FEDERAL UNIVERSITY OF TECHNOLOGY – PARANÁ
GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

LEANDRO TAKESHI HATTORI

CONTRIBUTIONS TO THE STUDY OF THE PROTEIN FOLDING PROBLEM USING DEEP LEARNING AND MOLECULAR DYNAMICS

THESIS

CURITIBA
2020

LEANDRO TAKESHI HATTORI

CONTRIBUTIONS TO THE STUDY OF THE PROTEIN FOLDING PROBLEM USING DEEP LEARNING AND MOLECULAR DYNAMICS

Contribuições para o estudo do problema de dobramento de proteínas usando métodos de aprendizado profundo e dinâmica molecular

Thesis presented to the Graduate Program in Electrical and Computer Engineering of the Federal University of Technology - Paraná as ártoaç fulfillment of the requirements for the title of "Doctor of Science (D.Sc.)" - Computer Engineering.

Advisor: Prof. Dr. Heitor Silvério Lopes
Co-advisor: Prof. Dr. César Manuel Vargas Benítez

CURITIBA

2020

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba

LEANDRO TAKESHI HATTORI

CONTRIBUTIONS TO THE STUDY OF THE PROTEIN FOLDING PROBLEM USING DEEP LEARNING AND MOLECULAR DYNAMICS

Trabalho de pesquisa de doutorado apresentado como requisito para obtenção do título de Doutor Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Computação.

Data de aprovação: 30 de Novembro de 2020

Prof Heitor Silverio Lopes, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Alceu De Souza Britto Junior, Doutorado - Pontifícia Universidade Católica do Paraná (Pucpr)

Prof Fabricio Martins Lopes, - Universidade Tecnológica Federal do Paraná

Prof Rafael Bertolini Frigori, - Universidade Tecnológica Federal do Paraná

Prof Rafael Stubs Parpinelli, Doutorado - Fundação Universidade do Estado de Santa Catarina (Udesc)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 30/11/2020.

RESUMO

HATTORI, Leandro Takeshi. **Contribuições para o estudo do problema de dobramento de proteínas usando métodos de aprendizado profundo e dinâmica molecular**. 2020. 130 f. Thesis (Doctorate em Engenharia de Computação) – Federal University of Technology – Paraná. Curitiba, 2020.

O *Protein Folding Problem* (PFP) é um dos principais desafios da área de Biologia Computacional. Acredita-se que as proteínas globulares evoluem de conformações iniciais aleatórias através de trajetórias de dobramento, alcançando, em quase todos os casos, uma estrutura nativa funcional. Estudos relacionados ao dobramento proteico estão relacionados a vários eventos anormais, como dobramento incorreto e agregação de proteínas. Portanto, várias abordagens computacionais têm sido propostas na literatura para este problema. Métodos de *Deep Learning* (DL) têm se destacado em estudos na área de Proteômica, dada a sua capacidade de extrair vetores de características e também pela sua eficiência após o processo de treinamento. *Recurrent Neural Network* (RNN) são métodos DL cíclicos que alcançaram desempenho do estado-da-arte para problemas sequenciais e temporais. Esta tese apresenta contribuições para o estudo das trajetórias espaço-temporais do enovelamento de proteínas utilizando métodos RNN. Para alcançar essas contribuições, os experimentos desta tese foram organizados em três etapas: desenvolver um *framework* para gerar grande quantidades de dados de dobramento de proteínas usando métodos sequenciais e paralelos de *Molecular Dynamics* (MD) no ensemble canônico; propor uma abordagem de *Neighbourhood List* (NL) para o método MD paralelo; aplicar redes RNNs ao PFP. Na primeira etapa, apresentamos um pacote chamado PathMolD-AB para simular e analisar trajetórias de dados de dobramento usando o modelo 3D-AB off-lattice para representar a estrutura da proteína. Os conjuntos de dados gerados a partir do PathMolD-AB correspondem à 3.500 trajetórias de dobras, abrangendo 35×10^6 estados de dobramento. A análise de *speedup* mostrou que a abordagem paralela obteve simulações mais rápidas quando se utilizaram sequências de proteínas com mais de 99 aminoácidos. Na segunda etapa, a abordagem NL com MD paralelo mostrou melhoria no desempenho de aceleração do que a versão MD puramente paralela com sequências de proteínas entre 99 a 1.000 aminoácidos, que abrange 80 % de todo o Protein Data Bank (PDB). Na última etapa desta tese, foi realizada uma análise comparativa entre as arquiteturas de RNNs utilizando o modelo *many-to-one* com conjuntos de dados gerados pelo PathMold-AB. Os resultados indicam que a *Long Short-Term Memory* (LSTM) obteve o melhor desempenho que as outras arquiteturas de RNNs em termos de erro de predição. A análise biológica indicou que a rede LSTM previu estruturas com características semelhantes ao alvo (MD), em termos de compactação hidrofóbica e polar, e também energias de torção e ligação, sugerindo que esta abordagem é auspiciosa para o estudo PFP.

Palavras-chave: Palavra-chave A. Palavra-chave B. Palavra-chave C. Palavra-chave D. Palavra-chave E.

ABSTRACT

HATTORI, Leandro Takeshi. **CONTRIBUTIONS TO THE STUDY OF THE PROTEIN FOLDING PROBLEM USING DEEP LEARNING AND MOLECULAR DYNAMICS**. 2020. 130 p. Thesis (DSc in Computer Engineer) – Federal University of Technology – Paraná. Curitiba, 2020.

The Protein Folding Problem (PFP) is one of the main challenges in the Computational Biology area. Globular proteins are believed to evolve from random initial conformations through folding pathways achieving, in almost all cases, to a functional native structure. Studies of the folding process are related to several abnormal events, such as misfolding and protein aggregation. Therefore, several computational approaches have been proposed in the literature for this problem. Deep Learning (DL) methods have been highlighted in studies in the Proteomics area, given their ability to extract features vectors and their efficiency after the training process. Recurrent Neural Networks (RNN) are cyclic DL methods that have achieved state-of-the-art performance for sequential and temporal problems. Therefore, this thesis presents contributions to studying the spatial-temporal pathways of the protein folding using RNN methods. To achieve these contributions, experiments of this thesis were organized in three steps: develop a framework to generate a massive amount of protein folding data using pure sequential and parallel Molecular Dynamics (MD) methods in the canonical ensemble; propose a Neighbourhood List (NL) approach to the parallel MD method; apply RNNs networks to the PFP. In the first step, we presented a package called PathMolD-AB to simulate and analyze folding data trajectories using the 3D-AB off-lattice model to represent the protein structure. The datasets generated from PathMolD-AB correspond to the MD evolution of 3,500 folding pathways, encompassing 35×10^6 states. The speedup analysis showed that the parallel approach obtained faster simulations when used protein sequences with more than 99 amino acids were used. In the second step, the NL approach with parallel MD showed higher improvement in the speedup performance than the purely parallel MD version with protein sequences between 99 to 1,000 amino acids, which covers 80% of the entire Protein Data Bank (PDB). In the last step of this thesis, a comparative analysis between RNNs architectures were carried out using the many-to-one model with datasets generated by the PathMold-AB. Results indicate that the Long Short-Term Memory (obtained the best performance than other RNNs architectures in terms of prediction error. The biological analysis indicated that the LSTM predicted structures with similar features to the target (MD), in terms of hydrophobic and polar compactness, and also torsion and bond energies, suggesting that this approach is auspicious for the PFP study.

Keywords: Deep Learning. Long Short-Term Memory. Protein Folding Problem. Molecular Dynamics. High-Performance Computing.

LIST OF ALGORITHMS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

INITIALISM	
ANN	Artificial Neural Networks
AE	Autoencoders
AC	Automata Cellular
BPTT	BackPropagation Through Time
BLSTM	Bidirectional Long Short-Term Memory
CM	Contact Map
CNF	Conditional Neural Fields
CNN	Convolutional Neural Networks
DL	Deep Learning
DNN	Deep Neural Network
DNA	DeoxyriboNucleic Acid
GD	Gradient Descendent
HPC	High-Performance Computing
IP	Integer Programming
LSTM	Long Short-Term Memory
MSM	Markov state model
NCBI	National Center for Biotechnology Information
NL	Neighborhood List
OPEC	Optimized Potential for Efficient Structure Prediction
PDB	Protein Data Bank
PFP	Protein Folding Problem
PRIMO	PRotein Interactive MOdeling
PSP	Protein Structure Prediction
ReLU	Rectified Linear Units
RNNs	Recurrent Neural Networks

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

The Protein Folding Problem (PFP) is one of the most challenging problems in the Computational Biology area. This research area comprises the study of state sequences of a protein's structure from the unfolded structure for its native conformation. The protein native structure exerts many biological functions, for instance, regulatory mechanisms, structural function, defense mechanisms, transportation, and other functions. Thus, computational methods have been focused on the PFP process to enlarge research on how proteins achieve their native state.

Regarding proteomics studies, several large-scale projects have been developed to drive this area, such as the Human Genome Project[1], Folding at Home[2], Critical Assessment of Protein Structure Prediction[3]. These projects promote discoveries on protein structures and new computational methods (HOU *et al.*, 2019). Besides these works, repositories have been developed, and the scientific community can publish new information about discovered proteins. UniProtKB/TrEMBL is one repository, containing about 84 million protein sequences (November / 2020).

The Protein Data Bank (PDB) is a repository for storing structural information containing over 150,000 structures (November / 2020), a relatively small amount compared to the number of discovered sequences ($180{,}179{,}667$). The small number of protein structures is related to the difficulties of performing experimental methods (LACAPÈRE *et al.*, 2018). Datasets with protein folding information are sparsely available, and when it is present, they are inconsistent and non-standardized.

Among the computational methods for protein folding simulation, some approaches do not require prior information, called *ab initio*. The Molecular Dynamics approach (MD) is an *ab initio* method extensively explored for the PFP study (PEREZ *et al.*, 2016). This approach simulates protein folding pathways based on physical principles using energy functions. Regarded their notability, packages have been proposed in the scientific literature, as GROMACS[4] and

[1] Avaiable in https://www.genome.gov/ (accessed 24 November 2020).
[2] Avaiable in http://folding.stanford.edu/ (accessed 24 November 2020)
[3] Avaiable in http://predictioncenter.org/ (accessed 24 November 2020)
[4] http://www.gromacs.org/ (accessed 24 November 2020)

AMBER[5] (ABRAHAM *et al.*, 2015; MERMELSTEIN *et al.*, 2018). However, when employing generalized ensembles, these packages usually preserve the energy sample landscape instead of the time-dependence of the folding trajectories. The MD algorithm in the canonical ensemble preserves these features to the time-dependence of the simulation (RAPAPORT, 2004), however, barely exploited in the current literature.

Due to the computational power required for the PFP problem, simpler but non-simplistic models named Coarse-Grained (CG) have been explored in the literature (KMIECIK *et al.*, 2019; BOIANI; PARPINELLI, 2020). These models can represent many biological behaviors at a meso-scale (TOZZINI, 2005), such as the hydrophobic core formation, presented in many protein domains (KALINOWSKA *et al.*, 2017), and the protein aggregation process, which is related to proteinopathies (FRIGORI, 2017). Among the many variants of CG models, the 3D-HP off-lattice (STILLINGER; HEAD-GORDON, 1995) is a representation where the conformation of the protein is contained in a tri-dimensional (3D) lattice, and each amino acid is represented by one beads. Each bead represent the C_α and the binding structure between the amino acids sequence, which can be either hydrophobic (A) or polar (B).

Other approaches have been proposed to overcome this computational complexity, such as parallel MD approaches using Graphics Processing Units (GPU) hardware and Neighbourhood List (NL) technique for the MD method. Otherwise, computational intelligence has been poorly explored for the protein folding problem (BENÍTEZ, 2015).

Recently, Deep Learning (DL) methods have been overcoming previous state-of-the-art approaches in different Bioinformatics problems(ANGERMUELLER *et al.*, 2016)(LI *et al.*, 2019). The ability to learn feature extraction from raw data is one factor that makes DL effective (LECUN *et al.*, 2015). Although the training process demands a high computational cost, trained network applications can be efficient and used in real-time systems (LI *et al.*, 2018; GUO *et al.*, 2018). Among DL approaches, Recurrent Neural Networks (RNNs) are methods that allow storing information from network inputs as memory. These networks can associate a sequence of information to improve prediction and classification tasks. For example, in the proteomics area, RNNs are commonly used to predict protein structures using only the sequential information of the amino acids (MIN *et al.*, 2017).

The Long Short-Term Memory network (LSTM) among RNNs approaches, can associate long and short dependencies between the sequential input data, unlike standard RNN

[5] https://ambermd.org/ (accessed 24 November 2020)

(sRNN) (HOCHREITER; SCHMIDHUBER, 1997). Due to this feature, LSTMs have been applied successfully to Bioinformatics problems, such as secondary and tertiary protein structure prediction (PALIWAL *et al.*, 2015; HANSON *et al.*, 2018). Others applications of the LSTM in proteomics include: classification of secondary protein structures (SøNDERBY; WINTHER, 2015), prediction of structural unstable regions of proteins (HANSON *et al.*, 2017), and prediction of protein functions (LIU, 2017). To the best of our knowledge, no recent work is using DL methods for the PFP.

1.2 OBJECTIVES

This thesis aims to develop a new computational method based on DL approaches for the Protein Folding Problem. This study concentrated on the one-step-ahead prediction approach. The specific objectives are:

- To present a package called, PathMold-AB, to generate in silico spatio-temporal protein folding trajectories datasets with the canonical ensemble Molecular Dynamics using a coarse-grained model.
- To propose a parallel approach of the canonical ensemble Molecular Dynamics method;
- To compare the performance of the sequential and parallel Molecular Dynamics methods;
- To compare the synthetically folded structures and re-scaled structure from the Protein Data Bank;
- To propose Recurrent Neural Networks using the many-to-one model to the Protein Folding Problem;
- To compare the performance of the Recurrent Neural Networks;
- To propose an encoding of the protein structure using Relative Spherical Coordinate;
- To present metrics to analyze the performance of Recurrent Neural Networks methods using Radii of gyration and Potential energy of predicted folded structures.

1.3 OUTLINE

The Thesis is divided into six chapters. Chapter 2 presents the theoretical foundations of proteins and their structures. Next, related works are presented in Chapter 3. Chapter 4 presents the proposed methods. Chapter 5 presents the results of the experiments performed. Finally, some conclusions and future works are presented in Chapter 6.

1.4 CONTRIBUTIONS

Along with the development of the Thesis, we have produced some works, contributing to this project's overall development.

In Hattori *et al.* (2017a) was presented an application of the Deep Learning method using LSTM for the protein secondary structure prediction.

An application of Molecular Dynamics to the protein folding problem using the 3D-AB off-lattice model with the NL method was presented by Takiguchi *et al.* (2017).

In Hattori *et al.* (2018) was presented a preliminary framework, results, and directions of this Thesis.

In Hattori *et al.* (2020b) was proposed a framework to generate Spatio-temporal data of protein folding trajectories, called PathMolD-AB.

In Hattori *et al.* (2020a) a *benchmark* was presented for protein structure predictor using 3D-HP-SC model using Integer Programming (IP) method. The most important aspect of this work was the method of development to compare predicted structures with the re-scaled biological protein structures.

The other nine researches were developed concomitantly to the work presented in this Thesis. Those articles are presented at the Annex A.

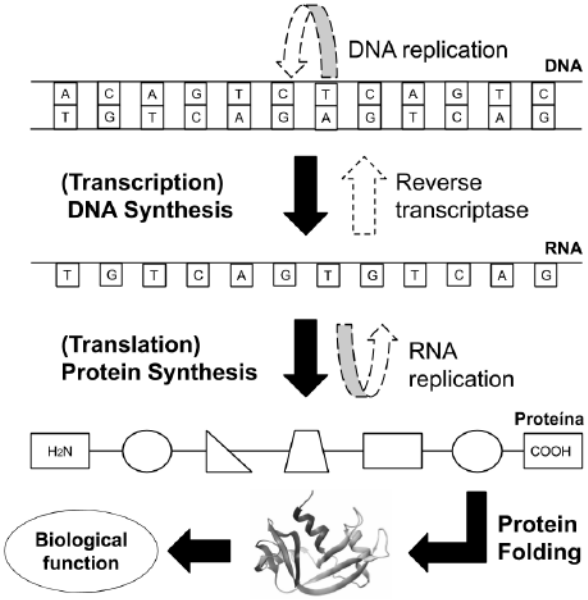
2 BACKGROUND

2.1 PROTEINS

The DeoxyriboNucleic Acid (DNA) is a structure that stores the biological information of an organism. The DNA is composed of two nucleotide strands coiled around each other. Each nucleotide of one strand is paired with the complementary nucleotide of the other strand, as represented in Figure 1.

All information in the DNA can be transcribed in nucleotide sequences called genes. The transcription of a gene is the beginning of the decoding process of the stored information to obtain the biological product, such as RiboNucleic Acid (RNA) and proteins.

Figure 1 – The central dogma of the Molecular Biology.



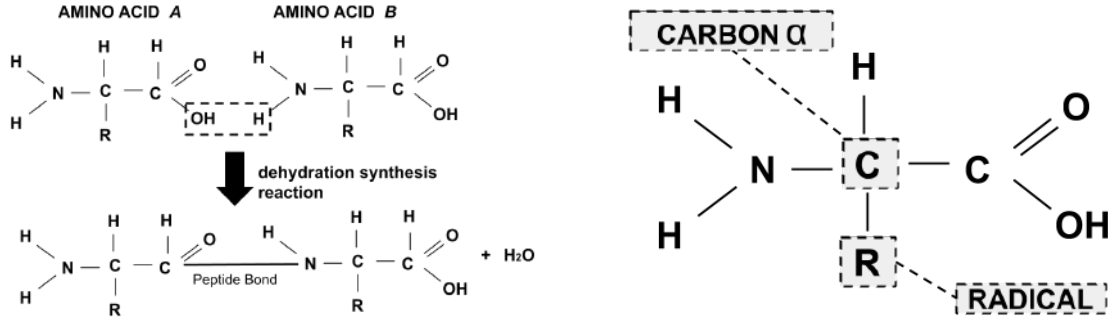
Source: LEHNINGER *et al.* (2008)

The first model describing the central dogma of Molecular Biology was proposed by Crick (1958). This model assumes only the cascade flow, such as a gene transcription to RNA, and the RNA translation to a protein sequence, as shown in Figure 1. The translated protein goes through a series of conformations, called the protein folding process, until it reaches its native structure.

