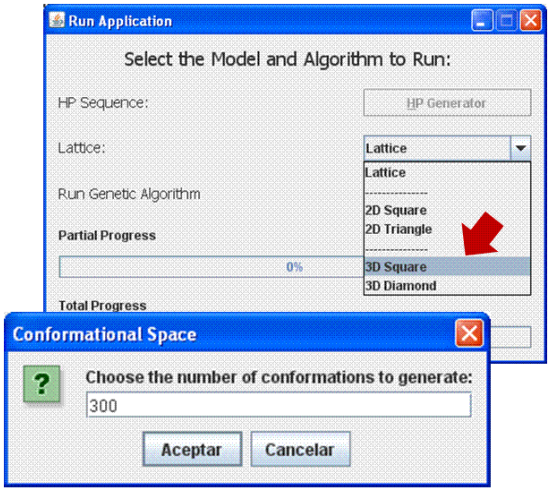


Fig. 9 Through the *Run Application* interface the 2D/3D lattice type is selected and the number of conformations to generate is specified. In the example, a 3D Square lattice has been chosen and a conformational space composed by 300 conformations will be created

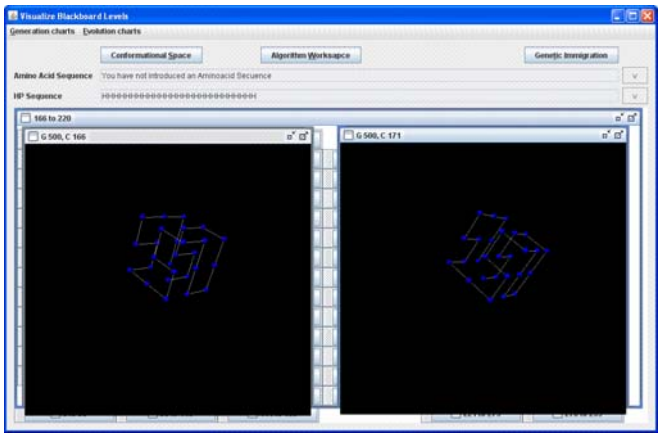


Fig. 10 Two optimal conformations reached for a 27-emer hydrophobic chain provided as input

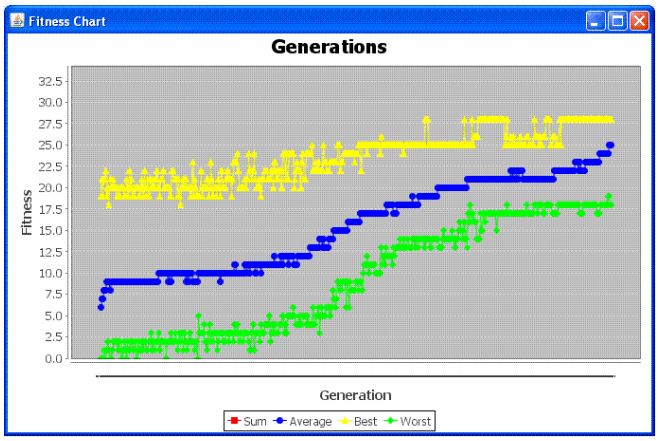


Fig. 11 Generation evolution chart, which shows the worst, the best and averaged fitness values through the different states (generations) produced during the heuristic search and optimization processes

V. RESULTS AND DISCUSSION

A. Though sequences to be folded in a 2D square lattice

Case of study A. A non-symmetric and turn motifs containing sequence: D-1, 45 fragments, max score -17, HP sequence: $P_2HPH_2PH_3PH_2P_2H_4PHPH_2P_5H_4P_7H_2P_3$.

This current sequence was taken from the contribution of Shmygelska et al. [24]. It has been reported a maximum score of -17 for this sequence, and its folded structure has been already known [24]. Herein we have observed that Evolution folds this sequence to reach a maximum score of -16, see Fig. 12. It succeeds in folding since this suboptimal solution has been obtained in a few seconds. Anyway some further improvements should be carried on in order to obtain a global minimum, since we were not able to obtain a close packed structure due to the particular placement of the P residues possibly belonging to turn motifs.

