A Novel Ab-initio Genetic-Based Approach for Protein Folding Prediction

Sergio R. Duarte
LISI
U. Nacional de Colombia
Bogotá, Colombia
srduartet@unal.edu.co

David C. Becerra
LISI
U. Nacional de Colombia
Bogotá, Colombia
dcbecerrar@unal.edu.co

Fernando Nino
LISI
U. Nacional de Colombia
Bogotá, Colombia
lfninov@unal.edu.co

Yoan J. Pinzón
LISI
U. Nacional de Colombia
Bogotá, Colombia
ypinzon@unal.edu.co

ABSTRACT

In this paper, a model based on genetic algorithms for protein folding prediction is proposed. The most important features of the proposed approach are: *i)* Heuristic secondary structure information is used in the initialization of the genetic algorithm; *ii*) An enhanced 3D spatial representation called cube-octahedron is used, also, an expansion technique is proposed in order to reduce the computational complexity and spatial constraints; *iii*) Data preprocessing of geometric features to characterize the cube-octahedron using twelve basic vectors to define the nodes. Additionally, biological information (torsion angles, bond angles and secondary structure conformations) was pre-processed through an analysis of all possible combinations of the basic vectors which satisfy the biological constrains defined by the spatial representation; and *iv*) Hashing techniques were used to improve the computational efficiency. The pre-processed information was stored in hash tables, which are intensively used by the genetic algorithm. Some experiments were carried out to validate the proposed model obtaining very promising results.

Categories and Subject Descriptors

J.3 [Life and Medical Sciences]: Biology and genetics

General Terms: Design, Algorithms

Keywords: Genetic Algorithms, Protein Folding Problem, Ab-Initio methods, 3D- FCC spatial representation

1. INTRODUCTION

Scientists have studied for decades the complex processes that determine the structure, properties and functionality of proteins. Nowadays, many of these investigation topics converge to the protein folding problem, and extremely challenging and complex issues still remain. The protein folding problem consists of determining the tertiary protein structure from its amino acid sequence; such three-dimensional conformation will allow the protein to carry out its function [1].

Understanding the complex process that determines the structure, properties and functionality of proteins is important because they carry out a wide variety of vitals functions developed in the living organisms; for example, proteins are involved in the catalysis of cellular chemical reactions, transport of molecules, transduction of signals, segregation of genetic material and production and use of energy [2].

A protein can be seen from four different structure levels. The primary protein structure is defined by the amino acid sequence. The secondary protein structure consists of local folding patterns, where the most common are α-helices and β-sheets. In contrast, the tertiary protein structure is the three-dimensional structure of the amino acids after proteins fold into its native state, at this level, proteins become functional. In the fourth protein structure, different polypeptide chains with tertiary structure interact to build a protein complex [3].

The laws of physics and the theory of evolution are the principles on which the techniques of protein structure prediction are based. Ab-initio methods use the laws of physics to predict a protein structure from its amino acid sequence. On the other hand, comparative methods rely on the folding similarity between the target protein and known protein structures [4].

Ab-initio computational approaches work at different levels of complexity, ranging from simple lattice models, where the amino acid residue of a protein is approximated as a point particle to all-atom models with explicit solvent. However, simpler models are useful because they can be exhaustively investigated with a modest amount of computer resources [5].

In this paper, a computational model for protein folding is proposed. The protein folding space is represented using a cubo-octahedral lattice, where the approach takes advantage of its geometric properties and biological feature representation [6]. Also, genetic algorithms are used to simulate the folding process because it has been shown that they are a good alternative in comparison with conventional Monte Carlo methods [7, 12]. Basis vector operations are used to guide possible location of an amino acid in the 3D-space during the protein structure search. In order to overcome inherent complexity of the problem, highly developed hashing techniques and data preprocessing are exploited. Additionally, second structure protein prediction is used as heuristic information. Furthermore, some experiments were designed to predict a set of proteins in order to evaluate the advantages and limitations of the model.

This paper is organized as follows. First, a biological background necessary to understand the computational simulation of the protein folding process is presented. Then, the proposed approach for protein folding prediction is described. Thus, the spatial representation, energy function and genetic algorithm characteristics are described. Subsequently, the experimental

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

framework and its results are discussed. Finally, some conclusions from this work are devised.

2. BACKGROUND

Proteins are formed from one or more amino acid sequences in a folding process in which a three-dimensional structure is obtained. This three-dimensional structure is highly important because it determines the function of the protein. In order to understand the structure and formation of proteins, it is convenient to consider four structural levels. Primary structure consists in the order of the amino acids in the sequence. Secondary structure that contains regular components like α-helices, β-sheets and β-turns, where these types of structures contribute to the stabilization of protein folding. Tertiary structure where the elements of secondary structure are folded forming an almost solid compact structure that is stabilized by weak interactions that involve polar groups as non-polar ones. Quaternary structure consists of several polypeptides chains with tertiary structure that are joined by weak connections – non-covalent – to form a protein complex [9].

