Protein Folding Modeling with Neural Cellular Automata Using Rosetta

Daniel Varela
Department of Computer Science
University of A Coruña
Campus de Elviña s/n, 15071 A Coruña (Spain)
daniel.varela@udc.es

José Santos
Department of Computer Science
University of A Coruña
Campus de Elviña s/n, 15071 A Coruña (Spain)
santos@udc.es

ABSTRACT

In this work the temporal and dynamic folding of proteins was modeled with neural cellular automata, on the contrary to the ample research performed on the prediction of the final protein structure. Using the Rosetta environment and its coarse-grained representation, starting from an unfolded or partially folded chain, a connectionist model acts like a cellular automaton to define the moves of the dihedral angles of the protein chain. The process is repeated for all the angles of the amino acids and through several time steps until the protein is folded. The neural cellular automaton uses as input information a partial view of the energy landscape, obtained through the consequences in the energy changes when an angle is moved. The neural model learns to decide the best move in each angle in order to minimize the energy of the final folded conformation. The neural cellular automata are automatically obtained by means of differential evolution. Initial results with short proteins are presented and discussed.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and medical sciences

Keywords

Protein folding, Cellular automata, Evolutionary computing, Differential evolution

1. INTRODUCTION

Protein folding is the dynamic physical process by which a protein structure assumes its functional shape or conformation. Levinthal's paradox [8] postulates that it is too time-consuming for a protein to randomly sample all the feasible confirmation regions for its native structure. However, proteins in nature can still spontaneously fold into their native structures (the whole process typically takes only milliseconds or even microseconds to finish). So, the folding pathway of a protein is unclear, and a general assumption is that the lower a structure is in the energy landscape, the closer the folding is to the native state of the protein [3].

An ample research was performed in the prediction of the final folded conformation of a protein, even in the challenging *ab initio* modeling [18], although ignoring the dynamic nature of the folding. There are few previous works in this line. Krasnogor et al. [6] used cellular automata (CA) and Lindenmayer systems to try to define the rules and dynamics of the folding process, with a very limited success. They used a one-dimensional cellular automaton with four states that correspond to the possible moves in 2D lattices, and the rules of the cellular automaton were obtained with a genetic algorithm. In an extension of their work, the rules took into consideration the specific amino acids the rules were being applied to, thus connecting the CA modeling with a particular primary sequence. For example, for a short sequence of 20 amino acids, only 50% of the runs led to a set of rules that allowed achieving the optimal configuration. For larger sequences, the results were even poorer. Their work with Lindenmayer systems was only focused on finding out sets of rules that captured a given folded structure but, again, without a connection between the rules and the nature of the amino acids of the primary sequence.

In an alternative work by Calabretta et al. [1], the authors tried to establish the tertiary structure modeling the folding process through matrices of attraction forces of the 20 amino acids. The matrices of 20x20 components were obtained with a genetic algorithm, where each component represented the attraction or repulsion force between two amino acids in a given distance (100 Å). The model included rotations of the side chains of amino acids around the backbone and the possibility that a local portion of the backbone bended itself. The fitness function was measured taking into account the discrepancies of the alpha-carbon bend and the torsion angles between the real known structure and the artificial folded one. They used the methodology with a short fragment of crambin (13 amino acids) that resulted in an alpha-helix, as in the real protein.

More recently Danks et al. [2] presented a Lindenmayer system model which used data-driven stochastic rewriting rules to fold protein sequences by altering the secondary structure state of individual amino acid residues. The state of each residue was rewritten in parallel across the whole protein. The change in a residue state depended on the amino acid type of that residue and the amino acid types and the current states of the neighboring residues on either side. Seven secondary structure states were employed, based on those used in the DSSP database of secondary structure

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO'16 Companion, July 20-24, 2016, Denver, CO, USA

© 2016 ACM. ISBN 978-1-4503-4323-7/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2908961.2931720

