Ab-initio Protein Folding using Molecular Dynamics and a Simplified Off-lattice Model

César M.V. Benítez[1] and Heitor Silvério Lopes[1]

[1]*Bioinformatics Laboratory*
Federal University of Technology - Paraná
Curitiba, Brazil
Emails: cesarvargasb@gmail.com, hslopes@utfpr.edu.br

***Abstract*—This paper presents an application of Molecular Dynamics (MD) to the Protein Folding Problem (PFP) using a simplified off-lattice model of proteins (3D-AB). To the best of our knowledge, this paper presents the first application of MD to the PFP using globular protein sequences represented with the 3D-AB model. The methodology is explained in details. Five synthetic sequences and four real globular proteins sequences were used for testing the approach. Results show that the method is capable of creating realistic folds of the proteins, displaying biological features such as hydrophobic core formation and protein breathing. Future works will investigate more efficient parallel processing methods and the creation of new benchmarks.**

***Keywords*-Molecular Dynamics; protein folding problem; 3D-AB model**

I. Introduction

Proteins are the basic structures of all living beings because they are responsible for performing important life-maintenance functions, such as: structural (e.g. fibrinogen); hormonal (e.g. insulin); defense (fibrinogen); enzymatic (e.g. amylase); and transport (e.g. hemoglobin), among others. Finding the proteins that make up an organism and understanding their function is the foundation of Molecular Biology. They are polymers composed by a chain of amino acids (also called residues) that are linked together by means of peptide bonds, and are synthesized in the ribosome of cells following a template given by the messenger RNA (mRNA).

The specific biological function of a protein is ultimately determined by its unique three-dimensional structure (also known as the native conformation), to which it folds under physiological conditions. This process is known as the protein folding. Due to its great importance for Medicine and Biochemistry, researchers have been focusing on the study of this process. Therefore, acquiring more knowledge about the three-dimensional structure of proteins and, consequently, about its functionality, is an important issue, since such knowledge can be used extensively in the development of new drugs with specific functionality.

A great motivation for studying the protein folding is the fact that ill-formed proteins can be completely inactive or even harmful to the organism. Furthermore, several diseases are believed to be the result of the accumulation of ill-formed proteins, such as Alzheimer's disease, cystic fibrosis, Huntington's disease and some types of cancer [1]–[3].

In recent years, a large number of new proteins have been discovered, thanks to the several genome sequencing projects being conducted in the world. However, only a small amount of such proteins have its 3-dimensional structure known. For instance, the
UniProtKB/TrEMBL repository of protein sequences has currently around 27 million records (as in November/2012), and the Protein Data Bank – PDB [4] has the structure of only 86,344 proteins

Benítez, C.M.V., Lopes, H.S. Ab-Initio Protein Folding Using Molecular Dynamics and a Simplified Off-Lattice Model. Journal of Bionanoscience, v. 7, p. 391-402, 2013. DOI:10.1166/jbns.2013.1153

(as in November/2012). This fact is due to the cost and difficulty in unveiling the structure of proteins, from the biochemical point of view. Here, the Computer Science can play an important role, developing computational models and approaches for the Protein Folding Problem (PFP). Due to the computational complexity, computational models that take into account every atom of the protein macromolecule are not feasible. Consequently, several simplified models for proteins have been proposed by using some biochemical properties, which can display some interesting features of the protein folding process and the protein structure.

II. The Protein Folding Problem

The protein folding is the process by which polypeptide chains are transformed into compact structures that perform biological functions. As mentioned before, under physiological conditions, the most stable three-dimensional structure is called the native conformation and actually allows a protein to perform its function.

In vitro experiments carried out by Anfinsen and colleagues [5] show that proteins can be denaturated by modifications in the environment where they are. Most proteins can be denaturated by temperature and pH changes, affecting weak interactions between residues (i.e.: hydrogen bonds). During the denaturation process, proteins lose their native shape and, consequently, their function. Anfinsen showed that some denatured (misfolded or unfold) proteins can refold into their native conformation. However, the spontaneous refolding only occurs for single-domain proteins. Failure to fold into the intended three-dimensional conformation usually leads to proteins with different properties that simply become inactive. In the worst case, such misfolded (incorrectly folded) proteins can be harmful to the organism.

A better understand of the protein folding process could help to: (a) accelerate drug discovery by replacing slow, expensive structural biology experiments with faster computational simulations, and (b) infer protein function from genome sequences. With the fast exponential growth of experimentally determined structures available in the PDB, the PFP has become as much a problem of inference and machine learning as it is of protein physics [6]. For instance, new theories have been developed in protein engineering [7] and structure-based drug design [8], [9].

Despite the considerable theoretical and experimental effort expended to study the protein folding process, there is not yet a detailed description of the mechanisms that govern the folding process.

Although the concept of the folding process arose in the field of Molecular Biology, this problem is clearly interdisciplinary, requiring support of many knowledge areas, and it is considered to be one of the most important open challenges in Biology and Bioinformatics [10]. In contemporary Computational Biology, there are two important problems regarding the folding of proteins. The first problem is to predict the protein structure (conformation) from sequence (primary structure) called the Protein Structure Prediction (PSP). The second one is to predict protein folding pathways, which consists in determining the folding sequence of events which lead from the primary structure of a protein (its linear sequence of amino acids) to its native structure. This is the Protein Folding Problem (PFP). It is commonly found in the literature both problems being referred as the PFP [11]. In this work we consider exclusively the second problem.

There are some computational methods to deal with the folding problem. However, the Molecular Dynamics (MD) approach (including all its variants) is the only computational methodology that really provides a time-dependent analysis of a system in Molecular Biology and, consequently, it can be employed to solve the PFP [12].

Several computational models have been proposed for representing protein structures with different levels of complexity and, consequently, computational feasibility. Ideally, both the protein and the solvent should be represented at the atomic level because this approach is the closest to experiments [13]. However, the simulation of computational models that take into account all the atoms of a protein is frequently unfeasible due to the multidimensionality of the system ($> 10^4$

degrees of freedom) [12], even with the most powerful computational resources (in nature, proteins can rapidly and reliably find their way into well-defined folded configurations). Generally, atomistic simulations of real-size proteins are usually limited to unfolding the native conformation of the proteins followed by refolding [13]. The dimensionality of a system containing the protein and the solvent can be reduced when the solvent is treated implicitly and a reduced coarse-grained model of proteins is used. In this scenery, several reduced (mesoscopic) models have been proposed. Although such reduced models are not realistic, their simulation can show some characteristics of real proteins. The success of reduced representations in reproducing several aspects of the folding process is due to the fact that this process has generally evolved to satisfy the principle of minimal frustration [14]. Computational studies of reduced models have provided several valuable insights into the folding process [15], [16].

The prediction of the structure of a protein is modelled as the minimization of the corresponding free-energy, following the Anfinsen's Thermodynamic Hypothesis. It is also known that the native conformation of a protein represents the folding state with minimal free-energy. A computational model that obeys this principle must have the following features:

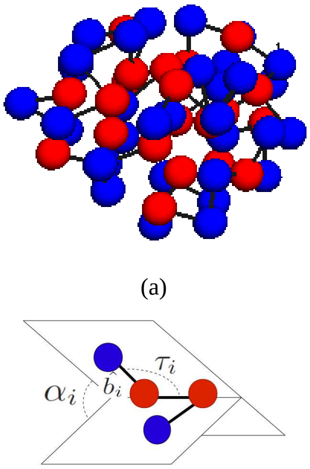
- a model of the protein, defined by a set of entities representing atoms and connections among them;
- a set of rules defining the possible conformations of the protein;
- a computationally feasible function for evaluating the free-energy of each possible conformation.

Whereas the protein structure prediction problem (PSP) is widely acknowledged as an open problem, the protein folding problem (PFP) has received little attention. It is important to note that the ability to predict the folding pathways can improve methods for predicting the native structure of proteins.

The total number of possible conformations of a protein is huge and it would take an astronomical length of time to find the native conformation by means of exhaustive search of the whole conformational space [17]. Nowadays, it is known that the folding process does not include mandatory steps between unfolded and folded states, but a search of many accessible conformations [17]. A possible approach to enumerate folding pathways is to start with an unfolded protein and consider the various possibilities for the protein to fold. The protein folds from a denatured conformation with a high free energy to its native conformation, following an energy landscape [18]. Notice that the free energy barrier between the native state and the multiple denature conformations is huge.

