Computational Biology and Chemistry 87 (2020) 107301

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/cbac

Research Article

PathMolD-AB: Spatiotemporal pathways of protein folding using parallel molecular dynamics with a coarse-grained model

Leandro Takeshi Hattori*, Bruna Araujo Pinheiro, Rafael Bertolini Frigori, César Manuel Vargas Benítez, Heitor Silvério Lopes

Bioinformatics and Computational Intelligence Laboratory (LABIC), Federal University of Technology Paraná (UTFPR), Av. 7 de Setembro, 3165, 80230-901 Curitiba, PR, Brazil

ARTICLE INFO

Keywords:
Canonical ensemble
CUDA
3D-AB off-lattice
Protein folding dataset

ABSTRACT

Solving the protein folding problem (PFP) is one of the grand challenges still open in computational biophysics. Globular proteins are believed to evolve from initial configurations through folding pathways connecting several thermodynamically accessible states in a free energy landscape until reaching its minimum, inhabited by the stable native structures. Despite its huge computational burden, molecular dynamics (MD) is the leading approach in the PFP studies by preserving the Newtonian temporal evolution in the canonical ensemble. Non-trivial improvements are provided by highly parallel implementations of MD in cost-effective GPUs, concomitant to multiscale descriptions of proteins by coarse-grained minimalist models. In this vein, we present the PathMolD-AB framework, a comprehensive software package for massively parallel MD simulations using the canonical ensemble, structural analysis, and visualization of the folding pathways using the minimalist AB-model. It has, also, a tool to compare the results with proteins re-scaled from the PDB. We simulate and analyze, as case studies, the folding of four proteins: 13FIBO, 2GB1, 1PLC and 5ANZ, with 13, 55, 99 and 223 amino acids, respectively. The datasets generated from simulations correspond to the MD evolution of 3500 folding pathways, encompassing 35×10^6 states, which contains the spatial amino acid positions, the protein free energies and radii of gyration at each time step. Results indicate that the speedup of our approach grows logarithmically with the protein length and, therefore, it is suited for most of the proteins in the PDB. The predicted structures simulated by PathMolD-AB were similar to the re-scaled biological structures, indicating that it is promising for the study of the PFP study.

1. Introduction

The protein folding problem (PFP) is an active area of research in Biophysics and aims at unveiling how proteins fold into their native form (Dill and MacCallum, 2012). Along time, many methods and algorithms have been proposed to find the native structure of a protein based only on the sequence and properties of their amino acids chain (Moult et al., 2018; Hattori et al., 2020). However, the dynamics of the protein folding is sparsely addressed in the literature, and very few datasets of protein pathways are available (Manavalan et al., 2019).

Molecular dynamics (MD) was developed in the 1950s (Alder and Wainwright, 1959) and, since then, MD has been the most important method for simulating the folding process of proteins (Levitt and Warshel, 1975). This approach is frequently used in proteomics research, not only for the study of the PFP but, also, for drug design (Hays et al., 2018) and the study of mechanisms leading to amyloid diseases (Lesgidou et al., 2018), cancer, diabetes, and Alzheimer's (Hsu and Schiøtt, 2019).

GROMACS and AMBER (Salomon-Ferrer et al., 2013; Abraham et al., 2015) are software packages widely used in the literature for MD simulations. They gained popularity among researchers by their robustness and flexibility. Both packages preserve the sample features by using generalized ensemble approaches during the simulation, such as replica exchange (RE) (Sugita and Okamoto, 1999), and umbrella sampling (US) (Torrie and Valleau, 1977) methods. However, in such ensembles, the spatiotemporal evolution of the folding trajectory is lost in favor of a faster sampling of the energy landscape. Among the MD variants, the canonical ensemble approach preserves the Newtonian dynamics of the protein trajectory (Stillinger and Head-Gordon, 1995; Rapaport, 2004) and, therefore, it is useful for the temporal analysis of

* Corresponding author.
E-mail addresses: leandrotakeshihattori@gmail.com (L.T. Hattori), www.bruna.a.p@gmail.com (B.A. Pinheiro), frigori@utfpr.edu.br (R.B. Frigori), hslopes@utfpr.edu.br (H.S. Lopes).

https://doi.org/10.1016/j.compbiolchem.2020.107301
Received 12 February 2020; Received in revised form 25 May 2020; Accepted 28 May 2020
Available online 05 June 2020

such systems. Moreover, this approach has been barely used together with coarse-grained (CG) models in the PFP literature (Benítez and Lopes, 2012; Benítez, 2015), which could enable much larger molecular models to be studied.

From the numerical perspective, at each iteration, the MD algorithm computes all the forces acting on each atom of a protein and, then, their updated positions in the 3D space resulting from the application of those forces. Since MD implies a huge computational burden for solving iterative equations, it seems to be appropriate to apply parallel methods, therefore minimizing the overall simulation time. In recent years, several computational approaches have been proposed to optimize the MD method (Salomon-Ferrer et al., 2013; Abraham et al., 2015), including parallelism support with GPUs (graphics processing units) (Phillips et al., 2011; Spellings et al., 2017; Yang et al., 2018). Although the parallelization of the DM method decreases the computational time of simulations, the larger the number of the beads representation of the model, the higher the computational power required to run simulations (Kmiecik et al., 2016; Kobayashi et al., 2017).

A successful strategy in the heart of multiscale modeling of proteins (Tozzini, 2009), frequently used for decreasing the computational complexity of the MD algorithm, employs CG models for representing proteins (Pierri et al., 2008; Poursina and Anderson, 2014; Hattori et al., 2020). Such modeling treats a set of atoms as less or just one element at a higher level of abstraction, which is suitable for understanding the essential aspects at the mesoscale level. The use of less complex representations leads to a significant reduction of the interaction between the elements of a protein chain, thus requiring less computational power (Llanes et al., 2016). Despite this gain, for long protein chains, multi-protein interactions, or when large amounts of computational experiments are required, the simulation is still challenging (Schneider and Müller, 2019).

There are many variants among CG models, for instance, according to the structure constraints, or the number of beads and force fields. Discrete models restrict the protein conformation in a lattice (Yanev et al., 2011; Hattori et al., 2020). On the other hand, off-lattice models allows continuous bond and torsion angles (Stillinger and Head-Gordon, 1995; He et al., 2013). The number of beads to represent each amino acid is another parameter, which can range from a more generalistic approach, such as the one-bead model (Stillinger and Head-Gordon, 1995; Cieplak and Hoang, 2000), to a more realistic representation using five or six beads that describe with more details mainly the side chain feature (Pierri et al., 2008; Tian et al., 2011). The force field can be derived either from physical or statistical analysis. The former includes electrostatics (Sheinerman et al., 2000), Lennard-Jones, torsion, and bond energies (Stillinger and Head-Gordon, 1995), whilst the latter includes frequency of atom contacts (Tanaka and Scheraga, 1976), short-range interactions (Amir et al., 2008), and hydrogen bonding (Levy-Moonshine et al., 2009; Kmiecik et al., 2016).

The $C\alpha$ represents the center of mass of each natural alpha amino acid, and the distance between two $C\alpha - C\alpha$ is near to 3.8 Å (Onofrio et al., 2014). Based on the definition of the above reported Calpha-Calpha distance, CG models are simplified models in which the amino acids are replaced by single beads. Each bead is positioned approximately in the center of mass of the corresponding amino acid, at the level of the Calpha of the native protein residue (Pierri et al., 2008; Onofrio et al., 2014).

Thanks to the growing computational power available nowadays, very large amounts of raw data can be produced. However, this requires software approaches to analyze data and create knowledge from it (Hourdel et al., 2016; Razban et al., 2018). In this context, the development of tools that enable the understanding of the information embedded in the folding pathways of proteins may be of great importance for research, allowing the analysis of the energy funnel (Dill and MacCallum, 2012), as well as identifying anomalies during the folding process (Brezovsky et al., 2016; Jurcik et al., 2018). Following this trend, we are introducing the PathMolD-AB software package. This tool simulates the spatiotemporal pathways of protein foldings and enables the visualization of the protein structure, energy, and radius of gyration along the folding process in a video format.

In this work, the folding process of four distinct proteins were simulated and analyzed as case studies. The MD simulation generated large datasets of protein folding pathways, thus encouraging research on computational methods for the PFP (Benítez, 2015; Hattori et al., 2018; Reinders et al., 2018). The availability of such datasets and software enables studies in correlated physico-chemical areas, such as predicting the structure based on energy and compactness, as well as predictions of amino acid contacts (Geng et al., 2019; Hanson et al., 2018) and so inferring realistic experimental B-factors of proteins by elastic network models (Mendonça et al., 2014).

The main contributions of the PathMolD-AB software package include:

- A parallel canonical ensemble MD using a CG model running in a master-slave CPU-GPU architecture.
- Analysis of the impact of different protein sizes in the performance of the proposed approach.
- A dataset of spatiotemporal pathways of protein folding, suitable for the study of the PFP.
- A method for comparing the predicted structures and the protein structure from the Protein Data Bank (PDB).

This paper is organized as follows: Section 3 describes in detail the PathMolD-AB software, including the CG model for representing proteins, the parallel molecular dynamics algorithm, the pathways datasets generated for four protein chains, and the method for comparing structures. Next, Section 4 shows the computational experiments and the analysis of the dataset created. Finally, in Section 5 conclusions and future directions are pointed out.

2. The 3D-AB off-lattice model

Depending on the biological information intended to be inferred from computational simulations, the amino acids in proteins, also known as residues, and their surrounding solvent shall be explicitly described at the atomic level (Tozzini, 2009). As mentioned before, this approach leads to a computationally expensive approach (Ngo et al., 1994) which may not always be required, for instance, when studying normal modes of proteins (Mendonça et al., 2014) or the general aspects of their structural phase transitions (Bachmann, 2014). For this reason, along time, several simplified CG models for representing proteins have emerged (Brown et al., 2003; Colombo and Micheletti, 2006; Lopes, 2008; Hills and Brooks, 2009; Kmiecik et al., 2016; Finkelstein, 2018), including the 3D-AB model, proposed by Stillinger and Head-Gordon (1995). This toy model turned out to be a flexible representation, compared to other popular lattice models, since it allows more arrangements of the structure (Pierri et al., 2008). Therefore, simulations with the 3D-AB model demand a lower computational cost, compared to atomic models. For instance, in aggregation studies, where it is required a higher computational effort, this model enabled realistic simulations of fibrillar aggregates (Frigori et al., 2013; Frigori, 2014, 2017). Nowadays, the model has been used in many benchmark works for the PSP problem (Lin et al., 2018; Zhou et al., 2018).