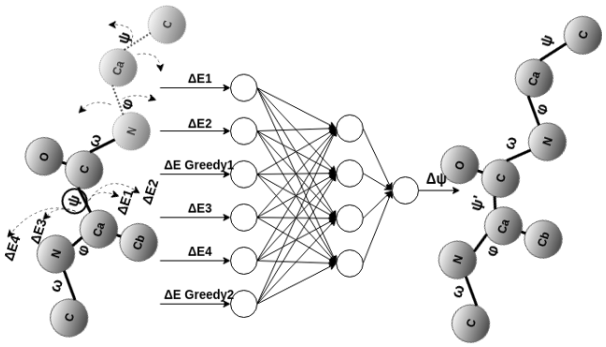

Figure 1: Neural-CA scheme for determining the change in angles ϕ and ψ. The inputs correspond to the energy increases (with respect to the energy with the current angle) when 4 perturbations are applied in the angle the neural-CA is applied to (ψ in the example). The two additional inputs correspond to energy increases when greedy moves are considered after the largest angle perturbations.

assignments, as well as their probabilities. Typical backbone torsion angles were obtained for each amino acid type in each of the seven states from the database and used to reconstruct the 3D structure of a protein at each derivation step. They showed results for four protein sequences from each major structural class. Local structure preference can be seen to emerge for some residues in a sequence. However, as indicated by the authors, the resulting structures did not converge to a preferred global compact conformation.

Our main goal is an attempt to model the temporal folding using CA-like systems, using evolutionary computing to automatically obtain the CA models that act over the multimodal energy landscape inherent to the protein folding problem [18], extending our initial work with simple lattice models like HP [13][14] to off-lattice models. The CA models will determine the moves of the dihedral angles of the amino acids through time. Such CA will be implemented with artificial neural networks (ANNs) that take input information about the energy landscape. A simple connectionist model was used to implement the CA-like systems, incorporating the advantage of the ANN generalization capability. Moreover, simulated evolution was used to optimize the connectionist models or neural cellular automata in order to obtain a folded conformation that minimizes the energy of the final folded protein. For the modeling the coarse-grained representation of the Rosetta environment was employed [12].

2. METHODS

2.1 Rosetta coarse-grained representation and protein conformational energy

The coarse-grained representation of Rosetta [12] was used. This centroid mode considers the location of the main backbone atoms, whereas each side chain is represented by a united pseudo-atom located at the side-chain center of mass. Each protein conformation is represented with the three dihedral angles, ϕ, ψ and ω, for each amino acid. The application of forward kinematics to this angular representation obtains the spatial information of the protein conformation.

Rosetta uses a physics and knowledge-based energy function [7]. Knowledge-based potential [17] refers to the empirical energy terms derived from the statistics of the solved structures deposited in PDB [9]. The physics-based energy function [5] contains terms associated with bond lengths, angles, torsion angles, Van der Waals and electrostatic interactions.

The Rosetta energy functions were used to calculate the free energy of each protein conformation. The Rosetta energy score of a protein is a linear combination of weighted terms that models molecular forces that act on and between all atoms in that conformation. There are energy terms such as solvation and electrostatics effects, repulsion, hydrogen bonding, and secondary structure scores such as strand pairing and helix-strand packing. Steric overlap of backbone atoms and side-chain centroids is penalized, but favorable Van der Walls interactions are modeled only by rewarding globally compact structures [11]. The Rosetta score function which takes into account all energy components, called *score3*, corresponds to the full coarse-grained energy function and it is used in all the energy calculations necessary for the folding process.

2.2 Neural cellular automata

A "neural cellular automaton" (neural-CA) is used for the modeling of the folding, since an artificial neural network is used to implement a scheme similar to a cellular automaton. The neural-CA is applied sequentially to each of the dihedral angles of the protein chain and in an iterative process. The neural-CA provides the next move of the dihedral angles ϕ and ψ of each amino acid, whereas the third dihedral angle (ω) is fixed (180°). The inputs of the neural-CA are determined by the consequences of possible moves of the ϕ or ψ angles to which the neural model is applied. The same neural network is applied sequentially to both angles of each amino acid of the protein sequence, being this process repeated through different temporal iterations or steps across the whole sequence of amino acids.