III. The AB Off-lattice Model

The AB off-lattice model was introduced by [19] to represent protein structures. In this model each residue is represented by a single interaction site located at the Cα position. These sites are linked by rigid unit-length bonds ($\hat{b_i}$) to form the protein structure. The three-dimensional structure of an N-length protein is specified by the $N-1$ bond vectors $\hat{b_i}$, $N-2$ bond angles τ_i and $N-3$ torsional angles α_i, as shown in Figure 1.


(a)

(b)

Figure 1. (a) Example of a hypothetic protein structure and (b) definition of $\hat{b_i}$, τ_i and α_i, adapted from [20]). Blue balls represent the polar residues and Red ones represent the hydrophobic residues. The backbone and the connections between elements are shown in black lines.

The 20 proteinogenic amino acids are divided into two classes, according to their affinity to water (hydrophobicity): 'A' (hydrophobic) and 'B' (hydrophilic or polar). Notice that this model does not describe the solvent molecules. However, solvent effects, such as the formation of the hydrophobic core, are taken into account through interactions between residues, according to their hydrophobicity (species-dependent global interactions).

When a protein is folded into its native conformation, the hydrophobic amino acids tend to pack inside the protein, in such a way to get protected from the solvent by an aggregation of polar amino acids that are positioned outwards. Interactions between amino acids take place and the energy of the conformation tends to decrease. Conversely, the conformation tends to converge to its native state, in accordance with the Anfinsen's thermodynamic hypothesis [5]. Therefore, the energy function of a folding is given by [20]:

$$\begin{aligned} &E(\hat{b_i};\sigma) = E_{Angles} + E_{torsion} + E_{LJ} = \\ &-k_1 \sum_{i=1}^{N-2} \hat{b_i} \cdot \hat{b_{i+1}} \\ &-k_2 \sum_{i=1}^{N-3} \hat{b_i} \cdot \hat{b_{i+2}} \\ &+ \sum_{i=1}^{N-2} \sum_{j=i+2}^{N} 4\varepsilon(\sigma_i, \sigma_j)(r_{ij}^{-12} - r_{ij}^{-6}) \end{aligned} \tag{1}$$

where: σ = σ_0, ..., σ_N form a binary string that represent the protein sequence, E_{Angles} and $E_{torsion}$ are the energies from bond angle and torsional forces, respectively. Where $\hat{b_i}$ represents the ith bond that joins the ($i-1$)th and the ith residues, and is represented by the vector $\hat{b_i} = \vec{r}_i - \vec{r}_{i-1}$, and $k1 = -1$; $k2 = +1/2$.

The species-dependent global interactions are given by the Lennard-Jones potencial (E_{LJ}); for pairs of ith and jth residues separated by a distance of r_{ij}. Where $\varepsilon(\sigma_i, \sigma_j)$ is chosen to favor the formation of the hydrophobic core ('A' residues). Thus, $\varepsilon(\sigma_i, \sigma_j)$ is 1 for AA interactions and 1/2 for BB/AB interactions.

IV. Molecular Dynamics

Molecular Dynamics (MD) is a computational simulation of physical movements of particles (atoms or molecules). The theoretical basis for MD embodies many of the important results produced by the great names of analytical mechanics – Newton, Euler, Hamilton and Lagrange. The basic form of MD involves little more than Newton's second law [21]. The idea of MD is to generate the trajectory of a system with N particles through numerically integration of the classical equations of motion.

MD is a deterministic approach, differently from Monte Carlo simulations that are stochastic Thus, a MD simulation will always generate the same trajectory from the same initial condition.

The pseudo-code of the Molecular Dynamics is shown in Algorithm 1.

Algorithm 1 Molecular Dynamics pseudo-code

```
Start
Set the initial conditions: positions $r_i(t_0)$, velocities $v_i(t_0)$ and accelerations $a_i(t_0)$
while $t < t_{max}$ do
    Compute forces on all particles
    Integrate equations of motion
    Perform ensemble control
    Compute geometric constraints
    Compute the desired physical quantities
    $t \leftarrow t + \delta t$
end while
End
```

The high level of detail in MD simulations gives general physical conclusions. However, these simulations are usually limited to short timescales (typically, ns) because the calculation of the physical forces is computationally expensive. Two solutions to overcome the computational cost are to use coarse-grained models and use faster hardware [22], [23] in MD simulations.

In the next subsections the steps for implementing MD for the 3D-AB model will be shown.

A. Set the initial conditions

In this step, initial positions, velocities and accelerations are assigned to all particles (i.e. amino acids). An initial unfolded or partially folded conformation is randomly generated. To represent the position of the amino acids, three-dimensional

Cartesian coordinates are defined by a vector $\overrightarrow{r_i}$, as shown in Equation 2.

$$\overrightarrow{r_i} = (x_i, y_i, z_i) \in \Re, i = 0, ..., N-1 \qquad (2)$$

Where, $\Re$ is the set of real numbers (in our program, we use the double precision representation); N is the number of amino acids; x_i, y_i and z_i represent the Cartesian coordinates.

The first amino acid of the primary structure is positioned at the origin of the Cartesian system and next amino acids are positioned at Cartesian coordinates relative to its predecessor and obtained from random spherical coordinates (see Figure 2 [1]), as shown in Equation 3:

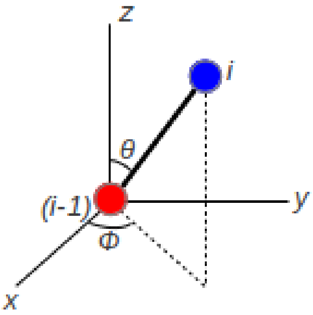


Figure 2. Example of spherical coordinates

$$\begin{aligned} x_i &= x_{i-1} + r_{ij} * sin\theta * cos\phi \\ y_i &= y_{i-1} + r_{ij} * sin\phi * sin\theta \\ z_i &= z_{i-1} + r_{ij} * cos\theta \end{aligned} \qquad (3)$$

Where $\phi \in [0, 2\pi]$ and $\theta \in [0, \pi]$.

The spherical coordinates r_{ij}, ϕ and θ are the radial distance, azimuth and inclination, respectively. It is important to recall that the AB model uses unity radial distances between residues, that is, unit-length bond, as shown in Equation 4.

$$r_{ij} = |\hat{b_i}| = 1 \qquad (4)$$

The initial velocities are generated in two steps. First, the initial velocities are assigned to random directions and a fixed magnitude based on the temperature, as shown in Equation 5.

$$v_i\Big|_{t=0} = velMag * \overrightarrow{\xi}; \qquad (5)$$

Where $velMag$ represents the magnitude, which is based on the temperature, as shown in Equation 6; $\overrightarrow{\xi}$ is a randomly oriented vector of unit length, generated by a random number generator with uniform distribution over the interval [-1, +1] [21].

$$velMag = \sqrt{3.(1 - \tfrac{1}{N}).T_0} \qquad (6)$$

Where N and T_0 represent the number of amino acids of the protein and the initial temperature, respectively.

For generating the unit length vectors, a rejection method proposed by [24] is used, where the probability distribution is related to the uniform distribution on a unit sphere, as shown in the Algorithm 2.

Algorithm 2 Random Unit length vector generation algorithm

```
Start
$s^2 \leftarrow 2$
while $s^2 > 1$ do
    $x \leftarrow 2 * rand() - 1$
    $y \leftarrow 2 * rand() - 1$
    $s^2 \leftarrow x^2 + y^2$
end while
$x \leftarrow 2 * (\sqrt{1 - s^2}) * x$
$y \leftarrow 2 * (\sqrt{1 - s^2}) * y$
$z \leftarrow 1 - 2 * s^2$
End
```

Where, $rand()$ is a Linear Congruential Random Number generator (LCG) [25].

Next, the velocities are also adjusted to ensure that the center of mass is zero is at rest, thereby eliminating any overall flow [21], as shown in Equation 7.

$$v_i\Big|_{t=0} = v_i\Big|_{t=0} - \tfrac{1}{N}\textstyle\sum_j v_j\Big|_{t=0} \qquad (7)$$

The initial accelerations initialized to zeroed.