Protein structure prediction methods may be classified in four main groups: *i*) comparative models; *ii*) fold recognition; *iii*) primary principle methods with database information; and *iv*) first principle methods without database information (ab-initio) [4].

Methods based on knowledge for predicting protein structures have been widely criticized because they do not provide information about the mechanisms and forces that direct the formation of such structures. On the other hand, since methods based on primary principles support their predictions on physical models, they can discriminate between correct and incorrect assumptions of the model and have a deeper understanding of protein folding mechanisms.

Ab-initio methods try to directly predict the three-dimensional structure without structural information of the target protein's family. Although such methods are very demanding at a computational level, they are extremely important because in some cases, it is not possible to find homologous structures related to a target protein; also, new structures that are discovered can have unique or different structural characteristic with respect to other proteins already reported.

A very important issue to consider in protein folding modeling is spatial representation, which refers to the space on which amino acid sequence is folded. Although protein folding in a three-dimensional space can consider all the possible degrees of freedom, the computational cost of this type of representation is extremely expensive; therefore, it is necessary to make some simplifications in the amino acid and space representation. Typically, specific spatial lattices or grids can be used to represent amino acid space and to allow folding having discrete degrees of freedom.

The folding model is another essential aspect to consider, since it determines the protein structure. Several folding models have been formulated to explain a protein folding given physical assumptions (Levinthal´s Paradox) [5, 9] and the biological conditions in which the protein is folded. Thus, protein folding can be conceived as the exploration of different protein structures in an energy landscape which has the form of a funnel and is highly irregular. 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.

In order to explore the energy landscape, several search methods can be applied. Such approaches differ in the way they modify the folding to produce changes in protein energy and to move in the search space. The exploration of the energy landscape is determined by an evaluation criterion based on an energy function that represents low-level interactions between amino acids. Typically, in ab-initio methods, the energy function is used to search for the protein native state. Particularly, in the genetic algorithm proposed in this work, the energy function represents the fitness of each individual.

3. PROPOSED APPROACH

The proposed model is based on the following four key features:

- the use of genetic algorithms for the evolution of protein folding populations;
- heuristic information based on secondary structure and a priori biological information;
- spatial representation with low computational cost and high biological significance;
- hash tables for efficient search and operations.

In Section 3.1, a discrete structure called cube-octahedron to represent the protein folding is described; in Section 3.2 the energy function used in the model is briefly explained; in the subsequent sections the main details of the genetic algorithm are presented.

Fig. 1 depicts the proposed protein folding prediction model. The process starts with the Pred2ary secondary structure prediction. Then that information is processed and used for the creation of the first population. Subsequently, the population is evolved using genetic operators and then the individuals for the following generation are selected. The previous step is repeated until a number of generations given by the user is reached or a stopping criterion is met. The process ends with the generation and visualization of a PDB file which contains the spatial information of the predicted model.

3.1 Spatial Representation

An architecture of residue packing called cube-octahedron that has 14 faces and 12 vertices developed by Raghunathan and Jernigan was used [6]. This grid is a face-centered cubic (FCC) lattice in which the connections between the 12 neighbors and its center have the same length (see Fig. 2).

Taking advantage of the geometry of the cube-octahedron, it is possible to define 12 vectors that will determine the possible discrete points of the space where amino acids may be located. Thus, such vectors form 0°, 60°, 90° or 120° bond angles between them; however, it is important to notice that only 90° and 120° angles are biologically allowed due to steric constraints. Consequently, three of such vectors can only define certain valid torsions angles, specifically, 54.7°, 109.5°, 125.3° and 180°.

Notice that the FCC lattice is formed by joining each two cube-octahedron units through six points on the surface of any layer of expansion as shown in Fig. 3. Thus, starting with a cube-octahedron, the lattice can be expanded in a radial manner by adding a new layer of cube-octahedrons around it, with the corresponding geometrical constraints. Hence, as shown in [8],

the number of additional points N for a particular layer L is given by:

$$N = 10\,L^2 + 2 \quad (1)$$

Thus, the number of points in an L layer FCC lattice is given by:

$$N = \frac{10L^3 + 15L^2 + 11L}{3} + 1 \quad (2)$$

In other words, storing the lattice information is computationally expensive because the number of nodes grows in a cubic way with the number of layers. Thus, in order to consider all possible protein foldings without geometric constraints, the FCC lattice should have a number of layers around the number of amino acids in the protein sequence. In order to avoid this problem, which usually makes most approaches to consider a very limited lattice size, in this work, no structure information for the lattice is directly stored; instead, only the 3D discrete points, where the amino acids are located, will be stored. Besides, some tables that store the vector information that specify how to reach a neighbor point from the current location, according to the biological constraints, (only 90° and 120° bond angles are allowed). Fig. 4 shows a vector diagram that illustrates how to reach the neighbor points from the current amino acid location; in Fig. 4 each vector was represented with a different color (the opposite vectors were not included in the figure). Clearly, the neighbor points can be easily computed by a vector sum operation using the information in Table 2, where the first four columns correspond to 120° bond angles and the last two define 90° bond angles.

The proposed implementation based on vectors uses some preprocessed data related to the spatial representation of the protein folding problem thus reducing to a great extent the computational complexity of the algorithm.