In the 3D-AB model, the residues are simplified and represented by spheres which, in turn, are implicitly categorized according to their affinity with water: either hydrophobic (represented by the letter “A”) or polar (represented with “B”). These features are fundamental for the formation of the native structure of proteins (Pierri et al., 2008). The distance between a given residue and the next one in the chain is always constant, equal to one. Such a constraint helps to decrease the computational load for extensive simulations. The gradient of the potential energy function (E_p) associated with the 3D-AB model, which ultimately drives the folding of the protein, is computed by Eq. (1)

(Rapaport, 2004; Benítez, 2015):

$$f = \nabla u(r) = \nabla E_p(\hat{b}_i; \sigma) = \nabla(E_{\text{Angles}} + E_{\text{Torsion}} + E_{\text{LJ}}). \quad (1)$$

The equations of motion are given according to Newton's second law, as shown in Eq. (2), where, N represents the number of residues:

$$f_i = m\ddot{\vec{r}_i} = \sum_{i=j=1(j\neq i)}^{N} f_{\text{ij}} \quad (2)$$

According to Newton's third law, which implies $f_{\text{ij}} = -f_{\text{ij}}$, the forces need to be calculated only once for each pair of particles. In particular, in this work the AB model uses $m = 1$, following Benítez (2015).

The bond-angle generate forces between three points residues ($j = i - 2, i - 1, i$), and the corresponding energy (E_{Angles}) is given by Eq. (3):

$$-\nabla_{rj} u(\tau_i) = -\frac{\text{du}(\tau_i)}{d(\cos\tau)}\Big|_{\tau=\tau_i} f_j^{(i)}, \quad (3)$$

where $u(\tau)$ is the bond-angle potential, and $f_j^{(i)} = \nabla_{rj}\cos(\tau_i)$. As proposed by Rapaport (2004), when $\sum_j f_j = 0$, the bond-angle can be represented by Eq. (4):

$$\begin{aligned} f_{i-2}^{(i)} &= (c_{i-1,i-1}c_{\text{ii}})^{-\frac{1}{2}}\left[\left(\frac{c_{i-1,i}}{c_{i-1,i-1}}\right)\vec{b}_{i-1} - \vec{b}_i\right], \\ f_i^{(i)} &= (c_{i-1,i-1}c_{\text{ii}})^{-\frac{1}{2}}\left[\vec{b}_{i-1} - \left(\frac{c_{i-1,i}}{c_{\text{ii}}}\right)\vec{b}_i\right] \end{aligned} \quad (4)$$

where c is the scalar product between the bond vectors of the ith and the jth pair. This pair is expressed by $c_{i,j} = \vec{b}_i \cdot \vec{b}_j$, where $\vec{b}_i$ indicates the ith bond of the joins between the ith and $(i-1)$th residues.

The potential associated with the bond-angle force for the AB model (E_{angles}) is described as:

$$-k_1 \sum_{i=1}^{N-2} \hat{b}_i \cdot \hat{b}_{i+1}, \quad (5)$$

where $k_1 = -1$ (Irbäck et al., 1997). Given that the AB model restricts the unit distance between consecutive residues of the protein structure, the derivative used for the forces in Eq. (3) might be calculated using

$$E_{\text{Angles}} = u(\tau) = -k_1 \sum_{i=1}^{N-2} \hat{b}_i \cdot \hat{b}_{i+1} = \sum_{i=1}^{N-2} \cos(\tau_i). \quad (6)$$

The bond-torsion potential is associated with four consecutive residues. For instance, the torsion in the ith residue causes force in the $j = i - 2, i - 1, \ldots, i + 1$. When $\sum_j f_j = 0$, the torsion force can be expressed by the following equations (Rapaport, 2004):

$$\begin{aligned} \vec{f}_{i-2}^{(i)} &= \frac{c_{\text{ii}}}{q_i^{\frac{1}{2}}(c_{i-1,i-1}c_{\text{ii}} - c_{i-1,i}^2)}[w_1\vec{b}_{i-1} + w_2\vec{b}_i + w_3\vec{b}_{i+1}], \\ \vec{f}_{i-1}^{(i)} &= -\left(1 + \frac{c_{i-1,i}}{c_{\text{ii}}}\right)\vec{f}_{i-2}^{(i)} + \left(\frac{c_{i,i+1}}{c_{\text{ii}}}\right)\vec{f}_{i+1}^{(i)}, \\ \vec{f}_i^{(i)} &= \left(\frac{c_{i-1,i}}{c_{\text{ii}}}\right)\vec{f}_{i-2}^{(i)} + \left(\frac{c_{i,i+1}}{c_{\text{ii}}}\right)\vec{f}_{i+1}^{(i)}, \\ \vec{f}_{i+1}^{(i)} &= \frac{c_{\text{ii}}}{q_i^{\frac{1}{2}}(c_{\text{ii}}c_{i+1,i+1} - c_{i,i+1}^2)}[w_4\vec{b}_{i-1} + w_5\vec{b}_i + w_6\vec{b}_{i+1}], \end{aligned} \quad (7)$$

where:

$$\begin{aligned} w_1 &= c_{i-1,i+1}c_{\text{ii}} - c_{i-1,i}c_{i,i+1}, \\ w_2 &= c_{i-1,i-1}c_{i,i+1} - c_{i-1,i}c_{i-1,i+1}, \\ w_3 &= c_{i-1,i}^2 - c_{i-1,i-1}c_{\text{ii}}, \\ w_4 &= c_{\text{ii}}c_{i+1,i+1} - c_{i,i+1}^2, \\ w_5 &= c_{i-1,i+1}c_{i,i+1} - c_{i-1,i}c_{i+1,i+1}, \\ w_6 &= -w_1, \\ q_i &= (c_{i-1,i-1}c_{\text{ii}} - c_{i-1,i}^2)(c_{\text{ii}}c_{i+1,i+1} - c_{i,i+1}^2). \end{aligned} \quad (8)$$

According to Irbäck et al. (1997), the potential associated by the torsion-angle (E_{Torsion}) force for the AB model is described by Eq. (9), where $k_2 = -0.5$.

$$E_{\text{Torsion}} = -k_2 \sum_{i=1}^{N-3} \hat{b}_i \cdot \hat{b}_{i+2} \quad (9)$$

The Lennard-Jones potential represents the interactions between residues based on their distance and hydrophobicity. Its gradient is defined by Eq. (10).

$$f_{\text{ij}} = \nabla E_{\text{LJ}} = 48\cdot\varepsilon(\sigma_i, \sigma_j)\left(r_{\text{ij}}^{-14} - \frac{1}{2}r_{\text{ij}}^{-8}\right)\cdot\vec{r}_{\text{ij}}, \quad (10)$$

where the distance between amino acids i and j is represented by r_{ij}, and $\varepsilon(\sigma_i, \sigma_j)$ weighs the interaction between amino acids based on hydrophobicity interaction. For example, hydrophobic interactions is weighted equal to 1.0, and all other interactions are weighted equal to 0.5 (Irbäck et al., 1997), as shown in Eq. (11).

$$\varepsilon(\sigma_i, \sigma_j) = \begin{cases} 1 & \text{if AA interaction,} \\ 0.5 & \text{if BB or AB interactions.} \end{cases} \quad (11)$$

Due to the constraints imposed on the model used in this work by the unit distance between subsequent residues of the chain, we used the Shake algorithm (see Algorithm 1) for updating the estimated coordinates (r) using a correction factor (γ). The velocities are also adjusted using the same approach, where the mass of each residue is equal to one ($m = 1$).

Algorithm 1. Shake algorithm (Benítez, 2015).

Start
Correction of coordinates:
$\gamma \longleftarrow \frac{\vec{r}_{\text{ij}}^{2'} - b_i^2}{4(\delta/2)^2(1/m_i + 1/mj)\vec{r}_{\text{ij}}\cdot\vec{r}'_{\text{ij}}}$
while $|\gamma| < 10^{-k}\cdot b_i^2$ **do**
$\vec{r}'_i \longleftarrow \vec{r}_i - \gamma\vec{r}_{\text{ij}}$
$\vec{r}'_j \longleftarrow \vec{r}_j + \gamma\vec{r}_{\text{ij}}$
$\gamma \longleftarrow \frac{\vec{r}_{\text{ij}}^{2'} - b_i^2}{4(\delta/2)^2(1/m_i + 1/mj)\vec{r}_{\text{ij}}\cdot\vec{r}'_{\text{ij}}}$
end while
Correction of velocities:
$\gamma = \frac{\dot{\vec{r}}_{\text{ij}}\cdot\vec{r}_{\text{ij}}}{2\vec{r}_{\text{ij}}^2}$
while $|\gamma| < 10^{-k}\cdot b_i^2$ **do**
$\dot{\vec{r}}_i \longleftarrow \dot{\vec{r}}_i - \gamma\vec{r}_{\text{ij}}$
$\dot{\vec{r}}_j \longleftarrow \dot{\vec{r}}_j + \gamma\vec{r}_{\text{ij}}$
$\gamma = \frac{\dot{\vec{r}}_{\text{ij}}\cdot\vec{r}_{\text{ij}}}{2\vec{r}_{\text{ij}}^2}$
end while

3. Description of PathMolD-AB

Fig. 1 presents an overview of the proposed end-to-end framework, called PathMold-AB. The core of the framework comprises three steps/modules, plus other add-ons for specific analyses. In step 1, the input file for the computational simulations is generated from raw data acquired from the Protein Data Bank (PDB).[1] In this step the Cartesian coordinates of the $C\alpha$ of all amino acids are extracted and the distance between $C\alpha - C\alpha$ is rescaled to one, aiming at comparing to the CG model. In step 2, the folding simulations are accomplished and pathways data are generated, based on parallel and sequential models of the canonical MD method using a CG model. For further details about the input and output files, see Appendix B. In step 3, results of the simulation are compared with the rescaled biological structure of the protein

[1] http://pdb.org/

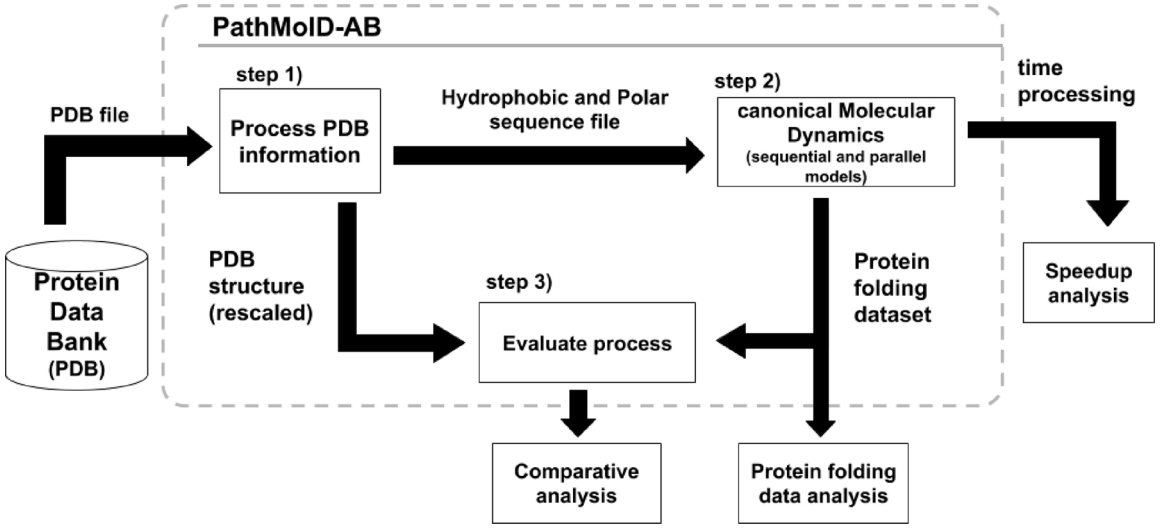


Fig. 1. An overview of the proposed framework (PathMolD-AB).