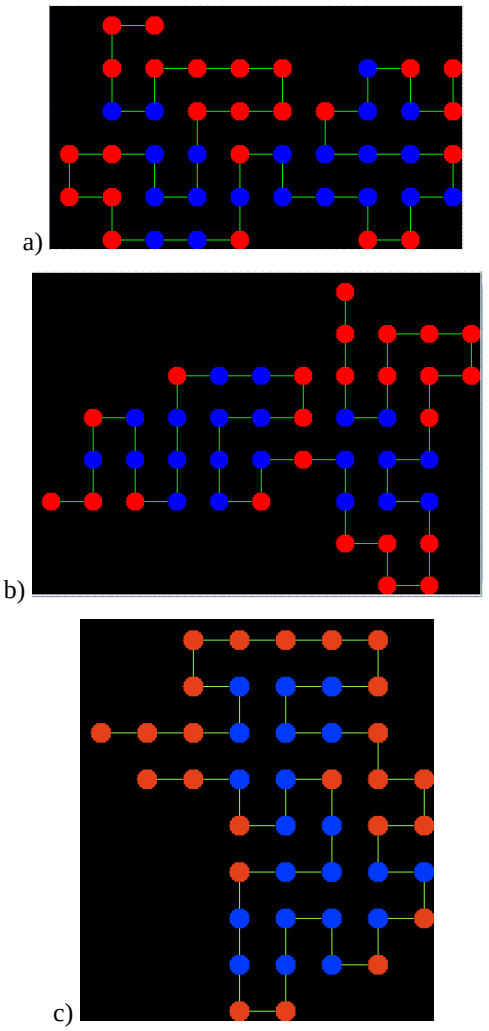


Fig. 12 Sequence D-1 folded into the 2D square lattice. These structures represent suboptimal solutions with maximum scores of a) -14, b) -15, and c) -16

Case of study B. A very symmetric and hydrophobic sequence with close starting-ending sides: SI-8, 64 fragments, max score -42, HP sequence: $H_{12}(PH)_2(P_2H_2)_2P_2HP_2H_2PPH_2P_2HP_2(H_2P_2)_2(HP)_2H_{12}$.

This current sequence was also taken from the contribution of Shmygelska et al. [24]. In this case it has been reported a maximum score of -42 as well as its folded 2D representation [24]. The Evolution mainframe achieved a maximum score of -37, anyway, other suboptimal folded structures were found, e.g. with maximum scores of -35 and -36, see Fig. 13. As seen in the work of Shmygelska, the optimal solution provides a completely packed and symmetric structure. This optimal solution also shows that the beginning of the chain is very close situated to the end of it. Since we were not able to obtain the optimal solution, our suboptimal structures quite resemble the behavior of placing starting-ending fragments also close in the net, but this is inherent to the solution of this particular sequence. Interestingly, Fig. 13 c is quite distant from this principle, but this is due to the local minima findings. These suboptimal solutions are indeed far away from the optimum, a distance of five energetic steps, hence the inclusion of new

variables into the fitness function, in order to find those deeper minima, are justified. The potential energy surface is visited in all these suboptimal states, and they may become kinetic pathways directed to the global minimum, therefore these states may also become trackers of protein engineering and design, but all this is matter under study.