Protein is a polymer of amino acids linked by peptide bonds, demonstrated in Figure 2. In the reaction of the peptide bonds, water molecules (H_2O) are formed. This reaction is also

called dehydration reaction, shown in Figure 2(a). An amino acid is composed of a central carbon, called Alpha Carbon ($C\alpha$), shown in Figure 2. $C\alpha$ has bonded with: a carboxyl, hydrogen, an amino group, and a radical.

Figure 2 – (a) A peptide bond scheme, that one amino acid loses an OH molecule and the other amino acid loses an H atom, producing at the end of the reaction a water molecule ($H2O$). (b) The basic structure of amino acids. In this structure, the amino acid has a $C\alpha$ with a carboxyl ($COOH$), hydrogen (H), an amino group (NH_2) and a radical (R) bonds.

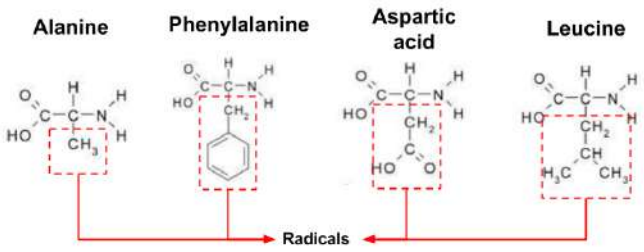


(a) **(b)**

Source: own authorship

Amino acids differ according to their radical, also known as a side chain. Figure 3 presents four examples of amino acids with their respective radicals. In total, there are 20 proteinogenic amino acids: Alanine (A, Ala), Arginine (R, Arg), Asparagine (N, Asn), Aspartic Acid (D, Asp), Cysteine (C, Cys), Glutamic Acid (E, Glu), Glutamine (Q, Gln), Glycine (G, Gly), Histidine (H, His), Isoleucine (I, Ile), Leucine (L, Leu), Lysine (K, Lys), Methionine (M, Met), Phenylalanine (F, Phe), Proline (P, Pro), Serine (S, Ser), Threonine (T, Thr), Tryptophan (W, Trp), Tyrosine (Y, Tyr), and Valine (V, Val). The amount and variation of these amino acids in the protein sequence characterize the native structure and drive the folding process.

Figure 3 – Sample of four radicals (Alanine, Phenylalanine, Aspartic acid, and Leucine) highlighted in dashed lines.



Source: www.mdsaude.com. Accessed in January 2021.

2.2 PROTEIN STRUCTURE

Proteins can be described at four levels of structure: primary, secondary, tertiary, and quaternary (LEHNINGER *et al.*, 2008). The primary structure represents the amino acid sequence of the protein. In this representation level, the number, nature, and amino acids of the sequence can be changed (LEHNINGER *et al.*, 2008).

Secondary structures are represented by local conformations of some polypeptide chain regions, as shown in Figure 4. The two most common structures at this level are the α-helix (PAULING *et al.*, 1951) and β-sheet (PAULING; COREY, 1951) structures. Other structures are irregular, called loops (LEWIS *et al.*, 1973). The secondary structures are described below:

- **α-helix**: These are highly stabilized helical-shaped structures, formed of a pattern of hydrogen bridges between the NH group and CO (NÖLTING, 2006). This structure does not involve the side chain, and different sequences form the helical structure, given its stability (ALBERTS *et al.*, 2002).

- **β-sheet**: These structures are formed from hydrogen bridges linked between adjacent peptide regions of the same or another molecule. In this structure, the attraction occurs between the NH and CO groups and does not involve the side chain. Adjacent β sheet sequences can be in the same direction (parallel β sheet) or in the opposite direction (antiparallel β sheet).

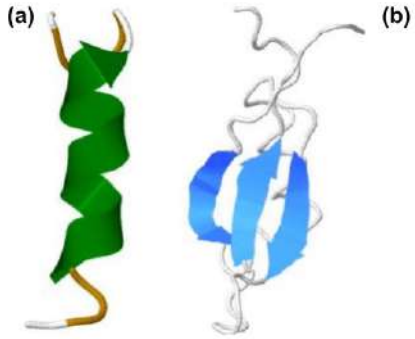
- **loops/handles**: Unlike α-helix structures and β-sheet (regular regions), loops are not presented with regular patterns. These structures are found after regular regions or where the polypeptide changes direction. Predictions of these structures can be difficult to identify due to their unstable feature.

Kabsch and Sander (1983) have subdivided secondary structures into eight types, including structures such as 3_{10}-helix, π-helix and β-bridge.

In the tertiary structure, conformations of the secondary structures in the three-dimensional space are considered, as shown in Figure 5. This representation considers interactions between amino acids and the environment. Based on these interactions, the protein structure seeks its thermodynamic stability.

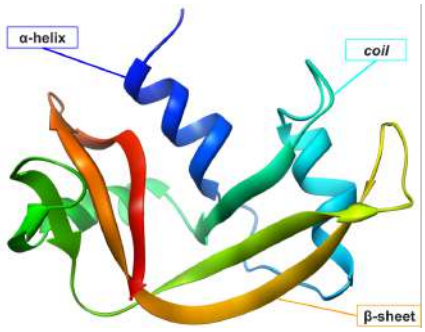
In the quaternary structure, two or more interacting polypeptide are considered to shape the functional structure of the protein. Each peptide in this model is called a subunit, as presented

Figure 4 – Representation of secondary structures: (a) α helix, and (b) β sheet.



Source: https://www.rcsb.org/. Accessed in January 2021.

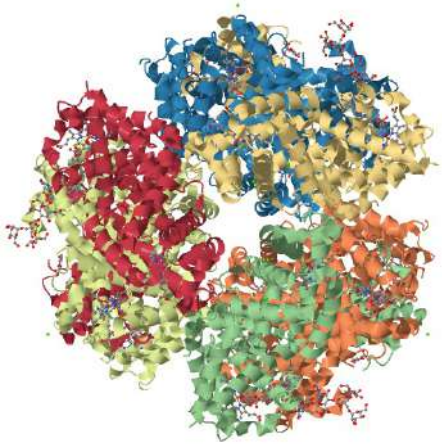
Figure 5 – Sample of a tertiary structure with $\alpha-$helix, $\beta-$sheets, and coil.



Source: https://www.rcsb.org/. Accessed in January 2021.

in Figure 6.

Figure 6 – Sample of a protein with quaternary structure, where each color represents one tertiary structure.



Source: https://www.rcsb.org/. Accessed in January 2021.

2.3 THE PROTEIN FOLDING PROBLEM

The Protein Folding Problem (PFP) is central in Science. It emerged around the 1960's, when the first protein structure at the atomic level was observable (DILL *et al.*, 2008). The PFP consists of the study of folding pathways, from the unfolded protein to its native conformation, defining the biological function of the protein. The PFP terminology is often misused for describing the Protein Structure Prediction (problem (LOPES, 2008), which is the prediction of the native three-dimensional conformation of a protein, based on its primary structure.

The PFP has raised some questions: What are the encoded physical forces in the amino acid sequence undertakening the folding process to its native structure? How can a computational method simulate protein folding? How do proteins fold so fast? Thus, the next sections will present the main assumptions of the PFP area.

2.3.1 The Thermodynamics Hypothesis of Protein Folding

The folding, desnaturation, and refolding process match experimentally to the thermodynamics hypothesis based on the principle of Molecular Biology proposed by Anfinsen (1973).

This hypothesis suggests that, under physiological conditions, the folding process occurs spontaneously, and the stability of the three-dimensional structure tends to achieve the lowest free energy (PEDERSEN, 2000). Also, the structure hardly loses its shape (DILL *et al.*, 2008). According to this hypothesis, all the necessary information to achieve a stable thermodynamic state is contained in the amino acid sequence features, represented by the primary structure of a protein (see Section 2.2).

Thermodynamically, the conformational space of a protein can be evaluated as a free energy function, the Gibbs free energy (MOREL; HERING, 1993). Such energy, or free enthalpy, is a thermodynamics principle: it defines the spontaneity of a chemical reaction according to the first and second laws. The Gibbs free energy ($\triangle G$) is presented in Equation 1. This scalar magnitude is defined by the enthalpy fluctuation ($\triangle H$) and temperature (T) multiplied by the entropy variation ($\triangle S$). The enthalpy variation represents energy fluctuation of a system, while

the entropy variation represents its disorder level variance.

$$\begin{aligned} \triangle G =& \triangle H - T\triangle S, \\ \triangle H =& H_2 - H_1, \\ \triangle S =& S_2 - S_1. \end{aligned} \tag{1}$$

The Gibbs free energy fluctuation discriminates between the folded and unfolded protein states (LEHNINGER *et al.*, 2008). The unfolded structure has a high entropy value. Usually, hydrogen bridges and hydrophobic interactions indicate a protein structure with low entropy, as in the native structure (DILL, 1990). Among these interactions, the hydrophobicity scale of the amino acids is the most influential factor in the protein folding process for globular proteins (DILL, 1999).

2.3.2 The Levinthal Paradox

In the Levinthal (1968) works, an important question was raised about protein conformation space. A protein can achieve all possible conformations until reaching the native structure, in the worst case. Nevertheless, the folding process reaches its native structure quickly. In this regard, how can proteins find their native structure so quickly?

Levinthal's (1986) arguments that a random search for a favourable conformation would not be possible because the magnitude of the search space would be impracticable. The calculation to define the time required to find all states of a protein is given by the number of amino acids in the polypeptide chain multiplied by the time required to obtain each conformation. For example, considering a biomolecule with 100 amino acids, the conformation space is 10^{70}, and a single conformation time would be 10^{-11}s (BENÍTEZ, 2015). The time required to find the native conformation would be the worst case, approximately 10^{59} years. Hence, a small protein would take ample time to explore every possible conformation. On the other hand, biological macromolecules achieve their native structure in seconds or even in microseconds depending on the number of amino acids in the chain (ENGLANDER, 2000).

2.3.3 Funnel, surface energy, and folding pathway

The term "folding funnel" was presented by Karplus (1992), and it is the study of the kinetic mechanism to understand the principles of sequence-structure self-organization, and explains how proteins can be folded quickly.

Wolynes *et al.* (1995) presented how the Anfinsen Hypothesis (Thermodynamics) and the Levinthal Paradox (Kinetics) are solved using a graphical approach. In Figure 7 (a), the vertical axis represents the free energy variation of protein conformations, and the horizontal axis is the entropy values of the conformations. The funnel shape of the energy landscape is given as follows: at the beginning of the folding process, the protein structure may assume several structures (states) until it reaches its native form. All initial conformations tend to have a high level of randomness (high entropy) and high free energy value compared with the features of the native state.

Along the folding process, free-energy values and the number of possible conformations decrease, and, at the end of the trajectory, all conformations tend to converge to the native conformation, as a funnel landscape, as shown in Figure 7 (b). This hypothesis is consistent with the rate at which protein folds as if the forces of the possible conformations slope reasonably quickly down to the free energy representing the native structure (DILL *et al.*, 2008).

The funnel representation reconciles with the thermodynamics and the kinetics approaches. The lowest free energy defines the native state in the funnel representation, as presented in the thermodynamics version. Besides, different unfolded structures bring to the same native conformation as in the kinetics description.

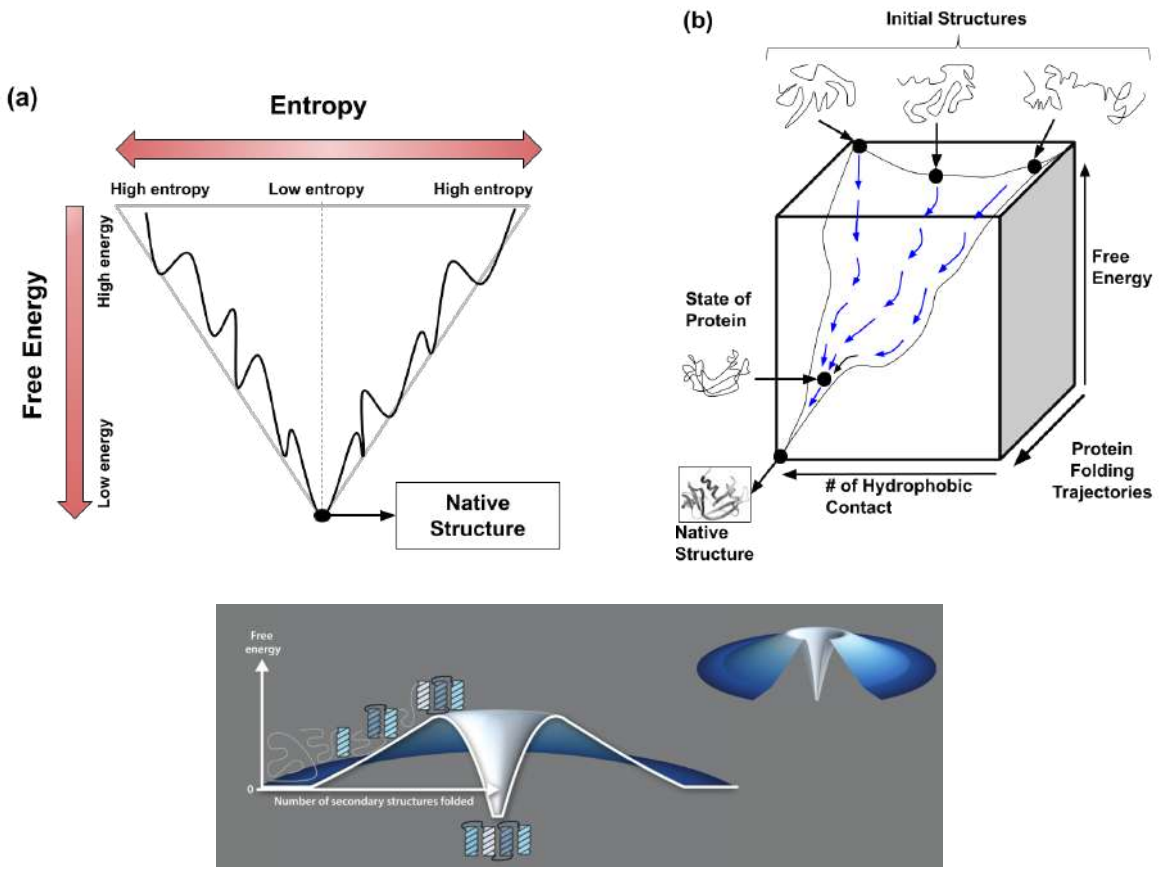
Rollins and Dill (2014) presented a complementary perspective of the funnel landscape resembling a volcano shape, as shown in Figure 7 (c). In this landscape, the beginning of the process has several energy barriers and may contain several local minima. After these barriers, the energy landscape has the shape of a funnel. The volcano landscape contributes to explain the energy barriers that separate the unfolded states to the folded structure.

2.3.4 Protein Properties

Yang *et al.* (2007) showed that the difference between the native state and unfolded states has 5-10 $kcal/mol$. Thus, the intramolecular interaction of proteins can change the folding process with different levels of influence. According to Dill *et al.* (2008), features with less influence on the structural stability of the protein include the amino acid charge, pH variation, and salt concentration.

On the other hand, the hydrogen interaction is a relevant property to the protein folding process (DILL; MacCallum, 2012), present in globular protein structures. Estimates showed that the energy of this interaction is between 1-4 $kcal/mol$ (BYRNE *et al.*, 1995). A relatively high

Figure 7 – (a) **Representation of entropy decay concerning free energy** (b) **Representation of folding trajectories by free energy and the number of hydrophobic contacts** (c) **Volcanic energy landscape.**



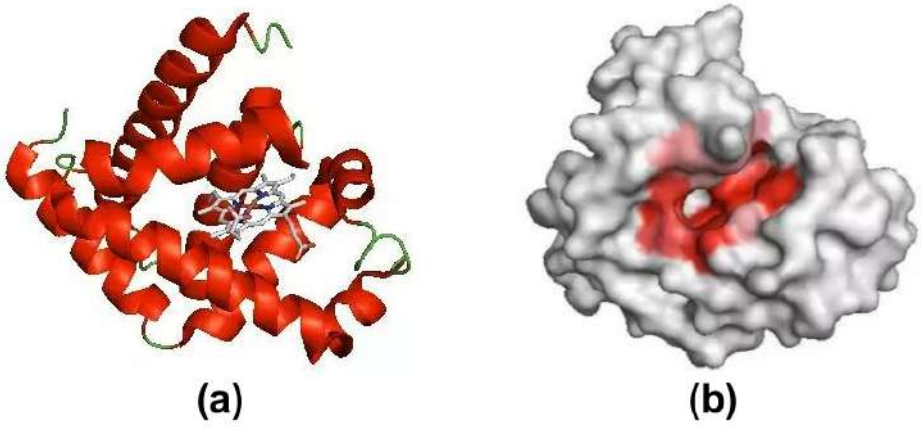
Source: (a) **Karplus (1992),** (b) **Vendruscolo *et al.* (2001) Dobson (2004) Benítez (2015), and** (c) **Rollins and Dill (2014)**

value when compared to the strength of the native structure.

Another aspect is the interactions between the hydrophobic amino acids. The hydrophobic kernel indicates the significant impact of this interaction inside the globular protein structure, as shown in Figure 8, and it was observed denaturation of proteins that have hydrophobic core when embedded in non-polar solvents (CHENG *et al.*, 2015).

There are different amino acid hydrophobicity scales in the literature. In Kyte's work (KYTE; DOOLITTLE, 1982), this scale is obtained from the physicochemical properties of amino acids side chains. Wimley and White (1996) defined their scale experimentally on membrane interface. Hessa *et al.* (2005) used a physicochemical process to measure the hydrophobicity scale. The Eisenberg scale (EISENBERG *et al.*, 1984) is based on normalized hydrophobicity scales. Janin's scale provides information about accessible amino acid residues in globular proteins (JANIN, 1979). Engleman's scale, also known as *GES-scale*, can be used to predict hydrophobicity using energy transfer calculation (TOLSTRUP *et al.*, 1994). In Hopp and Woods

Figure 8 – (a) **Secondary structures representation of the Protein 1WLA** (b) **Surface representation of the 1WLA protein (red color represents the hydrophobic surface).**



Source: https://www.quora.com/What-are-hydrophobic-pockets. Accessed in January 2021.