(performed in step 1), aiming to compare the predicted structures with the corresponding "biological" structure. In the following sections, each of these steps will be detailed.

3.1. PDB data processing

To properly extract useful information from the PDB files, two procedures are necessary: first, the AB sequence is obtained (for simulating the folding pathways of the protein); second, the rescaled biological structure is constructed (for comparing with the predicted structures).

The conversion of an amino acid sequence to the corresponding hydrophobic-polar (AB) sequence is shown in Algorithm 2. We used the Python programming language together with the Biopython[2] framework. Starting from a PDB ID, the program downloads the PDB file and extracts the sequence of amino acids. Next, this sequence is converted to the AB model using a hydrophobicity conversion table. Following a previous work (Benítez, 2015), here we used the hydrophobicity scale proposed by Alberts et al. (2002) (see Table 1) for converting the 20 different amino acid types to either A or B. Next, the AB sequence is saved in a file together with other features to run the MD simulation (see Appendix B).

Algorithm 2. Protein sequence conversion procedure. Based on Hattori et al. (2020).

```
Input PDB ID
Download PDB File
Read PDB File
for i = 0: N do
   Extract Amino Acid AA_i ∈ SEQRES
   Add AA_i to Sequence[i]
end for
Read AB Classification Table
for i = 0: N do
   if Sequence[i] == 'A' then
      AB_Sequence[i] ⟵ 'A'
   else
      AB_Sequence[i] ⟵ 'B'
   end if
end for
Save AB_Sequence
```

In the rescaling process, the Cartesian coordinates of the protein structure are extracted from PDB file and, from them, the coordinates of the $C\alpha$ of each amino acid (Pierri et al., 2008). The distances between each consecutive C_α is rescaled, dividing by 3.8 Å, to obtain the normalized distance (equal to 1) according to the 3D-AB off-lattice model (Chan and Dill, 1990; Pierri et al., 2008; Kolinski, 2011; Onofrio et al., 2014). Then, it is obtained the target structure represented on the CG model to be compared with structures predicted (Pierri et al., 2008).

Table 1
Hydrophobicity scale by Alberts et al. (2002).

Amino acid	Hydrophobicity classification	Amino acid	Hydrophobicity classification
ALA	A	MET	A
CYS	A	ASN	B
ASP	B	BRO	A
GLU	B	GLN	B
PHE	A	ARG	B
GLY	A	SER	B
HIS	B	THR	B
ILE	A	VAL	A
LYS	B	TRP	A
LEU	A	TYR	B

3.2. Parallel molecular dynamics

Based on previous works (Benítez, 2015; Benítez and Lopes, 2012), the MD implementation of PathMolD-AB uses the canonical NVT ensemble, where the number of residues (N) and volume (V) are constants, and the temperature (T) is controlled at a specific value. The parallelization proposed in this work is based on a CPU-GPU master-slave computation model. A master process running on CPU manages the sequential part of the algorithm, whilst slave processes running on GPU cores execute the main computations in parallel.

The initial procedures are naturally serial or require low computational effort, therefore they are run on a CPU. The first step is the generation of a structure randomly positioned in the space. Next, the main part of the MD algorithm is the computation of the torsion, bond, and Lennard-Jones energies, as described in Section 2. The computation of each energy function is parallelized in GPU, such that, for each energy term, the computation is assigned to a thread. The results are then stored in an array position. After computing these energies, the partial energies are summed by parallel reduction to sequential addressing, as shown in Fig. 2. The reduction algorithm is accomplished by a paired sum of the array positions, and these computations are done in parallel.

[2] http://biopython.org/

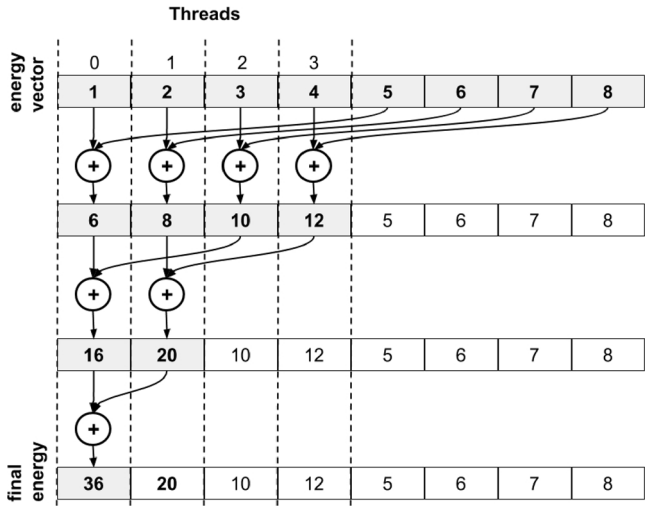


Fig. 2. Parallel reduction to sequential addressing.

The results of the sums are saved in the first indices of the array. This process takes place iteratively until all values are summed up to a single position of the array.

In the sequence, velocities and accelerations of all the residues are computed. These computations are independent each other and they can be performed in parallel. Then, as a consequence of the physical forces acting on the residues, they are pushed to another position in the 3D space, and, again, such computation can be accomplished in parallel in GPU. Next, the temperature of the system is adjusted by using the weak coupling to a thermal bath method, as proposed by Berendsen et al. (1984). Finally, the geometric constraints are applied to the structure in order to adjust coordinates and velocities (see Algorithm 1).

Additionally, for evaluating the compactness of protein conformations, the radius of gyration (Khokhlov, 1994) is computed at each step_size. The smaller the radius of gyration, the more compact is the set of residues. Three radii of gyration are provided RgAll (all the structure), RgH (only hydrophobic residues), and RgP (only polar residues). Notice that the observation of the temporal changes of RgH and RgP may indicate the formation of the hydrophobic core, typical of globular aqueous and cytosolic proteins (not membrane proteins) (Chan and Dill, 1990; Dill and MacCallum, 2012). Eq. (12) shows how the radius of gyration is computed:

Table 2
Information about the protein sequences used to generate the datasets.

ID	# amino acids	AB sequence
3FIBO	3	$AB^2(AB^2AB)^2$
GB1	6	$AB^3A^3BAB^2ABAB^5(A^2B)^2AB^2A^2$ $(B^3A)^2(AB)^3(B^3A)^2BAB^2$
PLC	9	$(ABA^5BB)^2(AB)^2A^2B^2A^3B^3A^4B^2$ $A^3B^4A^2BA(AB)^2(BA)^2B^4A(AB)^32$ $(BA)^4A(B^2BA)^2(BA)^2B^2A^6(BA)^2B$
NAZ	29	$(BA^2)^2A^2B^8(A^2B)^2(AB)^3(BA^2)$ $(BA)^2B^3(AB)^2(BA^2)^3(B^2A)^2A^5BA$ $B(BA^2)^2B^7A^2B(BA)^2A^3(BA)^2$ $(AB)^2(BA)^2A^2B^2A^4BA^8B^4A(BA^2)^3$ $B^2A^4B^2A^3B^3A^3(BA)^3(A^2B)^3$ $BAB^2A^4(BA)^4B^3AB^4A^3(AB)^3BA^2$ $B^2AB^3(BA)^2(AB)^2(B^2A)^2BA^2B^3$

$$\mathrm{RgAll} = \sqrt{\frac{\sum_{i=0}^{N-1}[(x_i-\overline{X})^2+(y_i-\overline{Y})^2+(z_i-\overline{Z})^2]}{N}}, \quad (12)$$

where x, y and z represent the Cartesian coordinates of each residue i, $\overline{X}$, $\overline{Y}$ and $\overline{Z}$ the average of each Cartesian coordinate, and N is the number of residues of the sequence.

At each pre-defined number of iterations (step_size) the program saves the state of the protein as a report file, saving the structure, energy, and the radii of gyration of the protein. The whole procedure is repeated until a stop condition is satisfied, such as a time-out criterion (a given number of iterations ($t_{\max}$) is reached) or a quality criterion (e.g. when the E_p stabilizes according to a predefined threshold) (Benítez, 2015). Algorithm 3 shows the main execution steps of PathMolD-AB.

Algorithm 3. Main execution steps of PathMolD-AB. The shaded lines are executed in parallel in GPU, whilst the other are executed in CPU.

```
Set the initial conditions of all particles of the proteins:
positions r_i(t_0), velocities v_i(t_0) and accelerations a_i(t_0)
for t = 0 : t_max do
    Compute Lennard-Jones energy
    Compute torsion energy
    Compute bond energy
    Summarize the partial energy (Parallel reduction to
sequential addressing)
    Update positions, velocities, and accelerations
    Adjust temperature (thermostat)
    Compute geometric constraints (Shake algorithm)
    if (t mod step_size) == 0 then
        Compute radii of gyration
        Save the state of the protein in a report file (struc-
ture, potential energy, and radii of gyration)
    end if
    t ← t + 1
end for
Store data
```

3.3. Generation of datasets for studying the protein folding dynamics

Using the PathMolD-AB software, four datasets of protein folding trajectories were produced as case studies. Four proteins were simulated, one artificially created and three real-world proteins with a growing number of amino acids, as shown in Table 2, and detailed as follows:

- 13FIBO: it has 13 amino acids, and was artificially created by Stillinger and Head-Gordon (1995), by distributing the hydrophobic amino acids according to the Fibonacci sequence.
- 2GB1[3] : this protein is in the group of the G proteins, which exerts signal transduction functions. The dysfunction of this protein is linked to diseases such as schizophrenia in humans (Mirnics et al., 2001).
- 1PLC[4] : this protein performs the function of electron transportation, which is related to the process of energy production in the cell. Its functional impairment results in cell death (Watabe and Nakaki, 2007).
- 5NAZ[5] : this is a globular structural protein of collagen, and it is related to the Goodpasture's and Alport's syndromes (Casino et al., 2018).