Inputs and output of the artificial neural network

The neural-CA is applied to the dihedral angles (ϕ and ψ) of each amino acid to decide their next moves. The ANN takes a decision based on a partial view of the energy landscape. This view is obtained considering perturbations in the corresponding angle the ANN is applied to. A value MAC ($Maximum\ Angle\ Change$) was considered for the changes (with respect to the current value) in an angle. The ANN has only 1 output that provides the increase of the angle to which the ANN is applied. The output of the ANN is constrained also to the range $[-MAC, MAC]$. The following 6 inputs are used in the ANN:

1. The ANN is applied to a particular angle (ϕ or ψ) of amino acid i of the chain (Fig. 1). This angle is perturbed in 4 quantities: MAC, $MAC/2$, $-MAC/2$ and $-MAC$. For each perturbation it is calculated the increase (positive or negative) of energy with respect to the current angle. These 4 energy changes (ΔE1, ΔE2, ΔE3 and ΔE4 in Fig. 1) are inputs to the neural-CA.

2. Since these inputs do not provide information about how is the possible energy landscape once a change

in an angle is taken, it is considered a (very limited) view of how would be the corresponding next energy landscape. For the two largest perturbations ($-MAC$ and MAC) in an angle in amino acid i, a greedy strategy several moves (N) forward is applied. That is, the next angle changes, in the next angles down the chain (ϕ or ψ), are defined by those that provide the minimum energy, considering as possible next moves $-MAC/2$ and $MAC/2$ (the average angle increases that the ANN can provide) in the next angles. Once these posterior angle changes are applied, the increases in energy (with respect to the energy of the current conformation) are also provided as inputs to the neural network (ΔE Greedy1 and ΔE Greedy2 in Fig. 1).

Hence, for each of the two largest angle perturbations ($-MAC$ and MAC), the network receives as input what would be the increase of energy if the neural network decided a greedy strategy in the next N angles (ϕ and ψ) of the chain. For the last amino acids only the possible moves in the final amino acids are taken into account (e.g., for the angle ψ of the last amino acid these two inputs are 0).

This way, the ANN has a view of the energy landscape in order to decide the most appropriate move in each situation, taking into account that the ANN can apply different changes than the greedy ones when it is applied to the next dihedral angles. Figure 1 shows a scheme of the feed-forward ANN used, summarizing the input information the ANN receives, using only a hidden layer. The output node determines the move in the corresponding dihedral angle, decoding its value to the range [$-MAC$,MAC]. The number of hidden nodes is selected between the number of inputs (6) and outputs (1), trying to obtain a good generalization capability and sufficient capacity to learn the association patterns (inputs - appropriate angle change).

2.3 Differential Evolution and fitness function

Simulated evolution was used to optimize a given neural-CA that provides the folding. Since the neural-CA is a feed-forward artificial neural network, each individual of the population encodes a possible ANN that generates a folding. As a standard and fixed transfer function (sigmoid) was used in the neural network nodes, every ANN is represented as its set of connection weights between the nodes of the layers. Differential Evolution (DE) [10] was used as evolutionary method, a population-based search method which needs a reduced number of parameters to define its implementation and which has proven efficiency in problems encoded with real parameters [4].

Each encoded neural-CA of the population is applied to a protein chain and we want that the iterative temporal folding, defined by that neural-CA, reaches a given conformation with a given and explicit fitness. The process begins with the protein sequences unfolded: all the dihedral angles ϕ and ψ are set to the same value (175°), whereas the dihedral angle ω is fixed (180°). Then, the neural-CA is applied sequentially to each of the dihedral angles ϕ and ψ of each amino acid of the chain (beginning with the angle ϕ of the first amino acid until the angle ψ of the last amino acid), where the neural-CA determines the next move of each angle. This procedure is repeated a maximum number of steps (neural-CA applied to all the angles ϕ and ψ) until a final conformation is reached, so the fitness of that individual (encoded neural-CA) is given by the final energy (*score3*) returned to the evolutionary method. At the end of each step a simple control is taken: If the energy of the final protein conformation is larger with respect to the energy of the final conformation of the previous step, then the folding process is ended, providing the final conformation (and fitness) of the last step.

Note that the fitness definition can be different to the definition of the energy used to calculate the energy increases that are the inputs to the ANN, although they are the same in the experiments presented here.