(1981), the energy of amino acid transfer from water to ethanol provides the necessary measure of its hydrophobicity scale.

Table 1 show the hydrophobic scales of each approach. Each hydrophobic scale is in a different range and direction. For example, in the Hessa scale, the hydrophobicity degree of each amino acid increases for lower values. In Kyle, Wimley, Eisenberg, Janin, Engelman, and Hoop scales, the amino acid hydrophobicity degree increases with the value of the scale. According to Alberts *et al.* (2002), the amino acid hydrophobicity scale can also be classified as hydrophobic (A) or polar (B) based on the mean behavior of the scales presented in Table 1.

The input of hydrophobicity classifications in simulations of the protein folding was studied by a comparative analysis of Benítez (2015). The statistical analysis of the potential energy showed no significant differences between these approaches. Then, since the Alberts classification represents the average behavior, it will be used in our approach.

2.4 COMPUTATIONAL MODELS FOR PROTEIN REPRESENTATION

The time-consumption of simulations can be an issue in the PFP. For example, detailed models require more computational power to calculate all interactions, while high-level representations tend to be more computationally feasible.

Overall, it is possible to classify protein representations as Atomic or Coarse-Grained (CG) models as presented in Figure 9. Although atomic models are the target, it is limited by the efficiency of algorithms and the current computational power (POBLETE *et al.*, 2018). In this approach, all-atom interactions need to be calculated at each time step. The CG models

Table 1 – Hydrophobicity scale and classification of each amino acid.

Aminoácidos	Kyte & Doolittle		Wimley		Hessa		Eisenberg		Janin		Engelman		Hoop	Alberts
	Scale	AB	Scale	AB	Scale	AB	Scale	AB	Scale	AB	Scale	AB	Scale	AB
Alanina (Ala)	1.8	A	-0.17	B	-0.11	A	0.62	A	0.3	A	1.6	A	-0.5	A
Cisteína (Cys)	2.5	A	0.24	A	0.13	A	0.29	A	0.9	A	2	A	-1.0	A
Ácido aspártico (Asp)	-3.5	B	-1.23	B	-3.49	B	-0.9	B	-0.6	B	-9.2	B	3.0	B
Ácido glutâmico (Glu)	-3.5	B	-2.02	B	-2.68	B	-0.74	B	-0.7	B	-8.2	B	3.0	B
Fenilalanina (Phe)	2.8	A	1.13	A	0.32	A	1.19	A	0.5	A	3.7	A	-2.5	A
Glicina (Gly)	-0.4	B	-0.01	B	-0.74	B	0.48	A	0.3	A	1	A	0.0	A
Histidina (His)	-3.2	B	-0.96	B	-2.06	A	-0.4	B	-0.1	B	-3	B	-0.5	B
Isoleucina (Ile)	4.5	A	0.3	A	0.6	A	1.38	A	0.7	A	3.1	A	-2.5	A
Lisina (Lys)	-3.9	B	-0.99	B	-2.71	B	-1.5	B	-1.8	B	-8.8	B	3.0	B
Leucina (Leu)	3.8	A	0.56	A	0.55	A	1.06	A	0.5	A	2.8	A	-1.8	A
Metionina (Met)	1.9	A	0.23	A	0.1	A	0.64	A	0.4	A	3.4	A	-1.3	A
Asparagina (Asn)	-3.5	B	-0.42	B	-2.05	B	-0.78	B	-0.5	B	-4.8	B	0.2	B
Prolina (Pro)	-1.6	B	-0.45	B	-2.23	B	0.12	A	-0.3	B	-0.2	B	0.0	A
Glutamina (Gln)	-3.5	B	-0.58	B	-2.36	B	-0.85	B	-0.7	B	-4.1	B	0.2	B
Arginina (Arg)	-4.5	B	-0.81	B	-2.58	A	-2.53	B	-1.4	B	-12.3	B	3.0	B
Serina (Ser)	-0.8	B	-0.13	B	-0.84	B	-0.18	B	-0.1	B	0.6	A	0.3	B
Treonina (Thr)	-0.7	B	-0.14	B	-0.52	B	-0.05	B	-0.2	B	1.2	A	-0.4	B
Valina (Val)	4.2	A	-0.07	A	0.31	A	1.08	A	0.6	A	2.6	A	-1.5	A
Triptofano (Trp)	-0.9	B	1.85	A	-0.3	A	0.81	A	0.3	A	1.9	A	-3.,4	A
Tirosina (Tyr)	-1.3	B	0.94	A	-0.68	A	0.26	A	-0.4	B	-0.7	B	-2.3	B

Source: Kyte and Doolittle (1982), Wimley and White (1996), Hessa *et al.* (2005), Eisenberg *et al.* (1984), Janin (1979), Eisenberg *et al.* (1984), Hopp and Woods (1981), Alberts *et al.* (2002).

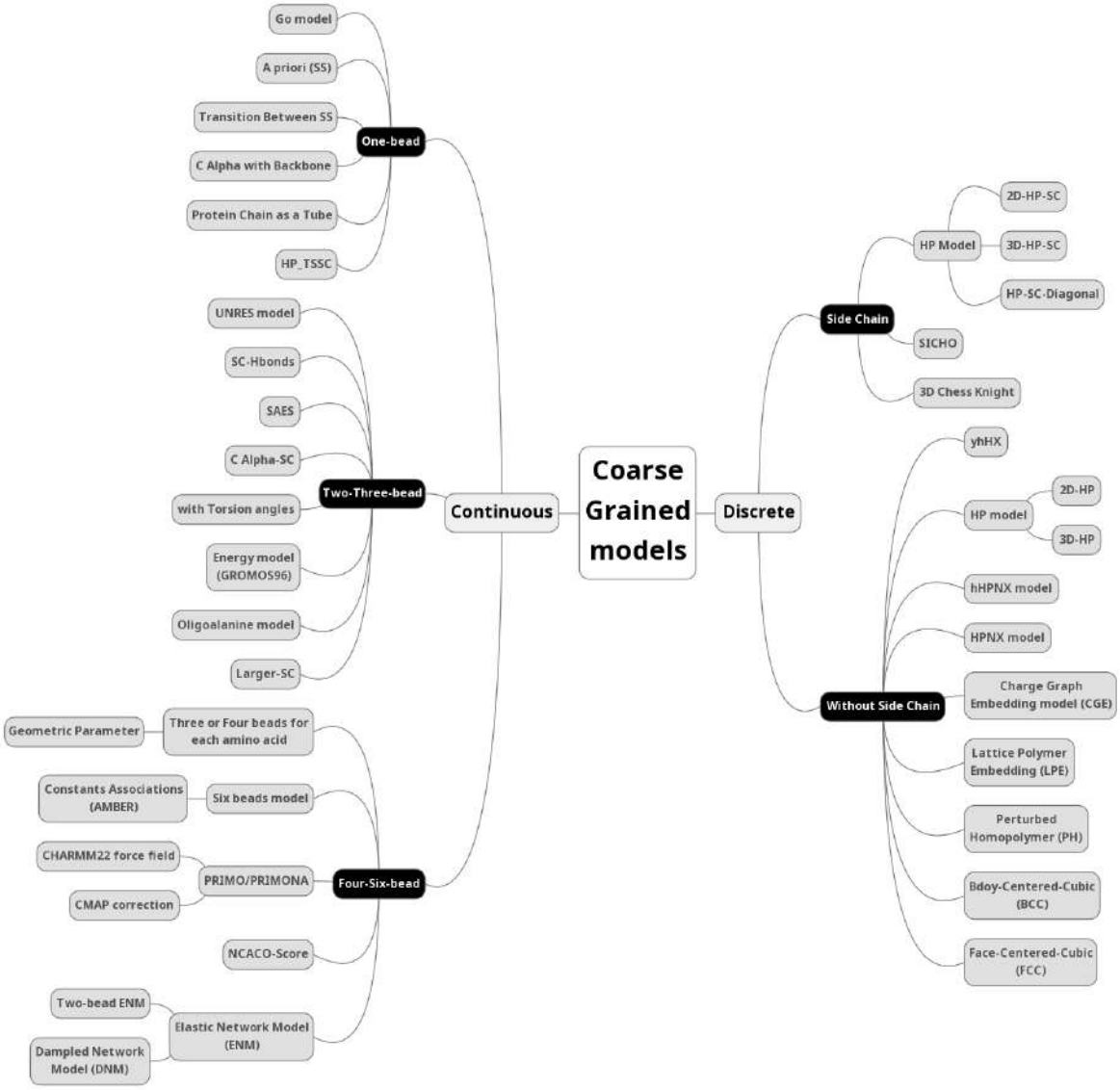
require less computational power compared to the atomic models, and represent each amino acid by beads (one to six). This high-level representation allows more extended folding simulations as well as the use of longer polypeptide chains. Since the lower amount of interactions that need to be calculated. Also, CG models can represent the folding behaviors satisfactorily at the mesoscale level (TOZZINI, 2005; DILL; MacCallum, 2012).

Proteins are formed by a sequence of amino acids linked by peptide bonds (see Section 2.1), called the backbone, which has degrees of freedom, torsion, and rotation between the bonds. Each amino acid can have a different side chain. Thus, amino acids have different sizes, freedom of movement, and interactions with the environment and other amino acids. Given these many variables represented in the model, the CG models intent to restrict such features to make more feasible simulations.

Due to the computational power issue, simplistic representations of protein structure are the most widely used in recent decades (GALVÃO *et al.*, 2012; BENÍTEZ; LOPES, 2013; NUNES *et al.*, 2016). Among these models, there are some with a discrete and continuous degree of freedom of conformational space. Discrete or lattice models are simpler since the protein conformation is limited to the space of a two-dimensional (2D) or three-dimensional (3D) lattice. Figure 10(a) presents an example of a three-dimensional 3D lattice, and an example of a protein with 10 amino acids (see Figure 10(b)).

Unlike the discrete model, the continuous or off-lattice model have a high degree of

Figure 9 – The CG model organization is represented in a mind map. It was divided into two main groups: discrete (lattice) and continuous (off-lattice). Discrete models were grouped in Side Chain and Without Side Chain classification. The continuous model was subdivided by the number of beads that represents an amino acid (one, two-three, and four-six).

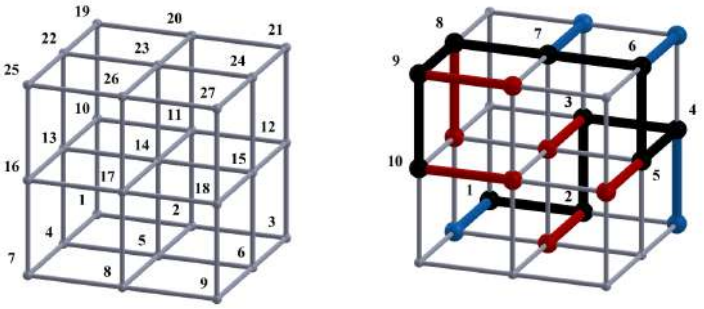


Source: own authorship

freedom, because the structure can assume continuous values of the bond and torsion angles. According to Tozzini (2005), CG off-lattice models can be classified by the number of beads, i.e., the number of elements that represent each amino acid of a protein structure. In general, models with few beads are less computationally expensive. However, the parameters of these force fields are challenging due to the generalization of various behaviors.

Continuous models can assume dihedral angles, representing the angles between two planes formed from three elements (amino acids in the case of CG models). This approach is usually applied to the Ramachandran diagram (RAMACHANDRAN *et al.*, 1963), and a tendency

Figure 10 – (a) **Sample of a 3x3x3 3D-lattice** (b) **Example of a protein with 10 amino acids using a 3D-HP-SC model (black vertex and edge represent backbone, blue and red vertex and edge represent side-chain).**



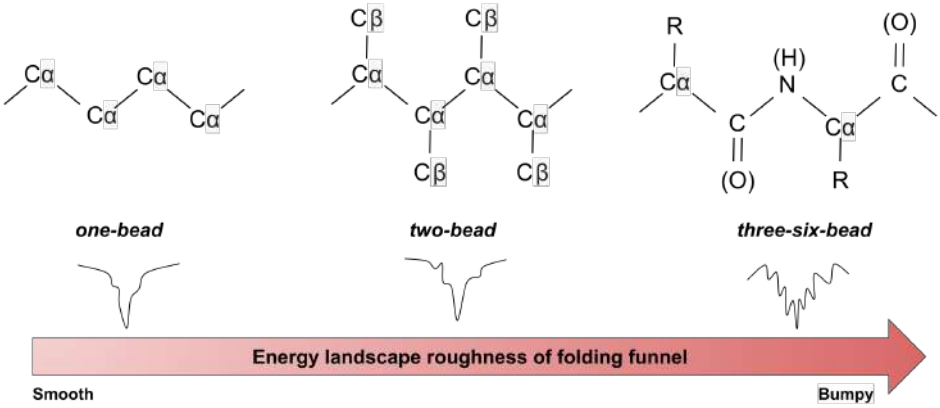
(a) **(b)**

Source: Nunes *et al.* (2016)

of secondary structures to be mapped in the diagram is observed.

Figure 11 presents the one-bead, two-bead, and tree-six-bead models. The one-bead model represents each amino acid as an element. This model has a smaller number of calculations, considering proteins with the same sequence, allowing simulations with large protein sequences in more feasible computation time. The two-bead model adds a second element to the centroid of the side chain, improving the specificity of local iterations. In the three-six-bead models, a single residue represents the side chain, and two or more elements represent the backbone.

Figure 11 – Representation of the models and their respective landscape energy.



Source: Tozzini (2005).

Other discrete and continuous approaches of the literature are presented in Sections 2.4.1 and 2.4.2, respectively.

2.4.1 Discrete models

Discrete and continuous models have been used for high-level behavior analysis (ZHENG; WEN, 2017). Benítez (2015) presented a survey of discrete and continuous models in his thesis, which are described below.

In discrete representation, the protein structure is restricted to a lattice. In this approach, the one-bead model represents each amino acid as a single element. These residues can assume hydrophobic (H) or polar (P) features. This approach describes high-level and less detailed protein conformation behaviors, such as the Go Model, proposed by Go and Taketomi (1978). The Go Model supported the grow of the computational biology area and fostered several studies on protein folding (CIEPLAK; HOANG, 2000; CLEMENTI *et al.*, 2000; KARANICOLAS; BROOKS, 2002). Although simple, its computational complexity is NP-complete (ATKINS; HART, 1999). Besides the H or P features, these models can be modeled in two-dimensional (2D-HP) or three-dimensional (3D-HP) lattices (DILL *et al.*, 1995). Lattice models can also have different binding freedom constraints, such as the "diamond" (SKOLNICK *et al.*, 1988) and the "chess-knight" (SKOLNICK; KOLINSKI, 1991). These designs still allow links between diagonal vertices, unlike the traditional GO model, which only allows connections between orthogonal neighbors.

Lattice models can also have different binding freedom constraints, such as the "diamond" (SKOLNICK *et al.*, 1988) and the "chess-knight" (SKOLNICK; KOLINSKI, 1991). These designs still allow links between diagonal vertices, unlike the traditional GO model, which only allows connections between orthogonal neighbors.

Besides, there is the high resolution two beads model, that considers the Side Chain (SC). They divide the representation of each amino acid into two elements, the backbone and the side chain. It divides variants among these representations SIde CHain Only (SICHO) (KOLIŃSKI; SKOLNICK, 1998), 2D-HP-SC, and 3D-HP-SC (BENíTEZ; LOPES, 2010; NUNES *et al.*, 2016; HATTORI *et al.*, 2020a).

2.4.2 Continuous models

Continuous or off-lattice models are one of the most classic approaches (STILLINGER; HEAD-GORDON, 1995). They differ from lattice representation because it is free in continuous space. Given the higher flexibility of the structure compared to discrete models, and less compu-

tational cost required than multi-beads, these models are very explored in the current literature (LI *et al.*, 2015; KAUSHIK; SAHI, 2017; LI *et al.*, 2017b).

There is the Elastic Network Model (ENM), which is a one bead-model (TIRION, 1996). This model represents amino acids of a protein as a set of particles interconnected by a network of elastic springs. Another approach is the Go model in its off-lattice representation (UEDA *et al.*, 1978), which includes more sophisticated energy equations. Brown *et al.* (2003) also presented a one-bead system that uses *a priori* information from secondary structures for the parametrization of dihedral angle terms.

Tozzini *et al.* (2006) and Wolff *et al.* (2011) employed the one-bead representation, presented other two models. Tozzini *et al.* (2006) proposed a new protein structure representation, using the pseudo-binding and dihedrals angles to produce the Ramachandran plot (RAMACHANDRAN *et al.*, 1963). The combinations of the dihedral angles in the Ramachandran plot, represent conformations of the protein structures. Wolff *et al.* (2011) proposed a model representing the protein in a tubular geometry to describe polypeptide chain evasion effects. In this approach, two parameters are used: one for effective model connectivity and another based on the Go model potential using Contact Map representation.

Gu *et al.* (2009) and Bezkorovaynaya (2011) also proposed Two-bead models. Gu *et al.* (2009) presented a potential energy calculation based on protein fold recognition. The measures use three interactions: contacts between residues, hydrophobicity, and pseudo-dihedral potentials (DEWITTE; SHAKHNOVICH, 1994). In the Bezkorovaynaya (2011) model, they consider simulations with protein and water molecules. Also, implicit interactions are represented between side chains, and CG-level water molecules using the one-bead model means the explicit ones.

The UNited RESidue model (LIWO *et al.*, 2005) is a three-bead representation and considers interactions between $C - \alpha$, side-chain, and the peptide bond. This approach yielded promising results for prediction of the protein structure (CASP10 (HE *et al.*, 2013)), protein-site linkage (SIERADZAN *et al.*, 2012), and for the protein aggregation study (ROJAS *et al.*, 2010). Zacharias (2003) presented another three-bead model with the flexible side-chain, aiming to work with complex protein structure on the protein coupling problems.