3.2 Energy Function

The next energy function is assumed for the protein folding [9]:

$$E = \sum_{ij} \Delta_{ij} B(\eta_i, \eta_j) \quad (3)$$

In this function, the variable Δ_{ij} can take either 1 or 0. If the amino acid i is located at a unit of distance from amino acid j and both of them are not consecutive in the sequence, variable Δij takes the value 1, otherwise it takes the value 0. In Eq. (3), the weight $B(\eta_i, \eta_j)$ refers to the energetic value that correspond to the interaction between each possible pair of essential amino acids. Therefore, a 20x20 matrix is provided.

In this work, two potential matrices are used; the first one was developed by Berrera *et. al.* [10] – this matrix will be referred as the BMF matrix–. The other matrix obtained by Miyazawa and Jerningan [11] is widely use in similar works – this matrix will be referred as the MJ matrix.

To compute the energy function, the amino acid sequence is scanned and for every amino acid, each one of the twelve basic vectors is checked in order to find another amino acid that is not consecutive in the sequence. Then, the energy is computed as the sum of the corresponding energetic values in the potential matrix for the neighbor amino acids. The final value is divided by two, since a same energetic value is considered twice in the computation.

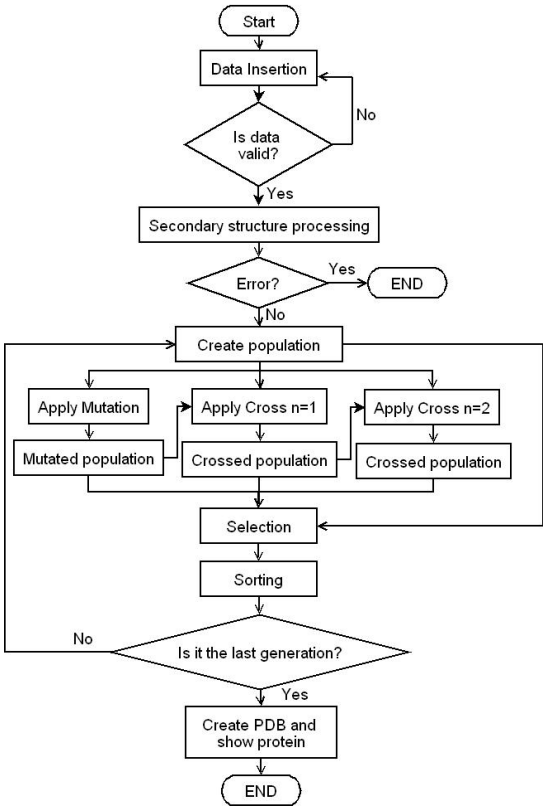

Figure 1. Flow diagram of the proposed protein folding model

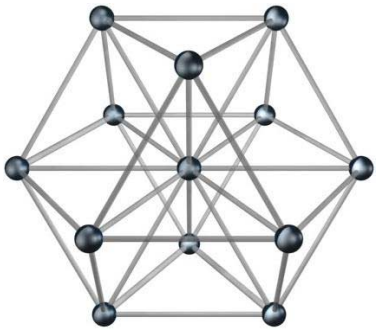

Figure 2. Cube-octahedron unit

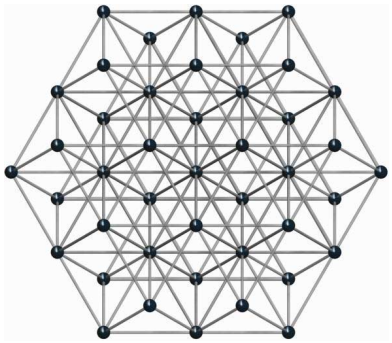

Figure 3. Face-centered Cube-octahedron lattice

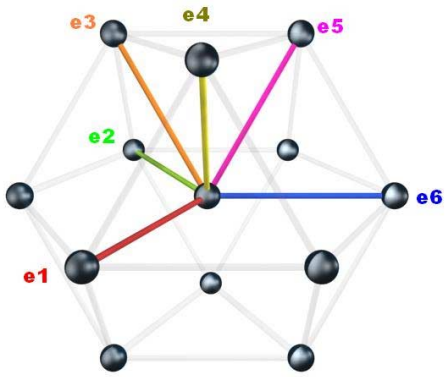

Figure 4. Face-Centered Cube-Vector Representation

Table 1. Vector Definition: Base on the geometry on octahedron and tetrahedron, the spatial representations for each vector defined by the indexing scheme V = (i, j, k) were found.

Vector	Value {*x, y, z*}
e1	{-0.5,0,0.866}
-e1	{0.5,0,-0.866}
e2	{-0.5,0,-0.866}
-e2	{0.5,0,0.866}
e3	{-0.5,0.8165,-0.288}
-e3	{0.5,-0.8165,0.288}
e4	{0,0.8165,0.577}
-e4	{0,-0.8165,-0.577}
e5	{0.5,0.8165,-0.288}
-e5	{-0.5,-0.8165,0.288}
e6	{1,0,0}
-e6	{-1,0,0}

Table 2. Vectors with valid bond angles: If e6 is the vector used to reach amino acid i, vectors –e1,-e2,-e3, -e4, e4 and e5 are candidates to be the next vector.

