Computational Modeling of Proteins based on Cellular Automata

Alia Madain, Abdel Latif Abu Dalhoum, Azzam Sleit
Department of Computer Science
King Abdulla II School for Information Technology
The University of Jordan
Amman, Jordan

***Abstract*—The literature of building computational and mathematical models of proteins is rich and diverse, since its practical applications are of a vital importance in the development of many fields. Modeling proteins is not a straightforward process and in some modeling strategies, it requires to combine concepts from different fields including physics, chemistry, thermodynamics, and computer science. The focus here will be on models that are based on the concept of cellular automata and equivalent systems. Cellular automata are discrete computational models that are capable of universal computation, in other words, they are capable of doing any computation that a normal computer can do. What is special about cellular automata is its ability to produce complex and chaotic global behavior from local interactions. The paper discusses the effort done so far by the researchers community in this direction and proposes a computational model of protein folding that is based on 3D cellular automata. Unlike common models, the proposed model maintains the basic properties of cellular automata and keeps a realistic view of proteins operations. As in any cellular automata model, the dimension, neighborhood, boundary, and rules were specified. In addition, a discussion is given to clarify why these parameters are in place and what possible alternatives can be used in the protein folding context.**

Keywords*—*Proteins 3D Folding; Bioinformatics; Computational Modeling; Cellular Automata; Theoretical Computer Science;

I. Introduction

Modeling any complex biological phenomenon is essentially a form of abstraction. The game of building a meaningful model usually falls back to making choices of what to keep and what to eliminate from available information. Nevertheless, Models have many advantages, as they tend to be more accessible and convenient for understanding the subject of study. Additionally, models can act as objects of further experimentation [1]. This perfectly applies to modeling proteins, because not only they are diverse, but also the simplest protein endures a huge amount of details.

Artificial Intelligence and image processing concepts are heavily used in the domain of modeling proteins such as neural networks [2], optimized evidence-theoretic K-nearest neighbor classifier [3], complexity measure factor [4], moments [5], in addition to fusing multiple classifiers [6].

A cellular automaton (CA) is a discrete model of computation that is studied in computability theory. CAs are simple since they are based on local interactions only but they are capable of exhibiting complex behavior [7].

Simply a CA has a collection of identical cells that are distributed spatially in one dimension, two dimensions or higher. Every cell in the CA has a finite number of possible internal states, the CA evolves from one iteration to the other based on transition rules that are applied simultaneously to all CA cells. The rules depend mainly on the cell neighborhood and may or may not consider the cell state itself.

There are many options for almost all aspect of CAs. CAs differ in their spatial distribution, cell neighborhood, transition rules, cell possible states, boundary, number of generations (iterations), cells shape, and the initial configuration from where the CA starts.

Although, all proteins composition is based on twenty amino acids, proteins are diverse and cover multiple functions in nature. Some proteins contain a surfeit of one amino acid whereas others may have one or two members of the twenty amino acids missing entirely [8]. Since there are many details in real proteins, simplified models called simple exact models (SEMs) were proposed. The most common one is the HP model, which consists of only hydrophobic (H) and polar (P) Monomers [9].

This paper discusses the CA potential in the domain of protein modeling and shed light on the possibilities offered by the CA concept. In addition, it focuses on the process involved in protein modeling when CA is used which is quite different from other computational paradigms. Finally, a 3D CA model is proposed and the challenges of protein modeling in terms of CA are discussed.

The remaining of this paper is organized as follows: Section II gives the related work; Section III includes background information about proteins; Section IV discusses the CA potential in the context of protein modeling; Section V presents the proposed model; Finally, section VI concludes the work done and gives direction to future work.

II. Related Work

The CA concept is related to many disciplines including mathematics, physics, biology, and computer science [10]. The idea of employing CA to the central dogma of molecular biology is not new and many attempts were made to model the central dogma in terms of structure, function, and evolution. CA models were used in modeling DNA sequences [11], evolution [12], mutation prediction [13], and gene networks [14].

One of the most attractive properties of CA is its ability to represent global behavior, and this is truly important in modeling the central dogma of molecular biology since the initial state of the protein synthesis process does not help in understanding the system behavior as a whole.

In this section, the discussion covers the work that depends on elementary cellular automata combined with pseudo-amino acid composition. In addition to the methods that combine CA with evolutionary algorithms, and finally work done in L-Systems is covered, since L-Systems were proved to be equivalent to CAs.

One work that is used in predicting multiple protein attributes is that based on elementary Rule 84 and pseudo-amino acid composition. This line of research depends on amino acid coding language proposed in [15] to act as the initial configuration of the elementary CA. This model is used to predict protein subcellular location [16], the G-protein-coupled receptor functional classes [17], and protein structural classes [18] [19].

The process starts with converting the protein amino acid sequence to the binary encoding and assumes the binary representation of each protein sequence as the initial configuration of the CA, after the CA runs for 100 generations, the resulting image parameters are extracted as given in Figure 1. These CA image parameters are then considered along with 20 more attributes to calculate the PseAA representation of each protein and each group of proteins as given in Figure 2.