3. RESULTS

3.1 Experimental setup

For automatically obtaining a neural-CA that provides the folding Differential Evolution was employed with a population size of 60 individuals. Standard values for the DE parameters [10] were used: $CR = 0.9$ (crossover probability) and $F = 0.9$ (differential weight), whereas the variant $DE/rand/1/bin$ was used to select the base vector to disturb in the DE process (this provides the lowest selective process). The number of generations was set to 100. DE individuals code the ANN weights in the range [-1,1] and are decoded multiplying the encoded value by a constant MAX_VALUE. The value $MAX\_VALUE = 3$ was used since it allows to saturate the nodes using a standard sigmoid function as transfer function of the ANN nodes.

For the neural-CA processing, a value $N = 2$ was used, that is, in the next 2 dihedral angles ϕ or ψ greedy moves are applied in order to calculate the 2 additional energy increases that are inputs to the ANN. The parameter MAC for the angle perturbations was set to 10°. Finally, the number of maximum steps was set to 20, that is, the neural-CA is applied sequentially (over all the ϕ and ψ angles of all the amino acids of the protein chain) 20 times at most.

For the initial experiments the PDB proteins 1j4m (14 amino acids) and 1d5q (27 amino acids) were used.

3.2 Results with proteins 1j4m and 1d5q

First, we evolved a neural-CA to define the folding of a very short protein (1j4m, two sheets). Figure 2 shows snapshots of the conformations at the end of different temporal steps in the folding process, using the best neural-CA at the end of the DE process. These conformations are visualized with Pymol. The final folded conformation has an energy (*score3*) of 22, 23 (the energy of the native structure is 24, 02), and the RMSD of the folded conformation is 2, 97.

Note that our main aim is not to obtain better values of RMSD in the final folded conformation with respect to other methods, but to experiment with the possibility of the modeling of the folding working only with the information of the energy landscape. Also, it must be taken into account the inaccuracies in the Rosetta energy function, full of local minima. For example, the results of Shmygelska and Levitt [16] reveal "clear deficiencies in the low-resolution Rosetta energy function in that the lowest energy structures are not necessarily the most native-like".

Figure 3 shows the energy evolution in the conformations obtained through the folding process shown in Figure 2. This figure specifies the energy of the conformation after each angle move decided by the evolved neural-CA. Figure 3