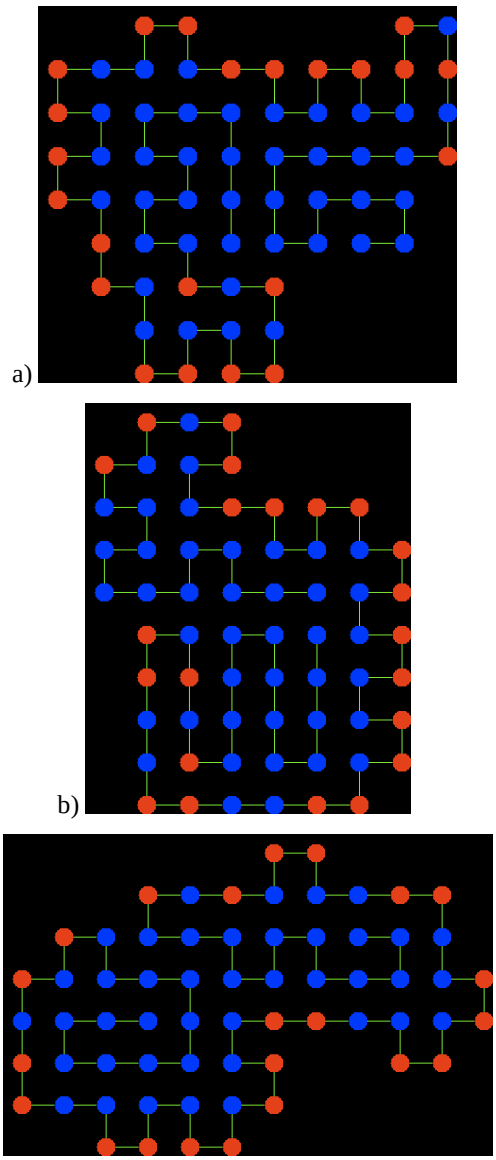


Fig. 13 Sequence SI-8 folded into the 2D square lattice. These structures represent suboptimal solutions with maximum scores of a) -35, b) -36, and c) -37

B. New molecular indexes: Radius of Gyration and Maximal Diameter

As can be seen in Fig. 12 and 13, with the current fitting function, suboptimal scores can be obtained with quite close packed (globular) sequences and with low density (relatively extended) ones, even for short sequences as the conformers shown in these figures, but lacking of the global minimum and also lacking the most compact structure. These suggest the inclusion of packing indexes in the fitting function as shown in (1). The molecular indexes included in this equation are Radius of Gyration (R_g) and Maximal Diameter (D_{max}), whose expressions can be seen in (2) and (3), respectively. The three terms in (1) (f(e), g(R_g) and h(D_{max})) are not completely independent as shown in Fig. 14. From this figure it can be appreciated that g(R_g) and h(D_{max}) decrease when f(e) augments. As has been stated in previous works [23-25], these two variables are indeed important structural variables to be not only tracked but also included into the fitness function; this last should be performed in a weighed fashion. This warning is to overcome that they may become the ruling out variables in the whole system. Due to its inherent difficulty of inclusion into the fitness function work in this line is under progress. Hence, this inclusion of new optimization variables into the fitness function merits its own contribution due to this weighed scheme of participation unknown until now.

$$F = f(e) + C_R g(R_g) + C_D h(D_{\max})$$

$$f(e) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} e_{ij} \Delta_{ij}$$

$$e_{HH} = -1.0 e_{HP} = 0.0 e_{PP} = 0.0 \tag{1}$$

$$\Delta_{ij} = \begin{cases} 1 if\ i\ and\ j\ are topological, \\ but\ not\ sequence\ neighbours \\ 0 otherwise \end{cases}$$

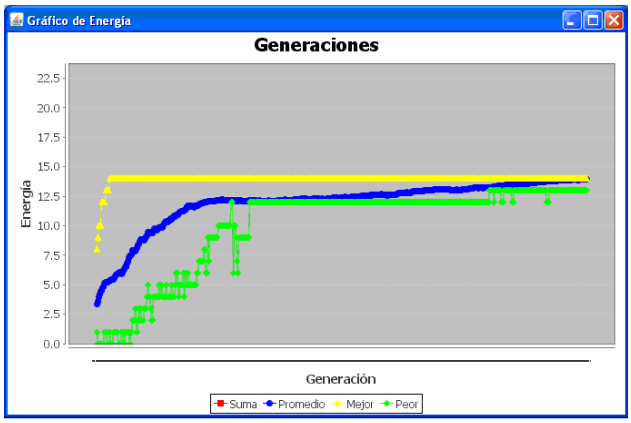
$$R_g : Radius\ of\ Gyration$$

$$D_{\max} : Maximal\ Diameter$$

$$R_g = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (r_k - r_{mean})^2} \tag{2}$$

$$\forall r_i, r_j \quad i, j = 1,2,\cdots,N, \quad where$$
$$r_i = (x_i, y_i, z_i), \quad r_j = (x_j, y_j, z_j)$$
$$calculate \quad d_{ij} = d(r_i, r_j) \tag{3}$$
$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$
$$D_{Max} = \max(d_{ij})$$

a)



b)