Input Vector	Bond Angle 120^0				Bond Angle 90^0	
-e6	e1	e2	e3	-e5	e4	-e4
-e5	e1	-e3	-e4	-e6	e2	-e2
-e4	-e1	e2	-e3	-e5	e6	-e6
-e3	-e2	-e4	-e5	e6	e1	-e1
-e2	e1	-e3	e4	e6	e5	-e5
-e1	e2	-e4	e5	e6	e3	-e3
e6	-e1	-e2	-e3	e5	e4	-e4
e1	-e2	e4	-e5	-e6	e3	-e3
e2	-e1	e3	-e4	-e6	e5	-e5
e3	e2	e4	e5	-e6	e1	-e1
e4	e1	-e2	e3	e5	e6	-e6
e5	-e1	e3	e4	e6	e2	-e2

3.3 Hashing

The use of hashing is highly important in the proposed approach because it drastically reduces the computational time, as a consequence of the efficiency offered by hash tables in search, updating and erasing processes. Additionally, hash tables have a constant size, depending on the protein sequence length, thus avoiding the hash table degeneration while keeping its efficiency. The hash tables used in this work are described next.

3.3.1 Amino acid Hash table

This hash table stores amino acid 3D coordinates; nonetheless, the main purpose of this table is to be able to determine if a 3D coordinate is already occupied by an amino acid. Consequently, this hash table is also used to calculate the energy function for each individual. Having access to this information in constant time allows designing very efficient genetic operators to modify individuals.

Each individual is associated one of these hash tables. Each hash table has an entry for each amino acid in the sequence. The key for this hash table corresponds to the concatenation of the Cartesian coordinates occupied by the amino acid. The key in these hash tables is formed by the concatenation of the coordinates seen as strings and separated by commas.

3.3.2 Secondary structure hash tables

Three hash tables are used to provide the information for all the possible combinations of three vectors that form secondary level structures. These hash tables were obtained by preprocessing bond and torsion angles information for all possible vector conformations with biological significance; for this reason, it is not necessary to calculate bond and torsion angles at run-time, thus reducing significantly computation time.

Therefore, there are two hash tables that contain the set of necessary vectors to form α-helices. This set of three vectors form alternate 120° and 90° bond angles with a torsion angle of 55° [6]. The vector combinations that form β-sheets are stored in another hash table that has 120° bond angles and a 180° torsion angle.

The key for each hash table corresponds to the string obtained as the concatenation of two vectors separated by commas and the key-value is the third vector necessary to complete the set of vectors with the bond and torsion angles required to complete the desired protein secondary structure.

3.4 The proposed genetic algorithm

Genetic algorithms are systematic methods based on biological evolution used to solve search and optimization problems. A population of individuals that typically represent the solutions to a particular problem is evolved, based on the survival of the fittest principle and introducing genetic variation in the individuals of the population. Thus individuals are encoded as chromosomes, and genetic operators are applied over them to introduce changes that allow an exploration of the solution search space. The individuals compete among them, and the environment that consists of other possible solutions produces a selective pressure over the population to favor the survival of the fittest individuals. Genetic information in the chromosomes will be preserved and transmitted to the next generations [13, 14]. A genetic algorithm can be described in a general way by the following pseudo code:

```
Begin Genetic Algorithm
  gc:=0  { generation counter }
  Initialize population P(gc)
  Evaluate population P(gc)
  while not stopping criterion met do
     gc:=gc+1
     Select P(gc) from P(gc-1)
     Mutate P(gc)
     Crossover P(gc)
     Evaluate P(gc)
  end while
end Genetic Algorithm
```

3.4.1 Individual Representation

The chromosome is a data array of the same length as the amino acid sequence. Each location of the array contains two integer numbers and three real values. The first integer identifies one of the vectors defined in Table 1. The second integer represents a bond angle between the current vector and its previous vector. On the other hand, the three real values correspond to a 3D coordinate (*x, y, z*) where an amino acid is located. It is important to notice that the coordinates and the torsion angle do not belong to the chromosome since genetic operators are not actually applied over them. Such coordinates are computed during the chromosome evaluation phase and it is convenient to store them together for implementation purposes.

The range for the real numbers that represent the points in the three-dimensional space is defined as follows: (-*n*, +*n*) for the *x* coordinate, (-0.8165×*n*, +0.8165×*n*) for the *y* coordinate and (-0.866×*n*, +0.866×*n*) for the *z* coordinate, where *n* is the length of the amino acid sequence. These intervals were computed by considering the longest length of a folding for a protein of length *n* (see Table 1).

A chromosome represents a particular folding for a given protein sequence, and it contains the spatial information of each amino acid and other information necessary to determine the protein folding.

The individual representation is quite convenient because all the necessary information to reproduce the three-dimensional structure of the protein is stored in a very simple data structure that maintains the flexibility of the model without adding computational constraints and is very efficient with respect to the use of computational resources.

