A protein folding model using the Face-Centered Cubic lattice model

Daniel Varela
University of A Coruña
A Coruña, Spain
daniel.varela@udc.es

José Santos
University of A Coruña
A Coruña, Spain
jose.santos@udc.es

ABSTRACT

In this work the temporal folding process with a cellular automaton like-scheme was modeled. The cellular automaton is implemented with an artificial neural network and evolved with Differential Evolution. This neural-CA model is applied sequentially to the amino acids of the protein chain to obtain, iteratively and through time, a final folded conformation. The Face-Centered Cubic lattice model was used for the protein conformation representation, using a relative encoding of the amino acid moves on the lattice. First results of different folded conformations with different proteins are presented and discussed.

ACM Reference format:
Daniel Varela and José Santos. 2017. A protein folding model using the Face-Centered Cubic lattice model. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 5 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082543

1 INTRODUCTION

One of the most important problems in molecular biology is to obtain the native structure of a protein from its primary structure, i.e., the amino acid chain. The gap between known protein sequences and known three-dimensional form of proteins is difficult to be closed using classical time-consuming and difficult methods such as X-ray crystallography and NMR spectroscopy. As a result, computational methods to predict the native structure are becoming more important. Ab-initio methods are a computational approaches that consists of finding the lowest energy structure using only the amino acid sequence. This approach is based on the Anfinsen's dogma [1] that specifies that the native structure of a protein is its minimum free energy conformation.

Ab-initio methods adopt different approaches for the protein structure representation. For example, lattice models impose the constraint that the location of amino acids must be in the lattice sites. In the ab-initio protein structure prediction problem (PSP) many authors have been working on the use of search methods, specially evolutionary algorithms, employing the simple HP lattice model [9][12][18] or more complex lattice models like the Face-Centered Cubic (FCC) model [11][16][17]

However, there is a very limited research in the modeling of the dynamic folding, taking into account the different amino acid interactions through the temporal folding process. We have modeled this temporal process as an emergent and dynamic process using the classical tool of cellular automata (CA), implemented with an artificial neural network (ANN) model.

In previous work, Krasnogor et al. [8] used CA and Lindenmayer systems to try to define the folding process in 2D lattices, with very limited success. Calabretta et al. [4] tried to establish the protein tertiary structure by modeling the folding process through evolved matrices of attraction forces of the 20 amino acids in an off-lattice model. In previous work we have used CA to model protein folding using the basic HP model [6], with the 2D square [13] and the 3D cubic lattices [14][15], as well as with the off-lattice coarse-grain model of the Rosetta system [19].

This work is a first attempt to extend the methodology to the Face-Centered Cubic lattice model. The next section summarizes the methods used for the modeling: The FCC model and the neural-CA that provides the folding. We used Differential Evolution (DE) [10] to evolve the neural-CA. The results section shows initial results of the folding provided by the methodology.

2 METHODS

2.1 Face-Centered Cubic lattice model

One of the most studied lattice models is the HP model [6]. This model simplifies a protein's primary structure to a linear chain of H's (hydrophobic, i.e. nonpolar) and P's (hydrophilic, i.e. polar) that represents the pattern of hydrophobicity in the protein's amino acid sequence. The model is widely used because of its simplicity and it is powerful enough to capture a variety of properties of actual proteins [7].

The HP model imitates the hydrophobic effect by assigning a negative (favorable) energy weight to interactions between adjacent (in the lattice topology) and non-consecutive H amino acids in the primary structure. Proteins that have minimum energy are assumed to be in their native state. The energy of a protein conformation is defined as:

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3082543

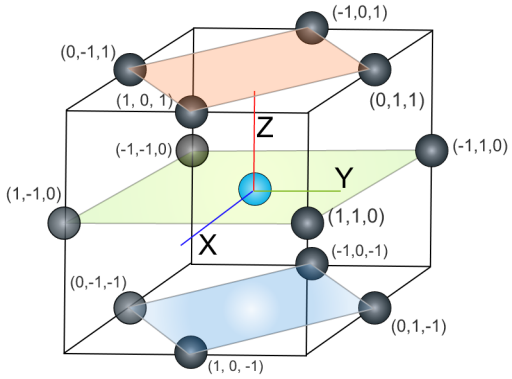

Figure 1: FCC lattice.