For each protein, a dataset was generated with 1000 (for 13FIBO, 2GB1 and 1PLC) or 500 (for 5NAZ) different pathways. Due to the length of the last protein, less simulations were done. As mentioned

[3] http://10.2210/pdb2GB1/pdb
[4] http://10.2210/pdb1PLC/pdb
[5] https://www.rcsb.org/structure/5NAZ

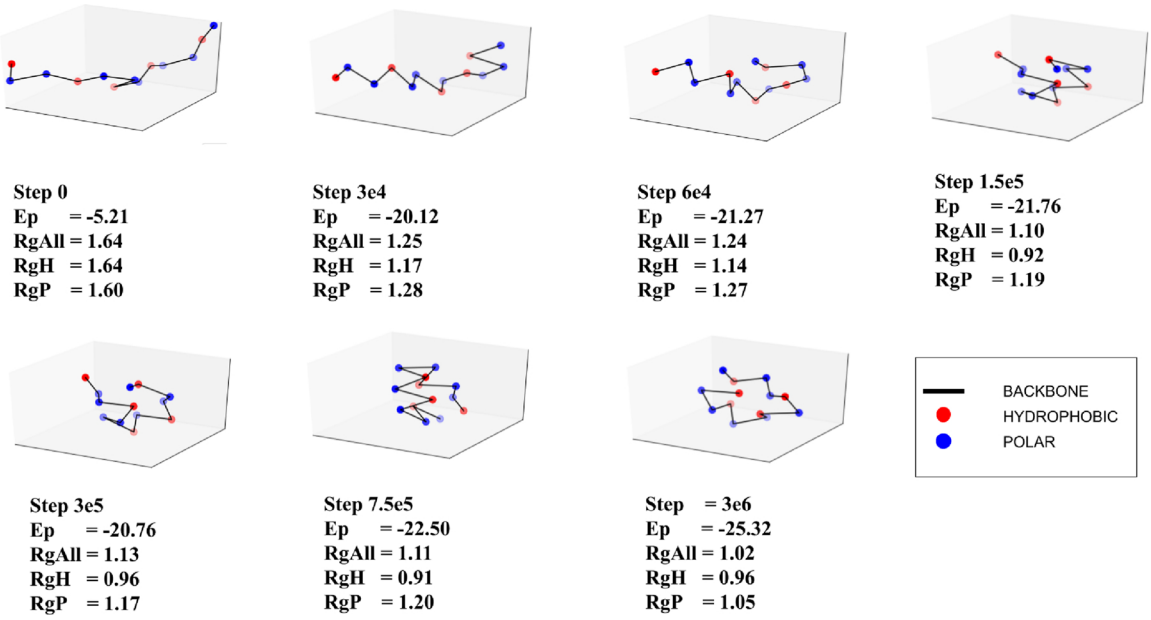


Fig. 3. Sample of a pathway for the protein 13FIBO.

before, all simulations start with structures randomly initialized in the 3D space, so as to enforce a high diversity of pathways, each one leading to the native conformation of the protein.

To guarantee a reliable stabilization of the native structure, the maximum number of time-steps ($t_{\max}$) for the simulations of the 13FIBO, 2GB1 and 1PLC proteins were set to 3×10^6 iterations, and 1×10^8 for the 5NAZ protein. Consequently, for standardizing the number of spatiotemporal states per pathway in each dataset, the step_size for 13FIBO, 2GB1 and 1PLC was 3000, and 8000 for 5NAZ. For each pathway, 1000 folding states were recorded. Fig. 3 illustrates snapshots of protein folding states.

3.4. Comparison with the biological structure from the PDB

In this section, we aim at comparing the structure of the crystallized proteins found in the PDB with the corresponding structure predicted by the our approach (see Fig. 1). This is accomplished indirectly, by computing the radii of gyration of both structures and, also, in a direct way, by using the Kasbch-RMSD measure described below.

Following the prepossessing, in step two some simulations are accomplished to produce the protein pathway dataset using MD (see Sections 2 and 3.2). After the simulations, the data were organized in such a way to enable the comparison with biological structures. As presented in Section 3.3, the three real-world proteins included in this case study were: 2GB1, 1PLC, and 5NAZ. However, due to the lack of information about the coordinates of the 5NAZ protein residues in the PDB file, the scaling process was unfeasible for this protein. Thus, the analyses were performed only for the first 2GB1 and 1PLC.

The comparison of the rescaled PDB structure and the MD predicted structure is a problem that can be modeled as an orthogonal Procrustes problem (Gower and Dijksterhuis, 2004). Kabsch (1976) proposed an algorithm to solve this problem by approximating two matrices P and Q, which represent the spatial coordinates of the two structures. In this work, the movement allowed is only the rotation of P and Q. First, residues of P and Q are superposed and, then, a rotation is applied to minimize the difference between these two matrices, based on the root mean square deviation (RMSD) (Kravraki, 2007), as shown in Eq. (13).

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (P_{\text{ix}} - Q_{\text{ix}})^2 + (P_{\text{iy}} - Q_{\text{iy}})^2 + (P_{\text{iz}} - Q_{\text{iz}})^2)}{N}}, \tag{13}$$

where N represents the number of amino acids of the protein, i is the ith amino acid, and x, y, z are the Cartesian coordinates of each amino acid.

4. Results and analysis

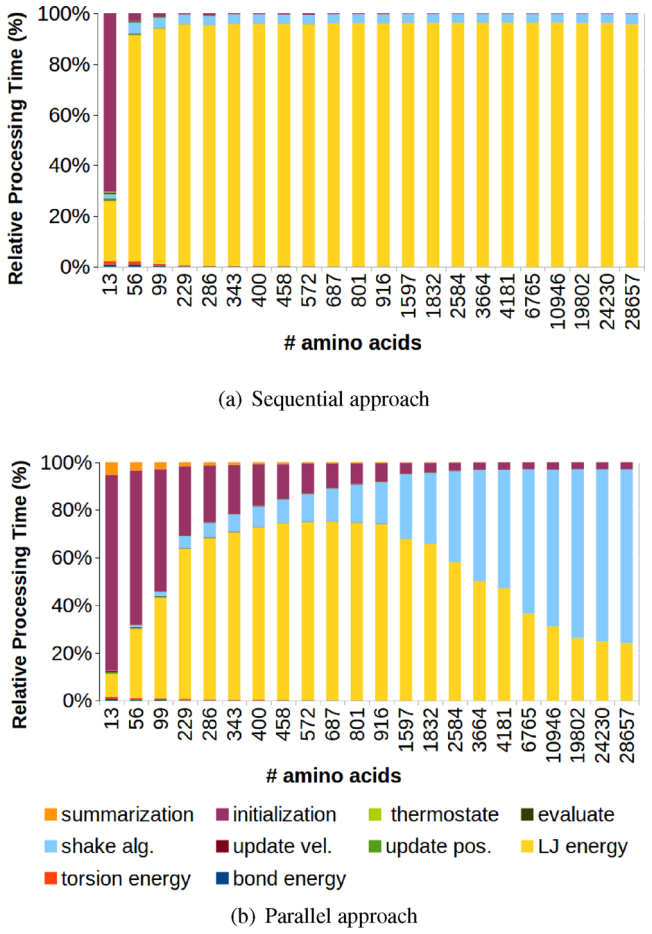
Experiments were run in a workstation running Ubuntu 18.04 LTS operating system, composed by an Intel i7-8700 processor at 3.2 GHz, 32 GBytes RAM, and a Nvidia Titan-Xp GPU (12 GBytes RAM DDR5 and 3840 CUDA cores at 1.6 GHz). The code was developed using the standard C programming language, and for the parallelization of the code, the CUDA library was used.

4.1. Performance of the parallel PathMolD-AB

This section aims at verifying the computational efficiency of the proposed parallel MD method of the PathMolD-AB software package. The reference for comparison is a purely sequential approach, previously introduced by Benítez and Lopes (2012).

The sequences used to evaluate the performance of the proposed parallel MD were the four proteins shown in Section 3.3. Other synthetic sequences, ranging from 286 to 28,657 amino acids, were also used specifically for assessing the scalability of the parallel approach.

The experiments performed were based on 3000 iterations of the MD method for both serial and parallel approaches. The comparison metric used was the speedup, that is, the processing time of the sequential approach divided by the corresponding processing time of the parallel approach. Fig. 4(a) and (b) shows the processing time of the MD functions (summarization, initialization, thermostat, evaluate, shake algorithm, update velocity, update position, Lennard-Jones energy, torsion energy, and bond energy) for both approaches. Considering the sequential approach, the most time-consuming part is computation of the LJ function. Only for the smallest protein (with 13 amino acids), the processing time of the initialization function exceeded



(a) Sequential approach

(b) Parallel approach

Fig. 4. Processing time of the PathMolD-AB functions, for both, sequential and parallel approaches.

the other functions. On the other hand, for the parallel approach, the processing time of the LJ function decreased significantly when compared to the sequential approach.

We observed that the computation of the geometric constraints (see Algorithm 1) tends to increase when compared to the sequential approach. Unfortunately, this algorithm is not parallelizable. That is, the adjustment of the $(i + 1)$th residue depends on the adjustment of the previous one. Also, the velocities update depends on the adjustment of the coordinates.

The speedup of the parallel model relative to the sequential model was evaluated using synthetic sequences of different growing sizes (see Fig. 5). Surprisingly, a speedup lower than one (the sequential approach was faster than the parallel version) was observed for sequences smaller than 99 amino acids, such as 13FIBO and 2GB1. Actually, this happened due to the time required for the communication between GPU-CPU and, more specifically, by the initialization function (see Fig. 4(b)). On the other hand, for sequences larger than 99 amino acids, such as 1PLC, a speedup higher than one was obtained, indicating that the parallel approach is faster than the serial one. The largest sequence used in this experiment had 28,657 amino acids, and the corresponding speedup was 23.27. This result clearly suggests that the parallel approach has high scalability for large sequences, when compared with the sequential approach. Regardless of the approach used, sequential or parallel, the processing time tends to follow a logarithmic curve, as shown in Fig. 4(b). This figure allows inferring that the function that most influences the speedup decay is that in charge of processing the geometric constraints (the Shake algorithm), which increases for the larger sequences, exceeding the time required by the LJ function.

Despite the speedup decay of the parallel approach for the large protein sequences, most of the real biological proteins are quite below that upper bound (Brocchieri and Karlin, 2005; Tiessen et al., 2012). In fact, the statistical information extracted from PDB, shown in Fig. 7, corroborates that this improvement covers more than 92% of proteins currently deposited in PDB.

Fig. 6 shows the speedup values for the three energy functions of the PathMolD-AB (torsion, bond, and Lennard-Jones). The highest speedup value was achieved for the Lennard-Jones energy (see speedup LJ). This result indicates that the parallelization of the LJ function contributed the most to the overall speedup. In fact, this result is quite important considering that the computation of this energy is the most time-consuming in the sequential approach. Although the bond and torsion energies (see speedup torsion and speedup bond) achieved lower speedup than LJ, some improvement in the speedup can be observed for large sequences too. Overall, the parallelization of these two functions also helped to increase the speedup value of the approach.

4.2. Data analysis of the case study

As proposed in Section 3.3, we generated a dataset of protein folding pathways for four case studies: 13FIBO, 2GB1, 1PLC and 5NAZ.

A high diversity of initial conformations is required to show that, starting from any initial spatial position, the structures will evolve toward their native structure. Therefore, the initial structures were

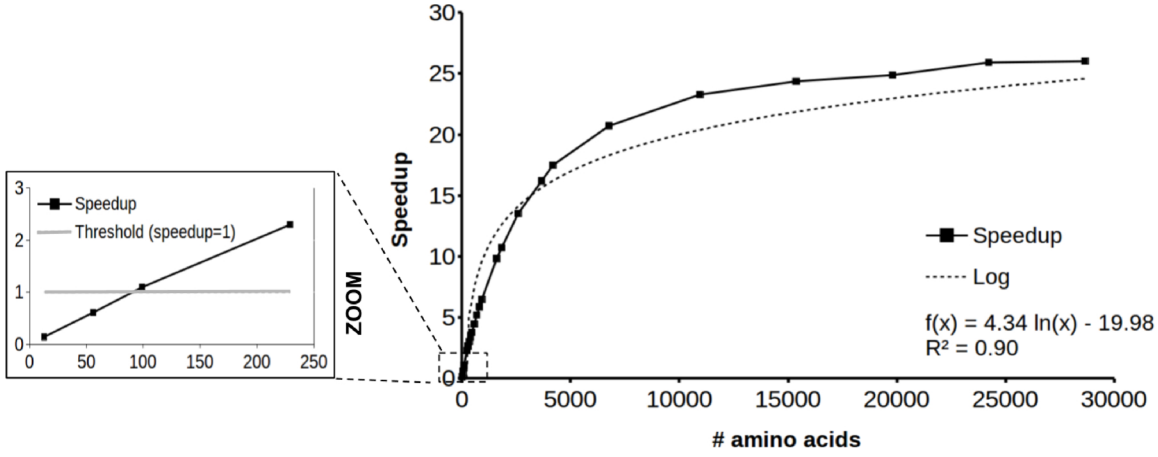


Fig. 5. Overall speedup for the simulation of a single pathway, considering the sequential and parallel approaches.