3.4.2 The initial population

A heuristics based on secondary structure information is used to generate the initial population. This heuristics helps the search process, thus increasing the precision and quality of the results. Moreover, the protein folding is modeled in a more realistic manner, because the formation of a secondary structure is an important step previous to a more complex process that leads to the native structure of the protein.

The generation of the secondary structure information is based on the method Pred2ary, developed by Chandonia and Karplus [15]. Pre2dary is implemented as a feedforward neural network that receives an amino acid sequence as input and outputs a secondary structure prediction. Thus, for each amino acid, a classification of helix, sheet or coil with its respective probabilities is obtained. This method reported a near to 79% accuracy [15].

The information of this secondary structure prediction process is then used to create the initial population. First, a set of two vectors using a uniform random distribution is generated, keeping in mind the restrictions given in the Table 2; the main objective of these two vectors is to create the first key necessary to get from the secondary structure hash table the next vector that will guarantee valid secondary structure formation.

Based on the predicted probabilities for each amino acid, they are classified as α-helix, β-sheet or coil – where coil is an unknown structure. Given the secondary structure classification, for each amino acid the next valid vector from the corresponding hash table is chosen or a random process is applied if the amino acid is classified as coil.

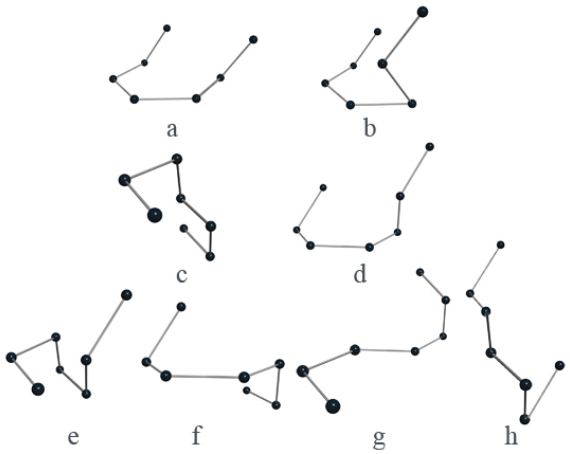

Figure 5. Genetic operators: a. Initial individual before mutation**; b**. Individual after mutation (amino acid 5)**; c-d.** Parents for crossover operators; **e-f.** Children after crossover *n* = 1 (amino acid 5); **g-h.** Children after crossover *n* = 2 (amino acid 3 and 6)

3.4.3 Fitness Function

Given that an individual represents a protein folding, the fitness function is equal to the energy function based on amino acid interaction described in Section 3.2.

3.4.4 Genetic Operators

In the genetic algorithm, tournament selection is used. In addition, genetic variation is introduced by particular crossover and mutation operators, which will be described next (see Fig. 5).

3.4.4.1 Mutation Operator

The mutation genetic operator is very important to explore the search space and to maintain the diversity between individuals. In general, by changing the local direction of an amino acid vector, drastic effects on the protein structure are produced.

A random number uniformly between 0 and *n* - where *n* is the number of amino acids in the sequence- is generated, and then the vector located in that position is replaced with a new vector randomly chosen from Table 2. This process ensures that a valid vector is generated with respect to its previous and contiguous neighbors. Then, the (*x, y, z*) coordinates of the amino acids that are located after that position are recomputed (see Fig. 5). The amino acid hash table is used to verify the absence of collisions and to know if the result of the mutation process is a valid individual. This guarantees that each amino acid occupies a unique Cartesian point in the space.

3.4.4.2 Crossover

N-point crossover operator is considered, for *n*=1 and *n*=2. The main objective of the first crossover is to exploit local minima in order to find better individuals; on the other hand, the main objective of the second crossover is to allow a better exploration of the search space.

In one-point crossover, a random number between 0 and the length of the amino acid sequence is generated, then the set of

vectors from this position to the last position of each individual are swapped. In order to exchange the information of each individual, it is necessary to check the vectors in Table 2 to ensure that the new set of neighbor vectors do not break the bond angle constraints. Using the new set of swapped vectors, the new positions of the amino acids for each individual are recalculated. If the new points are valid, the two new individuals are kept to be selected as part of the new population (see Fig. 4).

On the other hand, in two-point crossover, two random numbers between (0, n/2) and (n/2, n) are generated. The set of vectors in the range between the first random number and the second random number for each individual are swapped. In order to exchange the information of each individual, again, it is necessary to check the vectors in the Table 2 in order to ensure that the new set of neighbor vectors of the two insertion positions do not break the bond angle constraints. Using the new set of vectors exchanged, the new positions of the amino acids for each individual are recalculated. If the new points collide, the process is repeated using another possible vector from Table 2 until a new valid individual is generated or all the possible vectors have been checked (see Fig. 4).

It is important to notice that the crossover operator produces unfeasible protein foldings than mutation, which is better observed as the protein attains a more globular structure.