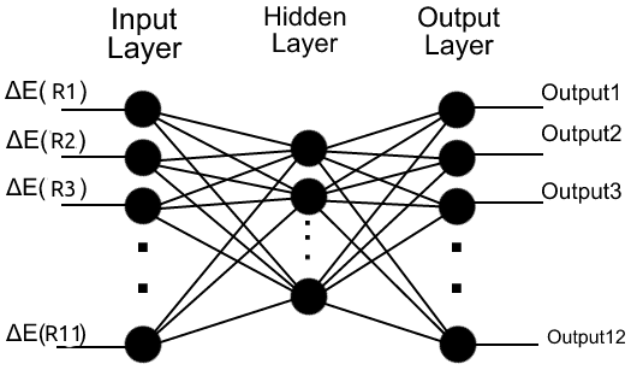

Figure 2: Neural CA scheme.

$$E = \sum_{i<j-1} c_{ij} \cdot e_{ij} \quad (1)$$

where $c_{ij} = 1$ if amino acids i and j are non-consecutive neighbors on the protein sequence and are neighbors (or in contact) on the lattice, otherwise 0; The term e_{ij} depends on the type of amino acids: $e_{ij} = -1$ if ith and jth amino acids are hydrophobic (H), otherwise 0.

The Face-Centered Cubic lattice has the highest average density compared to other lattices like the cubic or the body-centered cubic [5]. In the FCC lattice, the amino acids are located in the center and in the middle of the edges of the cubic unit cell, as shown in Figure 1. As a result, each lattice point has 12 neighbors with 12 basis vectors.

A protein of i amino acids can be encoded as a sequence of $i-1$ basis vectors that defines the 3D form. It will be a valid conformation if the sum of coordinates of each point is even and it consists of a self-avoiding walk. In the FCC lattice, two points p = (x, y, z) and q = (x', y', z') are adjacent in the lattice if and only if $|x - x'| \leq 1$, $|y - y'| \leq 1$, $|x - x'| \leq 1$ and $|x - x'| + |y - y'| + | z - z'| = 2$.

To represent a protein conformation, relative moves were used in this work. This means that the next move depends on (or it is relative to) the previous one, rather than relative to the axes defined by the lattice. Thus, there are 11 relative moves in the FCC lattice (further details can be found at [3]). This has the advantage that there is not a "back move", so there are not conflicts (collisions) between the next amino acid and the previous one.

2.2 Neural cellular automata

A simple feed-forward neural network is used as a cellular automaton to decide the moves of the amino acids in the FCC lattice and through time. We call it Neural Cellular Automaton (neural-CA) because the ANN implements the rule set of a classical CA.

The modeling of the protein folding process is as follows: the ANN is applied sequentially to each amino acid i of the protein chain, beginning with the unfolded protein conformation (protein in a straight line). The ANN receives input information from the energy landscape and the output decides the next move. For obtaining the appropriate information from the energy landscape, all possible moves between amino acids i and $i+1$ are considered (Figure 2), calculating the energy differences between the current protein conformation and the alternative conformations with these possible relative moves between i and $i+1$.

Therefore, there are 11 ANN inputs that correspond to the energy changes (positive, 0 or negative) when each of the relative moves are considered. For those energy calculations, only the close amino acids to the central amino acid i are considered, using the Euclidean distance with a given radius. Additionally, in these energy calculations, the HH contacts are weighted with -1 and the HP or PH contacts with a value 0.1. This provides a more detailed view of the energy landscape, which is useful when few H amino acids are located in that closest area to the central amino acid. Moreover, if a move implies a collision, the energy conformation is penalized with a high positive value. Finally, the ANN inputs are normalized in the range (-1, 1) in each situation.

This input information provides a partial view of the energy landscape to the ANN and can be associated with the central element and its neighborhood states in a classical CA. The output layer has 11 outputs that correspond to each of the 11 possible relative moves. The ANN node with the highest activation value determines the relative move to apply in each situation.

The process with the ANN is repeated several steps, that is, the ANN is applied sequentially to all the amino acids of the chain several times.