Monticelli *et al.* (2008) presented a four-bead model called MARTINI. This model considers four amino acid features (hydrophobic, charged, polar, and non-polar). The PRIME is an intermediate resolution protein model for simulations of polyalanine and polyglutamine aggregation (CHEON *et al.*, 2010). The PRIME 20 model, later extended, includes 20 types

of side chains. This model contains 14 representation groups, according to the hydrophobicity, polarity, size, charge, and hydrogen bonding potential. The Optimized Potential for Efficient Structure Prediction (is a different approach compared to the previous ones, because it considers all backbone atoms, while the side chains are represented as CG models (CHEBARO *et al.*, 2012).

Moreover, Maupetit *et al.* (2007) also presented a six-bead model that applies the OPEC energy function. This function includes terms for hydrogen bridges interactions, potential locations related to torsion, rotation angles, and long-range potentials.

Gopal *et al.* (2010) presented a model called Protein Interactive Modelling (. The protein backbone is represented by three-beads, $C\alpha$ and N elements, and a carbonyl (CO) functional group placed at the geometric center of the C and O atoms. Side-chains are represented by different amounts of elements, from one to five (one-bead to five- beads), depending on the residue type. For example, Alanine and Valine have only one element representing the side-chain, while five elements represent Arginine.

Tian *et al.* (2011) proposed a five-bead template, the backbone of this model is represented by four elements and one for the side chain. The function comprises four energy terms: element contact potentials, sequence-dependent location, solvation, and the β sheet geometry propensity.

Pierri *et al.* (2008) proposed a procedure for re-scaling the biological structures from PDB to compare with the predicted structure in CG models. The results suggest that the CG model can represent a similar structure with the re-scaled physical conformation.

2.4.3 The 3D-AB *Off-lattice* Model

In the 3D-AB model, the residues are simplified and represented by spheres, 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 and equal to one. This constraint helps to decrease the computational cost for extensive simulations. Energetic terms drive the models during the protein folding process, and they could have opposite directions. For example, Angle and Torsion energies are short-distance terms that guide the stretched of the structure. In contrast, the LJ energy is a long-distance term that tends to lead to the compaction of the conformation. 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 Equation 2 (RAPAPORT, 2004; BENÍTEZ, 2015):

$$f = \nabla u(r) = \nabla E_p(\hat{b}_i; \sigma) = \nabla \left(E_{Angles} + E_{Torsion} + E_{LJ} \right), \tag{2}$$

the terms E_{Angles}, $E_{Torsion}$ and the gradient of the E_{LJ} are described below in Equations 7, 11 and 12, respectively.

The equations of motion are given according to Newton's second law, as shown in Equation 3, where N represents the number of residues:

$$f_i = m\ddot{\vec{r}}_i = \sum_{i=j=1(j\neq i)}^{N} f_{ij} \tag{3}$$

According to Newton's third law, which implies $f_{ij} = -f_{ij}$, 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 Equation 4:

$$-\nabla_{r_j} u(\tau_i) = -\frac{du(\tau_i)}{d(\cos\tau)}\Big|_{\tau=\tau_i} f_j^{(i)}, \tag{4}$$

where $u(\tau)$ is the bond-angle potential, and $f_j^{(i)} = \nabla_{r_j}\cos(\tau_i)$. As proposed by (RAPAPORT, 2004), when $\sum_j f_j = 0$, the bond-angle can be represented by Equation 5:

$$\begin{array}{rcl} f_{i-2}^{(i)} & = & (c_{i-1,i-1}c_{ii})^{-\frac{1}{2}} \left[\left(\frac{c_{i-1,i}}{c_{i-1,i-1}} \right) \vec{b}_{i-1} - \vec{b}_i \right], \\ f_i^{(i)} & = & (c_{i-1,i-1}c_{ii})^{-\frac{1}{2}} \left[\vec{b}_{i-1} - \left(\frac{c_{i-1,i}}{c_{ii}} \right) \vec{b}_i \right] \end{array}, \tag{5}$$

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

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

$$-k_1 \sum_{i=1}^{N-2} \hat{b}_i \cdot \hat{b}_{i+1}, \tag{6}$$

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 Equation 4 can be calculated using

$$E_{Angles} = u(\tau) = -k_1 \sum_{i=1}^{N-2} \hat{b}_i \cdot \hat{b}_{i+1} = \sum_{i=1}^{N-2} \cos(\tau_i). \tag{7}$$

The bond-torsion potential is associated with four consecutive residues. For instance, the torsion in the $i-$th 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):

$$\begin{aligned}
\vec{f}_{i-2}^{(i)} &= \frac{c_{ii}}{q_i^{\frac{1}{2}}\left(c_{i-1,i-1}c_{ii}-c_{i-1,i}^2\right)}\left[w_1\vec{b}_{i-1}+w_2\vec{b}_i+w_3\vec{b}_{i+1}\right], \\
\vec{f}_{i-1}^{(i)} &= -\left(1+\frac{c_{i-1,i}}{c_{ii}}\right)\vec{f}_{i-2}^{(i)}+\left(\frac{c_{i,i+1}}{c_{ii}}\right)\vec{f}_{i+1}^{(i)}, \\
\vec{f}_{i}^{(i)} &= \left(\frac{c_{i-1,i}}{c_{ii}}\right)\vec{f}_{i-2}^{(i)}+\left(\frac{c_{i,i+1}}{c_{ii}}\right)\vec{f}_{i+1}^{(i)}, \\
\vec{f}_{i+1}^{(i)} &= \frac{c_{ii}}{q_i^{\frac{1}{2}}\left(c_{ii}c_{i+1,i+1}-c_{i,i+1}^2\right)}\left[w_4\vec{b}_{i-1}+w_5\vec{b}_i+w_6\vec{b}_{i+1}\right],
\end{aligned} \tag{8}$$

where:

$$\begin{aligned}
w_1 &= c_{i-1,i+1}c_{ii}-c_{i-1,i}c_{i,i+1}, \\
w_2 &= c_{i-1,i-1}c_{i,i+1}-c_{i-1,i}c_{i-1,i+1}, \\
w_3 &= c_{i-1,i}^2-c_{i-1,i-1}c_{ii}, \\
w_4 &= c_{ii}c_{i+1,i+1}-c_{i,i+1}^2, \\
w_5 &= c_{i-1,i+1}c_{i,i+1}-c_{i-1,i}c_{i+1,i+1}, \\
w_6 &= -w_1, \\
q_i &= \left(c_{i-1,i-1}c_{ii}-c_{i-1,i}^2\right)\left(c_{ii}c_{i+1,i+1}-c_{i,i+1}^2\right).
\end{aligned} \tag{9}$$

where:

$$\begin{aligned}
w_1 &= c_{i-1,i+1}c_{ii}-c_{i-1,i}c_{i,i+1}, \\
w_2 &= c_{i-1,i-1}c_{i,i+1}-c_{i-1,i}c_{i-1,i+1}, \\
w_3 &= c_{i-1,i}^2-c_{i-1,i-1}c_{ii}, \\
w_4 &= c_{ii}c_{i+1,i+1}-c_{i,i+1}^2, \\
w_5 &= c_{i-1,i+1}c_{i,i+1}-c_{i-1,i}c_{i+1,i+1}, \\
w_6 &= -w_1, \\
q_i &= \left(c_{i-1,i-1}c_{ii}-c_{i-1,i}^2\right)\left(c_{ii}c_{i+1,i+1}-c_{i,i+1}^2\right).
\end{aligned} \tag{10}$$

According to (IRBÄCK *et al.*, 1997), the potential associated by the torsion-angle ($E_{Torsion}$) force for the AB model is described by Equation 11, where $k_2 = -0.5$.

$$E_{Torsion} = -k_2\sum_{i=1}^{N-3}\hat{b}_i\cdot\hat{b}_{i+2} \tag{11}$$

The Lennard-Jones potential represents the interactions between residues based on their distance and hydrophobicity. Its gradient is defined by Equation 12.

$$f_{ij} = \nabla E_{LJ} = 48\cdot\varepsilon\left(\sigma_i,\sigma_j\right)\left(r_{ij}^{-14}-\frac{1}{2}r_{ij}^{-8}\right)\cdot\vec{r}_{ij}, \tag{12}$$

Algorithm 1 – Shake algorithm

Start
Coordinates correction:
$\gamma \leftarrow \frac{\vec{r}_{ij}^{2'} - b_i^2}{4(\delta/2)^2(1/m_i+1/m_j)\vec{r}_{ij}\cdot\vec{r}'_{ij}}$
while $|\gamma| < 10^{-k} \cdot b_i^2$ **do**
  $\vec{r}'_i \leftarrow \vec{r}_i - \gamma\vec{r}_{ij}$
  $\vec{r}'_j \leftarrow \vec{r}_j + \gamma\vec{r}_{ij}$
  $\gamma \leftarrow \frac{\vec{r}_{ij}^{2'} - b_i^2}{4(\delta/2)^2(1/m_i+1/m_j)\vec{r}_{ij}\cdot\vec{r}'_{ij}}$
end while
Velocities correction:
$\gamma = \frac{\dot{\vec{r}}_{ij}\cdot\vec{r}_{ij}}{2\vec{r}_{ij}^2}$
while $|\gamma| < 10^{-k} \cdot b_i^2$ **do**
  $\dot{\vec{r}}_i \leftarrow \dot{\vec{r}}_i - \gamma\vec{r}_{ij}$
  $\dot{\vec{r}}_j \leftarrow \dot{\vec{r}}_j + \gamma\vec{r}_{ij}$
  $\gamma = \frac{\dot{\vec{r}}_{ij}\cdot\vec{r}_{ij}}{2\vec{r}_{ij}^2}$
end while

Source: Benítez (2015)

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 Equation 13.

$$\varepsilon\left(\sigma_i, \sigma_j\right) = \begin{cases} 1 & \text{if } AA \text{ interaction,} \\ 0.5 & \text{if } BB \text{ or } AB \text{ interactions.} \end{cases} \tag{13}$$

Due to the constraints imposed on the model used in this work by the unit distance between subsequent residues of the chain, the Shake algorithm was employed (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 ($m = 1$).

2.5 METHODS APPLIED TO THE PROTEIN FOLDING PROBLEM

To predict the protein structure is not a trivial task, and it is still an open problem (DILL *et al.*, 2008). The development of computational methods is a necessary to the progress of this problem, given the difficuties of the experimental approaches. Overall, these methods can be divided into three categories: comparative modelling (homology), fold-recognition (threading) and first-principles (*ab initio*) (MALDONADO-NAVA *et al.*, 2018).

Methods for protein structure prediction using the homology approach depend on a set of protein samples. This method uses known structures from a database, such as PBD or

Table 2 – Main protein databases.

Database	Description	Web Address
PDB	Biological molecular structure	http://www.rcsb.org/
UniProtKB	Information about protein sequence and functionality	http://www.uniprot.org/
PIR	Integrated information that support genomic, proteomic and sysstem biology	http://pir.georgetown.edu/
PROSITE	Information about protein domains, families and functionalities sites	http://www.expasy.org/
Prints	Information about protein sequence, focus on protein 'figerprinting'	http://www.bioinf.man.ac.uk/
BLOCKS	A homology database (not update)	http://blocks.fhcrc.org/
eMOTIF	A database from protein motifs, it is derived from BLOCKS and PRINTS database	http://motif.stanford.edu/
PRODOM	Protein domain families information extracted from Uniprot	http://prodom.prabi.fr/
InterPro	Database with protein families, domain and sites informations	http://www.ebi.ac.uk/

Source: Benítez (2015)

IniProtKB (see Table 2), to predict the conformation of a new protein is used single or multiple alignments of the primary representation. According to Orengo *et al.* (2003), for the alignment between sequences higher than 70 %, this is a more manageable and reliable process. For the alignment with lower value, it requires multiple sequence alignments. A practical roadmap for this method is the increasing amount of data available in databases (BOHNUUD *et al.*, 2017).

The threading or fold-recognition approaches use statistical analysis of previously known structures (JONES *et al.*, 1992). This approach is based on the principles of the limited number of conformation that can be assumed by an amino acid sequence. The threading method complements the comparison method, as it requires a priori protein structures homologous to the target sequence. The target sequence is compared to portions in other sequences containing known structures and reasonable similarity. Several similarity factors can be used, such as the energy level, the similarity between amino acid sequences, alignment penalties, among others (BONETTI, 2015).

The *ab initio* method predicts the protein structure without previous knowledge by minimizing the free energy of the protein structure using physical principles. Unlike comparison and threading methods, the *ab initio* approach assumes no prior information. This feature allows its applicability when no information is available to support the structure prediction or when the information is limited or precarious. The predicted conformation is evaluated with an objective function related to a force field (ZAKI *et al.*, 2004). Several approaches have been published with different energy evaluation functions and optimization methods (see Section 3.2). The commonality between these approaches, the Molecular Dynamics (MD) method, has been

highlighted (DILL; MacCallum, 2012).

2.6 MOLECULAR DYNAMICS

The MD method is an *ab initio* computational approach that uses physical forces to simulate the particles motion (atoms or molecules). The theories of motion mechanics are the theoretical base of the MD algorithm. However, this canonical approach exceeds Newton's Second Law. This approach is applied to all atoms or particles simultaneously over t iteration, generating a protein folding trajectory information. With the folding data information, it is described and analyzed how protein structures evolve (KMIECIK *et al.*, 2016).

McCAMMON *et al.* (1977) presented one of the first studies that applied the MD algorithm. Since then, several works have been developed (LEVITT, 1983; KARPLUS; McCAMMON, 2002; MIAO *et al.*, 2015; RYCKBOSCH *et al.*, 2017). The MD algorithm is still currently explored, for example, the MD algorithm for PFP using the 3D-AB off-lattice model in the canonical ensemble, and also in other studies involving dynamic folding processes (BARRETO-OJEDA *et al.*, 2018; MICHELARAKIS *et al.*, 2018b) were presented by Benítez and Lopes (2012).

The MD method aims predicting the a native structure based on the protein folding simulation. In its canonical ensemble, the simulation is deterministic, unlike other approaches, such as Monte Carlo (MC) (LI; SCHERAGA, 1987), Replica Exchange (RE), and Umbrella Sampling (US). In this regard, a protein fold trajectory will always be the same considering the same initial structure, velocities, and temperature. Besides, the round-off error is noteworthy, since they can lead to a difference in the final values, and this can generate inconsistent values with different calculation sequence. Then, the canonical ensemble is also dependent on the calculation sequence, consequently, by the hardware used to perform this calculation (FLEISCHMANN *et al.*, 2019; IAKYMCHUK *et al.*, 2020). Differently from the sampling/statistic approaches, the canonical ensemble preserves the Newtonian feature, which is essential for the dynamic study of the PFP and folding trajectories (RAPAPORT, 2004).

The MD method requires a high computational cost to perform the simulation, given the number of calculations of all elements interactions at each simulation snap (MERMELSTEIN *et al.*, 2018). The Lennard-Jones (LJ) energy calculation is an example of an process that require a high computational power (HOWARD *et al.*, 2019a), demanding significant time for the simulation (HATTORI *et al.*, 2020b). In simulations of complex systems, such as in the protein

aggregation, the computation cost problem still increases since the number of force calculations escalates (WEN *et al.*, 2017).

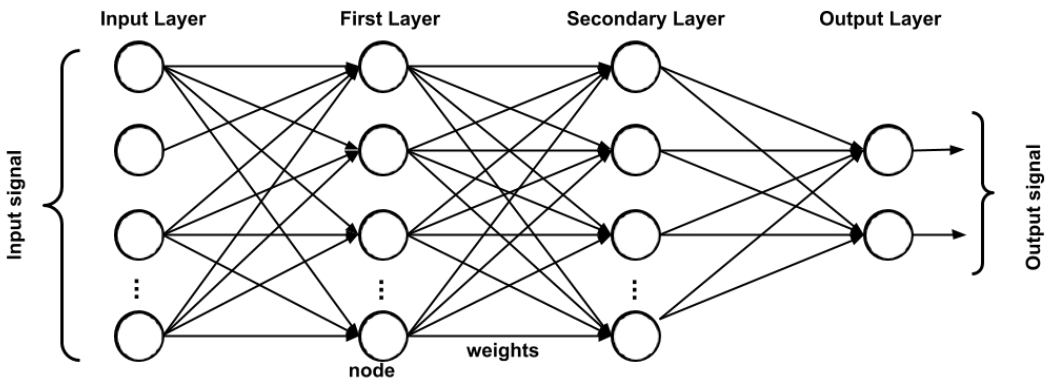
CG models and parallelization of sequential codes have been used to minimize the computation cost (ABRAHAM *et al.*, 2015). Commonly, high-performance architectures employ Graphics Processing Unit (GPU) to accelerate the MD method (STONE *et al.*, 2017). Besides, it has been applied techniques to minimize the amount of calculations, such as the Neighbourhood List algorithm (PENNYCOOK *et al.*, 2013), which omits pairwise iterations based on distance threshold.

2.7 DEEP LEARNING

Deep Learning (DL) methods are based on Artificial Neural Networks (, which are mathematical models for information processing inspired in biological neurons (MCCULLOCH; PITTS, 1988; SCHNEIRLA *et al.*, 1963; RUMELHART *et al.*, 1986). DL methods are multilayer networks, where each layer acts as a feature extractor, adding different levels of abstraction at each layer.

A simple ANN, called multilayer perceptron, is composed of nodes and weighted links (w) that connect two nodes of a layer to the next one, as shown in Figure 12. The data input accomplishes the activation of each node or neuron of the input layer. These values are propagates forwardly across the network through the weights, allowing information to stream with more or less intensity from the hidden layer to the output layer. This process is called *feed-forward* because of the data flow propagated across the network.

Figure 12 – Scheme of a MultiLayer Perceptron Network.