3.5 Computational Complexity

The use of hash tables and data preprocessing significantly speeds up the proposed algorithm. Each iteration of the algorithm runs in linear time in the length of the amino acid sequence. Specifically, the fitness function is computed in time O(n), corresponding to checking the twelve basic vectors at each amino acid in the sequence in order to compute the energy function. On the other hand, the genetic operators can be computed in O(n), corresponding to the worst case n evaluations performed in order to validate the spatial constrains for each amino acid in the sequence.

4. EXPERIMENTATION

The main objectives of the experiments that were carried out were: *i*) To evaluate the prediction accuracy of the proposed approach; *ii*) To analyze and compare the results obtained using the MJ potential matrix and the BMF potential matrix; *iii*) To quantify the conservation of the secondary structure in the predicted protein foldings produced by the proposed model with respect to the Pred2ary prediction; *iv*) To study the behavior of genetic algorithms as a search method in the energy landscape.

4.1 Experimental framework

The set of proteins used in the experiments were obtained from CASP7 and they corresponded to the category of free modeling. CASP (Critical Assessment of Techniques for Protein Structure Prediction) establishes the current state of the art in protein structure prediction. It identifies what progress has been made based on the prediction of a set of known structures using different approaches and techniques of research groups worldwide [17].

In this work, CASP is considered as a good way to evaluate the proposed approach and compare it to other techniques, since CASP contains the state of the art in PSP problems and it reflects the biological significance by ranking the solutions with respect to other approaches. The main characteristics of the proteins used in the experiments are listed in Tables 3 and 4.

In order to accomplish the first objective, the measure given by the CASP-LGA Server, particularly the RMSD (Root-mean-square deviation) measure was used [16]. This measure is one of the measures used to rank participant models in CASP.

Tests were performed using several runs of the model with different parameters for each potential matrix and the results obtained for each one of them were evaluated.

Tables 5 and 6 show the results of all the experiments carried out. In the genetic algorithm, populations of 50 individuals were evolved for a maximum of 20000 generations.

For each experiment, the probability of mutation was 0.7 and 0.2 for each type of crossover. It is important to emphasize that these probabilities were necessary to maintain the population diversity, thus avoiding premature convergence. This can be explained by the low probability of crossover to create feasible individuals as opposed to the mutation operator.

It is also important to emphasize that the preservation of secondary structure was measured taking as a reference the prediction done by the preprocessing with the method Pred2ary. Such measure is based on a correspondence for each amino acid of the predicted secondary structure and the Pred2ary prediction.

The behavior of the genetic algorithm with respect to the evolution of the populations and its energy significance was plotted in order to understand the search process on the energy landscape. Besides, the behavior of the RMSD measures was evaluated in order to study the biologic significance of the evolved populations.

4.2 Experimental Results

Analyzing the results of the experiments with respect to the results reported in CASP7 [17], it can be stated that the proposed model obtained good predictions; the results were ranked in the best third of the reported predictions, although the approach is classified in the free modeling (FM) category, where the use of protein database information is not allowed.

It should be stressed that an advantage of the proposed approach is that it generates not only one protein folding but, in general, it may produce a set of foldings.

Even though, a solution set corresponds to neighbors of the possible native state (their energy and RMSD values are similar), the individuals have different structures. As an example, in Fig. 9 reports a box plot of the family of foldings corresponding to the last evolved generation. The absence of outliers in Fig. 9 means that the final set of solutions is near to one possible native state of the protein.

In addition, the experiment shows that there is a strong relation between their accuracy and their energy values in the predicted foldings. In Fig. 7 and 8, it is clear that lower energy values imply better predicted models measured by the RMSD value. Although not all the energetic interactions are included and strong simplifications exist in biological conditions of the protein environment, the genetic algorithm was able to obtain accurate predictions.

The experiments carried out using the two different potential matrices produced similar results (see Tables 5 and 6). However, the best results were obtained using the BMF potential matrix. It

is important to mention that BMF Matrix was computed more recently than MJ potential matrix and it was tested using CASP 4 experiments [10].

On the other hand, regarding the conservation of secondary structure information, it can be stated that the proposed model conserved a good proportion of information predicted by Pred2ary (see Table 7). Nonetheless, it is clear that the α-helix conformation is more preserved than the β-sheets structure. These results show that the proposed model takes into account heuristic information provided by the pre-processed data and takes advantage of the heuristics to solve more accurately the protein folding problem (see Fig. 7, 8 and Table 7).

From the genetic algorithm chart (Fig. 6), it is clear that the population set evolves through the generations towards populations with better energy values approaching the native state of the protein. Sample prediction obtained by the proposed approach is shown in Fig. 10. This figure was obtained using the molecule viewer called JMOL.