[1] This figure is available at http://www.wolframalpha.com

$$a_i\Big|_{t=0} = 0\,\forall i \leq N-1 \tag{8}$$

Where, N represents the number of particles (i.e. amino acids).

B. Compute forces on all particles

The forces f_i that act on the particles are usually derived from the potential energy, which is presented in Equation 1. The force corresponding to $u(r)$ is $f = \nabla u(r)$, where $u(r) = E_{Angles} + E_{torsion} + E_{LJ}$. The equations of motion are written according to Newton's second law, as shown in Equation 9.

$$m\,\ddot{\vec{r}_i} = \textstyle\sum_{j=1(j\neq i)}^{N} f_{ij} \tag{9}$$

Where, N represents the number of amino acids. The Newton's third law implies that $f_{ji} = -fij$ Thus, each particle pair need to be examined only once. The AB model does not represent the mass value of residues. Thus, we used the unity dimensionless mass in this work ($m = 1$).

As shown in Equation 1, the force field has three terms: bond-angle forces, bond-torsion forces and forces corresponding to the Lennard-Jones potential.

- **Lennard-Jones potential**: The force that the jth amino acid exerts on the ith amino acid, corresponding to the Lennard-Jones potential is:

  $$f_{ij} = 48 * \varepsilon(\sigma_i, \sigma_j)(r_{ij}^{-14} - \tfrac{1}{2}r_{ij}^{-8}) * \vec{r}_{ij} \tag{10}$$

- **Bond-angle forces**: A change in the bond-angle (τ_i) produces forces on three neighbor residues $j = i-2, i-1, i$ given by:

  $$-\nabla_{r_j} u(\tau_i) = -\frac{du(\tau)}{d(cos\tau)}\bigg|_{\tau=\tau_i} f_j^{(i)} \tag{11}$$

  where $u(\tau_i)$ is the angle potential and $f_j^{(i)} = \nabla_{r_j} cos(\tau_i)$
  As $\sum_j f_j = 0$, the forces can be expressed by:

  $$\begin{gathered} f_{i-2}^{(i)} = (c_{i-1,i-1}c_{ii})^{-1/2}[\vec{b}_{i-1}(c_{i-1,i}/c_{i-1,i-1}) - \vec{b_i}] \\ f_i^{(i)} = (c_{i-1,i-1}c_{ii})^{-1/2}[\vec{b}_{i-1} - \vec{b_i}(c_{i-1,i}/c_{ii})] \end{gathered} \tag{12}$$

  $c_{i,j}$ represents the scalar product of the ith and the jth bond vectors and it is represented by the vector $c_{i,j} = \vec{b_i} \cdot \vec{b_j}$.
  The potential associated with the bond angles for the AB protein model (E_{Angles}) is shown in Equation 1. This equation can be written in cosine form because the AB model uses unit-length bonds, as follows:

  $$u(\tau_i) = -k_1\hat{b_i} \cdot \hat{b}_{i+1} = -k_1 * cos(\tau_i) \tag{13}$$

  and the derivative used for the forces is given by $-\frac{du(\tau)}{d(cos\tau)} = -k_1$.

- **Bond-torsion forces**:
  The force associated with a torsional degree of freedom is defined in terms of the relative coordinates of four consecutive residues.
  The torque caused by a rotation about the ith bond generates forces on four neighbor residues ($j = i-2, ..., i+1$) and it is defined as shown in Equation 11, but replacing the argument τ_i by α_i. Where $u(\alpha_i)$ is the angle potential and $\vec{f}_j^{(i)} = \nabla_{r_j} cos(\alpha_i)$.
  As $\sum_j f_j = 0$, the forces are expressed as shown in Equation 14.
  where:

  $$\begin{gathered} w_1 = c_{i-1,i+1}c_{ii} - c_{i-1,i}c_{i,i+1} \\ w_2 = c_{i-1,i-1}c_{i,i+1} - c_{i-1,i}c_{i-1,i+1} \\ w_3 = c_{i-1,i}^2 - c_{i-1,i-1}c_{ii} \\ w_4 = c_{ii}c_{i+1,i+1} - c_{i,i+1}^2 \\ w_5 = c_{i-1,i+1}c_{i,i+1} - c_{i-1,i}c_{i+1,i+1} \\ w_6 = -w1 \\ q_i = (c_{i-1,i-1}c_{ii} - c_{i-1,i}^2)(c_{ii}c_{i+1,i+1} - c_{i,i+1}^2) \end{gathered} \tag{15}$$

The potential associated with torsion for the AB protein model ($E_{torsion}$) is shown in Equation 1. This equation can also be written in cosine form as shown in Equation 13. The derivative used for the forces is given by $-\frac{du(\alpha)}{d(cos\alpha)} = -k_2$.

$$\begin{aligned}\vec{f}_{i-1}^{(i)} &= -(1 + c_{i-1,i}/c_{ii})\vec{f}_{i-2}^{(i)} + (c_{i,i+1}/c_{ii})\vec{f}_{i+1}^{(i)}\\ \vec{f}_{i}^{(i)} &= (c_{i-1,i}/c_{ii})\vec{f}_{i-2} - (1 + c_{i,i+1}/c_{ii})\vec{f}_{i+1}^{(i)}\\ \vec{f}_{i-2}^{(i)} &= \frac{c_{ii}}{q_i^{1/2}(c_{i-1,i-1}c_{ii}-c_{i-1,i}^2)}[w_1\vec{b}_{i-1} + w_2\vec{b}_i + w_3\vec{b}_{i+1}]\\ \vec{f}_{i+1}^{(i)} &= \frac{c_{ii}}{q_i^{1/2}(c_{ii}c_{i+1,i+1}-c_{i,i+1}^2)}[w_4\vec{b}_{i-1} + w_5\vec{b}_i + w_6\vec{b}_{i+1}] \qquad (14)\end{aligned}$$

Further information about bond-angle and bond-torsion forces calculation (with an example of an alkane chain) can be found in [21].

C. Integrate equations of motion

In this work, we use the velocity-verlet algorithm [26]. The implementation scheme of this algorithm is:

$$\begin{aligned}\vec{r}_i(t+\delta t) &= \vec{r}_i(t) + \vec{v}(t)\delta t + \tfrac{1}{2}\vec{a}(t)\delta t^2\\ \vec{v}_i(t+\delta t/2) &= \vec{v}_i(t) + \tfrac{1}{2}\delta t\vec{a}_i(t)\\ \vec{v}_i(t+\delta t) &= \vec{v}_i(t+\delta t/2) + \tfrac{1}{2}\vec{a}_i(t+\delta t)\delta t \qquad (16)\end{aligned}$$

Where, $\vec{r}_i(t)$,$\vec{v}_i(t)$ and $\vec{a}_i(t)$ are the position, velocity and acceleration of the ith residue, respectively; t and δt are the time and the timestep.

D. Perform ensemble control

Our MD simulation performs the canonical ensemble (also referred to as the ensemble NVT), where the number of particles (residues), the volume and the temperature are controlled at desired values. The temperature is controlled using the method of weak coupling to a thermal bath proposed by [27]. In this approach, coupling removes or adds energy to the system to maintain an approximately constant temperature. The velocities are scaled at each step using the scaling factor α, as follows

$$\vec{v}_i(t) = \lambda * \vec{v}_i(t) \qquad (17)$$

$$\lambda = \sqrt{1 + \frac{\delta t}{\tau_T}(\frac{T_{sp}}{T} - 1)} \qquad (18)$$

Where λ, τ_T, T_{sp}, T are the scaling factor, the coupling constant, the desired temperature (set-point) and the current temperature, respectively.

E. Compute geometric constraints

As mentioned before, a protein with the AB model is subject to geometrical constraints due to the fixed unit-length bonds between amino acids ($|\vec{r}_i - \vec{r}_j|^2 = b_i^2 = 1$).

Considering a protein with N residues, there are a total of $n_c = N - 1$ geometric constraints. In this work, we use the SHAKE algorithm [28] to deal with constraints.