In fact, PseAA proposed in [20] is widely used in protein modeling, which differs from traditional AAC in that it adds the protein sequence order effect in a set of discrete numbers. Surveys tailored to methods depending on the pseudo amino acid composition are given in [21] and [22].

In addition, CA was combined with evolutionary algorithms. An interesting work is the one that proposes a CA-like structure or a neural CA, where the cellular automaton is implemented by means of a simple feed forward neural model. The artificial neural network output correspond to the possible relative movements.

The idea was implemented in two-dimensions (2D) [23] and three-dimensions (3D) [24]. In 2D case the possible movements are forward, left and right while in the 3D case, the possible movements are forward, up, down, left, and right.

The work done in [25] combines CA with genetic algorithms to predict the protein secondary structure, where the genetic algorithm is used to optimize the parameters (Rules) involved. The authors summarizes what effects the prediction to three factors: the neighborhood, weights assigned to the neighborhood and the number of generations.

Specially designed cellular automata were proposed to model the chemical reactions of DNA replication, mRNA transcription, and splicing process in [26], where the protein synthesis process was left for future work.

Moreover, Systems proven to be equivalent to cellular automata such as L-systems [27] [28] were used to model proteins in [29] [30] and [31].

In this paper, the design of a 3D CA proteins model is discussed. The model is meant to be as simple as possible and within the known CA configurations. It does not require any combined representation nor does it require searching capabilities and evolutionary algorithms. The discussion also covers challenges in such design and present alternative solutions to each.

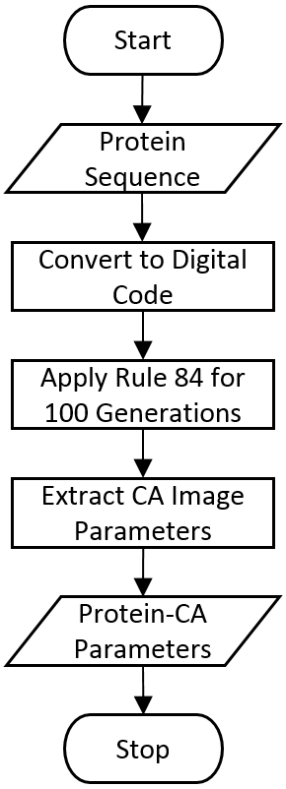


Fig. 1. Workflow of Protein Parameters Extracted from CA Images

III. Proteins

Proteins are the end product of the DNA decoding process. The central dogma of molecular biology states that DNA is transcribed into messenger RNA (mRNA), which is translated into proteins. This way of viewing the process is quite simplified, in reality this biological process is a rich and complex set of events [32].

In a cell, proteins are the workhorses and lead performers of cellular functions [33], they can be considered as specialized machines, each of which fulfills its own task. All the complex molecules of the cell are proteins except DNA and RNA which are not proteins and considered complex as well [32].

To simplify things, proteins are all united through their reliance on the same group of twenty amino acids, they consist of a linear arrangement of amino acid residues assembled together into a polypeptide chain and the order of linking the residues together is ultimately derived from the genes information.

Amino acids contain amine (-NH2) and carboxylic acid (-COOH) functional groups, usually along with a side-chain usually referred to as an R group that is specific to each amino acid. The key elements of an amino acid are carbon, hydrogen,

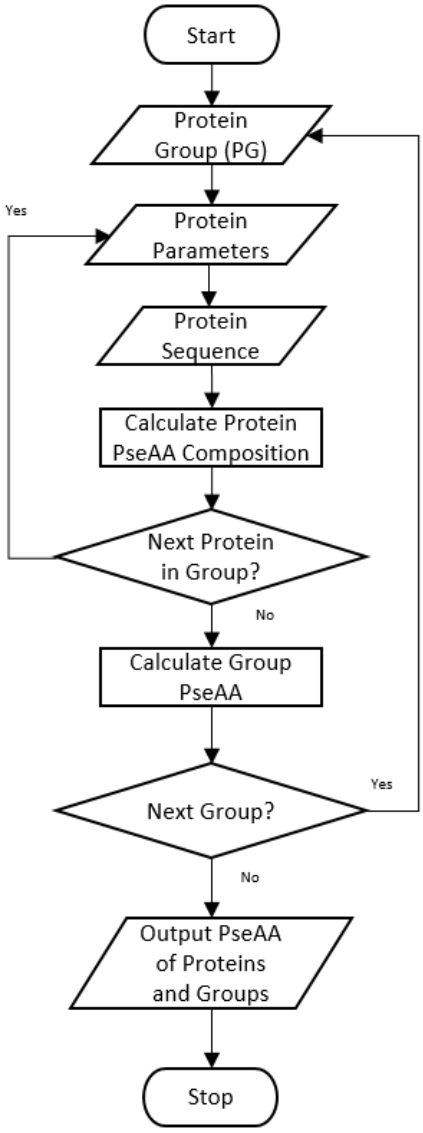


Fig. 2. Workflow of PseAA Composition of Proteins and Their Groups

oxygen, and nitrogen, though other elements are found in the side-chains of certain amino acids [34].