Table 3: Protein 2J6A description

Protein's name	TRM112 (2j6a)
Organism name	Saccharomyces cerevisiae
Amino acid length	135
Amino acid sequence	MKFLTTNFLKCSVKACDTSNDNFPLQYDGSKCQLVQD ESIEFNPEFLLNIVDRVDWPAVLTVAAELGNNALPPTKP SFPSSIQELTDDDMAILNDLHTLLLQTSIAEGEMKCRNC GHIYYIKNGIPNLLLPPHLV
Experimental method	Ray structure

Table 4: Protein 2HFQ description

Protein's name	NeT5 (2hfq)
Organism name	Nitrosomonas europaea
Amino acid length	85
Amino acid sequence	MQIHVYDTYVKAKDGHVMHFDVFTDVRDDKKAIEFA KQWLSSIGEEGATVTSEECRFCHSQKAPDEVIEAIKQN GYFIYKMEGCN
Experimental method	NMR

Table 5. Protein 2j6a (135 residues)

P. Matrix	Generation	Population	Energy	RMSD
BMF	20000	50	-249.193	15.470
MJ	20000	50	-815.82	16.747

Table 6. Protein 2hfq (85 residues)

P. Matrix	Generation	Population	Energy	RMSD
BMF	20000	50	-158.97	12.151
MJ	20000	50	-543.17	12.243

Table 7: Secondary structure conservation

		Helix		Sheet		Coil	
Protein 2j6a	Predicted	41	100%	14	100%	77	100%
	Conservation	19	46.3%	2	14.2%	48	62.3%
Protein 2hfq	Predicted	21	100%	22	100%	39	100%
	Conservation	7	33.3%	1	4.5%	23	58.9%

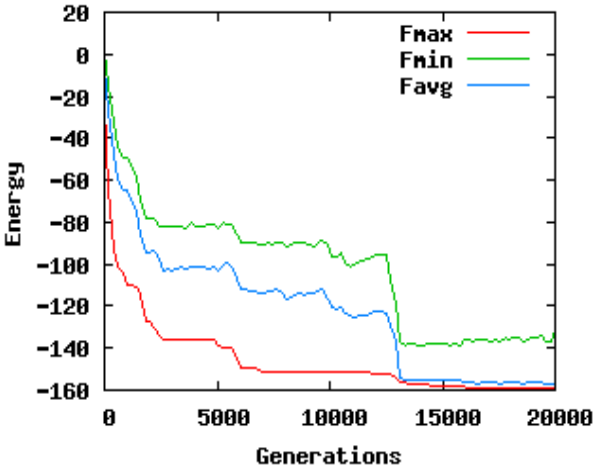

Figure 6: Genetic algorithm analysis using BM potential matrix for 2hfq protein

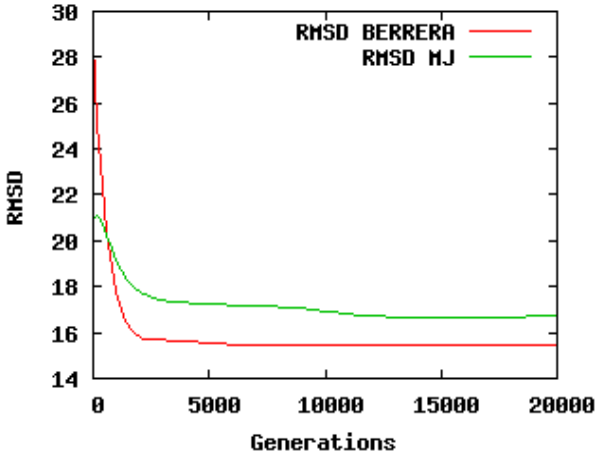

Figure 7: RMSD analysis using 2j6a protein

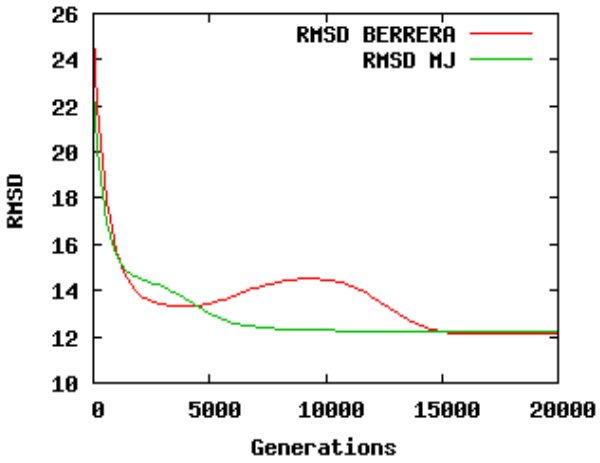

Figure 8: RMSD analysis using 2hfq protein

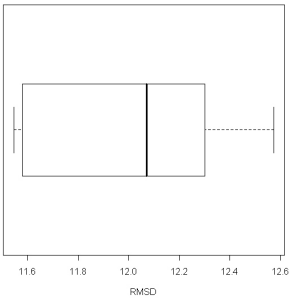

Figure 9: 2HFQ protein last evolved population box plot

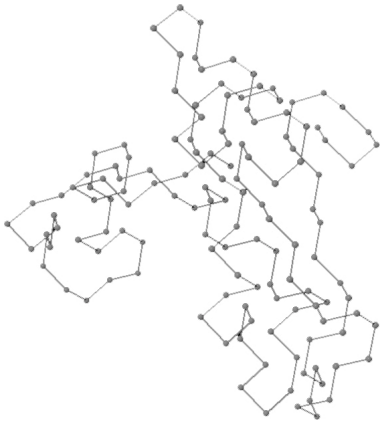

Figure 10: 2J6A tertiary structure prediction

5. CONCLUSIONS AND FURTHER WORK

In this work, an ab-initio genetic-based protein folding prediction approach was developed. Even though ab-initio predictions are less accurate than comparative methods, they have the advantage of providing a better understanding of protein folding principles. Analyzing the results, it is clear that the proposed model is a good protein folding predictor, although, more work needs to be done to improve the quality of the energy function and the spatial representation.