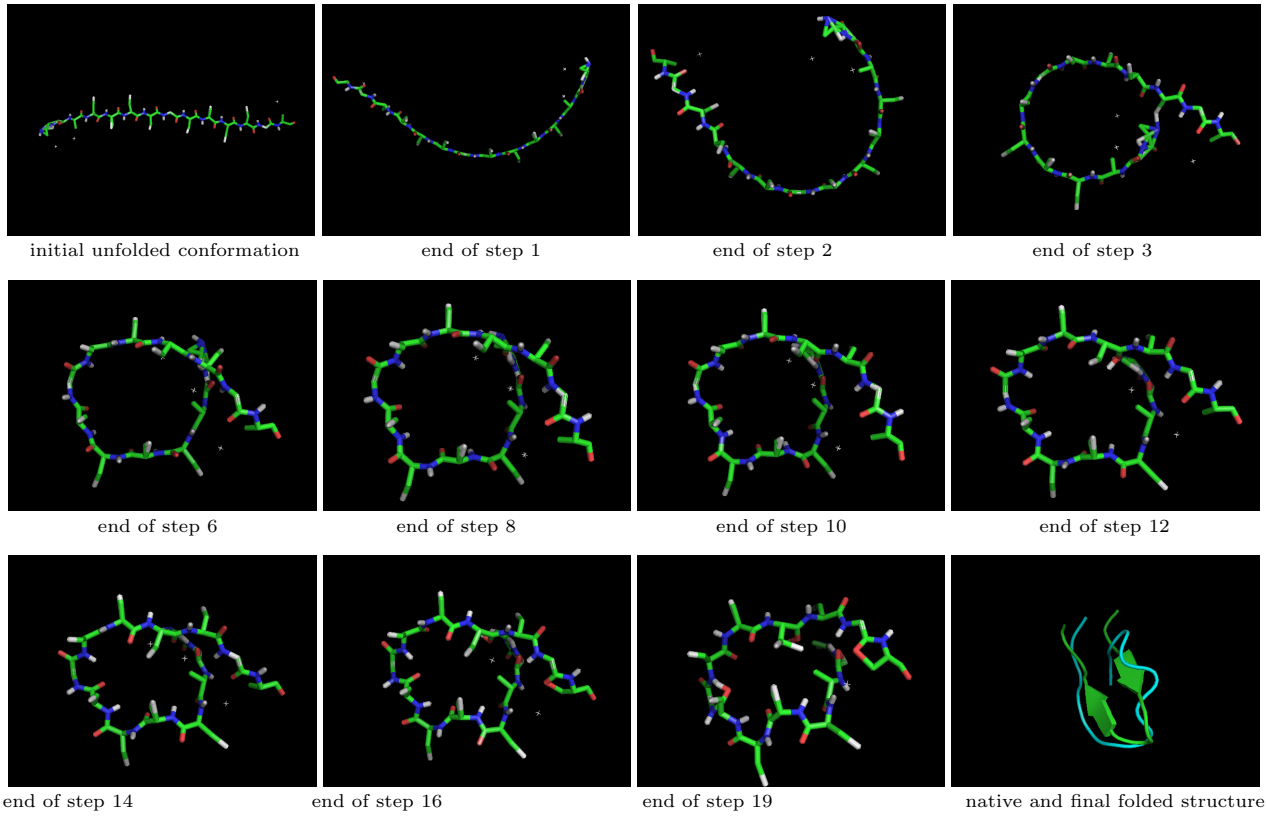

Figure 2: Snapshots of the folding at the end of different temporal steps with protein sequence 1j4m. The last snapshot shows the native structure (green) and the final folded structure at the last step (blue).

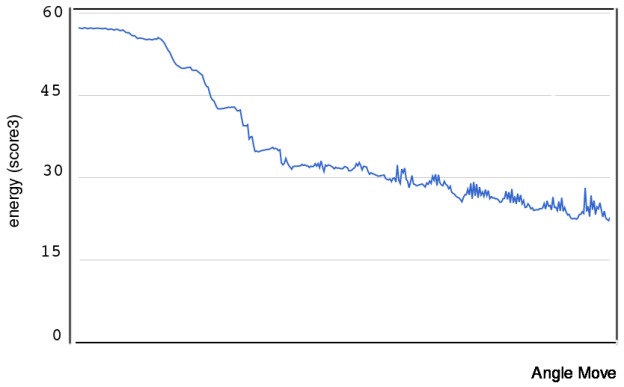

Figure 3: Evolution of energy (*score3*) during the folding process. The x-axis corresponds with the sequential moves of the ϕ and ψ angles during the temporal steps.

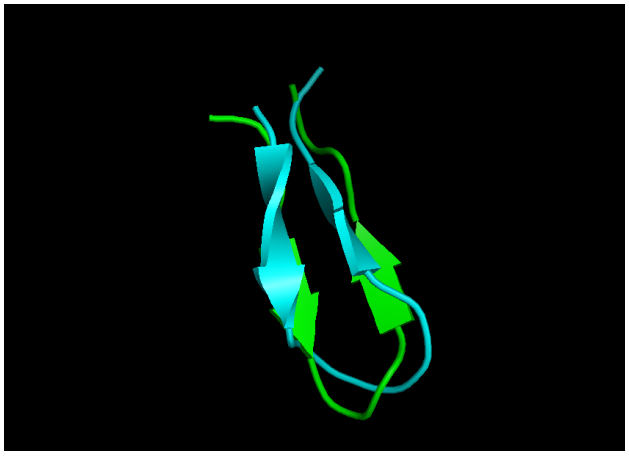

Figure 4: Final folded conformation for protein 1j4m (blue) and the corresponding native structure (green) when the energy term associated with secondary structure elements is taken into account.

shows that the conformations progress towards lower energy regions. However, it is clear that the best evolved neural-CA does not follow a strategy with angle moves that always decrease the energy. There are moves that can imply energy increases, but the next moves provide conformations towards the regions with decreasing energies.