The translation to proteins starts by the ribosomes which proceed along the mRNA one codon at a time incorporating one amino acid at each step and finally leaving the mRNA from the last codon.

This process results in a chain of amino acids called the primary structure of a protein. Mathematical and computational modelling of ribosomal movement along with a discussion of the impact of modeling studies on experimentalists is summarized in [35].

Protein folding is the process that folds the protein primary chain to its native three-dimensional structure, which is a specific and stable structure. The three-dimensional structure of a protein defines its function [33].

IV. Cellular Automata

A. Cellular Automata Potential

Starting this section with some historical information may sound redundant but it is essential to get a feel of why CA has potential to model proteins properly. The use of CA as a model is justified by its roots in biology and that it is especially relevant to the problem of protein modeling.

CAs were originally proposed as formal models of self-reproducing organisms. In the forties, John Von Neumann wanted to design a machine that can reproduce itself. He suggested a programmable assembly machine that can build a copy of itself, and he defined two phases in the machine blueprint, which are translation and transcription. The problem of this machine is in its components, which are sophisticated logical units. This is when Stanislaw Ulam suggested that Von Neumann use cellular automata, which Ulam used to study the growth of crystals at the time.

Also, the theory of Konrad Zuse is very relevant in this context, which suggests that physics is just computation. Zuse tried to apply an information and automata theory approach to certain problems of physics [36] in his article written in German (Rechnender Raum) which literally means space that is computing. In 1969, he published the book Rechnender Raum [37] which was translated into English as ”Calculating Space”, Zuse proposes that the universe is computed by some sort of CA or other discrete computing machinery.

In addition to being a suggested framework for researching connections between biology and automata theory, CAs design is open and flexible, there is no restrictions or mathematical formulas that restricts the construction of CAs. Another advantage of using CAs is the different behavior dynamics resulting from different rules namely, stable, periodic, chaotic and complex ones.

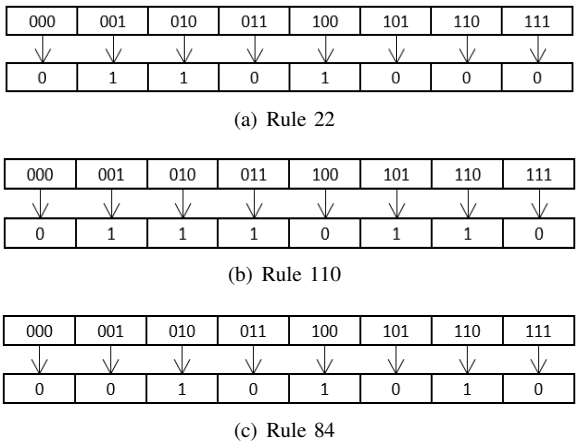
Finally, CAs are parallel in their nature which can be applied in many different ways using commercially-available parallel computers where the state of cells can be updated simultaneously, or using specialized CA machines.

B. Cellular Automata Technical Details

CA can be described as a set of cells arranged in any dimension, for example, cells can be arranged in a two dimensional grid or a one dimensional array. These cells can take a finite number of states and the states can be of any type for example the set of states can be binary (0,1) or an integer number or any other finite set of states.

The state of each cell may change or stay the same at every generation, the stability or change of the states depends on predefined rules (a transition function). The rules use the state of a cell neighbors as input and may or may not use the cell state itself to determine the cell state in the next generation.

According to Wolfram, it seems that the patterns which arise from different types of cellular automata can almost always be assigned to one of just four basic classes [38] [39] [10]. In class 1, patterns evolve into a stable, homogeneous state; in class 2, patterns evolve to a periodic state; in class 3 a chaotic behavior appears; and in class 4, configurations contain structures that interact in complex ways.



(a) Rule 22

(b) Rule 110

(c) Rule 84

Fig. 3. Different CA Rules

Langton [40] defines the Lambda parameter, which is a way to quantify the qualitative approach of Wolfram. In ideal cases, transition rules with the same Lambda evolve to a similar behaviour [41].

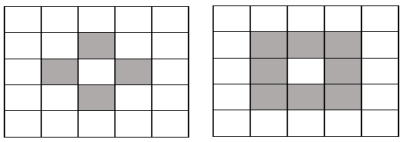
Although it is widely accepted that the power of CA is its ability to exhibit fascinatingly complex behavior from local rules, there is no superior behavior in CAs. In biological modeling, the appropriate behavior is the one that represents reality. In practical applications, the best behavior is the one that achieves the goals of the application, for example, work done in [42] uses chaotic behavior in image security and chaotic elementary CAs had an equivalent effectiveness as complex game of life in a multimedia related application [43] [44].

The simplest cellular automata (elementary CA) is one dimensional and the rules depend on the cell state and the state of its left and right neighbors (values of the nearest neighbor). So the combinations of each cell and its neighbors have 8 possibilities only. There are only 256 elementary cellular automata, each of which can be indexed with an 8-bit binary number. All the behavioral cases defined by Wolfram are covered within the 256 rules of the elementary CA.