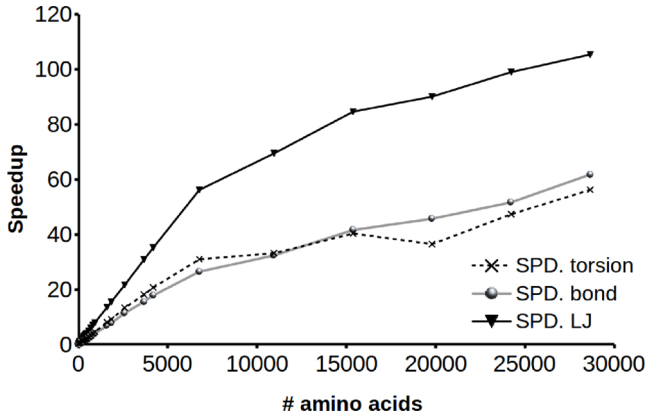


Fig. 6. Energy functions speedup for the simulation of a single pathway, considering the sequential and parallel approaches.

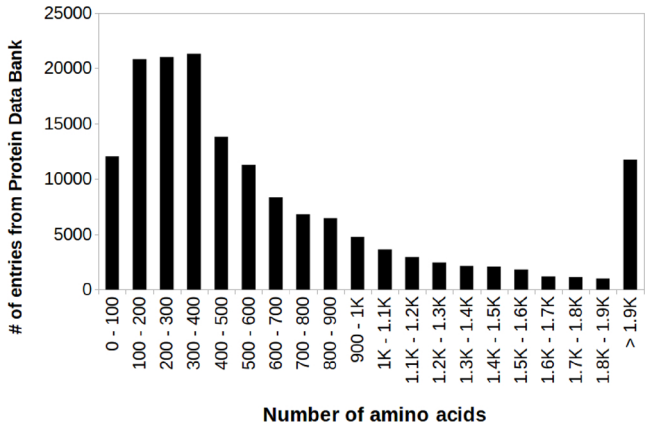


Fig. 7. Number of entries per protein size range. From https://www.rcsb.org/ (accessed in October 2019).

randomly initialized before running the PathMolD-AB simulation. In such situation, it is essential to evaluate how different the initial structures generated are and, conversely, how similar are the final ones after the simulation. For each case study, all the 1000 protein structures were compared one each other, in the first and the last step of the pathways. The comparison of two structures is not trivial, since they must be previously aligned using the Kabsch-RMSD method (see Section 3.4). Results were normalized in the range [0...1] and plotted in the heatmaps shown in Fig. 8, for the initial and final structures.

Each point of the horizontal and vertical axes of the heatmaps represent a protein structure at a given point of the pathway (in this case, either the initial or the final point). The darker the color in the heatmap, the closer to 1 it is according to the Kabsch scale, meaning that the structures tend to be more different. The opposite holds, meaning similar spatial structures.

As mentioned before, the values of the potential energy and the radii of gyration were also recorded along the pathway. They give additional insights about the compactness of the protein and convergence of the folding process toward the native structure of the protein. Fig. 9 illustrates the potential energy (E_p), normalized in the range [1...0], at each pathway time step. It is shown that the energy starts near one and decreases along the iterations until the number simulation.

Fig. 10 shows the radii of gyration (RgP, RgH, and RgAll) of the proteins, normalized in the range [0...1]. It is shown that, in the beginning, all the radii of gyration are high but, soon, decay exponentially, and later stabilize at low values.

Additional information is provided in Table 3. It presents the average and standard deviation values of the energy and the radii of gyration, computed at the final step of the pathways. Final values of RgH are lower than RgP, suggesting the formation of a hydrophobic core (Dill and MacCallum, 2012). Notice that the standard deviations are low for all cases, confirming that proteins converged to quite similar compact structures at the final step of the pathways, as previously shown by the heatmaps.

4.3. Comparison with biological structures

As mentioned in Section 3.4, we proposed a procedure for comparing the structures predicted by the MD method with structures re-scaled from the PDB.

Fig. 11 shows the results for the 2GB1 and 1PLC proteins in terms of RgAll, RgH, and RgP (see Eq. (12)). The results showed that the protein folding simulation yielded compactness values closer to the native structure. This behavior suggests that the method tends to bring the unfolded structure closer to the native biological structure. We also observed that the predicted structures tended to be more compact than those of the PDB, and the radii of gyration of hydrophobic and polar were not as distinct as those of the prediction. Possibly, the lower values of the compactness of the predicted structures may have been caused by the weight of the hydrophobicity interactions in Eq. (11). Overall, results suggest a further refinement of those parameters to improve the model representation. In addition, it depends on the degrees of freedom of their simplified systems and the convergence criterion of PathMolD-AB.

The predicted and the re-scaled biological structures were directly compared using the Kabsch-RMSD method (see Section 3.4), as shown in Fig. 12. Kabsch-RMSD values were observed to be more distinct in the initial iterations than in the final ones. Similarly, the standard deviation is higher in the initial iterations than in the final ones. These results reinforce the analysis of compactness presented before, and the conclusion that the simulation produces results structurally similar to the biological structure (see, also, the diagram of Fig. 13).

5. Conclusions

The protein folding problem (PFP) is still an open challenge in the area of computational biophysics, and it is related to unveiling how proteins fold toward their native (functional) structure. The mechanistic understanding of the PFP may shed light on the genesis of many human diseases related to misfolded proteins as well as to amyloidogenic aggregates.

MD is a widely used approach for simulating the mechanistic behavior that takes place during the protein folding. However, MD is computationally intensive and the processing time increases exponentially as the number of amino acids increases. This justifies the development of more efficient methods, such as the PathMolD-AB package proposed in this work. This software package uses MD with the canonical ensemble that deals with the Newtonian evolution of protein models. In addition, this software uses a CG model for representing proteins and a parallel master-slave computing architecture that enables experiments for tracking the spatiotemporal pathways of protein folding. Such pathways can be useful for analyzing the changes of the structure along with the folding and visualizing important events, such as misfolding and structural instability, typical of intrinsically disordered structures.

The parallel MD is faster than the sequential version for protein sequences larger than 99 amino acids. Above such a number of amino

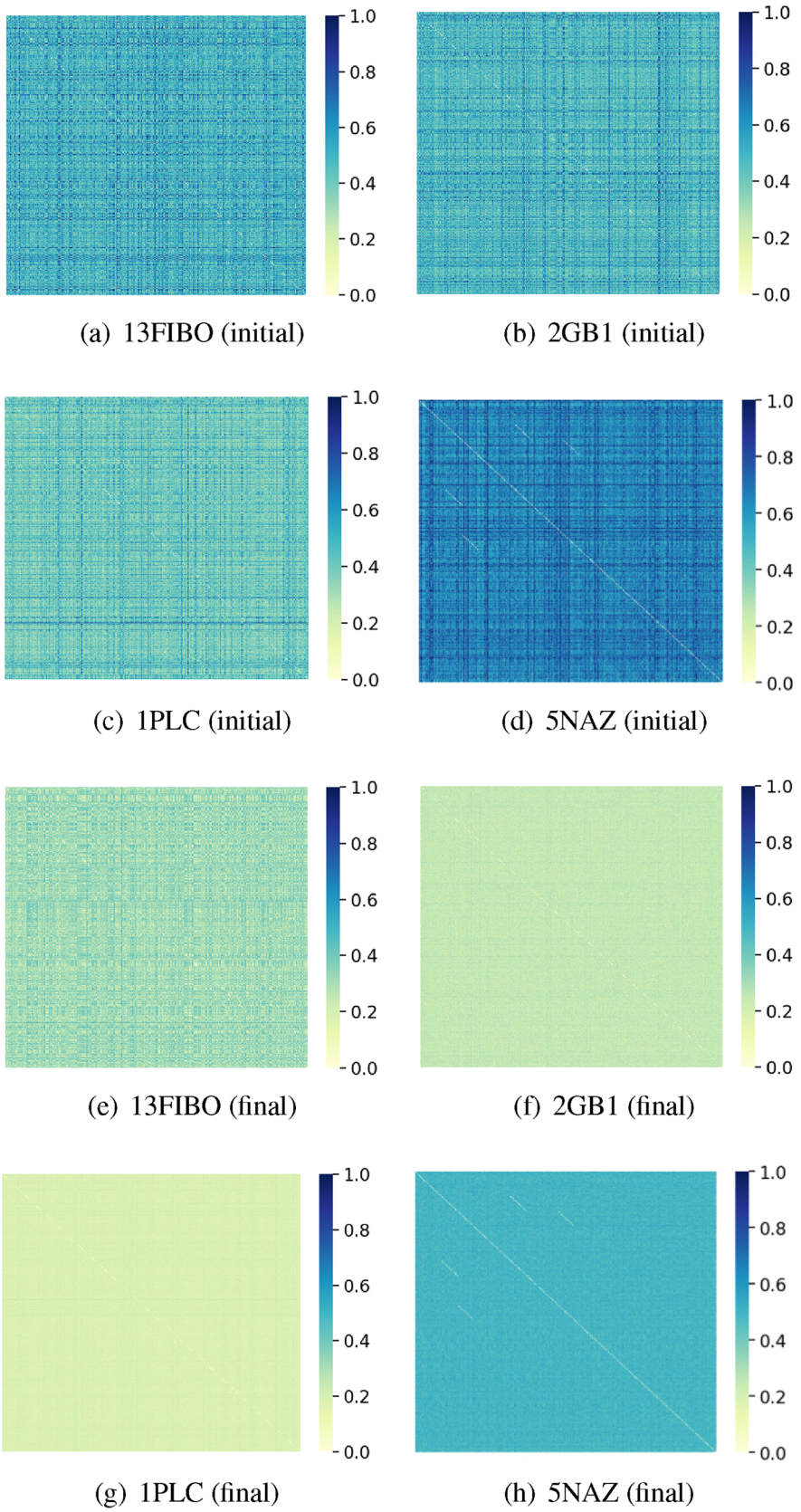
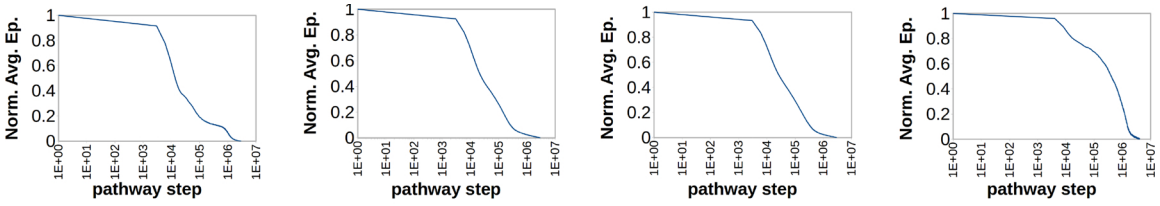
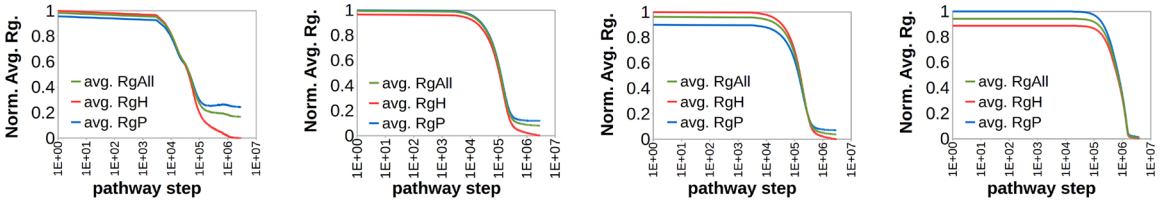


Fig. 8. (a–d) Normalized Kabsch RMSD between the 1000 initial structures of the four datasets, and the final structures similarity (e–h).