These neural cellular automata that perform the folding process are optimized by means of Differential Evolution [10]. The genotypes of the population define possible feed-forward ANNs (the connection weights), whereas the fitness function is defined as the energy (Eq. 1) obtained once the folding process has ended. The population size was 500 and the evolutionary algorithm was run for 1000 generations. In the folding process defined by each encoded ANN, if the encoded ANN decides a move that implies an immediate collision (in the next amino acid $i+1$), then the folding process is

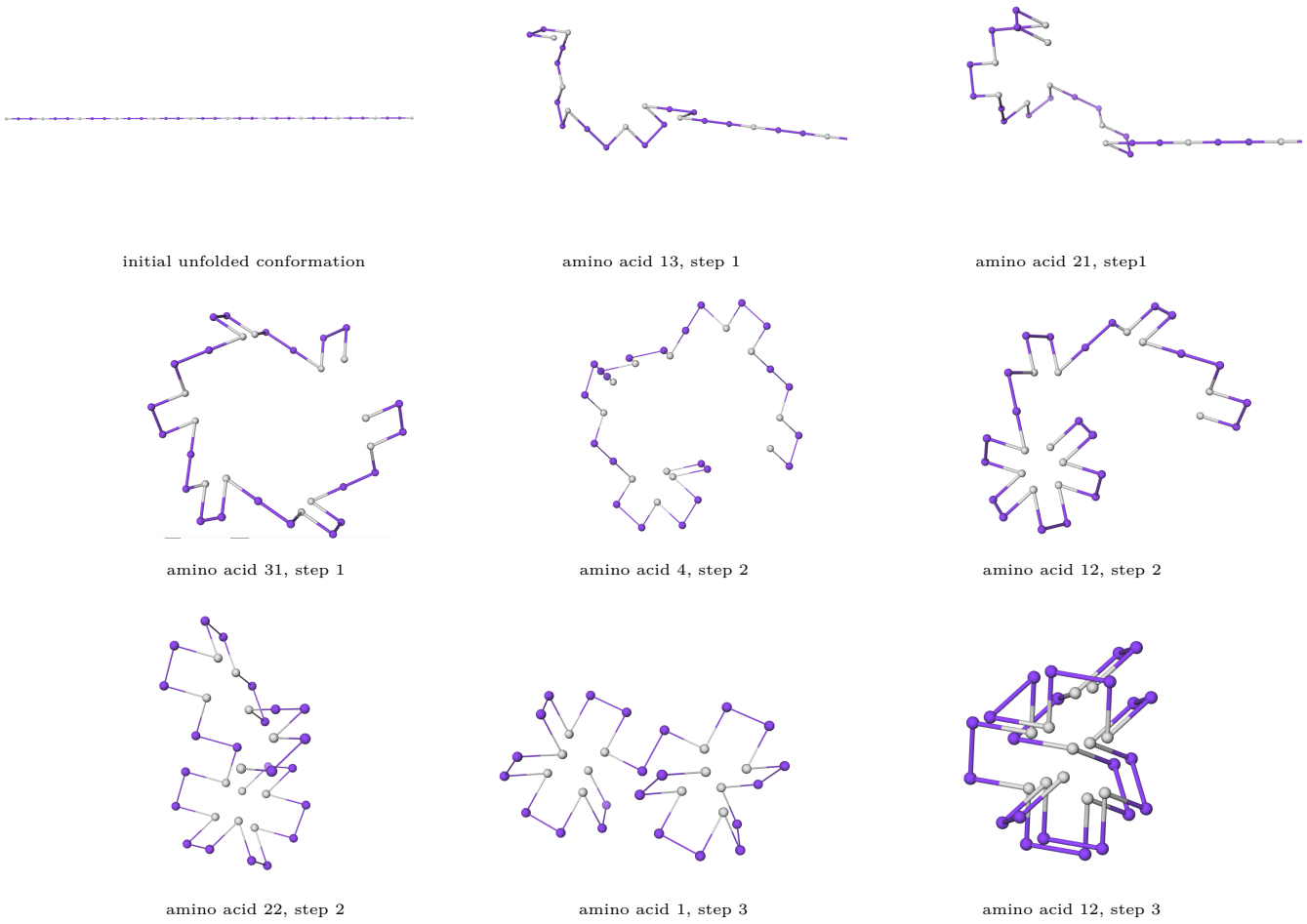

Figure 3: Different snapshots in the folding process with protein sequence HPPHPPHPPHPPHPPHPPHPPHPPHPPHPPH-PPH. The evolved neural-CA was applied to define the relative move between the amino acid specified in the caption of each subfigure and the next one.

ended and the energy before the collision is returned to the evolutionary algorithm. This helps that the encoded solutions (ANNs) can be refined progressively to obtain final folded conformations that maximize the number of HH contacts (minimize the energy conformation).