The CA is referenced by its rule number, which can be easily computed in the case of elementary CA. The rule number is simply the decimal number representing the rule output, so for every combination of the three cells (core cell and its neighbors) the rule give an output that is either zero or one, this output is then concatenated to a binary string and converted to a decimal number representing the rule number. Figure 3 shows the 8 states of rules 22, 110, and 84.

In a two-dimensional context (2D) some parameters are different. In 2D CA Moore and von Neumann are two widely used neighborhood configurations. In von Neumanns neighborhood, every cell has four neighbors: the cells at its North, South, East, and West, whereas in Moores neighborhood the cells at the four diagonals are also considered, as given in Figure 4.

One famous two-dimensional CA is the Game of life proposed by John Conway [45]. The rules of Conways game of life are simple and assumes a Moore Neighborhood.



(a) von Neumann (b) Moore

Fig. 4. Different CA Neighborhood

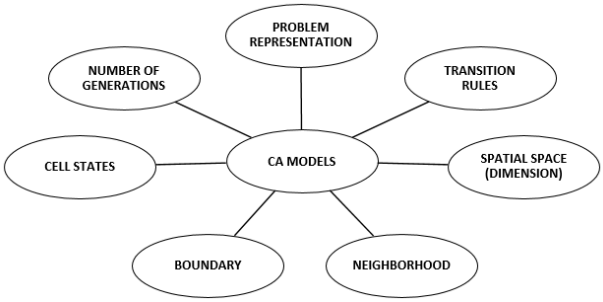


Fig. 5. CA based Modeling

V. Proposed Model

The problem of modeling proteins using cellular automata requires representing the problem in a way that maps to reality and that is suitable for cellular automata at the same time, it also requires setting multiple parameters such as the dimension, neighborhood, boundary, number of generations, and most importantly the transition rule as given in figure 5. In addition, there must be a way to check the validity of the CA used and to make sure it represents a realistic behavior, maybe by finding certain attributes specified by this process.

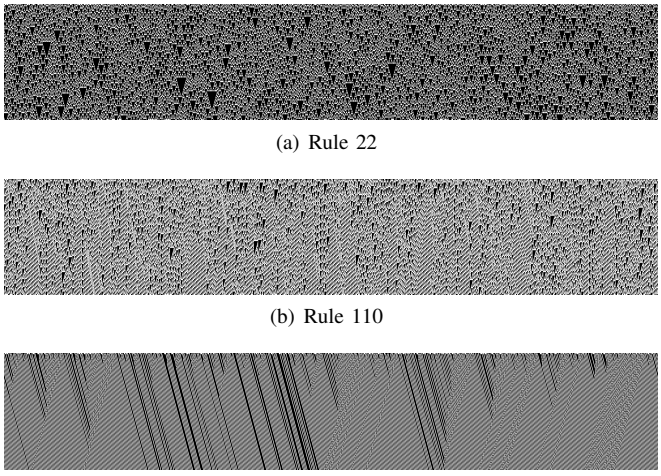
A. Problem Representation

The process of modeling proteins starts with the challenge of representing the problem in a set of finite, discrete values. Proteins in reality are full of details. Until today, the functional motions of proteins usually operate at timescales and conditions that are beyond the limits of current technology [46].

One way to specify the values that each cell in the CA uses, is to convert the twenty amino acids to a five digits binary representation. The binary representation has many advantages in the context of CA since the properties of CA are mostly studied in the binary domain.

The conversion between amino acids and binary representation is not random. Authors in [15] and [47] makes use of similarity rule, complementarity rule, molecular recognition theory, and information theory to give the digital coding of amino acids. Figure 6 shows the use of this coding in protein modeling using rules 22, 110, and 84. The figure shows the initial configuration and 100 generations of the same protein. The reason why these rules were chosen is that rule 22 is known for the chaotic behavior and rule 110 proved to being capable of complex behavior and rule 84 was used before in the context of protein modeling.

A comparison of four coding methods is given in [48], the binary codes presented are either based on biochemical properties or generated by artificial intelligence (AI) methods.



(a) Rule 22

(b) Rule 110

(c) Rule 84

Fig. 6. Different CA Rules representing the same protein

Another approach is the use of the HP model, which classifies the amino acids into two classes, the hydrophobic ones and the polar ones. Although this is a very useful abstraction, keeping the 20 amino acids with their one letter representation might allow for adding more rules of biochemistry.