Source: own authorship

The process that represents the feed-forward is presented in Equations 14, 15, 16, and

17.

$$a_j = \sum_{i=1}^{I} w_{ij} x_i \qquad (14) \qquad\qquad h_j = act\,(a_j) \qquad (15)$$

$$a_k = \sum_{j=1}^{J} w_{jk} h_j \qquad (16) \qquad\qquad y = act\,(a_k) \qquad (17)$$

where I represents the number of feature vector $\vec{x}$, act is a non-linear function, called the activation function, a_j and a_k represent the sum of the neurons in the input to hidden layers, and the hidden to the output layers, respectively. The outputs of the hidden layer and output activation functions are represented by h_j and y, respectively. There are several activation functions used in DL, such as sigmoid (Equation 18), hyperbolic tangent (Equation 19) and Rectified Linear Units ((Equation 20).

$$sigm(z) = \frac{1}{1+e^{-z}} \quad (18) \qquad tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} \quad (19) \qquad ReLU(z) = max(0,z) \quad (20)$$

In order to minimize the network output error it is necessary to update its weights. For this optimization, the Gradient Descendent method (is used (HECHT-NIELSEN, 1989). The GD is based on the partial derivative of a cost function Q concerning the weights, and it is presented in Equation 21.

$$w^{new} = w^{old} - \eta \frac{\partial Q}{\partial w}, \qquad (21)$$

where w^{old} represents the weight before the update, and w^{new} represents the updated weight, η is the learning rate, which is higher than zero and represents the step size of the GD algorithm. Weights can be updated using the mean error of a sample set, called batch. With its application, the training process is more efficient because the updates in the network weights are less frequent, and the error tends to decrease, given the update is performed on the errors average.

There are several ways to update weights, such as the Stochastic Gradient Descent, which uses a subset of the dataset, called a batch, randomly chosen to calculate the GD average. On the other hand, the Momentum (QIAN, 1999) and RmsProp (TIELEMAN; HINTON, 2012) add a momentum term to calculate the update. There are still methods that allow η to be auto-adjusted, theses approaches considering the values of the network weights, such a AdaGrad (DUCHI *et al.*, 2011) and Adam (KINGMA; BA, 2014). Similarly, AdaDelta (ZEILER, 2012) uses a limited amount of past gradients to calculate the current parameter setting.

One of the challenges for training a DL network is the over-fitting problem (LECUN *et al.*, 2015). In this problem, the network can achieve satisfactory results in the training set, but the

predicted results do not achieve similar performances when new data is presented for testing in this trained model. In this regard, regularization methods to increase the generality of models have been proposed in the literature. Dropout can be understood as a regularization technique of the neural network by adding noise to its hidden units. This technique is done by randomly removing connections between network nodes during the training process (SRIVASTAVA *et al.*, 2014). Other forms of regularization are the L1 and L2 (NG, 2004), where a term in the cost function is added based on the values of the network weights, forcing them to have small values. Moreover, classical machine learning techniques, such as the data augmentation (AQUINO *et al.*, 2017a), or the acquisition of more data to train the network could be also used to avoid the over-fitting behaviour.

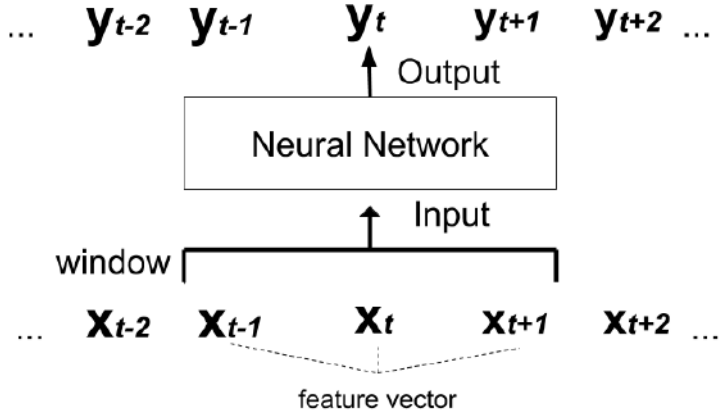
Several methods in DL have been proposed in recent years, networks can be classified into acyclic (feed-forward) and cyclic (recurrent). Among the acyclic approaches, there are the Deep Neural Network (, Auto-encoders, and Convolutional Neural Networks. In the recurrent methods, there are the traditional Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997), Gated Recurrent Unit (GRU) (CHO *et al.*, 2014), among others networks.

2.7.1 Recurrent Neural Network

In the beginning, the feed-forward network was used for sequential problems. This network used the sliding window technique (PALIWAL *et al.*, 2015). In this technique, a fixed-size window slide across the feature vectors ($\vec{x}$) (i.e., $t-1$, t, and $t+1$), and insert them concatenated in the input layer. Usually, the prediction/classification focus is the feature vector of the center of the window (Figure 13). Although it allows the use of feed-forward in sequential problems, the information of the prediction is restricted to the size of the window.

Recurrent Neural Networks (RNN) or "vanilla" are MultiLayer Perceptron (MLP) networks with a memory (Figure 14). In this network persists the information is processed during the sequence of data input presented to the network. These approaches are commonly used in problems where the sequence of data inserted into the network brings additional information, such as speech recognition, language processing, translation, and image subtitling. They also appear in many Bioinformatics problems, such as pattern recognition in nucleotide sequences and amino acid patterns in proteins (MIN *et al.*, 2017). Several networks have been presented in the literature, turning this method popular, such as Elman (ELMAN, 1990), Jordan (JORDAN,

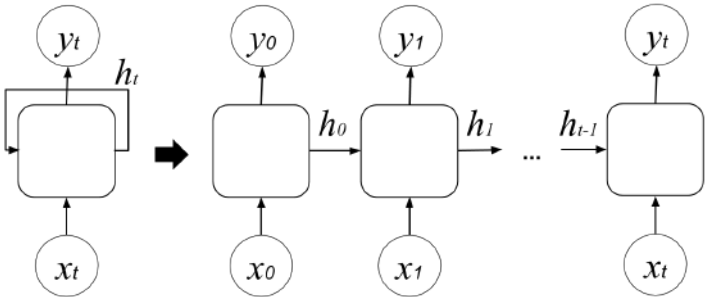
Figure 13 – Sliding window method at the instant t.



Source: own authorship

1990), and time delay neural network (LANG *et al.*, 1990) networks.

Figure 14 – Recurrent Neural Network in a simplified and extended representation.



Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed in January 2021.

In RNNs, inputs sequence ($\vec{x}$) is provided to the network, as shown in Figure 14. At each network input, the information is processed, and an output is produced (y). The information generated by the network is stored in the memory (h) and insert in the next input step. This process occurs iteratively until reach the end of the input sequence. The calculations used to obtain the RNN output are presented in Equations 22, 23, 24, and 25.

$$a_j^t = \sum_{i=1}^{I} w_{ij} x_i^t + \sum_{i=1}^{I} w_{jj} h_j^{t-1} \quad (22) \qquad h_j^t = act\left(a_j^t\right) \quad (23)$$

$$a_k^t = \sum_{j=1}^{J} w_{jk} h_j^t \quad (24) \qquad y^t = act\left(a_k^t\right) \quad (25)$$

where h_j^t represents the hidden layers and y^t represents the output layer. Also, w_{ij}, w_{jj}

and w_{jk} represent the weights of the RNN.

Several adjustments of RNN weights are present in the literature, such as Real-Time Recurrent Learning (ROBINSON; FALLSIDE, 1987), and BackPropagation Through Time ((WERBOS, 1990). However, the BPTT algorithm stands out for being computationally faster and simpler (GRAVES *et al.*, 2005). Equation 26 presents the BPTT algorithm.

$$w^{epoch} = w^{epoch-1} - \eta \frac{\partial Q_t}{\partial w}, \tag{26}$$

where $epoch$ represents an iteration in the training process, η represents the learning rate of the GD algorithm, and $\frac{\partial Q_t}{\partial w}$ represents the partial derivative of the output cost function t concerning the RNN weights.

As the GD algorithm, the BPTT algorithm consists of a repeated application of the chain rule (CHEN, 2016), as presented in Equation 27.

$$\frac{\partial Q_t}{\partial w} = \sum_{k=0}^{t} \frac{\partial Q_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial w}, \tag{27}$$

where $\frac{\partial Q_t}{\partial y_t}$, $\frac{\partial y_t}{\partial h_t}$ and $\frac{\partial h_k}{\partial w}$ represent their partial derivatives of the cost with the output function, output with the memory weights, and memory weights concerning the network weights. The BPTT calculation for recurrent networks depends not only on the hidden layer but also on previous influences inputs. This difference is reflected in the partial derivative term $\frac{\partial h_t}{\partial h_k}$ that depends on the previous input and the current state, see Equation 28.

$$\frac{\partial h_t}{\partial h_k} = \sum_{i=0}^{t} \frac{\partial h_i}{\partial h_{i-1}} \tag{28}$$

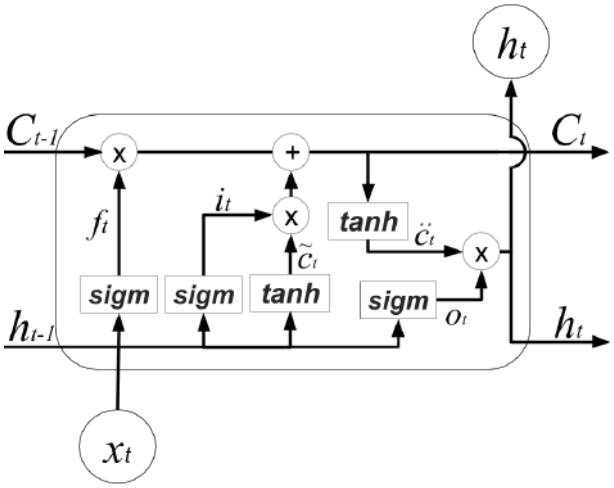
Although traditional RNNs are a suitable method to deal with sequential problems, they can not store information in memory for a long time, affecting the capability to connect distant inputs in the sequence. Then, a particular RNN approach was developed to solve this problem, called Long Short-Term Memory.

2.7.2 *Long Short-Term Memory*

The Long Short-Term Memory (LSTM) network proposed by Hochreiter and Schmidhuber (1997) was developed with the intention of decreasing the impact of the long-term dependency learning problem.

LSTM networks have a different structure from those of traditional RNNs, as shown in Figure 15. The LSTM framework contains a link system (peephole connections) connected through gates. This system allows the gradient to flow during inputs without losing long term memory (C_t) (HOCHREITER; SCHMIDHUBER, 1997). The LSTM structure, also called cell, is made up of three gates: forgot gate (f_t), update gate (i_t), and output gate (O_t). In order for the gates to achieve their respective objectives, the concatenated input information (x_t) and short-term memory (h_{t-1}) are imputed.

Figure 15 – Cell of the Long Short-Term Memory.



Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed in January 2021.

The forgot gate weighs the information that remains in long-term memory. Meanwhile, the update gate aims to control the amount of information entered there. Finally, the output gate defines how long-term information generates short-term memory information and network output.

As the traditional RNN, the variable x_t represents the network´s input while h_t represents the output. The layers with sigmoid activation function and hyperbolic tangent are represented respectively by $sigm$ and $tanh$. Equations 29 - 35 present mechanisms of LSTM.

$$f_t = sigm(W_f[h_{t-1},x_t] + b_f) \quad (29) \qquad i_t = sigm(W_i[h_{t-1},x_t] + b_i) \quad (30)$$

$$\widetilde{C}_t = tanh(W_C[h_{t-1},x_t] + b_C) \quad (31) \qquad C_t = f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t \quad (32)$$

$$o_t = sigm(W_o[h_{t-1},x_t] + b_o) \quad (33) \qquad \ddot{C}_t = tanh(C_t) \quad (34)$$

$$h_t = o_t \circ \ddot{C}_t \quad (35)$$

where, f_t, i_t, C_t, o_t, and h_t ($\in \Re^k$) are activations of the forgot gate, update gate, LSTM internal memory at time t, output gate, and network output. Besides, W_f, W_i, W_C and W_o

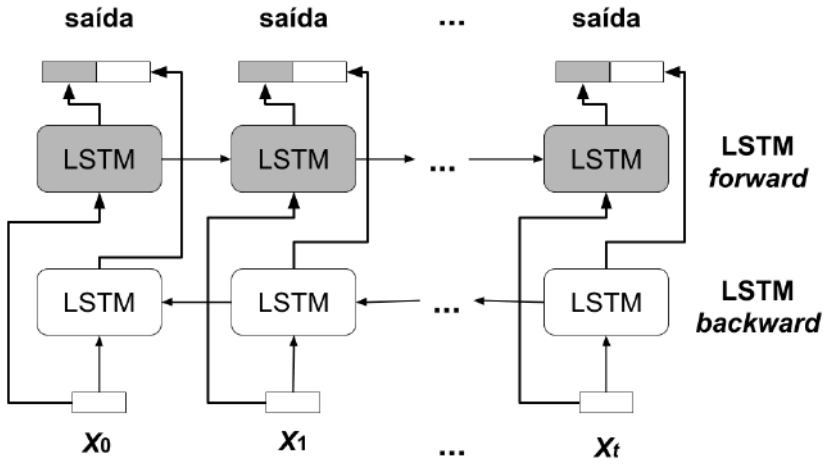
($\in \Re^{3q \times k}$) are network weights; and b_f, b_i, b_C and b_o ($\in \Re^k$) represent the bias.

Other studies proposed new models to modify the peephole connections and the gate system of the cell with the popularization of the LSTM. Gers and Schmidhuber (2000) added two connections: one at the forgot gate and the other in the output gate. The long-term memory information links to both connections (forgot and output gates) in this network. Cho *et al.* (2014) presented another method called Gated Recurrent Unit (GRU). In this case, it is placed together with the forgot and the output gate. Unlike LSTM, the long and short term memories in this approach are merged into a single memory. Given the more straightforward nature, GRU has been receiving attention (LI; YU, 2016), but overall, the LSTM network is giving more satisfactory results (BREUEL, 2015).

New models are still emerging, including LSTM-like structures, such as Depth-Gated Recurrent Neural Networks, and divergent structures, such as Clockwork. A question arises whether these proposed networks are relevant and whether these extent variants are beneficial. According to Greff *et al.* (2017) and Jozefowicz *et al.* (2015) these models have only advantages in specific problems over LSTM.

In addition to the recurrent network variants, there are also the bidirectional networks, such as the Bidirectional LSTM network, which allows the merge information from future inputs (backward) with the current input information (forward) (GRAVES *et al.*, 2005), as shown in Figure 16. For example, a secondary structure classification can be enhanced with the knowledge of subsequent and previous amino acid features (HATTORI *et al.*, 2017a).

Figure 16 – Scheme of the Bidirectional Long Short-Term Memory.



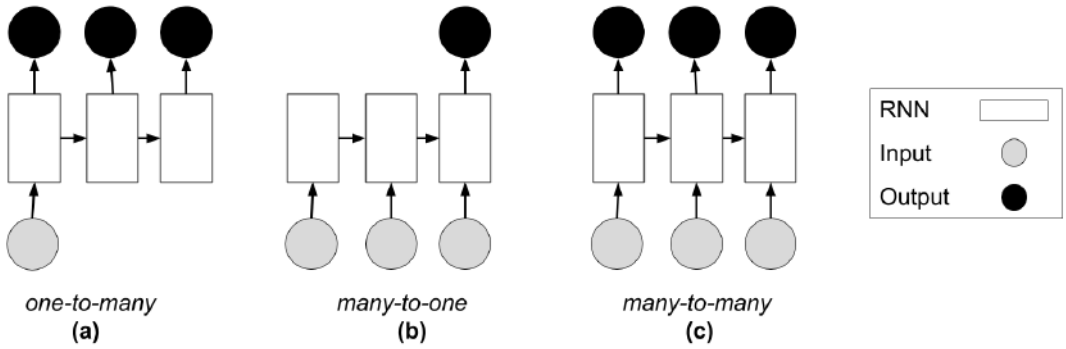
Source: own authorship

Despite the highlights of the RNN approach in sequential problems, the flexibility of

problems that they can handle using different models was fundamental for disseminating these techniques. Figure 17 presents these models.

- One-to-many model: it obtains a sequence of outputs from a network input. This model is usually applied to generate sequences (ALEMI *et al.*, 2017).
- Many-to-one model: it receives an output from a string input. This model was used for the prediction/classification of feelings (MARGARIT; SUBRAMANIAM, 2016).
- Many-to-many model: it receives an input data sequence and generates an output sequence. This model was applied to predict protein torsion angles (LI *et al.*, 2017).

Figure 17 – Recurrent Neural Network models one-to-many, many-to-one, and many-to-many.



Source: own authorship

2.8 HIGH-PERFORMANCE COMPUTING

The High-Performance Computing (area is related to the progress of computational methods (HOWARD *et al.*, 2019b), given that the HPC hardware allows executions in a more feasible time and with a higher number of calculations. When tasks are potentially parallelizable, concurrent approaches decrease time-consumption and increase the speedup performance compared to sequential methods (HATTORI *et al.*, 2020b).

Currently, several architectures allow parallelization of sequential methods. For example: multi-core processors, Field Programmable Gate Array (FPGA), cluster Beowulf, and Graphics Processing Unit (GPUs) (GIZOPOULOS *et al.*, 2019). This hardware has multiple processing cores that can be used for parallel calculations in different ways. Among these architectures, GPUs have been outstanding due to their cost-effectiveness. The purchasing and maintaining of a GPU are not costly compared to a cluster Beowulf. The GPU hardware has also a large number

of processing cores when compared to CPU multi-core hardware. Another factor are the libraries available for the GPU architecture development, such as Compute Unified Device Architecture (CUDA), Keras, Tensorflow, Torch, among other frameworks (GUTOSKI *et al.*, 2018).

The CUDA library has been fostered the experience of parallel programming in the NVIDIA GPU platform. Consequently, applications that required intensive processing are being converted to the use of this high-performance hardware (ESSAID *et al.*, 2019). An advantage of using GPU is that the developer and operational system do not require managing the organizational processing architecture. This processing architecture is composed of threads organized by blocks, and these blocks are arranged in a grid. Besides, the blocks of threads can be executed concurrently, and they could be performed in a different order (HENNESSY; PATTERSON, 2011). A thread in GPU is accessible using the coordinates of block index ($BlockID$) and thread index ($ThreadID$). The grid and block dimensions can be accessible using $gridDim$ and $blockDim$, respectively.