3 RESULTS

The methods explained in the previous section are used to obtain the folded conformations in different proteins were applied. The algorithmic setup is as follows. In DE, standard parameters were used (weight factor $F = 0.9$ and crossover probability $CR = 0.9$) [10]. Regarding the ANN, the ANN weights were set in the range [-1,1], sufficient to saturate the nodes of the ANN (sigmoid functions). The topology of the ANN is 11:5:11, trying to define a trade-off between the ANN capability to memorize and generalize. For the calculation of the energy increases, a spherical neighborhood (centered on the amino acid i to which the ANN is applied, Figure 2) with a radius ($r = 3$) was used in the different proteins, and the maximum number of steps (application of the ANN through the all the amino acids of the chain) was set to 3.

Figure 3 summarizes the neural-CA process with a sequence of 34 amino acids. The best evolved ANN is applied sequentially to the amino acids, through the maximum number of steps, beginning with the unfolded conformation. The different subfigures show partially folded conformations when the ANN applied the selected move in different amino acids and temporal steps. The optimal final folded conformation has 32 HH contacts with an internal hydrophobic core which maximizes the number of HH contacts.

Figure 4 shows the final folded conformation after the folding process defined by the best evolved neural cellular automaton for a benchmark sequence of 48 amino acids. This is one of the Harvard instances (H2) [20] commonly used with the FCC model. Again, there is a clear central core, where the H amino acids are located, which provides 55 contacts.

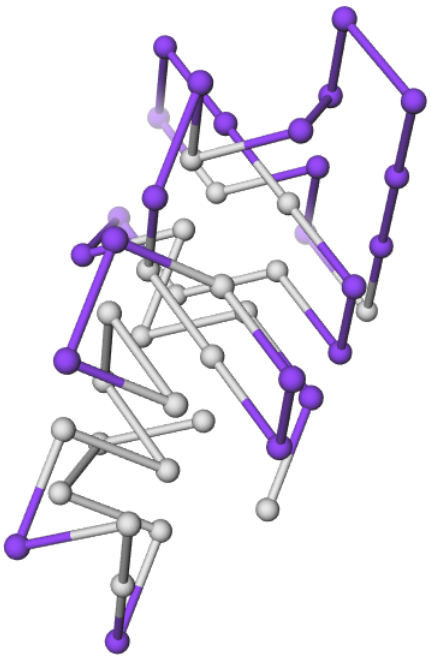

Figure 4: Final folded conformation of sequence H2.

However, the maximum number of possible HH contacts was not obtained. One of the methods that produces best results on the FCC model is the constraint-based protein structure approach (CPSP) by Backofen and Will [2]. The CPSP approach computes maximally compact sets of points used as hydrophobic cores. Then, it searches for a structure constraining the H-monomers to the H-core positions of the optimal hydrophobic core, obtaining a minimal energy conformation [7]. The CPSP approach obtains 69 HH contacts for that sequence. It should be taken into account that our objective is to define a possible modeling of the folding process and not to compete with the direct prediction methods of the final folded structure. But at the same time, this result indicates that the information the ANN receives may be insufficient in order to obtain the best optimized conformation regarding the number of HH contacts.

Figure 5 is another example with a PDB protein (3mse) with 179 amino acids. As in the previous case, the folded conformation provided by the best evolved neural cellular automaton does not obtain an optimal conformation, although many HH contacts are obtained in localized areas of the folded conformation. This conformation has 86 contacts out of 323 contacts of the optimal structure, obtained in this case by an exhaustive search [16]. This suggests that the radius considered to check the contacts around the central amino acid i (Figure 2), in order to inspect the dynamic energy landscape, should be tuned in relation, for example, to the protein length.

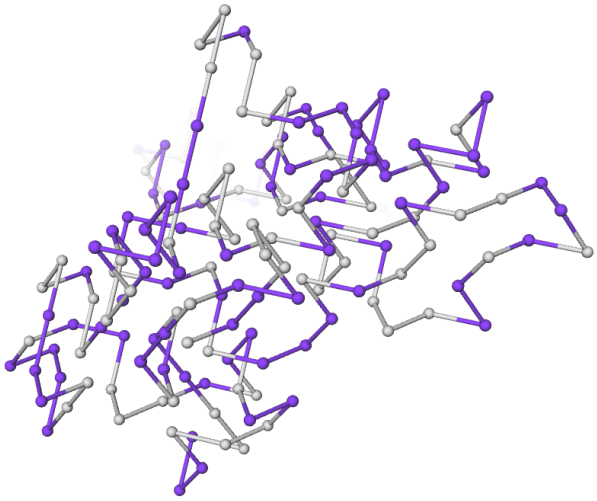

Figure 5: Final folded conformation of pdb sequence 3mse.