The SHAKE algorithm starts after advancing the system over a single timestep, while ignoring the constraints [21]. Thus, a set of uncorrected coordinates is obtained that are represented by Equation 19.

$$\begin{aligned}&\vec{r}'_i(t+\delta t/2) = 2\vec{r}_i(t) - \vec{r}_i(t-\delta t)\\ &+(\delta t/2)^2/m_i \cdot \vec{f}_i(t) \qquad (19)\end{aligned}$$

Algorithm 3 shows the SHAKE algorithm. The SHAKE algorithm has two parts. First, corrections along the direction of $\vec{r}_{ij}(t)$ are done. The estimated coordinates $\vec{r}'_i$ and $\vec{r}'_j$ are updated by using the correction factor γ, which is determined as shown in lines 1 and 8 of the algorithm. Next, velocities are corrected in a similar manner. Here, it is important to recall that m_i and m_j are the masses of the ith and jth amino acids, respectively. The AB model does not represent the mass value of residues. Thus, we used the unity dimensionless mass in this work (i.e. $m_i = m_j = 1$). In addition, b_i represent the bond length between the ith and jth amino acids which, as mentioned, are unit-length bonds in the AB model. The process is repeated for both direction and velocity corrections until all the constraints are satisfied.

The precision of the SHAKE algorithm is given by $|\vec{r}_0 - \vec{r}|/|\vec{r}_0| < 10^{-k}$, where 10^{-k} is the desired precision. Our implementation has a precision of 10^{-6}.

Algorithm 3 SHAKE algorithm

```
Start
Coordinates correction:
$\gamma \leftarrow \frac{\vec{r}^{2'}_{ij} - b_i^2}{4(\delta t/2)^2(1/m_i + 1/m_j)\vec{r}'_{ij}\cdot\vec{r}_{ij}}$
while $|\gamma| < 10^{-k} \cdot b_i^2$ do
    $\vec{r}'_i \leftarrow \vec{r}_i - \gamma\vec{r}_{ij}$
    $\vec{r}'_j \leftarrow \vec{r}_i + \gamma\vec{r}_{ij}$
    $\gamma \leftarrow \frac{\vec{r}^{2'}_{ij} - b_i^2}{4(\delta t/2)^2(1/m_i + 1/m_j)\vec{r}'_{ij}\cdot\vec{r}_{ij}}$
end while
Velocities correction:
$\gamma = \frac{\ddot{\vec{r}}_{ij}\cdot\vec{r}_{ij}}{2\vec{r}^2_{ij}}$
while $|\gamma| < 10^{-k} \cdot b_i^2$ do
    $\ddot{\vec{r}}'_i \leftarrow \ddot{\vec{r}}_j - \gamma\vec{r}_{ij}$
    $\ddot{\vec{r}}'_j \leftarrow \ddot{\vec{r}}_j + \gamma\vec{r}_{ij}$
    $\gamma = \frac{\ddot{\vec{r}}_{ij}\cdot\vec{r}_{ij}}{2\vec{r}^2_{ij}}$
end while
End
```

F. Compute the desired physical quantities

Besides the total energy (see Equation 1) of the obtained conformation, we also compute the radius of gyration [29] Radius of gyration is a measure of compactness of a set of points (in this case, the residues of the protein). The more compact the set of points, the smaller the radius of gyration is. The radius of gyration is computed by Equation 20:

$$Rg = \sqrt{\frac{\sum_{i=0}^{N-1}[(x_i - \overline{X})^2 + (y_i - \overline{Y})^2 + (z_i - \overline{Z})^2]}{N}} \tag{20}$$

In this equation, x_i, y_i and z_i are the coordinates of the residues. $\overline{X}$, $\overline{Y}$ and $\overline{Z}$ are the average of all x_i, y_i and z_i; and N is the number of residues.

G. General comments

- The simulation takes place in a cubic container, using periodic boundary conditions. The periodic boundary conditions are equivalent to considering an infinite array of identical copies of the simulation region [21]. There are two consequences of this periodicity: particles (i.e. amino acids) that leave the simulation region through a particular bounding face immediately reenters the region through the opposite face, and particles lying within a distace of a boundary interact with particles in an adjacent copy of the system (i.e. particles near the opposite boundary). The second consequence is considered to be a wraparound effect. If a particle have moved outside the region its coordinates are adjusted to bring it inside the simulation region, as shown in Equations 21, 22 and 23.

  $$x_i = \begin{cases} x_i - L_x & \text{if } x_i \geq L_x/2 \\ x_i + L_x & \text{otherwise} \end{cases} \tag{21}$$

  $$y_i = \begin{cases} y_i - L_y & \text{if } y_i \geq L_y/2 \\ y_i + L_y & \text{otherwise} \end{cases} \tag{22}$$

  $$z_i = \begin{cases} z_i - L_z & \text{if } z_i \geq L_z/2 \\ z_i + L_z & \text{otherwise} \end{cases} \tag{23}$$

  Where x_i, y_i and z_i represent the Cartesian coordinates of the amino acids; L_x, L_y and L_z are the region size in the x, y and z directions, respectively.
  The components of the distance between amino acids are determined in a similar manner, as shown in Equations 24, 25 and 26.

  $$r_{ij_x} = \begin{cases} r_{ij_x} - L_x & \text{if } r_{ij_x} \geq L_x/2 \\ r_{ij_x} + L_x & \text{otherwise} \end{cases} \tag{24}$$

  $$r_{ij_y} = \begin{cases} r_{ij_y} - L_y & \text{if } r_{ij_y} \geq L_y/2 \\ r_{ij_y} + L_y & \text{otherwise} \end{cases} \tag{25}$$

  $$r_{ij_z} = \begin{cases} r_{ij_z} - L_z & \text{if } r_{ij_z} \geq L_z/2 \\ r_{ij_z} + L_z & \text{otherwise} \end{cases} \tag{26}$$

  Where r_{ij_x}, r_{ij_y} and r_{ij_z} are the components of the distance between the ith and jth amino acids.
- We do not use real physical units because they are not defined for the AB model of

proteins. Thus, the energy, temperature and length are shown in reduced (or dimensionless) units.

It is important to note that in simulations of real molecular systems is convenient to express physical quantities, such as temperature and pressure, in reduced units, and to use basic units in order to translate them to real units. The basic units depend on experimental data and they are: length (σ), energy (ϵ), mass (m) and temperature (ϵ/K_B), where K_B is the Boltzmann constant [30]. Moreover, the main reason for using dimensionless units in simulations with real physical units is related to scaling. Thus, properties that have been measured in dimensionless units can be scaled to the physical units for the problem of interest. From a practical point of view, the use of dimensionless units removes any risk of problems with data representation.

V. Computational Experiments

All experiments reported in this work were run in a desktop computer with a Intel processor Quad Core, running Arch Linux All algorithms were implemented in ANSI-C programming language.

VI. Benchmarks

Sections VI-A and VI-B present the synthetic and real protein sequences, respectively, used in this work. On one hand, the synthetic sequences were only used for simple HP models [31], [32]. On the other hand, the real protein sequences are first introduced here and, consequently, there are no reference values for these sequences.

A. Synthetic sequences

Synthetic sequences were used, which were proposed by [32] for the 3DHP model, used by [31] for the 3DHP-SC model. They have either 27, 31, 36 and 48 amino acids and are shown in Table I.

Table I
Benchmark sequences for the AB off-lattice model, proposed by [32] for the 3DHP models.

id	N	Sequence
5	27	$AB^4A^4B^2ABABA^3BAB^2A^2B^2A$
6	27	$ABBBAAAABABAABBBABAABABBBAB$
7	27	$AB(AABB)^2A^4(BBBA)^2A^2B^2A$
8	31	$(AAB)^2A^6(BBAAAAA)^2A^2$
9	36	$BA(BBA)^{11}B$

B. Real protein sequences

Real proteins sequences and structures were also used in this work. Table II shows the list of real protein sequences that were used in this work. In this table, the second column, third and fourth columns identify, respectively, the PDB code, name and the size (N) of the proteins.

Table II
List of real proteins

id	PDB code	Name	N
10	2gb1	Protein G	56
11	1pcy	Oxidized Poplar Plastocyanin	99
12	2trx	*Escherichia coli's Thioredoxin*	108
13	3fxn	*Clostridium Beijerinckii's Flavodoxin*	138

These proteins were extracted from PDB files [2]. The PDB format has 12 sections, where in each section 46 different records are listed in a specific order. In this work, the amino acid sequence and the coordinates of the amino acids of the protein are required. Thus, we used the SEQRES and ATOM records of the PDB file. It is important to recall that we use the backbone of the proteins, which is formed by the Cαs of the amino acids. Hence, our approach reads the Cα coordinates from the PDB file.