The cube-octahedron is a computational feasible spatial representation that can be implemented using very efficient techniques and data structures. Additionally, this spatial representation has biological significance allowing to model secondary structures and some spatial characteristic of the protein such as torsion and bond angles. The construction of the spatial model from the radial expansion can be widely improved using the proposed representation, *i.e.*, infinite radial expansion through each node in the cube-octahedron applying a set of basic vectors.

The implementation developed in this research drastically decreased the algorithmic complexity of the protein folding construction and search. Specifically, strategies such as data preprocessing, hashing techniques and spatial vector representation made possible a highly efficient model in terms of time and computational resources.

The use of hash tables provides an excellent computational technique to model amino acid spatial occupancy, because the number of collisions are reduced to zero and the insertion, erasing and search are very efficient.

Secondary structure information is fundamental for the accuracy of the predicted models, given the importance of those conformations in the protein folding process present in nature.

Though the results obtained in this work were very encouraging, further exploration is necessary. The use of hash tables for efficient search, updating, and erasing operation to speed computation of secondary structure prediction conformance, bump checking, and fitness calculation has shown to be an excellent alternative to speed up the algorithm. Future work will focus on the implementation of all-atom three-dimensional coordinates to represent the polypeptide chain, the use of a more complex potential energy function, and the use of this research to predict 20 polypeptides used by the Colombia Institute of Immunology (FIDIC) about their research on malaria vaccines.

7. ACKNOWLEDGEMENTS

The authors would like to express their gratitude to IBUN (Bioinformatics research group at National University of Colombia) and FIDIC (Colombia Institute of Immunology) for all their helpful comments and support.

8. REFERENCES

[1] D. L. Nelson and M. Cox. Lehninger. Principles of Biochesmistry, Palgrave Macmillan, Chapter 1, 2004.

[2] R. K. Murray, D. K. Granner, P. A. Mayes, and V. W. Rodwell. Harper's Biochemistry, Appleton & Lange, 4:29-51, 1996.

[3] F. Allen et al. Blue Gene: A vision for protein science using a petaflop supercomputer, IBM Systems Journal, 40 (2): 310-327, 2001.

[4] A. Fiser and A. Sali, Comparative protein structure modelling, Pels Family Center for Biochemistry and Structural Biology, The Rockefeller University.

[5] C.A. Floudas, H.K. Fung, S. R. McAllister, M. Mönnigmann, R. Rajgaria. Advances in protein structure prediction and de novo protein design: A review. Chem. Eng. Sc., 28 (11): 2109-2129, 2004.

[6] G. Raghunathan and R. L. Jernigan, Ideal architecture of residue packing and its observation in protein structures. Protein Sci., 6:2072-2083, 1997.

[7] R. Unger and J. Moult, Genetic Algorithms for Protein Folding Simulations, J . Mol. Bid., 231: 75-81, 1993.

[8] J. Kappraff. Connections. The geometric bridge between art and science. McGraw-Hill, 1991.

[9] A. Sali, E. Shakhnovich, M. Karplus, Kinetics of Protein Folding: A lattice model study of the requirements for folding to the native state, J. Mol. Biol., 235, 1614-1636, 1994.

[10] M. Berrera, H. Molinari, F. Fogolari. Amino acid empirical contact energy definitions for fold recognition in the space of contact maps. BMC Bioinformatics 4:8, 2003.

[11] S. Miyazawa, R. Jernigan , Residue–Residue Potentials with a favorable Contact Pair term and an Unfavorable High Packing Density Term, for Simulation and Threading. J. Mol. Biol., 256:623-644, 1996.

[12] A. A. Rabow, H. A. Scheraga , Improved genetic algorithm for the protein folding problem by use of a cartersian combination operador, Protein Sci., 5: 1800-1815, 1996 .

[13] D. Whitley. An overview of evolutionary Algortihms: Practical Issues and Common Pitfalls, Journal of Information and Software Technology 43:817-831, 2001.

[14] J. Shapiro. Genetic Algorithms in Machine Learning: Machine Learning and Its Applications, Advanced Lectures. 146-168, 2001.

[15] J. M Chandonia and M. Karplus, Neural networks for secondary structure and structural class predictions. Protein Sci., 4:275-285, 1995

[16] A. Zemla. LGA: A method for finding 3D similarities in protein structures, Nucleic Acids Res., 31:3370-3374, 2003.

[17] Critical Assessment of Techniques for Protein Structure Prediction, Asilomar Conference Center, Pacific Grove, November 26-30, 2006: http://predictioncenter.org/casp7.