The parallelization with GPU hardware allows a massive number of threads when compared with the number of CPU hardware. The threads in GPU are organized in grids, which in turn are organized in blocks (Lindholm *et al.*, 2008). At each level of organization, they can share memory space, while having local memory. Threads can be initialized from a process that runs on a CPU, as the master-slave technique, where the process (master) manages the threads (slaves). The massive parallelism allowed by the GPU hardware can be a useful technique to decrease the computation time of a program. However, the communication and initialization of threads generate overhead on the communication time (TANGHERLONI *et al.*, 2018). Therefore, the parallelized tasks must be performed massively, and each task needs to require a high computational cost to counterbalance the time spent in the management (YANG *et al.*, 2018).

3 RELATED WORKS

This chapter presents a literature review of related studies. The research highlights open problems and the current state-of-the-art methods, unfolding promising research directions. This Chapter is divided into two Sections: DL methods applied to proteomics and computational methods applied for the protein folding problem.

We considered electronic databases in the Computer Science and Technology fields, such as the ACM Digital Library[1], IEEE Xplore [2], Science Direct[3], Scopus[4], Google Scholar[5] (accessed 24 November 2020)), and a specific Bioinformatics-related basis, the National Center for Biotechnology Information (. The Nature journal was also considered, given its scope in the Bioinformatics research field.

3.1 DEEP LEARNING APPLIED TO PROTEOMICS PROBLEMS

To the best of our knowledge, there have been no studies using DL for the protein folding problem up to this point. Thus, the focus of this Section has become to research DL in proteomics, which is a broader (or closely related) subject. In proteomics, DL has been used in secondary and tertiary structure prediction.

There are many pieces of research related to DL applied to the protein structure prediction problem. In order to group some of these works, Paliwal *et al.* (2015) presented a brief review of the literature. The authors divide the research into two types of mechanisms: Feed-forward DNNs, which uses the sliding window strategy, and Other Architectures, which are RNNs. Another present factor is the impact of RNNs in these problems. They were developed for issues which the data sequence adds information about the problem, such as the primary structure information being included in the classification of proteins. Two networks are pointed out as methods that may stand out in the future: Neural Turing Machines (GRAVES *et al.*, 2014) and *Memory Networks* (SUKHBAATAR *et al.*, 2015).

Min *et al.* (2017) presented a literature review of DL applied to Bioinformatics. The authors highlighted DL techniques applied to the protein structure prediction and classification.

[1] dl.acm.org/ (accessed 24 November 2020)
[2] ieeexplore.ieee.org/ (accessed 24 November 2020)
[3] sciencedirect.com/ (accessed 24 November 2020)
[4] scopus.com.br/
[5] scholar.google.com/

According to the authors, one of the main open problems is the large number of unbalanced datasets found in the literature.

Similar to Min *et al.* (2017), Angermueller *et al.* (2016) presented a survey of Bioinformatics applications using DL. The main difference from Min *et al.* (2017) 's work is the breadth of Bioinformatics studies and issues. For instance, the authors presented issues such as the prediction of secondary and tertiary structures, identification of disorder regions, and the prediction of contact between amino acids. The authors also presented suitable questions about DL methods, such as how to define the DL model architecture, how to best train them, avoiding over-fitting, ways to train in GPUs, among other questions. Finally, the work reinforces DL obstacles, such as identifying a generic DL architecture that works for different issues, and that the optimization of hyperparameters could be computationally costly and sometimes unfeasible.

As presented in the previous survey, the secondary protein structure prediction has gained a lot of space in DL applications, where for each amino acid, its secondary structure class is predicted. Wang *et al.* (2011) proposed a new probabilistic method based on the Conditional Neural Fields (for the secondary structure prediction problem. Subsequently, Zhou and Troyanskaya (2014) presented his approach named Generative Stochastic Network, which aims to improve the prediction of local secondary structures. A similar technique was also proposed by Yaseen and Li (2014b), who explores template methods using a DL approach, named C8-SCORPION.

Recently, in the secondary structure prediction problem, an approach called Deep Convolutional Neural Fields (DCNF) was proposed by Wang *et al.* (2016). This work obtained similar results compared to the SSpro (MAGNAN; BALDI, 2014), PSIPRED (SPENCER *et al.*, 2015), JPRED (CUFF *et al.*, 1998) and RaptorX-SS8 (WANG *et al.*, 2011) approaches. Another paper, presented by Li and Yu (2016), used an approach named Cascaded Convolutional and Recurrent Neural Networks (CCRN) based on RNNs. This method employs stacked and convolutional GRU cells. This is the current state-of-the-art work for the secondary structure classification problem. According to the authors, the results suggest that the RNN-based approach is appropriated for working with protein structure prediction.

Given the protein structure representation, the Contact Map [6] (is one of the representations used by predictors. Lena *et al.* (2012a) developed a 2D RNN for amino acid contacts forecast integrating spatial and temporal information. In Lena *et al.* (2012b) a DL approach was

[6] The Contact Map is a 3D structure of a protein in a 2D matrix, where the position (i, j) represents the contact between amino acids i and j of the protein.

applied for predicting the amino acid contacts of a protein. The technique presented here proposes a three-dimensional stacked neural network NN_{ij}^{k}, where i and j represent the spatial coordinate indicators of the CM, whereas k is the index of time (Deep Spatio- Temporal Network). The results indicated that the forecast of the proposed method was superior compared to the literature methods (MLP and RNN).

Wang *et al.* (2017) presented a DL technique for predicting CMs, called ultra-deep residual convolutional networks. The advantage of this approach is due to its use of information from the primary and secondary structures to infer the CMs. The approach suggested superior results when compared to the previous most modern methods: CCMpred (SEEMAYER *et al.*, 2014), and MetaPSICOV (JONES *et al.*, 2014). Hanson *et al.* (2018) proposed to predict CMs using RNNs, specifically, the Convolutional Bidirectional LSTM network. The results reached competitive values with highly advanced ones presented by Wang *et al.* (2017). The proposed approach highlighted the importance of predicting linkage between amino acids which are distant in the sequence.

Although the CM representation is simple, because it is a 2D representation, the reconstruction of the 3D structure is not trivial (BENÍTEZ *et al.*, 2015). An alternative procedure is to work with a three-dimensional structure using twisted angles. According to Staples *et al.* (2019), DL approaches have been shown promising results in the protein structure prediction problem during this decade, which is described below. Therefore, Wood and Hirst (2005) proposed a predictor to forecast the psi angles of the protein conformation, aiming to identify secondary structures using MLP networks, called DESTRUCT. Subsequently, Xue *et al.* (2008) presented another approach, named Real-SPINE2.0, which applies a set of MLP networks, based on the work of Dor and Zhou (2007), to predict torsion angles. Wu and Zhang (2008) also presented an approach that uses a set of prediction methods, MLP with Support Vector Machine, called ANGLOR.

DL approaches have also been applied to protein torsion angles prediction and overcame the results compared to the previous studies. Torsion angle prediction is commonly used in the Ramachandran diagram for secondary structure prediction and sequence alignment. Yang *et al.* (2017) presented a DL method based on multiple fully connected stacked layers, called SPIDER2. In this work, the method was tested in several case studies, including the prediction of secondary structures and torsion angles. Subsequently, Li *et al.* (2017) presented a benchmark with four DL methods for the torsion angle prediction problem. These four methods include Deep

Neural Network (DNN), Deep Recurrent Neural Network (DRNN), Deep Restricted Boltzmann Machine Network (DRBM), and Deep Recurrent RBM network (DRRBM). The paper compares its method with two approaches from the literature: TANGLE (SONG *et al.*, 2012) and SPIDER2. The tested methods achieved superior and comparable results to state-of-the-art methods. Among the tested approaches, RNN achieved higher results than non-recurrent networks.

Fang *et al.* (2018) [7] presented another work for predicting torsion angles using a method called DeepRIN. The composition of the DeepRIN approach is based on a network for image processing, and its architecture comprises Inception layers (SZEGEDY *et al.*, 2017) and Residual Networks (HE *et al.*, 2016). When compared with the SPIDER3 (HEFFERNAN *et al.*, 2017) method, the approach presented superior results considering the MAE value. AlQuraishi (2019) proposed a protein structure prediction method based on a co-evolution using neural networks called Recurrent Geometric Network (RGN). On all CASPs, RGNs obtained the best performance when compared to servers that use co-evolution data (KRYSHTAFOVYCH *et al.*, 2016) and benchmarked to both short and long multi-domain proteins (SCHAARSCHMIDT *et al.*, 2018).

3.1.1 *Long Short-Term Memory* (LSTM)

Among DL approaches, LSTM has shown the ability to work with sequential data in different problems, with promising results (LI; CAO, 2018; SUN; GONG, 2020; SHEN *et al.*, 2020).

Sønderby and Winther (2015) applied the Bidirectional LSTM Network (to the secondary structures classification, considering eight types of structures. In this work, the CB6133 dataset was used for training and the CB513 dataset was used for the test procedure. The proposed method was compared to the other three approaches in literature (BRNN, CNF, and GSN). The BLSTM obtained better results compared to the others in terms of accuracy per class.

Sønderby and Winther (2015) applied the Bidirectional LSTM network (to classify the secondary structures, considering eight types of structures. In this study, the CB6133 dataset was used for training, and the CB513 dataset was used for the test procedure. The proposed method was compared to the other three approaches (BRNN, CNF, and GSN). The BLSTM obtained better results when compared to the others in terms of accuracy per class.

In Lipton *et al.* (2015), a RNN survey was presented based on the last three decades, and how these computational methods have become a practical and robust model for dealing

[7] available at http://dslsrv8.cs.missouri.edu/~cf797/MUFoldAngle/ (accessed 24 November 2020)

with sequential problems. This article raises questions about sequential models, for example: information about traditional RNNs as well as its training, and the most current RNNs methods, such as LSTM, BLSTM, and Neural Turing machines.

Fleming *et al.* (2016) presented an application of the LSTM to predict the protein thermal stability after mutation. The work used simulation data extracted from the Molecular Dynamics (MD) approach. The results of this model achieved similar predictions when compared to Machine Learning approaches and the other two off-the-shelf software.

For the protein function prediction,Liu (2017) proposed the use of the LSTM. The method employs the classification protein functionality based on the primary sequence information feature. When compared with softwares in the literature, such as BLAST and HMMER, the RNN approach showed better results. The author cites the benefits of the LSTM, such as its ability to extract relationships between intricate patterns in data that comparative approaches were not capable of doing it.

Heffernan *et al.* (2017) conducted three case studies on proteomics applying LSTM networks, called SPIDER3. This research includes the secondary structure prediction, protein structure angle prediction, and the protein solvent accessibility forecast. Overall, the LSTM network achieved superior state-of-the-art results for all case studies (DCNF, SPIDER2, PORTER 4.0 (MIRABELLO; POLLASTRI, 2013), SCORPION-C3 (YASEEN; LI, 2014a), PSIPRED 3.3 (JONES, 1999), SPINE-X (FARAGGI *et al.*, 2012) and Jpred4 (DROZDETSKIY *et al.*, 2015)).

Li *et al.* (2017a) applied the BLSTM network to predict protein homology. The authors showed that DL methods automatically extract characteristics from raw data, with any or few pre-processing steps, especially when compared to traditional machine learning techniques. The comparative analysis using Receiver Operating Characteristic (ROC) Curve indicated that the proposed approach obtained superior results when compared to the ones found in literature. An extension of this work was produced, and it merged the new approach with Ranking methods (LIU; LI, 2018). The results were similar to the previous method.

LSTMs have also been applied for Protein Fold Recognition (PFR) (TSUBAKI *et al.*, 2017). The process was divided into two stages: application of the feature extractors of dataset samples (MIKOLOV *et al.*, 2013; PENNINGTON *et al.*, 2014), and a supervised approach using the LSTM network. In addition, the LSTM method had a superior performance when compared to those based on the Support Vector Machine (GHANTY; PAL, 2009; DING; DUBCHAK, 2001; YANG *et al.*, 2011; SHARMA *et al.*, 2013), which were the most advanced ones nowadays.

Recently, Villegas-Morcillo *et al.* (2020) proposed an Convolutional layers with GRU layers to the PFR. This approach indicated superior performance compared to literature methods, such as DeepFRpro, (ZHU *et al.*, 2017), CEthreader (ZHENG *et al.*, 2019), DeepSVM-fold (LIU *et al.*, 2019), among others.

3.1.2 Analysis of the related works

As presented in the beginning of this Section, the protein folding problem using DL methods has not been explored by other researches. On the other hand, we observed that DL has proved to be successful in many problems related to the PFP. Table 3 provides a summary of proteomics problems, protein representations, work references, and methods used in these problems found in the literature review. Besides, there is a lack of datasets for the protein folding study. Hence, we also present new datasets for this study.

Among scientific researches related to the DL applied to the proteomics problems, DL techniques proved to be effective. Specifically, for the protein secondary and tertiary structure predictions, RNN has been emphasized when compared to the literature approaches, including other Machine Learning methods and *ab initio* approaches. Among the RNNs, LSTM networks have been highlighted in these proteomics problems. Other RNN methods have been identified, such as GRU and standard RNN.

Additionally, two types of protein structure representations in the tertiary space (CM and torsion angles) were identified. Although CM representation is a simplified approach to represent protein structure, the reconstruction of this three-dimensional representation is an open problem. On the other hand, there is no problem to reconstruct the three-dimensional structure with the angle representation. Then, angle representations based on Spherical Relative Coordinate (SRC) were employed in this work. According to our research, no recent study has been using DL methods for the PFP with this representation.

3.2 THE PROTEIN FOLDING PROBLEM

In this section, we present computational methods applied to the PFP.

Table 3 – Deep Learning methods applied to the Proteomics study.

Problem	Representation	Authors	Methods
Secondary Structure Prediction	Secondary Structure Classification	Zhou and Troyanskaya (2014)	GSN
		Spencer *et al.* (2015)	PSIPRED
		Sønderby and Winther (2015)	BLSTM
		Wang *et al.* (2016)	DCNF
		Li and Yu (2016)	CCRN
		Heffernan *et al.* (2017)	BLSTM
Tertiary Structure Prediction	Contact Map	Lena *et al.* (2012b)	CMAPpro
		Seemayer *et al.* (2014)	CCMpred
		Jones *et al.* (2014)	MetaPSICOV
		Wang *et al.* (2017)	DST network
		Heffernan *et al.* (2017)	BLSTM
		Hanson *et al.* (2018)	BLSTM/CNN
	Torsion and Angle	Song *et al.* (2012)	TANGLE
		Heffernan *et al.* (2015)	SPIDER
		Yang *et al.* (2017)	SPIDER2 (DNN)
		Heffernan *et al.* (2017)	SPIDER3 (BLSTM)
		Li *et al.* (2017)	DRNN/DNN/DRBM/DRRBM
		Fang *et al.* (2018)	DeepRIN
		AlQuraishi (2019)	RGN
Protein Folding Problem	Relative Spherical Coordinates	proposed method	Deep Learning

Source: own authorship

3.2.1 Computational methods

Experimental techniques have been supporting the development of the proteomics area (FERSHT, 2017). Among these techniques there are X-ray Crystallography (LIZAK *et al.*, 2017), Nuclear Magnetic Resonance (ROGAWSKI; MCDERMOTT, 2017), and recently, Microscopy cryoelectronics (ZEYTUNI *et al.*, 2017). However, they are costly and time-consuming processes (MCPHERSON; GAVIRA, 2014; LACAPÈRE *et al.*, 2018). For these reasons, computational methods are justifiable in this area.

Although computational methods are an alternative for studying proteomics problems, they are an NP-complete problem for protein structure predictions and they have been open for over 50 years (DILL; MacCallum, 2012; FINKELSTEIN, 2018).

Several approaches have been proposed to explore this problem. Table 4 presents methods used in the PFP study. According to Morriss-Andrews and Shea (2015), the computational methods used for the protein folding study can be divided into thermodynamics, kinetics, trajectory analysis, and systematic approaches.

Thermodynamic approaches allow simulating physical and chemical processes in different systems based on energy to drive simulations. According to Dill *et al.* (2008), thermodynamic methods commonly used for PFP simulation are MD (KARCZYŃSKA *et al.*, 2017) and MC (FARRIS *et al.*, 2018) approaches. Due to popularization of these methods, different scientific programs have been developed and published. GROMACS and AMBER are examples of state-of-the-art softwares in the literature (SALOMON-FERRER *et al.*, 2013; ABRAHAM *et al.*, 2015). However, these programs commonly use heuristic strategies in the simulations. In other words, they may lose the Spatio-temporal feature of the folding pathway to enhance the sampling of the energy landscape.

The popularization of MD method brings discoveries to the pharmacology study applied to drug design (VIVO *et al.*, 2016; GANESAN *et al.*, 2017; MICHELARAKIS *et al.*, 2018a). These approaches have been instrumental tools to explore the propensity for protein interaction and finding 'hot-spots' for receptor binding (proteins that interact with other substances). Despite these insights, the computational time required by the MD is a problem. Some features contribute to increase the time-consumption of MD simulations. For example: the size of the protein chain, extensive time-step simulations, and detailed models, such as the atomic model (EASTMAN *et al.*, 2017; MERMELSTEIN *et al.*, 2018). According to Lee *et al.* (2017), the MD approach

would take over 1000 years to run 300 micro-seconds using a distributed system for a protein in the atomic model.

As presented in Section 2.6, the MD method has a high computational cost. In the last decade, several computational approaches have been proposed to optimize it (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 MD method decreases time cost, it can be computationally expensive for atomic model simulations for larger polypeptide chains.

The main time-consumption of the MD is the Lennard-Jones (LJ) energy calculation (HOWARD *et al.*, 2019a). Neighbourhood List (NL) has been used to decrease the number of LJ calculations. The NL restricts the number of pairwise interactions to the nearest neighbors in the space of each residue. This approach has achieved high success in many studies (HOWARD *et al.*, 2019a; BAILEY *et al.*, 2017; EASTMAN; PANDE, 2010).

The 3D-AB off-lattice model, a CG representation, can be used to decrease such time-consumption. The 3D-AB off-lattice model is a one-bead model used to represent proteins at a high-level to simulate globular protein behavior (STILLINGER; HEAD-GORDON, 1995; DILL; MacCallum, 2012). This toy model turned out to be a flexible representation, when 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 lower computational cost than atomic representation and other multi-beads models. For instance, in aggregation studies, where a higher computational effort is required, this model enabled realistic simulations of fibrillar aggregates (FRIGORI *et al.*, 2013; FRIGORI, 2014; FRIGORI, 2017). It was also able to represent a similar final structure of the re-scaled biological native structure (HATTORI *et al.*, 2020b). Nowadays, the 3D-AB model has been used in many benchmark studies for the PSP problem (LIN *et al.*, 2018; ZHOU *et al.*, 2018).