(a) 13FIBO (b) 2GB1 (c) 1PLC (d) 5NAZ

Fig. 9. Average potential energy (E_p) per iteration.



(a) 13FIBO (b) 2GB1 (c) 1PLC (d) 5NAZ

Fig. 10. Average radii of gyration (*RgAll*, *RgP* and *RgH*) per iteration.

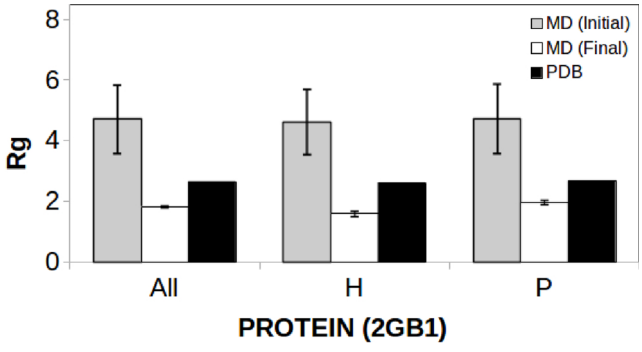
Table 3
Average and standard deviation energy and radii of gyration of the final state for the four proteins (13FIBO, 2GB1, 1PLC and 5NAZ).

	Protein structure predicted (avg. $\pm \sigma$)	
	13FIBO	2GB1
Ep	− 24.921 ± 0.831	− 156.117 ± 3.884
RgAll	1.080 ± 0.027	1.840 ± 0.035
RgH	0.896 ± 0.090	1.600 ± 0.093
RgP	1.164 ± 0.069	1.970 ± 0.058
	1PLC	5NAZ
Ep	− 331.246 ± 7.136	− 808.516 ± 12.08
RgAll	2.306 ± 0.080	3.192 ± 0.175
RgH	2.147 ± 0.120	2.929 ± 0.155
RgP	2.452 ± 0.081	3.443 ± 0.211

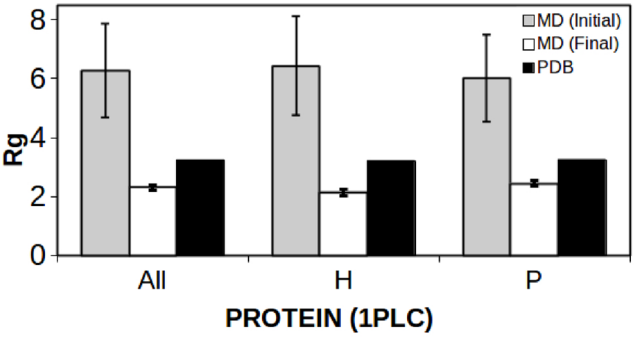
acids, the speedup increases significantly for the parallel version. We showed that, among the several functions of PathMolD-AB, the LJ function is the most computationally expensive. Notwithstanding, we achieved the highest speedup in this function, decreasing the bottleneck of MD.

The speedup curve suggests a logarithmic trend, that is, the larger the protein sequence, the more the PathMolD-AB slows down and stabilizes speedup. The decay of speedup is the result of the concurrency between processing threads in the CUDA cores of the GPU and the Shake algorithm. However, this is not a drawback, since the distribution of the lengths of proteins deposited in the PDB shows that the proposed software can be useful for simulating the folding of most of the biological proteins in the PDB.

PathMolD-AB was applied to case studies and generated a large amount of simulation data which analyzes indicated that at the final state of the spatiotemporal trajectories led to similar conformations, starting from distinct initial structures, as suggested by the energy funnel theory. The high similarity between structures in the final state, compared to the initial state, also indicate that the simulated folding pathways led to structurally similar final structures. In addition, the thermodynamic characteristics are coherent with the energy curve that



(a) 2GB1



(b) 1PLC

Fig. 11. Radii of Gyration of the crystallized structure (from the PDB) and predicted structure by PathMolD-AB, at the initial and final step of the simulation.

decays at the initial iterations and stabilizes later.

Furthermore, we showed that the predicted structures simulated by PathMolD-AB were similar to the re-scaled biological structures. Even considering that we used a CG model, such a “biological-like” validation is an important step toward more realistic simulations.

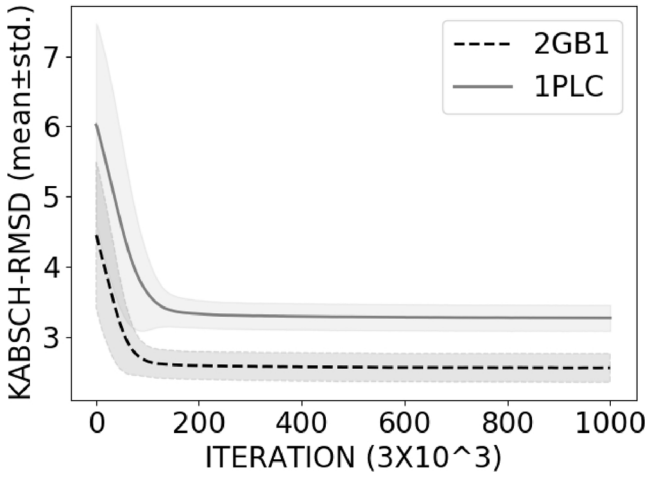


Fig. 12. Kabsch-RMSD (mean and standard deviation) between the biological sequence and the predicted structure along of the iterations.

The main drawback of our simulations is that the compactness of the predicted structures was smaller than those of the re-scaled biological structures. This fact suggests the need for optimization of the hydrophobicity interaction weights between the residues, as first proposed by Irbäck et al. (1997). A natural further step in this direction would consist of implementing, still, in the scope of CG models, knowledge-based structural information incorporated in the so-called "LBN-model" (Brown et al., 2003). The LBN-model is a straightforward generalization of the potential energy of the AB-model for hydrophilic, hydrophobic, and neutral residues that especially includes additional local (torsional) terms. This might enable more accurate structural analysis at a small extra computer burden, without requiring significant changes in the implementation of the MD-evolution equations.

Future works will explore other parallel MD approaches, such as the neighbor-list MD, so as to enable the generation of folding pathways more efficiently. The protein sequences presented at the critical assessment of methods of protein structure prediction (CASP) event will be included in future experiments. Also, other studies to fine-tune the methods will be carried out, such as the impact of the weights of the short-range interactions in the PathMolD-AB simulation in order to achieve results closer to the rescaled biological structure, as reported by (Onofrio et al., 2014).

Finally, we believe that the software developed, as well as the datasets created in this work, will be useful for fostering further research in the areas related to the PFP and MD. Therefore, both, software and datasets will be made available for research purposes.

Author contributions

Leandro Takeshi Hattori and Heitor Silvério Lopes: Conceptualization, methodology, software. Leandro Takeshi Hattori and Bruna Araujo Pinheiro: Data curation, writing – original draft preparation. Leandro Takeshi Hattori and Bruna Araujo Pinheiro: Visualization, investigation. Heitor Silvéio Lopes, Rafael Bertolini Frigori, César Manuel Vargas Benítez: Supervision. Leandro Takeshi Hattori and Bruna Araujo Pinheiro: Software, validation. Leandro Takeshi Hattori, Heitor Silvéio Lopes, Rafael Bertolini Frigori, César Manuel Vargas Benítez: Writing – reviewing and editing,

Conflict of interest

The authors declare that there is no conflict of interest.

Acknowledgments

L.T. Hattori thanks CAPES and CNPq for the PhD and IC scholarships, respectively; C.M.V. Benítez and H.S. Lopes thank CNPq for the research grants no. 311785/2019-0, 311785/2019-0, 424957/2016-7, 311778/2016-0, and 440977/2015-0. R.B. Frigori thanks the Brazilian National Laboratory for Scientific Computing (LNCC) for the research grant (PHAST2) at the Santos Dumont supercomputer. All authors thank NVIDIA Corporation for the donation of the GPU Titan-Xp used in this work.

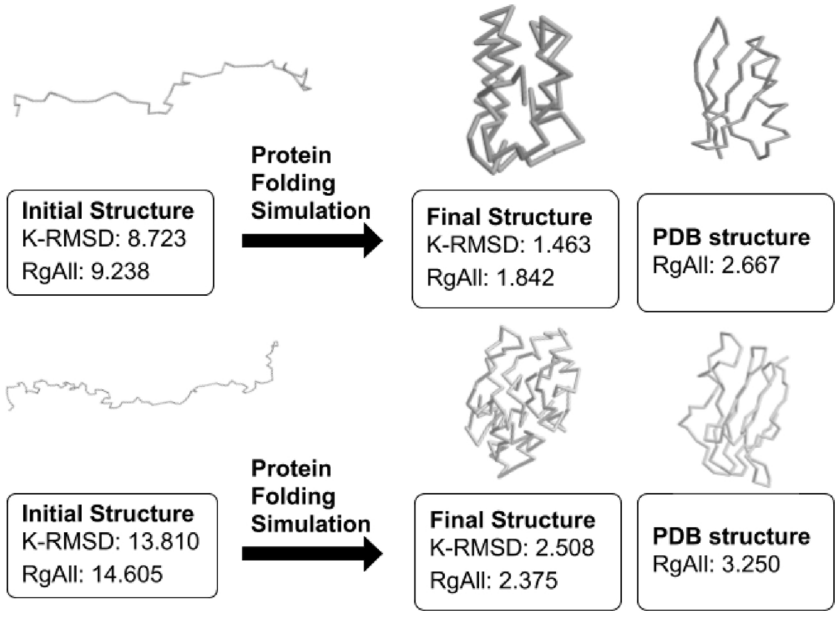


Fig. 13. Sample of a folding pathway simulation of 2GB1 and 1PLC proteins compared with the re-scaled biological structures from the PDB.