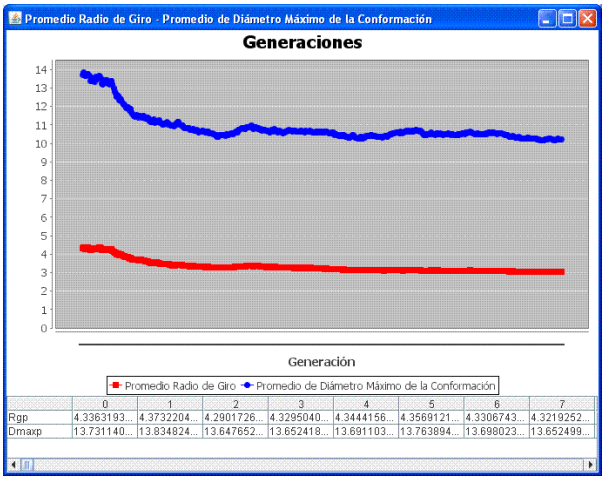


Fig. 14 Inverse proportionality in Equation 1, among a) augment of f(e), and b) decrease of R_g and D_{max}

VI. Conclusion

In this contribution, a novel bioinformatics framework, built upon multi-agent system architecture, was presented. Evolution, the bioinformatics framework, was designed as a virtual laboratory to be used for studying and exploring complex problems such as those from structural biology. As a case study, the version of Evolution presented here was dedicated to the study of the protein folding, initially using the well-known lattice bead models. The overall computational constitution of the framework is herein described in a complete point of view, showing the particular tools needed to gather a robust core, which indeed is feed through experimentation. The laboratory concept of the platform was exploited when the framework was exposed to fold two though sequences, one of 45-mer (non-symmetric and with turn motifs) and another of 64-mer (symmetric and fully compact), described in an HP 2D-square lattice level of description. These two experiments have shown the needs to include radius of gyration and maximal diameter as weighed optimization variables into the fitness function. As is thoroughly stated in the abstract, the backwards philosophy herein is that the continuous studying of sequences provides itself important points to be added into the platform, to any time improve its efficiency, as is demonstrated herein.

Acknowledgment

We acknowledge for support from Acuerdos del Rector General UAM. A.R-D. thanks for financial support from CONACyT México.

References

[1] Jennings, N.R., *CONTROLLING COOPERATIVE PROBLEM-SOLVING IN INDUSTRIAL MULTIAGENT SYSTEMS USING JOINT INTENTIONS.* Artificial Intelligence, 1995. **75**(2): p. 195-240.

[2] Jennings, N.R., *On agent-based software engineering.* Artificial Intelligence, 2000. **117**(2): p. 277-296.

[3] Jennings, N.R., et al., *Agent-based business process management.* International Journal of Cooperative Information Systems, 1996. **5**(2-3): p. 105-130.

[4] Jennings, N.R., et al., *TRANSFORMING STANDALONE EXPERT-SYSTEMS INTO A COMMUNITY OF COOPERATING AGENTS.* Engineering Applications of Artificial Intelligence, 1993. **6**(4): p. 317-331.

[5] Wooldridge, M., *The Gaia methodology for agent-oriented analysis and design.* Autonomous Agents and Multi-Agent Systems, 2000. **3**(3): p. 285-312.

[6] Wooldridge, M.J., *Software engineering with agents: Pitfalls and pratfalls.* IEEE Internet Computing, 1999. **3**(3): p. 20-+.

[7] Mach, R. and F. Schweitzer, *Multi-agent model of biological swarming.* Advances in Artificial Life, Proceedings, 2003. **2801**: p. 810-820.