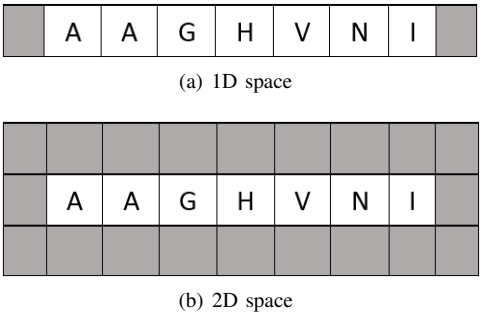
Notice that some approaches combines two methods together. In [16] [18], [17] and [19], the binary representation is followed by the Pseudo-amino acid composition (PseAA). In those Papers, CA is applied to the binary representation, then the CA image parameters are extracted by methods such as the geometric moments of Hu [49] and the GLCM texture features [50]. As explained before in the related works section, the discrete numbers identifying the protein are added to 20 attributes to form the PseAA composition of proteins.

B. Environment and CA parameters

The environment effects the protein folding, for example, the hydrophobic and hydrophilic properties of the amino acids forming the proteins are important in the context of protein folding since the environment surrounding the protein contains water.

The work done in [29] [30] and [31] models the folding of protein-like structures using local rewriting rules with environmental interaction. In the context of cellular automata, the dynamic environmental factors may be modeled by the CA rules whereas the static environmental factor such as the existence of water may be assumed as a part of the initial configuration.

It is assumed that CAs have an infinite grid then every cell has neighbors. Nevertheless, the actual implementation of space is usually finite and therefore there must be a way to handle the neighborhood over the edges. The proposed model uses water as the boundary of the initial configuration. Therefore, if the CA is implemented in a one-dimensional space the neighbors of the first cell and the last cell are two cells of water and in the case of two-dimensional space, the extreme cells in the four dimensions are assumed water cells, as shown in Figure 7.



(a) 1D space

(b) 2D space

Fig. 7. Suggested CA initial configuration in 1D and 2D spaces

The proposed model uses 3D CA space, so now the CA used is more like a cube. The reason why the 3D model was chosen is that it maps to proteins reality. At first, the finite spatial space and the water boundary are assumed, then the flexibility of expanding the spatial space is given.

One main issue here is that the amino acids move and since CA is highly parallel, one might expect that these amino acids will end up in the same cell, which makes the possible state of a cell more complicated. Based on the basic principles of CA, there must be a finite set of states for the CA cells. In order to overcome this problem there are many possible solutions. The first thing that comes to mind is to restrict the possible movement of each cell, but that might effect the model accuracy.

Another work around is to change the cell shape to become hexagonal (still a homogeneous grid). What is gained from this conversion is that it is possible to map the hexagonal cell to a group of square cells with adding the radius of neighborhood and without effecting the assumption of a finite set of states in each cell. This work around adds to the complications of the design.

Another alternative solution is the use of a timed CA where neighboring cells or competing cells to the same position does not calculate the transition function at the same time. Each cell has its turn based on its position. So the cells in the second round can check the availability of the position. A similar solution is to use block CA or a partitioning CA, where groups of cells are divided into non-overlapping blocks and instead of applying the transition rule to each cell individually, it is applied to a whole block at each time step.

An interesting challenge in modeling proteins in terms of CA is keeping the amino acid connected to its neighbors in the primary structure, one way to do this is to number the primary chain of the protein, and check that there is no separation between originally connected amino acids. This problem can be partially fixed after the final generation or it can be ignored. A more complicated solution is to backtrack illegal moves after each generation.

The actual neighborhood chosen is usually crucial for the global behavior of a CA. most CA studies restrict the neighborhood to Moore or von Neumann [51]. The details of the proposed 3D CA is given in Table I. The use of Moore neighborhood has an advantage in modeling protein folding, since the presence of certain amino acids and the connections between them effect the folding. In addition,

TABLE I. CA PARAMETERS

No.	Parameter	Value
1	Dimension	3D
2	Boundary	Water
3	Neighborhood	3D Moore Neighborhood
4	Spatial Space	Infinite (finite and can be extended)
5	Number of Generations	Specified based on the protein stabilization
6	Possible States	Amino acids of the primary structure and water

Moore neighborhood might serve the heuristic rules described in section V-C and gives a meaningful abstraction of each cell environment.

C. CA Rules

Section V-B discusses the CA configuration before moving from one generation to the other. The rules or the transition function is what cause the global behavior to occur; they are essential to understand the behavior of proteins.

According to Chou [52], the three main strategies developed in structural bioinformatics, are pure energetic approach, heuristic approach, and homology modeling approach. Pure energetic approach depends on the thermodynamics principle. The heuristic approach on the other hand collects the physical, chemical, and biological principles as much as possible.

Finally, the homology modeling approach, which is a well-known method of modeling proteins, compares the protein in hand (target protein) with related proteins stored in a database (template proteins). When the target and template proteins are closely related, homology modeling can produce accurate structural models with more reliable results than other methods. Nevertheless, the quality of the homology model depends on the data used and the quality of the sequence alignment and template structure.

In the CA context, the heuristic approach seems to be the most relevant. the priority is given to the chemical properties of hydrophobic and hydrophilic amino acids. Therefore, if the amino acid is hydrophobic and is surrounded by water, it must change its position preferably towards other hydrophobic amino acids.

The following subsections discuss the possible use of simple rules and principles of chemistry and thermodynamics.