Appendix A. Running parameters

In principle, PathMolD-AB just needs to read an input text file before starting any run. However, the software can be reconfigured before the compilation by editing the following files:

- `define.h`: the main configuration file that sets the MD parameters and constants. This includes, but it is not limited to, the energy variables (E_{Torsion}, E_{Angle} and E_{LJ}), radii of gyration variables (RgAll, RgH and RgP), the number of the MD iterations, as well as the maximal size and number of proteins.
- `functions.h`: contains the routines declaration (utilities, initialization, power functions, assembly control, as well as those related to the I/O process).
- `main.c`: this is the main routine of the simulation module. It receives the program arguments such as the input file, GPU type and the seed for the initialization.
- `function.cu`: contains the implementations of the routines defined in the `functions.h` file is contained in this file. For instance, the routines used for the simulation of MD, contains all GPU communication, I/O functions, and ensemble control.

To improve the usability of the program, a script file (`Makefile`) was developed for the program execution. This script can be run in the command line using `make all`. All procedures available in this package will be run with that command, including: download of the protein file from PDB file, extraction of the AB sequence, creation the MD input file, compilation the parallel and sequential models of the MD program, execution of the simulation in both models, and generation a visualization movie of the protein trajectory.

Appendix B. Input and output files

To simulate the protein folding trajectories using PathMolD-AB it is necessary to configure only an input text file containing information about the simulation and the protein to be folded, as shown in Table 4. Other control parameters of the program were centralized in the function loadFile (in file `function.cu`) for future modifications, such as the Shake algorithm (to deal with algorithm' constraints), the mass and distance between each residue of the model (mass and bond_len).

To obtain the AB sequence information, the Python script `ab_sequence.py` is provided to extract and convert the amino acids sequence directly from a FASTA file (downloaded from the PDB) to an AB sequence based on the hydrophobicity scale proposed by Alberts et al. (2002).

The output text file generated by PathMolD-AB contains spatiotemporal information about the residues of the protein along the folding process. At each time step t of the simulation (i.e. step_size), the Cartesian coordinates of all residues are recorded along with the overall E_p energy. The format of the records in the dataset is shown in Fig. 14(a).

To make the pathway data generated in the simulations humanly interpretable, the PathMolD-AB software package provides a visualization tool (`pathway_print_multisubplot.py`). This program produces a video using the information contained in the folding data (see Section 3.3) showing the protein structure evolving along many iterations. Other information are also presented, including the plots of potential energy (Ep) and the radii of gyration (RgAll, RgP, RgH). A sample of a video frame generated by this program is shown in Fig. 14(b). This software was developed for the Linux operating system using the Python programming language.

Table 4
Input file parameters for the protein folding simulation.

Parameter	Description	Example
sequence	Hydrophobic-polar sequence of the protein	ABBABBA BABBAB
ProtLen	Number of amino acids	13
LV	Box size of the simulation	26
stepLimit	Maximum number of MD iterations	3,000,000
savepathways	If yes (y), save the pathway data	y
pathwaysstep	Number of iterations between saving partial results	3000
temperature	Temperature of the simulation	0.1

N	x	y	z
0	34.999992	35.000009	34.999967
1	34.703832	34.344024	34.305726
2	35.373810	33.944306	34.931309
3	35.458964	33.808000	35.918309
4	34.697299	33.364096	36.390340
5	35.112200	33.847174	37.161376
6	36.007293	34.254166	37.343494
7	36.668778	33.527120	37.159525
8	36.663247	33.247217	38.119537
9	36.774896	32.838795	39.025475
10	36.271704	33.585294	39.460832
11	37.101658	34.079531	39.719490
12	36.883032	34.948974	39.276465

protein structure information (step zero)

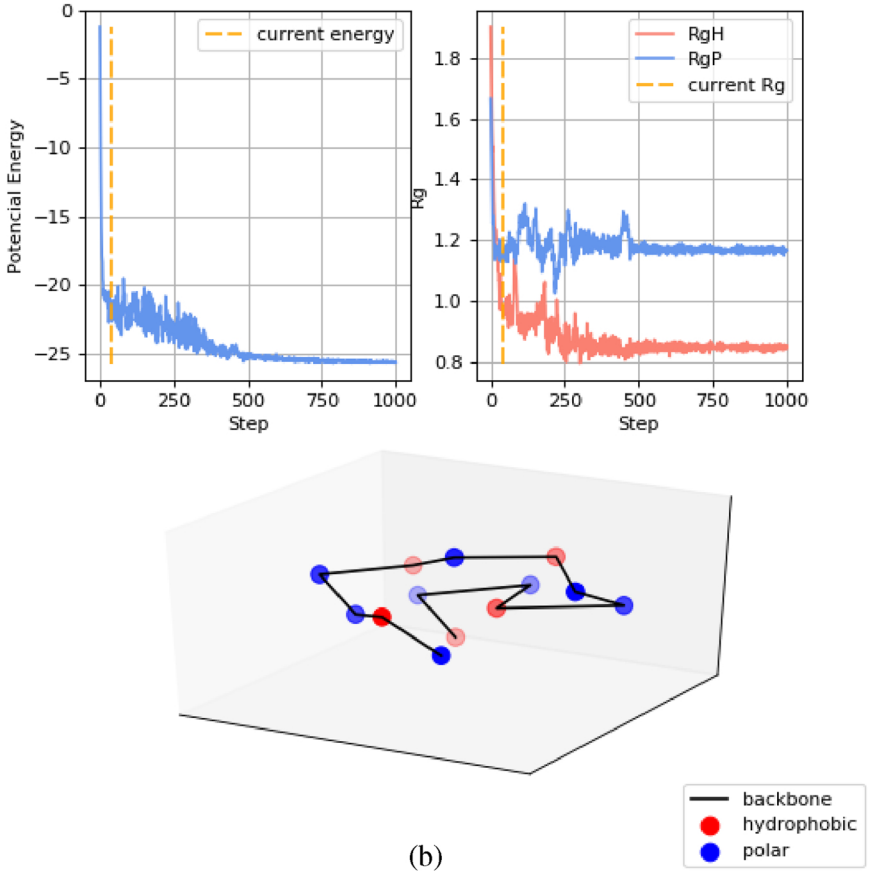
Ep= -0.339261
Step = 0
RgAll = 2.050609
RgH = 1.911790
RgP = 2.120789

information about the above protein structure at current time step

N	x	y	z
0	35.258793	35.125736	35.535713

....

(a)



(b)

Fig. 14. (a) A sample of the pathway data format. (b) Sample of a video frame generated by the visualization program. The image represents a protein structure at a given folding step, along with the plots of energy and radius of gyration.

Appendix C. Software-hardware compatibility

Table 5 presents the PathMolD-AB software compatibilities in terms of CUDA toolkit version, programming language version, and compute capability,[6] in which comprehends a set of features related to NVIDIA devices, including hardware and software features support. All experiments were run under the Ubuntu 18 LTS operating system.

Table 5
PathMolD-AB package compatibility. CC = Compute capability.

GPU model	CC	CUDA7	CUDA8	CUDA9
GTX660	3	No	No	No
K40	3.5	No	No	No
GTX750	5	No	No	No
Titan X	5.2	No	Yes	Yes
GTX 1080	6.1	No	Yes	Yes
Titan Xp	6.1	No	Yes	Yes
GCC/G++		4.8	5.3	6.5
Python		2.7/3.6		

References

Abraham, M.J., Murtola, T., Schulz, R., Páll, S., Smith, J.C., Hess, B., Lindahl, E., 2015. GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 1, 19–25.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P., 2002. Molecular Biology of the Cell, 4th ed. Garland Science, New York.

Alder, B.J., Wainwright, T.E., 1959. Studies in molecular dynamics. I. General method. J. Chem. Phys. 31, 459–466.

Amir, E.A.D., Kalisman, N., Keasar, C., 2008. Differentiable, multi-dimensional, knowledge-based energy terms for torsion angle probabilities and propensities. Proteins: Struct. Funct. Bioinform. 72, 62–73.

Bachmann, M., 2014. Thermodynamics and Statistical Mechanics of Macromolecular Systems, 1st ed. Cambridge University Press.

Benítez, C., 2015. Contributions to the Study of the Protein Folding Problem using Bioinspired Computation and Molecular Dynamics (PhD thesis). Federal University of Technology Parana (UTFPR), Curitiba, Brazil.

Benítez, C., Lopes, H., 2012. Molecular dynamics for simulating the protein folding process using the 3D AB off-lattice model. In: In: de Souto, M.C., Kann, M.G. (Eds.), Advances in Bioinformatics and Computational Biology, vol. 7409. Springer, Heidelberg, pp. 61–72.

Berendsen, H.J.C., Postma, J.P.M., van Gunsteren, W.F., DiNola, A., Haak, J.R., 1984. Molecular dynamics with coupling to an external bath. J. Chem. Phys. 81, 3684–3690.

Brezovsky, J., Babkova, P., Degtjarik, O., Fortova, A., Gora, A., Iermak, I., Rezacova, P., Dvorak, P., Smatanova, I.K., Prokop, Z., et al., 2016. Engineering a *de novo* transport tunnel. ACS Catal. 6, 7597–7610.

Brocchieri, L., Karlin, S., 2005. Protein length in eukaryotic and prokaryotic proteomes. Nucleic Acids Res. 33, 3390–3400.

Brown, S., Fawzi, N.J., Head-Gordon, T., 2003. Coarse-grained sequences for protein folding and design. Proc. Natl. Acad. Sci. U. S. A. 100, 10712–10717.

Casino, P., Gozalbo-Rovira, R., Rodriguez-Diaz, J., Banerjee, S., Boutaud, A., Rubio, V., Hudson, B.G., Saus, J., Cervera, J., Marina, A., 2018. Structures of collagen IV globular domains: insight into associated pathologies, folding and network assembly. Int. Union Crystallogr. J. 5, 765–779.

Chan, H.S., Dill, K.A., 1990. Origins of structure in globular proteins. Proc. Natl. Acad. Sci. U. S. A. 87, 6388–6392.

Cieplak, M., Hoang, T.X., 2000. Scaling of folding properties in Gō models of proteins. J. Biol. Phys. 26, 273–294.

Colombo, G., Micheletti, C., 2006. Protein folding simulations: combining coarse-grained models and all-atom molecular dynamics. Theor. Chem. Acc. 116, 75–86.

Dill, A., MacCallum, J.L., 2012. The protein-folding problem, 50 years on. Science 338, 1042–1046.

Finkelstein, A.V., 2018. 50+ years of protein folding. Biochemistry 83, S3–S18.

Frigori, R.B., 2014. Breakout character of islet amyloid polypeptide hydrophobic mutations at the onset of type-2 diabetes. Phys. Rev. E 90 052716.1-052716.7.

Frigori, R.B., 2017. PHAST: protein-like heteropolymer analysis by statistical thermodynamics. Comput. Phys. Commun. 215, 165–172.