In order to convert the protein sequences of the PDB into the AB model alphabet (i.e.: 'A' and 'B' for hydrophobic and hydrophilic residues, respectively) we used the amino acid type classification presented in [33]. Table III shows the equivalent AB sequences of the proteins.

[2] Available in *http://www.pdb.org*

Table III
SEQUENCES

Protein	Sequence
2gb1	$AB^3A^3BAB^2ABAB^4B(AAB)^2AB^2A^2(BBBA)^3A(BA)^2B(BBBA)^2BAB^2$
1pcy	$A(BAAAAABBA)^2(BA)^2AB^2A^3B^3A^4B^2A^3B^4(AAB)^2AB(BA)^2B^4A^2$
2trx	$B^3A(AB)^2(BBBA)^2(AB)^2A^2(AAAB)^2A^5BA^6B^2A^2B^4(AB)^2A^2(BA)^2$
3fxn	$B^4A^3B^3(AAB)^2A^5B(BA)^2A^3BA^5(BA)^2B^2A^2(BA)^2A$

C. Structure alignment and evaluation

The applicability of a coarse-grained model can be evaluated by comparing the obtained structures with real protein structures (i.e. protein structures extracted from PDB). Therefore, we assess the quality of the obtained structures by comparing them with real structures, using the Algorithm 4.

Algorithm 4 Structure evaluation algorithm

```
Start
$AB\_like_a \leftarrow$ fitting($P_1$)
$AB\_like_b \leftarrow$ fitting($P_2$)
$RMSD \leftarrow kabsch(AB\_like_a, AB\_like_b)$
End
```

Basically, Algorithm 4 has three steps, where the first two steps are fitting procedures and the last one represents a quality assessment.

- **Fitting procedures**: In step 1 ($AB\_like_a \leftarrow$ *fitting(P_1)*), the PDB file coordinates (P_1) are fitted to an off-lattice structure (called "AB_like"), where all bond lenghts are scaled to 3.8 Å (AB_like_a), which is the mean distance between consecutive Cα atoms [34]. In step 2 ($AB\_like_b \leftarrow$ *fitting(P_2)*), the coordinates of the obtained AB model structure (P_2 are fitted to the "AB_like' structure, where all unit-length bonds are also scaled to 3.8 Å (AB_like_b). Algorithm 5 shows the fitting procedure.
  Where, $sin(\theta)$, $cos(\theta)$, $acos(x)$, $atan2(x, y)$ are the sine, cosine, inverse cosine and the inverse tangent, respectively. It is important to recall that the $atan2$ function returns a positive value for counter-clockwise angles, and a negative value for clockwise angles.

Algorithm 5 Fitting procedure – *fitting(p)*

```
Start
Let $N$ be the protein size (number of amino acids)
Let $p$ be the input coordinates (from PDB or AB)
Let $a$ be the output "$AB\_like$" coordinates
Let $dx$, $dy$ and $dz$ Let $r$ be the bond lenght between $i$ and $(i+1)$ amino acids
for $i = 1 \rightarrow N-1$ do
    $dx \leftarrow p[i+1].x - p[i].x$
    $dy \leftarrow p[i+1].y - p[i].y$
    $dz \leftarrow p[i+1].z - p[i].z$
    $r \leftarrow \sqrt{(dx^2 + dy^2 + dz^2)}$
    $\theta[i] \leftarrow acos(dz/r)$;
    $\phi[i] \leftarrow atan2(dx, dy)$;
end for
$a[0].coord \leftarrow p[0]$
for $i = 2 \rightarrow N$ do
    $a[i].x \leftarrow a[i-1].x + 3.8 * sin(\theta[i-1]) * cos(\phi[i-1])$
    $a[i].y \leftarrow a[i-1].y + 3.8 * sin(\theta[i-1]) * sin(\phi[i-1])$
    $a[i].z \leftarrow a[i-1].z + 3.8 * cos(\theta[i-1])$
end for
return $a$
End
```

- **Quality assesment:**
  According to [34] RMSD is used to assess protein model quality. It measures the similarity of two structures from coordinates, as shown in Equation 27.

$$RMSD = \sqrt{\frac{\sum_{i=0}^{N-1} |P_{1_i} - P_{2_i}|}{N}} \qquad (27)$$

  Where, N, P_{1_i} and P_{2_i} represent the number of amino acids, the Cartesian coordinates of the first protein structure P_1 and the Cartesian coordinates of the second protein structure P_1, respectively.

The RMSD evaluation depends on the superpositioning of the protein structures. Since the RMSD is a rotation-dependent measure, a RMSD-optimised is done using the Kabsch method [35] in order to obtain the lowest RMSD. The main idea of the Kabsch method is to calculate the rotation matrix (U), which is used to minimize the RMSD. Basically, the Kabsch method algorithm has three steps: a translation to the origin of both structures, the computation of a covariance matrix and the computation of the rotation matrix.

From the structure similarity point of view, [36] pointed that for small proteins with size up to 150 amino acids, RMSD values less than 3 Å (i.e. RMSD ¡ 3 Å) indicate that the model presents a good quality. In addition, RMSD values between 3 and 5 Å (i.e. 3 $\leq$ RMSD $\leq$ 5) are considered acceptable and useful, and predictions with deviations above 5 Å are considered to be uninformative. [37] also stated that models with RMSD values up to 6.5 Å can be informative and useful.

VII. Results and Analysis

Table IV shows the results obtained for the synthetic sequences. In this table, the second and last columns identify, respectively, the best results obtained and the average processing time.

Table V shows the results obtained for the real sequences.

Table IV
Results for the synthetic sequences.

N	$Energy$		$t_p(s)$
	Best	Avg±stdev	
27	-75.8225	-71.44±3.38	302.56
27	-73.0161	-67.96±3.52	264.34
27	-74.3461	-68.63±3.56	325.72
31	-103.4963	-99.36±3.08	247.09
36	-94.0439	-89.92±2.59	271.53

As shown in Table V, the average RMSD values obtained is greater than 9 Å, indicating that the AB model, using the dimensionless unit length bonds, is uninformative. This may be caused by lack of information essential to describe secondary structures, such as hydrogen bonding (i.e. non-bonded interactions between the NH group of the ith amino acid and C=O group of the $i+4$th amino acid), that is the most prominent characteristic of α-helices. Moreover, non-bonded interactions based on the hydrophobicity of the side chains, which are included in the energy equation of the AB model only allow the formation of a hydrophobic core inside the proteins. The conformation of α-helices is also driven by the environment (solvent). For instance, [38] explained that it seems reasonable to assume that the conformation of α-helices located in hydrophilic environments, such as water, differs from those located in hydrophobic environments, such as the cell membrane. Here, it is important to recall again that the AB model does not consider the environment. Overall, the main weakness of the AB model is related to the lack of a clear representation of secondary structures, despite the formation of a hydrophobic core, which is also an important aspect of the protein folding.

Overall, the processing time is a function of the length of the sequence, growing as the number of amino acids of the sequence increases. This fact, by itself, strongly suggests the need for high performance approaches for dealing with this problem. With the advantage of parallel processing, it will be possible to simulate several folding pathways, which will allow us to explore the energy landscape of the AB model.

A. Pathways

Figures 3(a), 3(b) and 3(c) show the time dependence of the total energy of the best conformation of each sequence, radius of gyration of the best conformation of each sequence and the radius of gyration of the hydrophobic (Rg_H) and hydrophilic (Rg_P) residues of protein 2gb1, using the conversion table following the classification by [33]. Such plot confirm the Anfinsen's thermodynamic hypothesis, where a denatured conformation has high energy and folding to the native

Table V
RESULTS OBTAINED FOR REAL SEQUENCES

Protein	Energy		RMSD		Avg(R_g)	Avg(T_p) (s)
	Best	Average	Best	Average		
2gb1	-166.96	-159.44 ± 3.46	7.35	9.52 ± 0.79	1.85	2621.50
1pcy	-350.97	-339.41 ± 6.76	10.5	12.34 ± 1.11	2.46	8377.56
2trx	-393.17	-379.57 ± 5.94	11.06	11.94 ± 0.75	2.36	12954.22
3fxn	-490.08	-474.30 ± 7.09	11.25	**12.54 ± 0.87**	2.61	21301.80

state, the free energy of the protein decreases significantly.