1) Chemistry: Usually the abstraction of the chemistry behind protein folding depends on the hydrophobic and hydrophilic amino acid properties. In the living cell, Ribosomes read the mRNA to produce the amino acid chain. After that, proteins are in an environment full of water (around 70% of the living cell), so it will spontaneously fold.

In the protein folding process, one can imagine the hydrophobic amino acids cluster in the core of the protein since those amino acids move away from the water in the environment. On the other hand, hydrophilic amino acids fold around this core as if they are trying to protect the hydrophobic amino acids.

Moreover, the interactions that stabilizes the protein can be added such as the salt bridge where positively charged side chains likes to be close to negatively charged side chains. The salt bridge is a combination of two non-covalent interactions, namely, hydrogen bonding and electrostatic interactions.

One more rule that can be added is the contribution of cysteine in folding. Similar to the case of salt bridges, cysteine plays an important role in stabilizing the protein because of the disulfide bridges.

2) Thermodynamics: Thermodynamics is the study of energy. The second law of thermodynamics states that in an isolated system, the total entropy always increases or remains the same but never decreases.

The measure of a molecule energy is Gibbs free energy (G), let the change in free energy be ΔG and the energy of the final state be G_f and the energy of the initial state be G_i, then ΔG is calculated as follows:

$$\Delta G = G_f - G_i \tag{1}$$

When $\Delta G<0$ the process goes from a high free energy state to a low free energy state which implies that the process is spontaneous and releases energy, so the process is a favored reaction and would happen if it could. On the other hand, if the $\Delta G>0$ the process is not spontaneous.

Gibbs energy takes into account the total energy or enthalpy (H), the total disorder or entropy (S), and the temperature (T), as shown in the following equation:

$$\Delta G = \Delta H - T\Delta S \tag{2}$$

Temperature plays a role in how much the entropy effects the change in (ΔG), if a process occurs in a high temperature environment then the entropy has a higher role in determining (ΔG) or how spontaneous the process occurs.

The role of thermodynamics laws in protein folding and stabilization is explained in [53]. In protein folding enthalpy changes from a high value in primary structures to a lower value in the 3D structures which makes ΔH negative, but entropy is also negative since it is higher in primary structures, so the temperature need to be low in order for the protein to fold. Thermodynamics are important since it is usually assumed that the protein's native state corresponds to its free energy minimum. This is tricky since it needs a global view and CA models work on the level of local rules. The point is that one should make the CA go towards this global energy model at each step in each cell.

D. General Steps

In this subsection, the proposed process of modeling proteins is summarized in some general steps. The input of the modeling process is the amino acid sequence of the target protein and the CA number of generations and the output of the algorithm is the final 3D CA representing the input protein. The process can be summarized as follows:

- Initialize the CA cells with the amino acid sequence and initialize the boundary with the water state
- Run CA rules for the number of generations
- at each step, if the spatial space needs extension then extend it
- Return the folded protein in the form of the input amino acids in their new positions in the 3D space

From an implementation point of view, it can be easier to define a 3D space that is double the size of the input chain so that the extensions are not needed often. The number of generations is assumed to be given in the input. Nevertheless, there are multiple methods that can be used to find an average number of generations that is suitable for the problem.

The output of the process shows the position of each Amino acid and how the hydrophobic core of the protein is created. The accurate results of the model means that CA is capable of modeling proteins without knowing the global view beforehand. It also means that proteins depend to some extent on some internal rules that results in its global behavior.

E. Yet More things to consider

Building a model that is based on CA starts with building a dataset, although protein information is generously available online, choosing the proteins or the benchmark needs to be accounted for. Many researchers depend on more than one benchmark to test their work. After applying the CA model, there must be a meaningful way to evaluate the model.

Although not using the CA model, some useful Literature compares between two or more distance matrices in terms of predicting protein attributes. For example, work done in [54] reports detailed results of protein structural classes prediction using Hamming, Euclidean, and Mahalanobis distances. Comparisons also exists in predicting protein subcellular Location [55].

VI. Conclusion and Future Work

The aim of many who work in the field of modeling natural phenomena is to add a step in discovering new things. Usually models try to capture the main properties and factors of the phenomenon to get results that are more meaningful in the sense of modeling one or more realistic attributes.

In this paper, protein modeling using cellular automata was discussed. Work in this area was analyzed and a suggested 3D model with heuristic rules was given. In addition, the general process of modeling proteins using cellular automata was discussed and alternative solutions to possible design issues were given.

The actual implementation of the design proposed in this paper is left for future work. In general, measuring the model effectiveness includes comparisons against data from laboratory experiments, but still the similarity between proteins is an interesting issue for further investigation.

References

[1] J. C. Wooley and H. S. Lin., *Catalyzing Inquiry at the Interface of Computing and Biology*. Washington (DC): National Academies Press (US), 2005, ch. Computational modeling and simulation as enablers for biological discovery, pp. 117–202.

[2] Y. Cai, J. Hu, Y. Li, and K. Chou, "Prediction of protein structural classes by a neural network method," *Internet Electronic Journal of Molecular Design*, vol. 1, no. 7, pp. 332–338, July 2002.