Another aspect to consider is that the secondary elements are not obtained using the evolved neural-CA. The last snapshot of Figure 2 shows that the strands are not obtained by the secondary structure assignment of the Rosetta environment (DSSP), so the strands are not visualized. In a second experiment more information related with secondary structure elements (SSEs) is added to the energy function, in order to improve the formation of SSEs. We added to the Rosetta energy function *score3* another term that provides a positive reinforcement (value -1) when the Rosetta assigned SSE of each amino acid corresponds to the one of the 3D native structure, and a negative reinforcement (value $+1$) on the contrary. We used the SSEs elements of the native structure, but a secondary structure predictor can be used instead. This new term is averaged over all the amino acids of the protein conformation. Figure 4 shows a snapshot of the final folded conformation using the best evolved neural-CA incorporating the new energy term (used for the ANN inputs and fitness). This inclusion clearly facilitates the formation of the SSEs during the folding process. The RMSD of the final folded conformation in Figure 4 is now $2,16$.

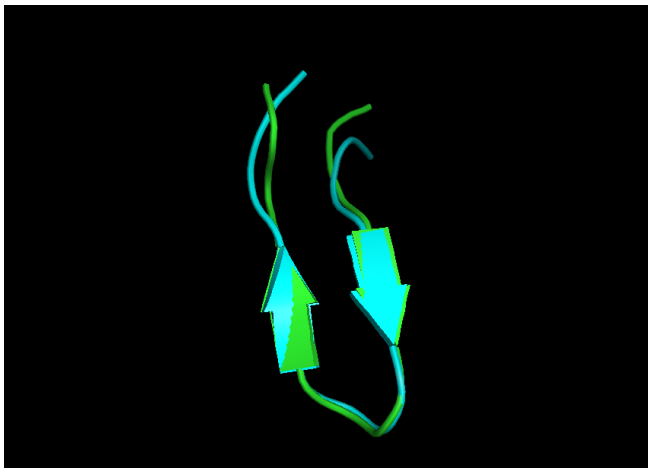

Figure 5: Final folded conformation for protein 1j4m (blue) and the corresponding native structure (green) when the energy term associated with secondary structure elements is taken into account and when the folding process begins with a partially folded conformation.

Next, a partially folded conformation was used to begin the folding process. The software TALOS+ (Torsion Angle Likeliness Obtained from Shift and Sequence Similarity) [15] was used to define an initial protein conformation. The program establishes an empirical relation between C, N and H chemical shifts and backbone torsion angles ϕ and ψ. TALOS+ uses a two-level feed-forward multilayer artificial neural network to predict the location in ϕ and ψ space based on a residue's NMR chemical shifts and amino acid type, and those of its adjacent residues. Beginning with the protein conformation given by TALOS+ (*score3* value $= 94,95$), a neural-CA is evolved, using again both terms (*score3* and SSEs correspondence) in the fitness function and energy calculations. With the best evolved neural-CA, the final folded conformation is shown in Figure 5. The RMSD value of this folded conformation is $0,99$ and its *score3* value is $22,61$.

Using another protein (1d5q, 1 helix, 1 sheet), Figure 6 corresponds to the final folded conformation using the best evolved neural-CA with the same setup of the previous Figure (using the additional term of the SSEs correspondence and an initial folded conformation obtained from TALOS+). The energy of the initial conformation is $85,69$, whereas the *score3* value of the final folded conformation, after the neural-CA folding, is $39,19$ (the energy value of the native structure is $14,52$). The RMSD of the folded conformation is $1,78$. In this case, the energy of the native structure was not improved, although a reasonable low value of RMSD was obtained.

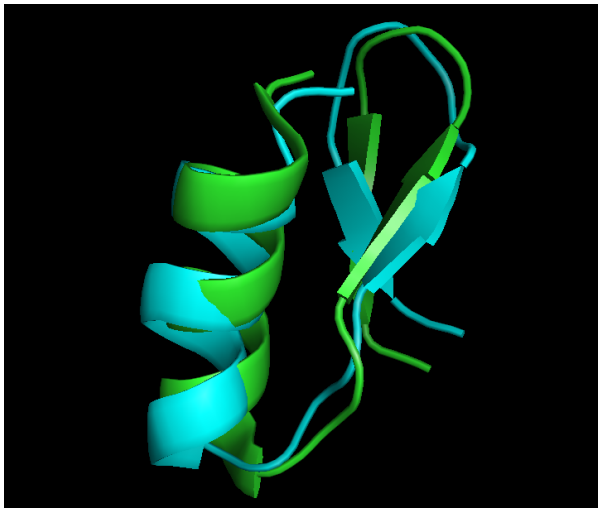

Figure 6: Final folded conformation for protein 1d5q (blue) and the corresponding native structure (green) when the folding process begins with a partially folded conformation.