4 CONCLUSIONS

The neural-CA methodology provides an alternative to define the temporal folding process of a protein. Unlike the ample research performed on the protein prediction problem, which is only aimed at predicting the final folded conformation, the methodology presented here is an attempt to model the folding process as an emergent and dynamic process, incorporating the constraints of the FCC lattice model used. The next steps in this research must be focused on obtaining a better detailed view of the dynamic energy landscape the ANN receives and on the training of the ANN with different proteins and on the validation with other proteins.

5 ACKNOWLEDGMENTS

This work was funded by the Ministry of Economy and Competitiveness of Spain (project TIN2013-40981-R) and Xunta de Galicia (project GPC ED431B 2016/035).

REFERENCES

[1] C.B. Anfinsen. 1973. Principles that govern the folding of proteins. *Science* 181, 96 (1973), 223–230.
[2] R. Backofen and S. Will. 2006. A constraint-based approach to fast and exact structure prediction in three-dimensional protein models. *Constraints* 11, 1 (2006), 5–30.
[3] R. Backofen, S. Will, and P. Clote. 2000. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding. In *Proceedings of the Pacific Symposium on Biocomputing*. Citeseer, 92–103.
[4] R. Calabretta, S. Nolfi, and D. Parisi. 1995. An artificial life model for predicting the tertiary structure of unknown proteins that emulates the folding process. *Proc. European Conference on Advances in Artificial Life - LNCS* 929 (1995), 862–875.
[5] J.H. Conway and N.J.A. Sloane. 1998. *Sphere Packings, Lattices and Groups*. Springer-Verlag.
[6] K.A. Dill. 1990. Dominant forces in protein folding. *Biochemestry* 29 (1990), 7133–7155.
[7] I. Dotu, M. Cebrián, P.V. Van Hentenryck, and P. Clote. 2011. On lattice protein structure prediction revisited. *IEEE/ACM*

Transactions on Computational Biology and Bioinformatics 8, 6 (2011), 1620–1632.

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

[9] W.P. Patton, W.F. Punch, and E. Goldman. 1995. A standard genetic algorithm approach to native protein conformation prediction. In *Proceedings of 6th International Conference on Genetic Algorithms*. 574–581.

[10] K.V. Price, R.M. Storn, and J.A. Lampinen. 2005. *Differential Evolution. A Practical Approach to Global Optimization*. Springer - Natural Comp. Series.

[11] M.A. Rashid, M.T. Hoque, M.H. Newton, D.N. Pham, and A. Sattar. 2012. A new genetic algorithm for simplified protein structure prediction. In *Proc. Australasian Joint Conf. on Advances in Art. Intell., LNCS 7691*. 107–119.

[12] J. Santos and M. Diéguez. 2011. Differential evolution for protein structure prediction using the HP model. *Lecture Notes in Computer Science* 6686 (2011), 323–323.

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

[14] J. Santos, P. Villot, and M. Diéguez. 2013. Protein folding with cellular automata in the 3D HP model. In *Proceedings International Workshop on Evolutionary Computation in Bioinformatics - BIO 2013 - Genetic and Evolutionary Computation Conference (GECCO 2013)*. 1595–1602.

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

[16] S. Shatabda, M.H. Newton, M.A. Rashid, and A. Sattar. 2013. An efficient encoding for simplified protein structure prediction using genetic algorithms. In *Proc. IEEE Congress on Evolutionary Computation - IEEE-CEC 2013*. 1217–1224.

[17] J.J. Tsay and S.C. Su. 2013. An effective evolutionary algorithm for protein folding on 3D FCC HP model by lattice rotation and generalized move sets. *Proteome science* 11, 1 (2013), S19.

[18] R. Unger and J. Moult. 1993. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology* 231(1) (1993), 75–81.

[19] D. Varela and J. Santos. 2016. Protein folding modeling with neural cellular automata using Rosetta. In *Proceedings of Genetic and Evolutionary Computation Conference - GECCO'16*. 1307–1312.

[20] K. Yue, K.M. Fiebig, P.D. Thomas, H.S. Chan, E.I. Shakhnovich, and K.A. Dill. 1995. A test of lattice protein folding algorithms. In *Proceedings of the Pacific Symposium on Biocomputing, Vol. 92, No. 1*. 325.