[3] H. Shen and K.-C. Chou, "Using optimized evidence-theoretic k-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types," *Biochemical and Biophysical Research Communications*, vol. 334, no. 1, pp. 288–292, 2005.

[4] X. Xiao, S. Shao, Y. Ding, Z. Huang, Y. Huang, and K.-C. Chou, "Using complexity measure factor to predict protein subcellular location," *Amino Acids*, vol. 28, no. 1, pp. 57–61, 2005.

[5] X. Zhou, X. Li, M. Li, and X. Lu, "Predicting protein functional class with the weighted segmented pseudo-amino acid composition moment vector," *MATCH Communications in Mathematical and in Computer Chemistry*, vol. 66, no. 1, pp. 445–462, 2011.

[6] K.-C. Chou and H.-B. Shen, "Predicting protein subcellular location by fusing multiple classifiers," *Journal of Cellular Biochemistry*, vol. 99, no. 2, pp. 517–527, Oct 2006.

[7] P. Sarkar, "A brief history of cellular automata," *ACM Comput. Surv.*, vol. 32, no. 1, pp. 80–107, 2000.

[8] D. Whitford, *Proteins: Structure and Function*. Wiley, 2005.

[9] E. Ferrada, "The amino acid alphabet and the architecture of the protein sequence-structure map. i. binary alphabets," *PLOS Computational Biology*, vol. 10, no. 12, pp. 1–20, December 2014.

[10] S. Wolfram, *A New Kind of Science*. Wolfram Media Inc., 2002.

[11] C. Burks and D. Farmer, "Towards modeling dna sequences as automata," *Physica 10D*, vol. 10, no. 1-2, pp. 157–167, 1984.

[12] G. Sirakoulis, I. Karafyllidis, C. Mizas, V. Mardiris, A. Thanailakis, and P. Tsalides, "A cellular automaton model for the study of dna sequence evolution," *Computers in Biology and Medicine*, vol. 33, no. 5, pp. 439–453, 2003.

[13] C. Mizas, G. Sirakoulis, V. Mardiris, I. Karafyllidis, N. Glykos, and R. Sandaltzopoulos, "Reconstruction of dna sequences using genetic algorithms and cellular automata: Towards mutation prediction?" *Biosystems*, vol. 92, no. 1, pp. 61–68, 2008.

[14] J. A. de Sales, M. L. Martins, and D. A. Stariolo, "Cellular automata model for gene networks," *Phys. Rev. E*, vol. 55, pp. 3262–3270, Mar 1997.

[15] X. Xiao, S. Shao, Y. Ding, and X. Chen, "Digital coding for amino acid based on cellular automata," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 5, Oct 2004, pp. 4593–4598.

[16] X. Xiao, S. Shao, Y. Ding, Z. Huang, and K.-C. Chou, "Using cellular automata images and pseudo amino acid composition to predict protein subcellular location," *Amino Acids*, vol. 30, no. 1, pp. 49–54, 2006.

[17] X. Xiao, P. Wang, and K.-C. Chou, "Gpcr-ca: A cellular automaton image approach for predicting g-protein-coupled receptor functional classes." *Journal of Computational Chemistry*, vol. 30, no. 9, pp. 1414–1423, 2008.

[18] X. Xiao and W. Ling, "Using cellular automata images to predict protein structural classes," in *Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on*, July 2007, pp. 346–349.

[19] X. Xiao, P. Wang, and K.-C. Chou, "Predicting protein structural classes with pseudo amino acid composition: An approach using geometric moments of cellular automaton image," *Journal of Theoretical Biology*, vol. 254, no. 3, pp. 691–696, 2008.

[20] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *PROTEINS: Structure, Function, and Genetics*, vol. 43, pp. 246–255, 2001.

[21] ——, "Some remarks on protein attribute prediction and pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 273, no. 1, pp. 236–247, 2011.

[22] X. Xiao, P. Wang, and K.-C. Chou, "Cellular automata and its applications in protein bioinformatics," *Current Protein and Peptide Science*, vol. 12, no. 6, pp. 508–519, 2011.

[23] J. Santos, P. Villot, and M. Dieguez, "Cellular automata for modeling protein folding using the hp model," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, June 2013, pp. 1586–1593.

[24] 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*, vol. 21, no. 11, pp. 823–845, 2014.

[25] P. Chopra and A. Bender, "Evolved cellular automata for protein secondary structure prediction imitate the determinants for folding observed in nature," *In Silico Biology*, vol. 7, no. 7, pp. 87–93, 2006.

[26] D. Takata, T. Isokawa, N. Matsui, and F. Peper, "Modeling chemical reactions in protein synthesis by a brownian cellular automaton," in

2013 First International Symposium on Computing and Networking, Dec 2013, pp. 527–532.

[27] A. L. A. Dalhoum, A. Ortega, and M. Alfonseca, "Cellular automata equivalent to d0l systems," in *3rd WSEAS International Conference on Systems Theory and Scientific Computation, Special Session on Cellular Automata and Applications*, 2003, pp. 15–17.