Frigori, R.B., Rizzi, L.G., Alves, N.A., 2013. Microcanonical thermostatistics of coarse-grained proteins with amyloidogenic propensity. J. Chem. Phys. 138, 015102.

Geng, C., Vangone, A., Folkers, G.E., Xue, L.C., Bonvin, A.M., 2019. iSEE: interface structure, evolution, and energy-based machine learning predictor of binding affinity changes upon mutations. Proteins: Struct. Funct. Bioinform. 87, 110–119.

Gower, J.C., Dijksterhuis, G.B., 2004. Procrustes Problems. Oxford University Press, Oxford, UK.

Hanson, J., Paliwal, K., Litfin, T., Yang, Y., Zhou, Y., 2018. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. Bioinformatics 34, 4039–4045.

Hattori, L., Benítez, C., Lopes, H., 2018. A novel approach to protein folding prediction based on long short-term memory networks: a preliminary investigation and analysis. Proc. IEEE World Congress on Computational Intelligence. IEEE Press, Piscataway, NJ, pp. 1–6.

Hattori, L.T., Gutoski, M., Benítez, C.M.V., Nunes, L.F., Lopes, H.S., 2020. A benchmark of optimally folded protein structures using integer programming and the 3D-HP-SC model. Comput. Biol. Chem. 84, 107192.

Hays, J.M., Irrgang, M.E., Kasson, P.M., 2018. gmxapi: a high-level interface for advanced control and extension of molecular dynamics simulations. Bioinformatics 34, 3945–3947.

He, Y., Mozolewska, M.A., Krupa, P., Sieradzan, A.K., Wirecki, T.K., Liwo, A., Kachlishvili, K., Rackovsky, S., Jagieła, D., Ślusarz, R., et al., 2013. Lessons from application of the UNRES force field to predictions of structures of CASP10 targets. Proc. Natl. Acad. Sci. U. S. A. 110, 14936–14941.

Hills Jr., R.D., Brooks III, C.L., 2009. Insights from coarse-grained Gō models for protein folding and dynamics. Int. J. Mol. Sci. 10, 889–905.

Hourdel, V., Volant, S., O'Brien, D.P., Chenal, A., Chamot-Rooke, J., Dillies, M.A., Brier, S., 2016. MEMHDX: an interactive tool to expedite the statistical validation and visualization of large HDX-MS datasets. Bioinformatics 32, 3413–3419.

Hsu, J.P.C., Schiøtt, B., 2019. Investigating C99 in amyloid formation using molecular dynamics: from simple to complex neuronal models. Biophys. J. 116, 493a–494a.

Irbäck, A., Peterson, C., Potthast, F., Sommelius, O., 1997. Local interactions and protein folding: a three-dimensional off-lattice approach. J. Chem. Phys. 107, 273–282.

Jurcik, A., Furmanova, K., Strnad, O., Kozlikova, B., Pavelka, A., Stourac, J., Bednar, D., Daniel, L., Kokkonen, P., Marques, S.M., Damborsky, J., Brezovsky, J., Byska, J., Manak, M., 2018. CAVER Analyst 2.0: analysis and visualization of channels and tunnels in protein structures and molecular dynamics trajectories. Bioinformatics 34, 3586–3588.

Kabsch, W., 1976. A solution for the best rotation to relate two sets of vectors. Acta Crystallogr. A32, 922–923.

Khokhlov, A.R., 1994. Statistical Physics of Macromolecules, 1st ed. American Institute of Physics, NY, USA.

Kmiecik, S., Gront, D., Kolinski, M., Wieteska, L., Dawid, A.E., Kolinski, A., 2016. Coarse-grained protein models and their applications. Chem. Rev. 116, 7898–7936.

Kobayashi, C., Jung, J., Matsunaga, Y., Mori, T., Ando, T., Tamura, K., Kamiya, M., Sugita, Y., 2017. Genesis 1.1: a hybrid-parallel molecular dynamics simulator with enhanced sampling algorithms on multiple computational platforms. J. Comput. Chem. 38, 2193–2206.

Kolinski, A., 2011. Lattice Polymers and Protein Models. Springer, New York, pp. 1–20.

Kravraki, E., 2007. Geometric Methods in Structural Computational Biology, 1st ed. Connexions, Houston.

Lesgidou, N., Vlassi, M., Eliopoulos, E., Goulielmos, G.N., 2018. Insights on the alteration of functionality of a tyrosine kinase 2 variant: a molecular dynamics study. Bioinformatics 34, i781–i786.

Levitt, M., Warshel, A., 1975. Computer simulation of protein folding. Nature 253, 694–698.

Levy-Moonshine, A., Amir, E.A.D., Keasar, C., 2009. Enhancement of beta-sheet assembly by cooperative hydrogen bonds potential. Bioinformatics 25, 2639–2645.

Lin, J., Zhong, Y., Li, E., Lin, X., Zhang, H., 2018. Multi-agent simulated annealing algorithm with parallel adaptive multiple sampling for protein structure prediction in AB off-lattice model. Appl. Soft Comput. 62, 491–503.

Llanes, A., Muñoz, A., Bueno-Crespo, A., García-Valverde, T., Sanchez, A., Arcas-Tunez, F., Pérez-Sanchez, H., Cecilia, J.M., 2016. Soft computing techniques for the protein folding problem on high performance computing architectures. Curr. Drug Targets 17, 1626–1648.

[6] https://developer.nvidia.com/cuda-gpus

Lopes, H.S., 2008. Evolutionary algorithms for the protein folding problem: a review and current trends. In: In: Smolinski, T.G., Milanova, M.M., Hassanien, A.E. (Eds.), Applications of Computational Intelligence in Bioinformatics and Biomedicine: Current Trends and Open Problems, vol. 1. Springer-Verlag, Heidelberg, pp. 297–315.
Manavalan, B., Kuwajima, K., Lee, J., 2019. PFDB: a standardized protein folding database with temperature correction. Sci. Rep. 9, 1588.
Mendonça, M.R., Rizzi, L.G., Contessoto, V., Leite, V.B.P., Alves, N.A., 2014. Inferring a weighted elastic network from partial unfolding with coarse-grained simulations. Proteins: Struct. Funct. Bioinform. 82, 119–129.
Mirnics, K., Middleton, F., Stanwood, G., Lewis, D., Levitt, P., 2001. Disease-specific changes in regulator of G-protein signaling 4 (RGS4) expression in schizophrenia. Mol. Psychiatry 6, 293–301.
Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., Tramontano, A., 2018. Critical assessment of methods of protein structure prediction (CASP) – Round XII. Proteins: Struct. Funct. Bioinform. 86, 7–15.
Ngo, J.T., Marks, J., Karplus, M., 1994. Computational Complexity, Protein Structure Prediction, and the Levinthal Paradox. Birkh”auser Boston, Boston, MA, pp. 433–506.
Onofrio, A., Parisi, G., Punzi, G., Todisco, S., Di Noia, M.A., Bossis, F., Turi, A., De Grassi, A., Pierri, C.L., 2014. Distance-dependent hydrophobic-hydrophobic contacts in protein folding simulations. Phys. Chem. Chem. Phys. 16, 18907–18917.
Phillips, C.L., Anderson, J.A., Glotzer, S.C., 2011. Pseudo-random number generation for Brownian dynamics and dissipative particle dynamics simulations on GPU devices. J. Comput. Phys. 230, 7191–7201.
Pierri, C.L., De Grassi, A., Turi, A., 2008. Lattices for ab initio protein structure prediction. Proteins: Struct. Funct. Bioinform. 73, 351–361.
Poursina, M., Anderson, K.S., 2014. An improved fast multipole method for electrostatic potential calculations in a class of coarse-grained molecular simulations. J. Comput. Phys. 270, 613–633.
Rapaport, D.C., 2004. 2nd ed. The Art of Molecular Dynamics Simulation, vol. 1 Cambridge University Press, Cambridge.
Razban, R.M., Gilson, A.I., Durfee, N., Strobelt, H., Dinkla, K., Choi, J.M., Pfister, H., Shakhnovich, E.I., 2018. ProteomeVis: a web app for exploration of protein properties from structure to sequence evolution across organisms' proteomes. Bioinformatics 34, 3557–3565.
Reinders, M.J.T., van Ham, R.C.H.J., Makrodimitris, S., 2018. Improving protein function prediction using protein sequence and Gō-term similarities. Bioinformatics 35, 1116–1124.
Salomon-Ferrer, R., Case, D.A., Walker, R.C., 2013. An overview of the Amber biomolecular simulation package. WIREs Comput. Mol. Sci. 3, 198–210.
Schneider, L., Müller, M., 2019. Multi-architecture Monte-Carlo (MC) simulation of soft coarse-grained polymeric materials: SOft coarse grained Monte-Carlo Acceleration (SOMA). Comput. Phys. Commun. 235, 463–476.
Sheinerman, F.B., Norel, R., Honig, B., 2000. Electrostatic aspects of protein-protein interactions. Curr. Opin. Struct. Biol. 10, 153–159.
Spellings, M., Marson, R.L., Anderson, J.A., Glotzer, S.C., 2017. GPU accelerated discrete element method (DEM) molecular dynamics for conservative, faceted particle simulations. J. Comput. Phys. 334, 460–467.
Stillinger, F., Head-Gordon, T., 1995. Collective aspects of protein folding illustrated by a toy model. Phys. Rev. E 52, 2872.
Sugita, Y., Okamoto, Y., 1999. Replica-exchange molecular dynamics method for protein folding. Chem. Phys. Lett. 314, 141–151.
Tanaka, S., Scheraga, H.A., 1976. Medium-and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. Macromolecules 9, 945–950.
Tian, L., Wu, A., Cao, Y., Dong, X., Hu, Y., Jiang, T., 2011. NCACO-score: an effective main-chain dependent scoring function for structure modeling. BMC Bioinform. 12, 208.
Tiessen, A., Pérez-Rodríguez, P., Delaye-Arredondo, L.J., 2012. Mathematical modeling and comparison of protein size distribution in different plant, animal, fungal and microbial species reveals a negative correlation between protein size and protein number, thus providing insight into the evolution of proteomes. BMC Res. Notes 5, 85.
Torrie, G.M., Valleau, J.P., 1977. Nonphysical sampling distributions in monte carlo free-energy estimation: umbrella sampling. J. Comput. Phys. 23, 187–199.
Tozzini, V., 2009. Multiscale modeling of proteins. Acc. Chem. Res. 43, 220–230.
Watabe, M., Nakaki, T., 2007. ATP depletion does not account for apoptosis induced by inhibition of mitochondrial electron transport chain in human dopaminergic cells. Neuropharmacology 52, 536–541.
Yanev, N., Milanov, P., Mirchev, I., 2011. Integer programming approach to HP folding. Serdica J. Comput. 5, 359–366.
Yang, L., Zhang, F., Wang, C.Z., Ho, K.M., Travesset, A., 2018. Implementation of metal-friendly EAM/FS-type semi-empirical potentials in HOOMD-blue: a GPU-accelerated molecular dynamics software. J. Comput. Phys. 359, 352–360.
Zhou, C., Sun, C., Wang, B., Wang, X., 2018. An improved stochastic fractal search algorithm for 3D protein structure prediction. J. Mol. Model. 24, 125.