Notice that, in Figure 3(c), it is possible to observe the formation of a compact hydrophobic core, surrounded by polar residues, during folding because the radius of gyration of the hydrophobic residues is much lower than that of the polar residues (that is, $Rg_H < Rg_P$). Also, in this figure, we can observe an evidence of conformational fluctuations (commonly known as breathing motion [39]) in the maximized plot.

An example of a folding trajectory of protein 2gb1, using the AB model, is presented in Figures 4 and 5. In these Figures, it is shown seven folding states that were obtained in a simulation. The figure captions below each protein structure show the energy (E), radius of gyration (R_g) and the RMSD (between AB_like structures obtained from real proteins and AB structures) at different times (t). Two amino acids i and j are taken to be in contact if $r_{ij}^2 < 1.75$ [20]. The number of local and global interactions is also shown. A contact between ith and jth amino acids is called local if $2 \leq |i - j| \leq 4$ and global if $|i - j| > 4$. Contact maps are also shown in this figure in order to observe the formation of secondary structures.

In addition, Figure 5(d) shows the backbone trace and the contact map of the protein 2gb1 obtained from PDB files. The backbone trace was obtained using the RasMol software [3]. The contact map was obtained using the CMView tool [40] [4], where we use 7 Å as the threshold distance and consider pairs of residues whose sequence separation is $|i - j| \geq 2$. In Figure 5(d), it is possible to observe an α-helix (amino acids 22–36) as a band along the main diagonal. β-sheets are also shown (amino acids 2–17 and 41–56), where anti parallel β-sheets are represented by thin bands orthogonal to the main diagonal, and the two central β-sheets are in parallel.

In Figures 4(b), 4(c) and 4(d), it is possible to observe the formation of an antiparallel β-sheet between the 24th and 35th amino acids, that is different from the real structure which, in turn, has an α-helix between the 22nd and 36th amino acids. Figure 4(d) also suggests the formation of an α-helix. However, in Figures 5(b) and 5(c) it is not easy to found the secondary structures from the contact maps. Moreover, it is possible to observe that the RMSD decreases during the folding process towards the native state. Notwithstanding, the RMSD measures are still high [36], [37] and the AB model is too simple to represent protein structures.

VIII. CONCLUSIONS

The PFP is still an open problem for which there is no closed computational solution. While most works used HP models, the off-lattice AB model is still poorly explored despite being a simplified model with a level more of biological expressiveness.

To the best of our knowledge, this work presents the first implementation of Molecular Dynamics using the off-lattice AB model. This work also offered new reference values for benchmark sequences that can be used in the future by other researchers for testing computational approaches applied to the same problem.

Future work will include simulations and analysis of folding pathways using other structures

[3] RasMol is a molecular visualization software. Available at http://www.rasmol.org

[4] CMView is available in http://www.bioinformatics.org/cmview

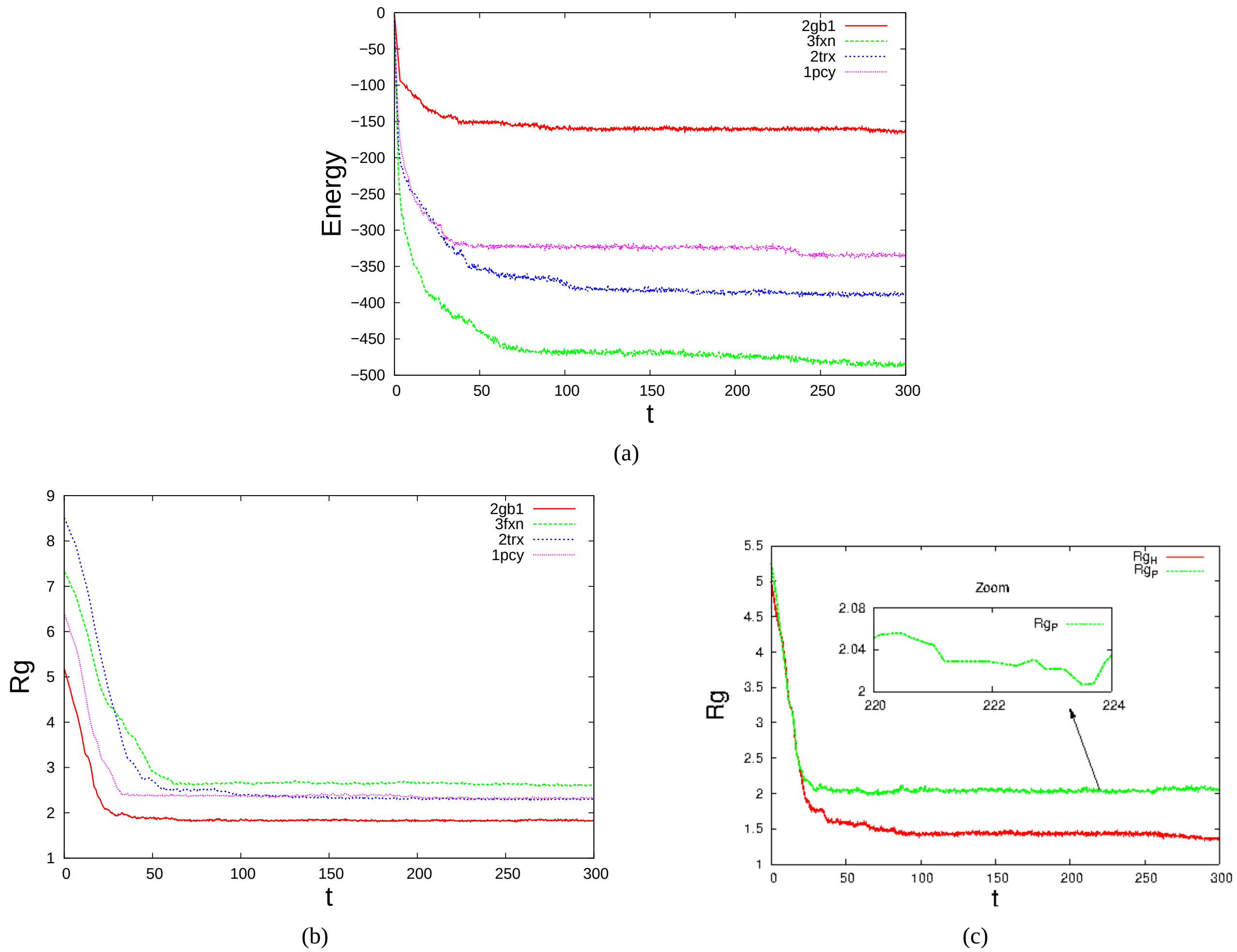


Figure 3. Properties (in dimensionless MD units): (a) Energy of the best conformation of each sequence, (b) radius of gyration of the best conformation of each sequence and (c) represent the radius of gyration of the hydrophobic (Rg_H) and hydrophilic (Rg_P) residues of sequence 2gb1.

drawn from real protein structures extracted from the PDB, using a more complex coarse-grained model for proteins.

Besides the energy, radius of gyration, RMSD and thermodynamic measures, such as the temperature, pressure, kinetic energy and the specific heat [21], [30], we intend to study other metrics in order to contribute to better understanding the process.

MD simulations with different thermodynamic ensembles will be done, which are characterized by the control of certain thermodynamic quantities using thermostats and barostats such as the canonical ensemble (NVT – moles, volume, temperature) and the isothermal-isobaric (NPT – moles, pressure, temperature) ensembles, including a thermodynamic analysis of the folding process.

An important drawback is regarding the processing time for the simulations. There is a strong increase of processing time as the length of the protein grows, following a polynomial complexity. This fact, by itself, strongly suggests that future research will need highly parallel approaches for dealing with the PFP, such as the use of GPGPU (General Purpose Graphics Processing Units) [23], [41] or hardware-based accelerators [42].