4. CONCLUSIONS

An alternative strategy to define the folding process with off-lattice models was presented. The alternative uses an evolved ANN, which acts like a cellular automaton through all the elements (angles) and through time to define the appropriate moves so that the protein conformations fold towards regions of low energy. Using short proteins, initial experiments with evolved neural-CA are reported, which show that an evolved neural-CA can model the folding process working only with the energy landscape information, even with the limitations of the Rosetta low resolution energy function. The next work will be focused on the use of only local information as input to the neural-CA (information obtained only with the local vicinity of the angle to be changed), so the neural-CA can act as a folding operator, evolved with a protein or proteins but defining the folding of other different proteins.

5. ACKNOWLEDGMENTS

This work was funded by the Ministry of Economy and Competitiveness of Spain (project TIN2013-40981-R).

6. REFERENCES

[1] R. Calabretta, S. Nolfi, and D. Parisi. An artificial life model for predicting the tertiary structure of unknown proteins that emulates the folding process. *Proc. Third European Conference on Advances in Artificial Life - LNCS*, 929:862–875, 1995.

[2] G. Danks, S. Stepney, and L. Caves. Protein folding with stochastic L-systems. In *Artificial Life XI: Proc.. of 11th Int. Conf. on the Simulation and Synthesis of Living Systems (MIT Press)*, pages 150–157, 2008.

[3] S. Duarte, D. Becerra, F. Nino, and Y. Pinzón. A novel ab-initio genetic-based approach for protein folding prediction. In *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO'07*, pages 393–400, 2007.

[4] V. Feoktistov. *Differential Evolution: In Search of Solutions*. Springer, NY, 2006.

[5] A. Hagler and S. Lifson. Energy functions for peptides and proteins, II: The amide hydrogen bond and calculation of amide crystal properties. *Journal of the American Chemical Society*, 96:5319–5327, 1974.

[6] N. Krasnogor, G. Terrazas, D. Pelta, and G. Ochoa. A critical view of the evolutionary design of self-assembling systems. *Proceedings of the 2005 Conference on Artificial Evolution, LNCS*, 3871:179–188, 2002.

[7] J. Lee, S. Wu, and Y. Zhang. Ab initio protein structure prediction. In *From Protein Structure to Function with Bioinformatics*, pages 3–25. Springer-London, 2009.

[8] C. Levinthal. Are there pathways for protein folding? *J. Chim. Phys.*, 65:44–45, 1968.

[9] Protein Data Bank. http://www.wwpdb.org.

[10] K. Price, R. Storn, and J. Lampinen. *Differential Evolution. A practical approach to global optimization.* Springer - Natural Comp. Series, 2005.

[11] C. Rohl, C. Strauss, K. Misura, and D. Baker. Protein structure prediction using Rosetta. *Methods in enzymology*, 383:66–93, 2004.

[12] Rosetta system. http://www.rosettacommons.org.

[13] J. Santos, P. Villot, and M. Diéguez. Cellular automata for modeling protein folding using the HP model. In *Proceedings IEEE Congress on Evolutionary Computation - IEEE-CEC 2013*, pages 1586 –1593, 2013.

[14] J. Santos, P. Villot, and M. Diéguez. Emergent protein folding modeled with evolved neural cellular automata using the 3D HP model. *Journal of Computational Biology*, 21(11):823–845, 2014.

[15] Y. Shen, F. Delaglio, G. Cornilescu, and A. Bax. TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts. *J Biomol NMR*, 44:213–223, 2009.

[16] A. Shmygelska and M. Levitt. Generalized ensemble methods for de novo structure prediction. *PNAS*, 106(5):1415–1420, 2009.

[17] M. Sippl. Knowledge-based potentials for proteins. *Current Opinion in Structural Bio.*, 5:229 –235, 1995.

[18] X. Zhao. Advances on protein folding simulations based on the lattice HP models with natural computing. *Applied Soft Comp.*, 8:1029–1040, 2008.