[28] A. Ortega, A. A. Dalhoum, and M. Alfonseca, "Grammatical evolution to design fractal curves with a given dimension," *IBM Journal of Research and Development*, vol. 47, no. 4, pp. 483–493, 2003.

[29] G. B. Danks, S. Stepney, and L. S. D. Caves, *Folding Protein-Like Structures with Open L-Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1100–1109.

[30] G. Danks, S. Stepney, and L. Caves, "Protein folding with stochastic l-systems," *Artificial Life XI*, pp. 150–157, 2008.

[31] G. B. Danks, S. Stepney, and L. S. D. Caves, *Cotranslational Protein Folding with L-systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 289–296.

[32] M. Gibson and E. Mjolsness, *Computational modeling of genetic and biochemical networks*. Cambridge MA: MIT Press, 2004, vol. 8, no. 1, ch. Modeling the Activity of Single Genes, pp. 3–48.

[33] P. Koehl, *Protein Structure Classification*. John Wiley & Sons, Inc., 2006, pp. 1–55.

[34] D. E. Krane and M. L. Raymer, *Fundamental concepts of bioinformatics*. San Francisco, London, Paris: B. Cummings, 2003.

[35] T. von der Haar, "Mathematical and computational modelling of ribosomal movement and protein synthesis: an overview," *Computational and Structural Biotechnology Journal*, vol. 1, no. 1, pp. 1–7, 2012.

[36] K. Zuse, "Rechnender raum," *Elektronische Datenverarbeitung*, vol. 8, pp. 336–344, 1967.

[37] ——, *Rechnender Raum*. Friedrich Vieweg & Sohn, Braunschweig, 1969.

[38] S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.*, vol. 55, pp. 601–644, Jul 1983.

[39] ——, "Universality and complexity in cellular automata," *Physica D: Nonlinear Phenomena*, vol. 10, no. 12, pp. 1–35, 1984.

[40] C. G. Langton, "Computation at the edge of chaos: Phase transitions and emergent computation," *Phys. D*, vol. 42, no. 1-3, pp. 12–37, 1990.

[41] Z. Aleksic, "Artificial life: growing complex systems," in *Complex Systems*, T. R. J. Bossomaier and D. G. Green, Eds. Cambridge University Press, 2000, pp. 91–126, cambridge Books Online.

[42] R. Ye and H. Li, "A novel image scrambling and watermarking scheme based on cellular automata," in *Electronic Commerce and Security, 2008 International Symposium on*, Aug 2008, pp. 938–941.

[43] A. Madain, A. Abu Dalhoum, H. Hiary, A. Ortega, and M. Alfonseca, "Audio scrambling technique based on cellular automata," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1803–1822, 2014.

[44] A. L. Abu Dalhoum, A. Madain, and H. Hiary, "Digital image scrambling based on elementary cellular automata," *Multimedia Tools and Applications*, pp. 1–16, 2015.

[45] M. Gardner, "Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life"," *Scientific American*, vol. 223, pp. 120–123, 1970.

[46] L. Orellana, "Protein dynamics studied by coarse-grained and atomistic theoretical approaches," Ph.D. dissertation, University of Barcelona, Department of Fundamental Physics, 2014.

[47] X. Xiao, S. Shao, Y. Ding, Z. Huang, X. Chen, and K.-C. Chou, "Using cellular automata to generate image representation for biological sequences," *Amino Acids*, vol. 28, no. 1, pp. 29–35, 2005.

[48] H. Fu and E. Mephu Nguifo, *Clustering Binary Codes to Express the Biochemical Properties of Amino Acids*. Boston, MA: Springer US, 2005, pp. 279–282.

[49] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, February 1962.

[50] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems Man and Cybernetics SMC3*, vol. 3, no. 6, pp. 610–621, 1973.

[51] H. Nishio, "How does the neighborhood affect the global behavior of cellular automata?" in *Cellular Automata*, ser. Lecture Notes in Computer Science, S. El Yacoubi, B. Chopard, and S. Bandini, Eds. Springer Berlin Heidelberg, 2006, vol. 4173, pp. 122–130.

[52] K.-C. Chou, "Structural bioinformatics and its impact to biomedical science," *Current Medicinal Chemistry*, vol. 11, no. 16, pp. 2105–2134, Aug 2004.

[53] A. Cooper, *Protein: A Comprehensive Treatise*. JAI Press Inc., 1999, vol. 2, ch. Thermodynamics of Protein Folding and Stability, pp. 217–270.

[54] K.-C. Chou and C.-T. Zhang, "Prediction of protein structural classes," *Critical Reviews in Biochemistry and Molecular Biology*, vol. 30, no. 4, pp. 275–349, 1995.

[55] K.-C. Chou and Y.-D. Cai, "Using functional domain composition and support vector machines for prediction of protein subcellular location," *The Journal of Biological Chemistry*, vol. 277, no. 48, pp. 45 765–45 769, November 2002.