Overall, we believe that the use of Molecular Dynamics for the PFP using coarse-grained mod-

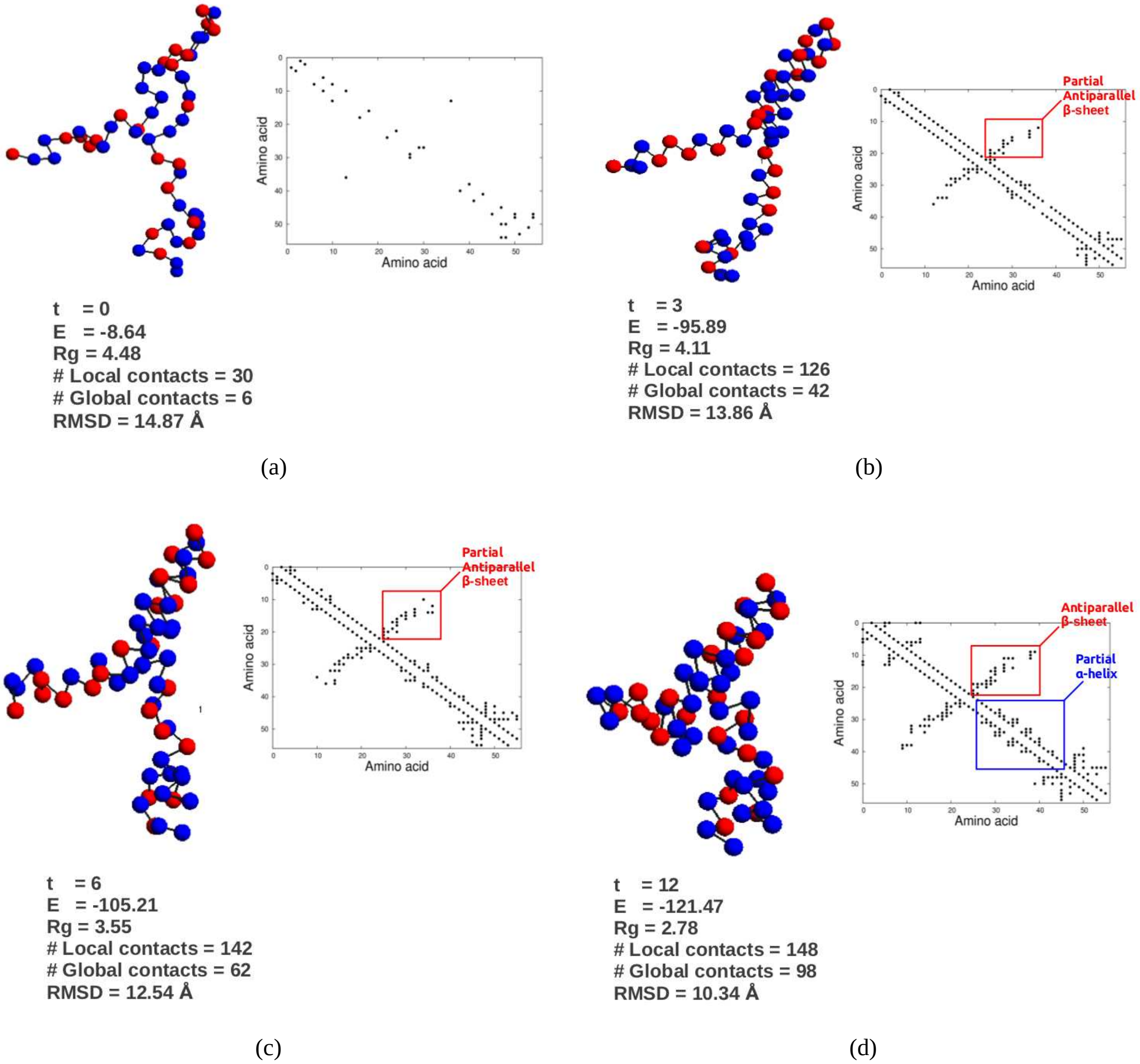


Figure 4. Folding Pathway example, protein 2gb1 (part 1 of 2).

els is very promising for this area of research.

Acknowledgments

Authors would like to thank the Brazilian National Research Council (CNPq) for the research grant to H.S. Lopes; and CAPES-DS scholarship to C.M.V. Benítez.

References

[1] L. Luheshi and C. Dobson, “Bridging the gap: From protein misfolding to protein misfolding diseases,” *FEBS Letters*, vol. 583, pp. 2581–2586, 2009.

[2] S. Broadley and F. U. Hartl, “The role of molecular chaperones in human misfolding diseases,” *FEBS Letters*, vol. 583, pp. 2647–2653, 2009.

[3] Y. Chen, F. Ding, H. Nie, A. W. Serohijos, S. Sharma, K. Wilcox, S. Yin, and N. D. Dokholyan, “Protein folding: then and now,” *Archives of Biochemistry and Biophysics*, vol. 469, pp. 4–19, 2008.

[4] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. P. E. Bourne, “UniProt archive,” *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[5] C. B. Anfinsen, “Principles that govern the folding of

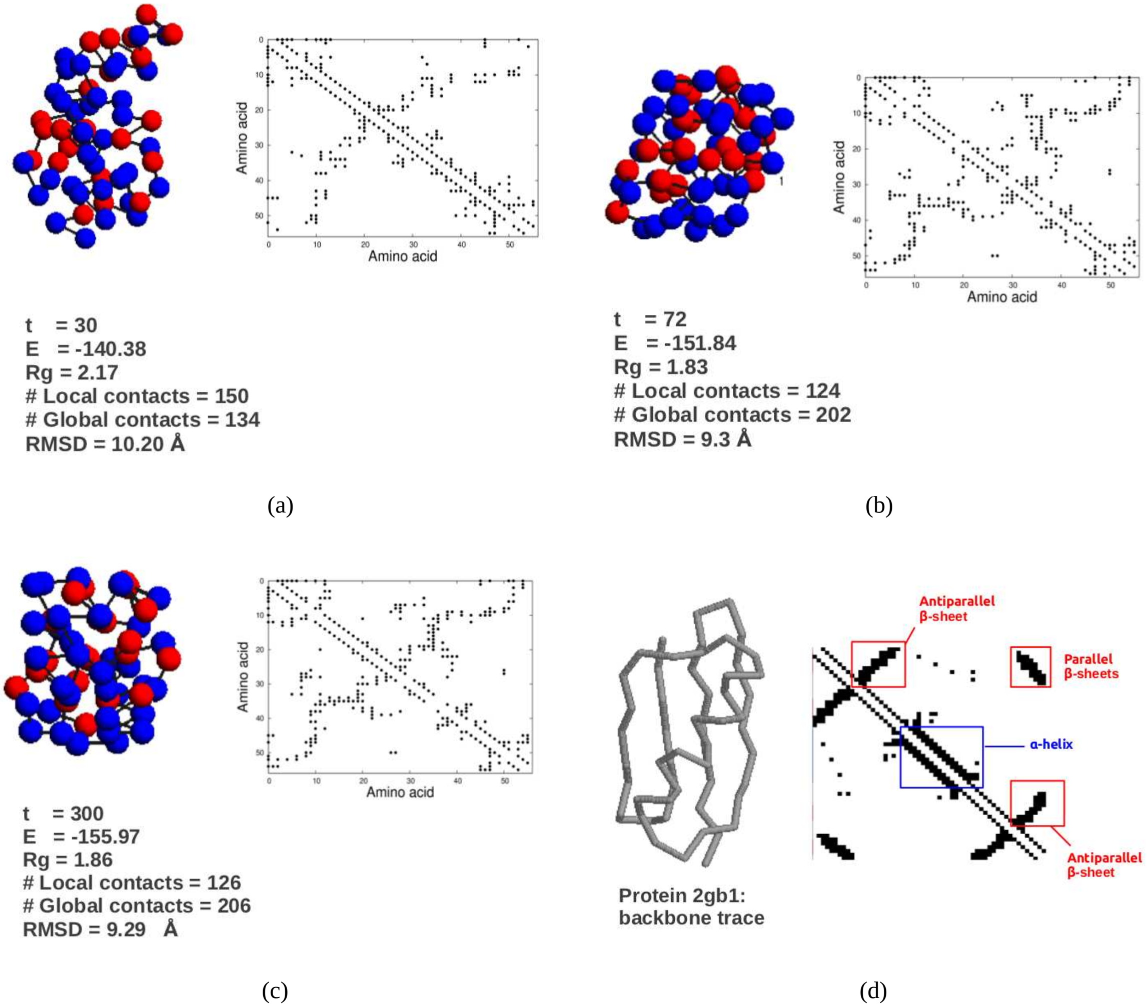


Figure 5. (a) to (c) : Folding Pathway example, protein 2gb1 (part 2 of 2). (d) Protein 2gb1: backbone trace and contact map.

protein chains," *Science*, vol. 181, no. 96, pp. 223–230, 1973.