The Replica exchange method is also a thermodynamic approach used to overcome the problem of energy barriers. In this method, several simulations/replicas occur in parallel and can change the temperatures of the simulations in each iteration. This technique is commonly applied to MC and MD methods (SWENDSEN; WANG, 1986; SUGITA *et al.*, 2000). Despite a more extensive landscape sampling than canonical approaches, they lose the Spatio-temporal feature of the folding pathway (STILLINGER; HEAD-GORDON, 1995). There are different Replica exchange variations, such as procedures that optimize the number of replicas required in

the simulation (KIM *et al.*, 2012).

The Markov State Model (is a complementary technique of thermodynamic methods. Unlike most, which lose kinetic characteristics in favor of energy landscape sampling, this approach preserves the simulation kinetics (SWOPE *et al.*, 2004). This strategy is modeled as a Markovian transition system set, generating a map of transition probabilities between states.

Free energy guided sampling (ZHOU; CAFLISCH, 2012) is another kinetic approach. Unlike MSM, it uses a free energy approximation to start simulation settings. In this method, small trajectories are executed iteratively in parallel, based on the landscape exploitation and the refinement of simulations settings. Another kinetic method called WExplore (DICKSON; BROOKS, 2014) uses settings that determine the simulation dynamically defined by sampling the hierarchically organized region. There are parallel simulations in the WExplore technique, and the trajectory is oriented to new directions of space configurations.

Transition path sampling (DELLAGO *et al.*, 1998) is an approach that involves the generation of a pathway based on the initial and final states, usually using an optimization function that maximizes the probability of obtaining reaction coordinates. The String method (MARAGLIANO *et al.*, 2006), in turn, performs multiple simulations in parallel and selects the pathway with the lowest energy barrier between two states.

In the systematic methods, the process follows the opposite directions of the previous methods. Native structures are predicted from the behavior of experimental data (MORRISS-ANDREWS; SHEA, 2015). Among the systematic techniques, the Relative Entropy approach uses the probability of structure conformations, and it is considered the structure with the minimal loss of information function (SHELL, 2008). Multi-scale is a variation approach that predicts the protein structure based on the minimization of the difference between the reference data and the predicted structure (IZVEKOV; VOTH, 2005). Another systematic approach is the Iterative Boltzmann Inversion method. This method was designed to reproduce Boltzmann statistics for structural prediction and the other systematic approaches comparing the reference results with the inferred structure (REITH *et al.*, 2003).

In order to reduce the computational cost of the protein folding simulation and present a new approach for PFP based on computational intelligence methods, Benítez (2015) used Cellular Automata (using CM representation. To infer the folding rules for the AC, it used the Gene Expression Programming method (BENÍTEZ *et al.*, 2015). The data used to infer the rules were based on the folding pathway simulation using the MD method with the 3D-AB off-lattice

model (BENÍTEZ; LOPES, 2013). Parallel Ecology-Inspired Optimization, which is a set of approaches based on evolutionary computation, is used to reconstruct the predicted structures from CM representation (PARPINELLI; LOPES, 2015). Regarding the reconstruction problem, Hattori *et al.* (2017c) proposed using Relative Spherical Coordinates (RSC) to avoid this step, which is simpler to convert to the Cartesian Coordinates, and its representation maintained the unitary distance constrain between each amino acid as required by the 3D-AB off-lattice model.

3.2.2 Analysis of the related works

As shown in this section, the PFP is an NP-complete problem. Given the computational cost required for folding simulation, several CG representations have been proposed in the literature. Among protein model representations, the 3D-AB off-lattice is still a common approach used in researches. The main reasons identified to use 3D-AB off-lattice include the higher degree of flexibility (continuous angles) than discrete models, as well as the lower computational cost compared to multi-beads and atomic models. It can also represent several behaviors of protein folding at the mesoscale. Given these features, the CG model was the representation chosen for this study.

Computational methods are essential for the study of PFP because of the difficulties of the experimental methods. Among these techniques, physical, chemical, and statistical approaches were identified. The Molecular Dynamics (MD) procedure is one of the most commonly used in Computational Biology researches. Despite its substantial computational burden, Molecular Dynamics (MD) is the leading approach for the PFP studies. Also, this technique in the canonical ensemble conserves the spatial-temporal feature of the protein folding pathway. Thus, we used this approach in our experiments. As it is computationally costly, we proposed the parallel MD using GPU architecture. Given the improvement of the NL mechanism to increase the MD method, we also proposed the parallel NL in the MD method with a canonical ensemble.

As shown in Table 4, the computational intelligence methods are not sufficiently explored in the literature for the PFP (BENÍTEZ, 2015). In this scenario, we also intended to explore the protein folding problem using computational intelligence methods, explicitly using the DL methods (see Section 3.1). Moreover, we presented the RSC representation of the protein structures instead of CM maps (BENÍTEZ, 2015), avoiding the reconstruction issue as presented in Hattori *et al.* (2017c).

It is worth noting that several works in the literature have been using the wrong nomen-

clature of the PFP. Many of these studies aim to predict the native conformation, also called the Protein Structure Prediction (PSP) problem. The researches presented by Li *et al.* (2015), Kaushik and Sahi (2017), and Li *et al.* (2017b) are examples of this erroneous nomenclature. This mistake is a persistent issue, and it has already been addressed by Lopes (2008). This problem actually makes it harder to identify studies that were related to this research.

Table 4 – Computational methods applied to the protein folding problem.
Source: Based on Morriss-Andrews and Shea (2015).

Computational Method	Method	**Ref.**
Thermodynamics	Metadynamics	(LAIO; PARRINELLO, 2002)
	Umbrella sampling	(TORRIE; VALLEAU, 1977)
	Molecular Dynamics	(McCAMMON *et al.*, 1977)
	Parallel tempering (temperature REMD)	(SUGITA; OKAMOTO, 1999)
	Replica exchange Molecular Dynamics (REMD)	(SUGITA *et al.*, 2000)
	Replica exchange Statistical Temperature molecular dynamics	(KIM *et al.*, 2012)
	Monte Carlo	(LI; SCHERAGA, 1987)
	Replica exchange Monte Carlo	(SWENDSEN; WANG, 1986)
Kinetic	Markov State Model	(SWOPE *et al.*, 2004)
	Free Energy Guided Sampling	(ZHOU; CAFLISCH, 2012)
	WExplore	(DICKSON; BROOKS, 2014)
	Transition Path Sampling	(DELLAGO *et al.*, 1998)
	String method	(MARAGLIANO *et al.*, 2006)
Systematic Methods	Relative Entropy Coarse graining	(SHELL, 2008)
	Multiscale Coarse Graining	(IZVEKOV; VOTH, 2005)
	Iterative Boltzmann Inversion	(REITH *et al.*, 2003)
Computational intelligence	Cellular Automata Genetic Programming	(BENÍTEZ *et al.*, 2015)
	Proposed method (Deep Learning)	(HATTORI *et al.*, 2018)

Source: own authorship

4 MATERIAL AND METHOD

This thesis presents a novel computational approach based on Deep Learning (DL) methods applied to the Protein Folding Problem study, concentrated on one-step-ahead prediction analysis.

The overview of the proposed approach is presented in Figure 18. It can be divided into two steps: generate in silico datasets and process it using DL methods.

In the first step, to generate in silico datasets, a package was developed called PathMolD-AB (HATTORI *et al.*, 2020b). This package provides an integration with the Protein Data Bank to perform the data acquisition of the amino acid sequence and convert it to the AB sequence. This package supplies variants of the MD in the Canonical Ensemble. PathMolD-AB offers a visualization tool to analyze the protein folding simulation. A comparative method is presented to analyze the similarity between the rescaled biological structure with the predicted structure (generated by the MD method).

In the second step, to train DL methods, a pre-processing step in datasets was performed. This pre-processing convert the Cartesian coordinates representation to the Relative Spherical Coordinate (RSC). These datasets are divided into three subsets: training, validation and test. Subsequently, it is performed train and test of DL methods. Lastly, the trained model is validated, and the differences and similarities between the target results (MD) and predicted structures (DL) are analyzed.

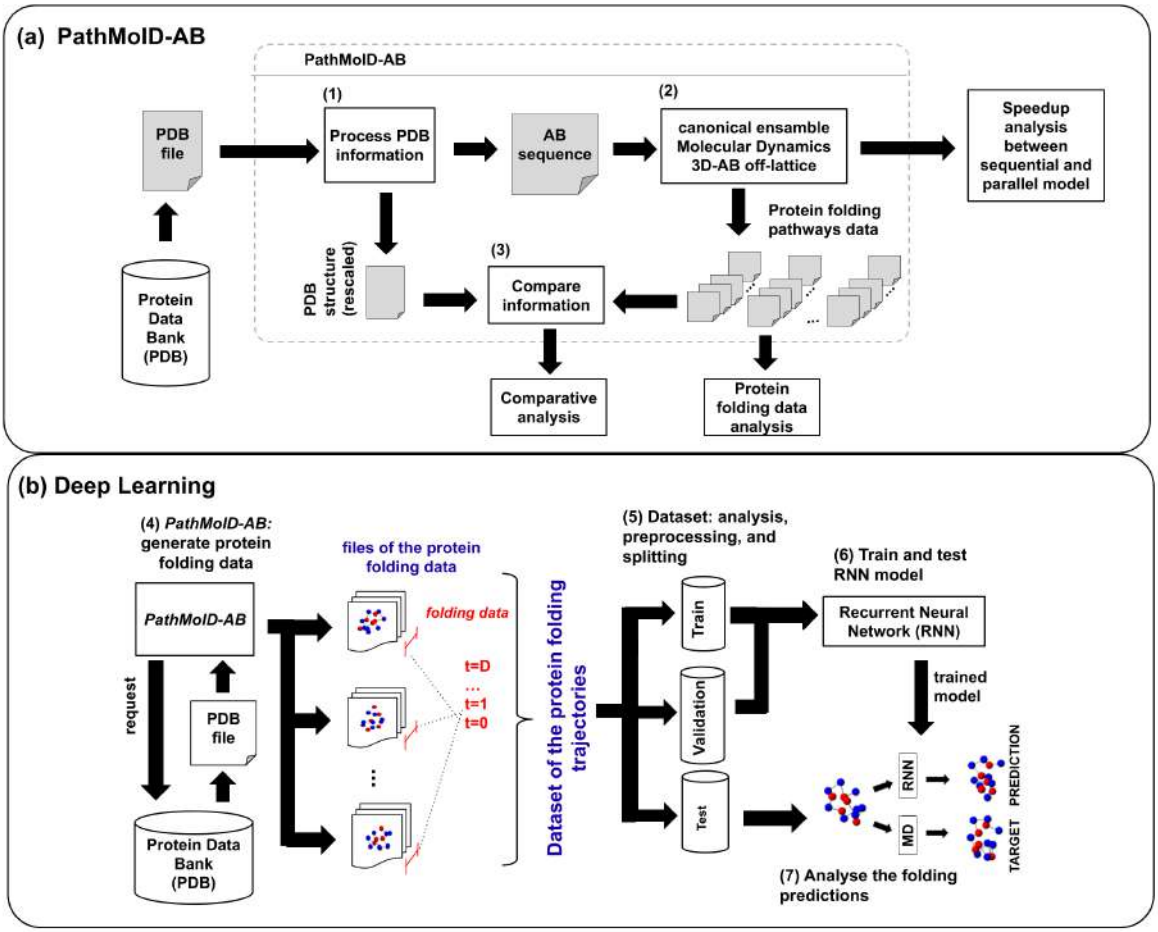
The following sections deepen these two steps.

4.1 GENERATION OF IN SILICO DATASETS USING PATHMOLD-AB

Figure 18(a) 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 analyzes. In step 1, it is generated the input file based on raw data acquired from the Protein Data Bank (PDB)[1]. In this step, the Cartesian coordinates $C\alpha$ of all amino acids are extracted, and the distance between $C\alpha - C\alpha$ is rescaled to one, aiming to compare with 3D-AB *off-lattice* 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

[1] http://pdb.org/ (accessed 24 November 2020)

Figure 18 – Overview of the proposed method for the protein folding problem using Deep Learning.



Source: own authorship

3D-AB *off-lattice*. In step 3, the simulation results are compared with the rescaled biological structure of the protein (performed in step 1), aiming to compare the predicted structures with the corresponding "biological" structure. In the following Sections, these steps will be detailed.

4.1.1 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. The program downloads the PDB file and extracts the amino acids' sequence starting from a PDB ID. Next, this sequence is converted into the AB model

[2] http://biopython.org/ (accessed 24 November 2020)

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 5) 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 C.2).

Algorithm 2 – Protein Sequence conversion procedure.

Input PDB ID
Download PDB File
Read PDB File
for $i = 0 : N$ **do**
  Extract Amino Acid $AA_i \in$ SEQRES
  Add AA_i to $Sequence[i]$
end for
Read AB Classification Table
for $i = 0 : N$ **do**
  if $Sequence[i]$ == 'A' **then**
    $AB\_Sequence[i] \leftarrow$ 'A'
  else
    $AB\_Sequence[i] \leftarrow$ 'B'
  end if
end for
Save $AB\_Sequence$

Source: Hattori *et al.* (2020a)

Table 5 – Hydrophobicity scale.

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

Source: Alberts *et al.* (2002)

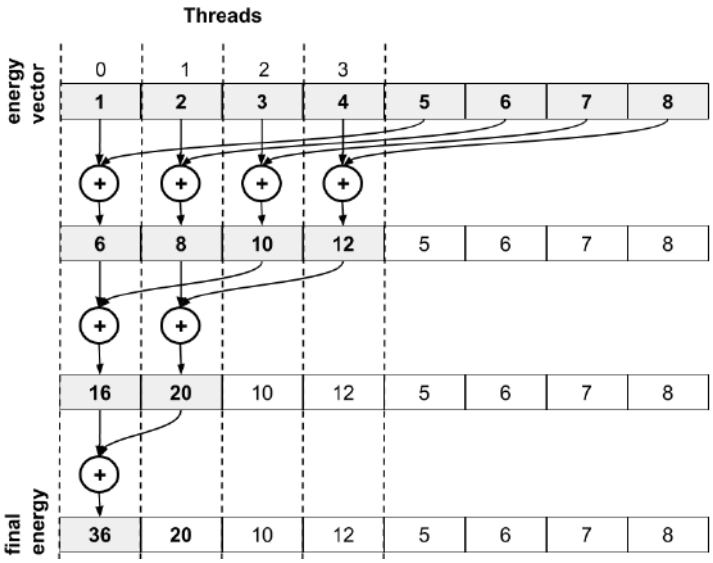
In the rescaling process, the Cartesian coordinates of the protein structure are extracted from the PDB file. Furthermore, from them, the coordinates of the $C\alpha$ of each amino acid (PIERRI *et al.*, 2008). The distances between each consecutive C_α is rescaled, dividing by 3.8 Å, to obtain the normalized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

4.1.2 Parallel Molecular Dynamics

The MD implementation of the PathMolD-AB software package was based on previous works described in Benítez (2015), Stillinger and Head-Gordon (1995), Benítez and Lopes (2012). This approach 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, while slave processes running on GPU cores execute the main computations in parallel.

The algorithm consists of a sequence of steps, starting with a structure randomly positioned in the space. The initial procedures are naturally serial or require low computational efforts. Therefore, they run on a CPU. Next, the central part of the MD algorithm is the computation of the torsion, bond, and Lennard-Jones energies, as described in Section 2.4.3. The computation of each energy function is parallelized separately. For each energy term, the computation to assigned to a thread, and the value is stored in an array position. After computing these energies, the partial energies are summed by parallel reduction to sequential addressing, as shown in Figure 19. The reduction algorithm is accomplished by summing in pairs, and these calculations are performed in parallel. The sums of each pair are saved in the memory position of the first partial value. This process takes place iteratively until all values are summed in a single array position.

Figure 19 – Parallel reduction to sequential addressing.



Source: own authorship

In the sequence, velocities and accelerations of all the residues are computed. These computations are independent of each other and are performed in parallel. Then, due to the physical forces acting on the residues, they are pushed to another position in the 3D space. As this step is highly parallelizable, it was also accomplished in parallel with a GPU. Next, The weak coupling adjustment into a thermal bath method provides the temperature system proposed by (BERENDSEN *et al.*, 1984). Finally, the geometric constraints are applied to adjust the coordinates and velocities (see Algorithm 1).

In order to evaluate 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). It is worth noting that the observation of the temporal changes of RgH and RgP may indicate the formation of the hydrophobic core, typical of many proteins. Equation 36 presents how the radius of gyration is computed:

$$RgAll = \sqrt{\frac{\sum_{i=0}^{N-1}[(x_i - \bar{X})^2 + (y_i - \bar{Y})^2 + (z_i - \bar{Z})^2]}{N}}, \tag{36}$$

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

At each pre-defined number of iterations ($step$), the program saves the protein information state as a report file. At each $step\_size$ are saved: the structure, the energy, and the radii of gyration of the protein. All the procedure is repeated until a stop criterion is satisfied. For example, a pre-defined number of iterations (t_{max}) or, eventually, when the E_p stabilizes according to a specific criterion. Algorithm 3 shows the main execution steps of PathMolD-AB.

4.1.3 Generation of Datasets for Studying the Protein Folding Dynamics

Four datasets of protein folding trajectories were produced as case studies using the PathMolD-AB software. Four proteins were simulated, one artificially created and three real-world proteins with a growing number of amino acids, as shown in Table 6, and detailed as follows:

- 13FIBO: it contains 13 amino acids, and it was artificially created by Stillinger and Head-Gordon (1995), by distributing the hydrophobic amino acids according to the Fibonacci

Algorithm 3 – Main execution steps of PathMolD-AB. The shaded lines are executed in parallel in GPU, while the others 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 $(t \bmod step\_size) == 0$ **then**
    Compute radii of gyration
    Save the state of the protein in a report file (structure, potential energy, and radii of gyration)
  end if
  $t \leftarrow t + 1$
end for
Store data

Source: own authorship

sequence.