[8] Gershenson, C., *Design and Control of Self-organizing Systems*. 2007, Vrije Universiteit Brussel: Brussel.

[9] Omicini, A., et al., *Coordination artifacts: Environment-based coordination for intelligent agents.* Proceedings of 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2004: p. 286-293.

[10] Conte, R., et al., *Sociology and Social Theory in Agent Based Social Simulation: A Symposium.* Comput. Math. Organ. Theory, 2001. **7**(3): p. 183-205.

[11] Nigel, G. and C. Rosaria, *Artificial Societies: The Computer Simulation of Social Life*. 1995: Taylor \& Francis, Inc.

[12] Davidsson, P., *Agent based social simulation: A computer science view.* Jasss-the Journal of Artificial Societies and Social Simulation, 2002. **5**(1).

[13] Van Dyke Parunak, H., R. Savit, and R.L. Riolo, *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide*, in *Multi-Agent Systems and Agent-Based Simulation*. 1998. p. 10-25.

[14] Gonzalez, P.P., et al., *Cellulat: an agent-based intracellular signalling model.* Biosystems, 2003. **68**(2-3): p. 171-185.

[15] Lagunez-Otero, J., et al., *Cellulat*, in *Artificial Life VIII: Proceedings of the Eight International Conference on Artificial Life*, R.K. Standish, M.A. Bedau, and H.A. Abbass, Editors. 2002: Sydney, Australia. p. 97-100.

[16] Corradini, F., E. Merelli, and M. Vita, *A multi-agent system for modelling carbohydrate oxidation in cell.* Computational Science and Its Applications - Iccsa 2005, Pt 2, 2005. **3481**: p. 1264-1273.

[17] Merelli, E., et al., *Agents in bioinformatics, computational and systems biology.* Briefings in Bioinformatics, 2007. **8**(1): p. 45-59.

[18] Corkill, D.D. *Collaborating Software: Blackboard and Multi-Agent Systems & the Future*. in *Proceedings of the International Lisp Conference*. 2003. New York.

[19] Lesk, A.M., *Introduction to bioinformatics.* 2002, Oxford ; New York: Oxford University Press. 283 p.

[20] Sadqi, M., *Atom-by-atom analysis of global downhill protein folding.* Nature, 2006. **442**(7100): p. 317-321.

[21] Creighton, T.E., *EXPERIMENTAL STUDIES OF PROTEIN FOLDING AND UNFOLDING.* Progress in Biophysics & Molecular Biology, 1978. **33**(3): p. 231-297.

[22] Snow, C.D., et al., *Absolute comparison of simulated and experimental protein-folding dynamics.* Nature, 2002. **420**(6911): p. 102-106.

[23] Mann, M., S. Will, and R. Backofen, *CPSP-tools - Exact and complete algorithms for high-throughput 3D lattice protein studies.* Bmc Bioinformatics, 2008. **9**.

[24] Shmygelska, A. and H.H. Hoos, *An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem.* Bmc Bioinformatics, 2005. **6**.
[25] Thachuk, C., A. Shmygelska, and H.H. Hoos, *A replica exchange Monte Carlo algorithm for protein folding in the HP model.* Bmc Bioinformatics, 2007. **8**.
[26] Clementi, C., *Coarse-grained models of protein folding: toy models or predictive tools?* Current Opinion in Structural Biology, 2008. **18**(1): p. 10-15.
[27] Jacob, E., A. Horovitz, and R. Unger, *Different mechanistic requirements for prokaryotic and eukaryotic chaperonins: a lattice study.* Bioinformatics, 2007. **23**(13): p. I240-I248.
[28] Blackburne, B.P. and J.D. Hirst, *Population dynamics simulations of functional model proteins.* Journal of Chemical Physics, 2005. **123**(15).
[29] Dill, K.A., *THEORY FOR THE FOLDING AND STABILITY OF GLOBULAR-PROTEINS.* Biochemistry, 1985. **24**(6): p. 1501-1509.
[30] Dill, K.A., *Polymer principles and protein folding.* Protein Science, 1999. **8**(6): p. 1166-1180.