[6] K. Dill, S. Ozkan, M. Shell, and T. Weikl, "The protein folding problem," *Annual Review of Biophysics*, vol. 37, pp. 289–316, 2008.

[7] V. Nanda and R. Koder, "Designing artificial enzymes by intuition and computation," *Nature Chemistry*, vol. 2, pp. 15–24, 2010.

[8] D. Röthlisberger, O. Khersonsky, A. Wollacott, L. Jiang, J. DeChancie, J. Betker, J. Gallaher, E. Althoff, A. Zanghellini, O. Dym, S. Albeck, K. Houk, D. Tawfik, and D. Baker, "Kemp elimination catalysts by computational enzyme design," *Nature*, vol. 453, no. 7192, pp. 190–195, 2008.

[9] R. Broglia and G. Tiana, "Physical models for protein folding and drug design," in *Proceedings Idea-Finding Symposium*. Frankfurt, Germany: Frankfurt Institute for Advanced Studies, 2003, pp. 23–33.

[10] G. Nicosia and G. Stracquadanio, "Generalized pattern search algorithm for peptide structure pediction," *Biophysical Journal*, vol. 95, no. 10, pp. 4988–4999, 2008.

[11] H. S. Lopes, "Evolutionary algorithms for the protein folding problem: A review and current trends," in *Computational Intelligence in Biomedicine and Bioinformatics*. Heidelberg: Springer-Verlag, 2008, vol. I, pp. 297–315.

[12] A. Liwo, M. Khalili, and H. A. Scheraga, "Ab-initio simulations of protein-folding pathways by molecular

dynamics with the united-residue model of polypeptide chains," *Proceedings of the National Academy of Sciences*, vol. 102, no. 7, pp. 2362–2367, 2005.

[13] R. Day and V. Daggett, "All-atom simulations of protein folding and unfolding," *Advances in Protein Chemistry*, vol. 66, pp. 373–403, 2003.

[14] D. Baker, "A suprising simplicity to protein folding," *Nature*, vol. 405, pp. 39–42, 2000.

[15] Z. Cui, D. Liu, J. Zeng, and Z. Shi, "Using splitting artificial plant optimization algorithm to solve toy model of protein folding," *Journal of Computational and Theoretical Nanoscience*, vol. 9, pp. 2255–2259, 2012.

[16] R. S. Parpinelli, C. M. V. Benítez, J. Cordeiro, and H. S. Lopes, "Performance analysis of swarm intelligence algorithms for the 3D-AB off-lattice protein folding problem," *Journal of Computational and Theoretical Nanoscience*, vol. 13, 2013, to appear.

[17] M. Karplus, "The Levinthal paradox: yesterday and today." *Folding & Design*, vol. 2, no. 4, pp. S69–S75, 1997.

[18] M. Gruebele, "Protein folding: the free energy surface," *Current Opinion in Structural Biology*, vol. 12, no. 1, pp. 161–168, 2002.

[19] F. Stillinger and T. Head-Gordon, "Collective aspects of protein folding illustrated by a toy model," *Physical Review E*, vol. 52, no. 3, pp. 2872–2877, 1995.

[20] A. Irback, C. Peterson, F. Potthast, and O. Sommelius, "Local interactions and protein folding: A three-dimensional off-lattice approach," *Journal of Chemical Physics*, vol. 1, pp. 273–282, 1997.

[21] D. Rapaport, *The Art of Molecular Dynamics Simulation*. Cambridge, UK: Cambridge University Press, 2004.

[22] T. Harder, M. Borg, S. Bottaro, W. Boomsma, S. Olsson, and J. Ferkinghoff-Borg, "An efficient null model for conformational fluctuations in proteins," *Structure*, vol. 20, no. 6, pp. 1028–1039, 2012.

[23] N. Nouri and S. Ziaei-Rad, "Mechanical property evaluation of buckypaper/epoxy composites using molecular dynamics simulations fully implemented on graphical processing units," *Journal of Computational and Theoretical Nanoscience*, vol. 9, pp. 2144–2154, 2012.

[24] G. Marsaglia, "Choosing a point from the surface of a sphere," *The Annals of Mathematical Statistics*, vol. 43, no. 2, pp. 645–646, 1972.

[25] D. Knuth, *The Art of Computer Programming*. USA: Addison-Wesley, 1981, vol. 2.

[26] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters," *The Journal of Chemical Physics*, vol. 76, p. 637, 1982.

[27] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gusteren, A. DiNola, and J. R. Haak, "Molecular dynamics with coupling to an external bath," *Journal of Chemical Physics*, vol. 81, p. 3684, 1984.

[28] J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, "Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes," *Journal of Computational Physics*, vol. 23, pp. 327–341, 1977.

[29] H. S. Lopes and M. P. Scapin, "A hybrid genetic algorithm for the protein folding problem using the 2D-HP lattice model," in *Success in Evolutionary Computation*, ser. Studies in Computational Intelligence, A. Yang, Y. Shan, and L. T. Bui, Eds. Berlin, Germany: Springer-Verlag, 2008, vol. 92, pp. 121–140.

[30] D. Frenkel and B. Smit, *Understanding Molecular Simulation. From algorithms to applications*. San Diego, USA: Academic Press, 2002.

[31] C. M. V. Benítez and H. S. Lopes, "Hierarchical parallel genetic algorithm applied to the three-dimensional HP side-chain protein folding problem," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. Piscataway, USA: IEEE Press, 2010.

[32] K. Yue and K. Dill, "Sequence-structure relationships in proteins and copolymers," *Physical Review E*, vol. 48, no. 3, pp. 2267–2278, 1993.

[33] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of The Cell*. New York, USA: Garland Science, 2002.

[34] M. Mann, R. Saunders, C. Smith, R. Backofen, and C. Deane, “Producing high-accuracy lattice models from protein atomistic coordinates including side chains,” *Advances in Bioinformatics*, vol. 2012, p. 148045, 2012.

[35] W. Kabsch, “A discussion of the solution of the best rotation to relate two sets of vectors,” *Acta Crystallographica*, vol. A34, pp. 827–828, 1978.

[36] R. Unger, “The building block approach to protein structure prediction,” ser. Cellular Origin, Life in Extreme Habitats and Astrobiology, J. Seckbach and E. Rubin, Eds. Amsterdam, The Netherlands: Kluwer Academic, 2004, vol. 8, pp. 177–188.

[37] M. Marti-Renom, B. Yerkovich, and A. Sali, “Comparative protein structure prediction,” in *Current Protocols in Protein Science*. New York, USA: John Wiley & Sons, 2001.

[38] M. Olivella, X. Deupi, C. Govaerts, and L. Pardo, “Influence of the environment in the conformation of α-helices studied by protein database search and molecular dynamics simulations,” *Biophysical Journal*, vol. 82, no. 6, pp. 3207–3213, 2002.

[39] A. Ramanathan and P. Agarwal, “Computational identification of slow conformational fluctuations in proteins,” *Journal of Physical Chemistry B*, vol. 113, no. 52, pp. 16 669–16 680, 2009.

[40] C. Vehlow, H. Stehr, M. Winkelmann, J. Duarte, L. Petzold, J. Dense, and M. Lappe, “CMView: Interactive contact map visualization and analysis,” *Bioinformatics*, vol. 27, pp. 1573–1574, 2011.

[41] M. H. Scalabrin, R. S. Parpinelli, C. M. V.Benítez, and H. S. Lopes, “Population-based harmony search using GPU applied to protein structure prediction,” *International Journal of Computational Science and Engineering*, vol. 8, 2013, to appear.

[42] N. B. Armstrong Junior, H. S. Lopes, and C. R. E. Lima, “Preliminary steps towards protein folding prediction using reconfigurable computing,” in *Proc. 3rd Int. Conf. on Reconfigurable Computing and FPGAs*. Piscataway, USA: IEEE, 2006, pp. 92–98.