- 2GB1 [3]: it contains 56 amino acids, and 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]: it contains 99 amino acids, and 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; NAKAKI, 2007);

- 5NAZ [5]: it contains 229 amino acids, and this is a globular structural protein of collagen, related to the Goodpasture's and Alport's syndromes (CASINO *et al.*, 2018).

For each protein, a dataset was generated with 1,000 (for 13FIBO, 2GB1, and 1PLC) or 500 (for 5NAZ) different pathways. The size of the 5NAZ protein sequence implied the run of fewer simulations. As earlier mentioned, all simulations start with structures randomly initialized in the 3D space to achieve higher diversity of pathways, each one leading to the native conformation of the protein.

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 to guarantee

[3] http://10.2210/pdb2GB1/pdb (accessed 24 November 2020)
[4] http://10.2210/pdb1PLC/pdb (accessed 24 November 2020)
[5] https://www.rcsb.org/structure/5NAZ (accessed 24 November 2020)

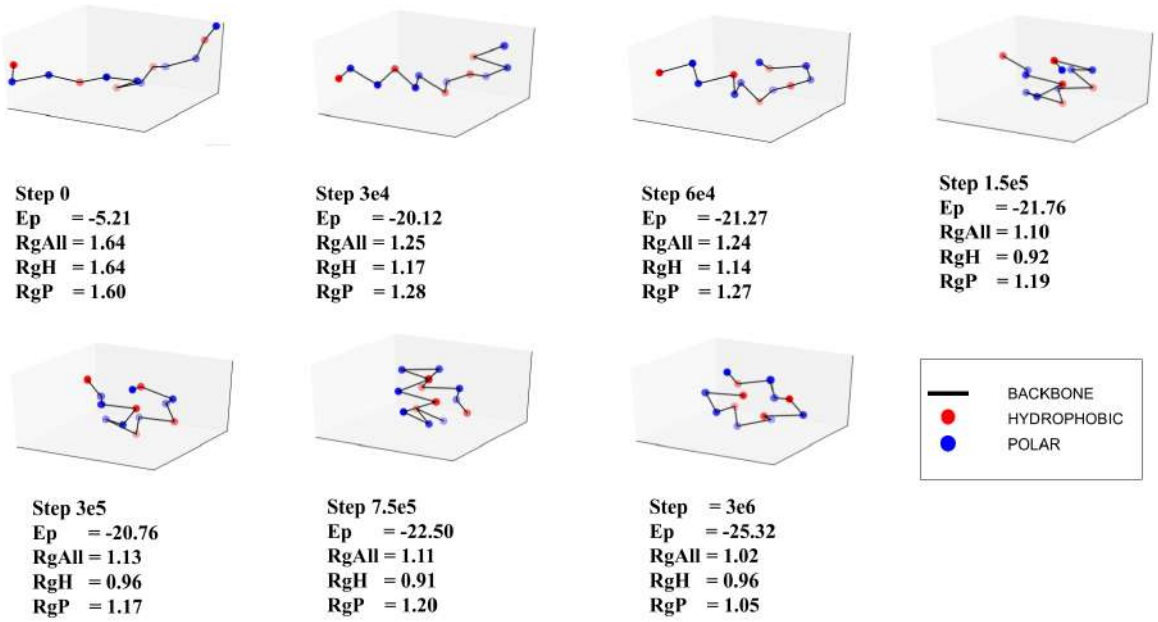
Table 6 – Information about the protein sequences used to generate the datasets.

ID	# amino acids	AB sequence
13FIBO	13	$AB^2(AB^2AB)^2$
2GB1	56	$AB^3A^3BAB^2ABAB^5(A^2B)^2AB^2A^2$ $(B^3A)^2(AB)^3(B^3A)^2BAB^2$
1PLC	99	$(ABA^5BB)^2(AB)^2A^2B^2A^3B^3A^4B^2$ $A^3B^4A^2BA(AB)^2(BA)^2B^4A(AB)^32$ $(BA)^AA(B^2BA)^2(BA)^2B^2A^6(BA)^2B$
5NAZ	229	$(BA^2)^2A^2B^8(A^2B)^2(AB)^3(BA^2)$ $(BA)^2B^3(AB)^2(BA^2)^3(B^2A)^2A^5BA$ $B(BA^2)^2B^7A^2B(BA)^2A^3(BA)^2$ $(AB)^2(BA)^2A^2B^2A^4BA^8B^4A(BA^2)^3$ $B^2A^4B^2A^3B^3A^3(BA)^3(A^2B)^3$ $BAB^2A^4(BA)^4B^3AB^4A^3(AB)^3BA^2$ $B^2AB^3(BA)^2(AB)^2(B^2A)^2BA^2B^3$

Source: own authorship

reliable stabilization of the native structure. Consequently, for standardizing the number of Spatio-temporal states per pathway in each dataset, the $step\_size$ for 13FIBO, 2GB1, and 1PLC were 3000 and 8000 for the 5NAZ. For each pathway, 1,000 folding states were recorded. Figure 20 illustrates snapshots of protein folding states.

Figure 20 – Sample of a pathway for the protein 13FIBO.



Source: own authorship

4.1.4 Comparison with the biological structure from the PDB

This section focus on the structure comparison of the crystallized proteins identified in the PDB with the corresponding structure predicted by our approach (see Figure 18(a)). This comparison is accomplished indirectly by computing the radii of gyration of both structures and, in a direct way, employing the Kasbch-RMSD measure described below.

Following the prepossessing, the step number two comprises some simulations that originate the protein pathway dataset using MD (see Sections 2.4.3 and 4.1.2), organized after simulations in a way to enable the comparison with biological structures. As presented in Section 4.1.3, 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 residues in the PDB file, the scaling process was unfeasible for this protein. Thus, the analyzes were performed only for the first 2GB1 and 1PLC. This section focus on the structure comparison of the crystallized proteins identified in the PDB with the corresponding structure predicted by our approach (see Figure 18(a)). This comparison is accomplished indirectly by computing the radii of gyration of both structures and, in a direct way, employing the Kasbch-RMSD measure described below.

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; DIJKSTERHUIS, 2004). Kabsch (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, followed by a rotation applied to minimize the difference between these two matrices, based on the Root Mean Square Deviation (RMSD) (KRAVRAKI, 2007), as shown in Equation 37.

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(P_{ix} - Q_{ix})^2 + (P_{iy} - Q_{iy})^2 + (P_{iz} - Q_{iz})^2)}{N}}, \tag{37}$$

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

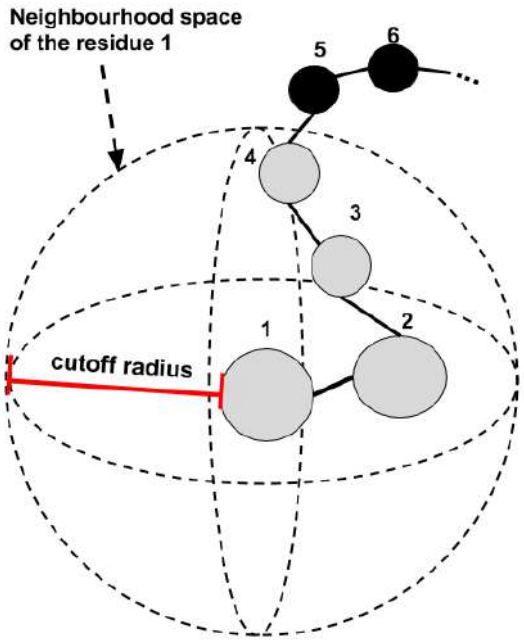
4.1.5 Parallel Molecular Dynamics with Neighbourhood List

The parallel Neighbourhood List (NL) technique is proposed in this work to decrease the time processing of the purely parallel MD simulation.

The NL approach works using only the nearest residues to calculate the potential energy

of each residue in the protein sequence. The neighborhood to each residue is defined by a radius cut-off, as represented in Figure 21. Then, if a residue is inside of this radius cut-off, it is a neighbor. Thus, pairwise interaction can be neglected in the simulation favoring the computation speedup, which needs broadening efforts as presented in the Section 3.2.

Figure 21 – The neighborhood space representation of the residue one (dashed line). The gray spheres represent the residues inside the neighborhood space of the residue 1 (residues 2, 3, and 4). Also, it is presented residues outside of this space by black spheres (residues 5 and 6).



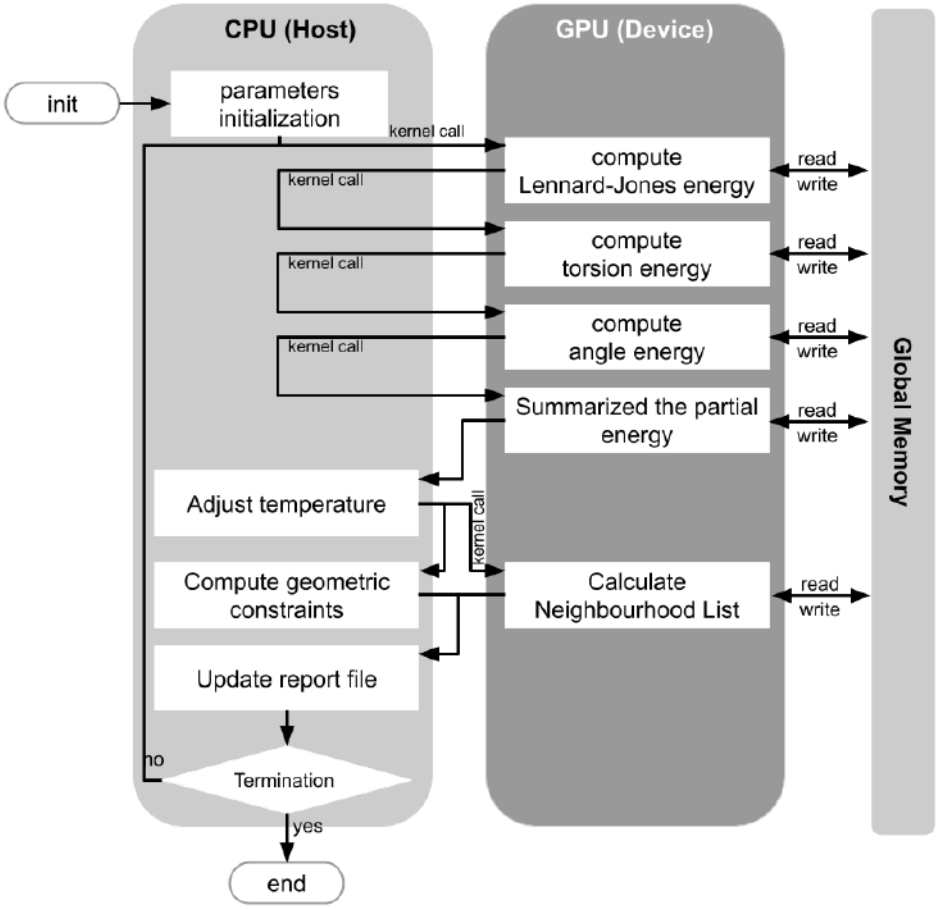
Source: own authorship

The NL mechanism presented is divided into three steps. The management of the update list using threads in CPU and GPU, the calculation of the NL, and NL application in the energy function calculation. The overview of this proposed approach is presented in Figure 22.

In the first step of the NL algorithm, a CPU-thread is stated to manage the NL update procedure. This procedure initiates N other threads in GPU. Each GPU-thread started is responsible for calculating the neatest residues of a specific residue. After these calculations, the main thread in CPU organized all these results and returned the NL to the process. This task is computational costly ($O(N^2)$)(HOWARD *et al.*, 2019a), then, this task is performed at each $step\_time$ (see Section 4.1.2). To take advantage of the time spent by the NL update, we execute the Shake algorithm (HATTORI *et al.*, 2020b) in parallel using another CPU-thread.

In the second step, it is presented the NL update procedure, see Algorithm 4. Each thread calculates the Euclidean distance of the i-th residue to the other $N - id$ residues, where i represents the current residue analyzed, and $N - id$ represents the other residues of the protein

Figure 22 – Overview of the proposed approach.



Source: own authorship

Algorithm 4 – Update the neighbour list in GPU.

UpdateNeighbourList
id ← ThreadID + BlockID x BlockSize
$i \leftarrow id + 1$
while $i < N$ **do**
  distance ← Euclidian distance between *id* and *i*
  if *distance* < *cut-off radius* **then**
    Neighbourlist$[id][i] \leftarrow 1$
  else
    Neighbourlist$[id][i] \leftarrow 0$
  end if
  $i \leftarrow i + 1$
end while
end procedure

Source: own authorship

chain. In this algorithm, if the pairwise Euclidean distance is lower than the cut-off radius, they are considered neighbors. Then, the position of the list correspondent to this pairwise receives the value equal to one, otherwise, receives the value equal to zero.

Finally, when LJ energy function is calculated, it is considered the NL calculated in

Algorithm 5 – Lennard-Jones Potential Energy Calculation with Neighbour List.

```
LJEnergy
id ← ThreadID + BlockID x BlockSize
for i=(id+2) : i < N do
   if Neighbourlist[id][i]=1 then
      Calculate the energy between the particle[id]
      Update particle[id].acceleration
      Update particle[i].acceleration
   end if
end for
LJEnergyVector[id] ← calculated energy
end procedure
```

Source: own authorship

Table 7 – Information about the protein sequence added.

ID	# amino acids	AB sequence
2QHT	210	$ABA_3B_2A_5BABA_3B_4AB_3A_4B_2A_2BA_4B(BA)_2A$ $BA_2B_2(AB)_2BA_4B_2AB_4A(AB)_2A_2B_2A(AB)_2B_2$ $A_3BA_3BA_6B_2(ABBA)_2B_2A_3BA_4BA(AB)_3A_2B$ $A_6B_3A_8B_2A(AB)_2A_4BAB_3AB_2ABA_8(BA)_2$ $(ABB)_2A_3B_3A_3B(BA)_3A_5BA_4BA_2$

Source: own authorship

the previous step. In this procedure, if this pairwise of residues are neighbors, the thread saves the value in the output vector (see Algorithm 5). Otherwise, the thread verifies the subsequent pairwise interaction. The summarization of these partial energies values are summarized using the parallel reduction to sequential addressing method (see Figure 19).

To analyze the speedup of the MD with the NL mechanism, we used 23 artificial protein sequences ranged from 13 to 28657 amino acids. The artificial sequences were generated using the Fibonacci sequence method to distribute hydrophobic and polar residues in the artificial sequence, as presented by Hsu *et al.* (2003).

Moreover, in the case study, we used three biological protein sequences (2GB1, 1PLC, and 2QHT) and an artificial sequence (13FIBO), as shown in Table 6. The method used to convert the amino acid sequence to the AB sequence was presented in Section 4.1.3. There is one protein sequence changed to test this mechanism when compared in the purely parallel MD algorithm experiment. It was changed the 5NAZ to the 2QHT (see Table 7), given miss values in the amino acid sequence data in the PDB. The 2QHT is a protein that has 210 amino acids, while 5NAZ has 229 amino acids.

4.2 DEEP LEARNING

In this section, it is described the DL method applied to the PFP. Figure 18(b) presents a simplified overview of the proposed approach.

4.2.1 Protein Folding Dataset

The protein folding dataset was generated using the PathMolD-AB[6] software package, which was previously presented in Section 4.1. Then, the parallel implementation of MD in the canonical ensemble and the minimalist model for representing proteins known as 3D-AB *off-lattice* were used in this step to simulate the protein folding datasets.

Three protein sequences were used to generate the folding pathways datasets, as shown in Table 6. The first sequence (13FIBO), created by Stillinger and Head-Gordon (1995), is a synthetic protein with 13 amino acids. The other two proteins are real-world biological sequences extracted from Protein Data Bank: 2GB1 (GRONENBORN *et al.*, 1991) and 1PLC (GUSS *et al.*, 1992), with 56 and 99 amino acids, respectively. The 20 different proteinogenic natural amino acids present in biological sequences were translated into AB (A for hydrophobic, and B for polar) sequence. This translation was accomplished based on the Alberts (ALBERTS *et al.*, 2002) scale, as presented in the Section 4.1.1.

Each dataset is constituted of a series of protein folding pathways. Besides, each folding pathway data is comprised of a sequence of folding states. Where a folding state is a representation of a protein conformation in a specific iteration of the simulation (see Figure 7). The dataset comprises 1,000 different folding pathways per protein sequence, each one starting from a different initial folding state. Each pathway is composed of 1,000 folding states equally spaced in time. Therefore, the final number of folding states is 10^6 for each protein dataset.

4.2.1.1 Evaluation Measures

An analysis of the structural similarity between subsequent folding states of the pathways is proposed in this work. For this purpose, we use the Root-Mean-Square-Deviation (RMSD) to evaluate the structural difference between two protein structures, as shown in Equation 38.

[6] https://github.com/bioinfolabic/protein_folding_datasets (accessed 24 November 2020)

$$RMSD = \sqrt{\frac{\sum_{i=1}^{S-1} |P_{1,i} - P_{2,i}|}{S}}, \tag{38}$$

where S represents the number of amino acids, and P_{1i} and P_{2i} are the Cartesian coordinates of the protein structures P_1 and P_2 at time stamp i, respectively.

Since the RMSD is a rotation-dependent measure, an optimized RMSD is done using the Kabsch algorithm (KABSCH, 1978) to obtain the smallest RMSD.

4.2.1.2 Pre-processing the Protein Folding Dataset

Each sample of the dataset is comprised by *inputs* and a *target*. *Input* data consist in τ subsequent folding states, for example, if the initial state is equal to 1 and $\tau = 4$, then $\overrightarrow{inputs} = \{\chi_1, \chi_2, \chi_3, \chi_4\}$. Therefore, the *target* data consists in the next folding state of the last input data ($target = \chi_5$).

We organized datasets as follows: separating τ states of folding subsequent of the pathways data as input data, and the next folding states as target data. Using a hold-out procedure, we split these samples into the train, validation, and test subsets (70% for the train, 20% for the validation, and 10% for the test) based on our previous work (HATTORI *et al.*, 2018).

DL methods are highly based on the data and the encoded of these data. In this work, it was proposed the Relative Spherical Coordinate (RSC) encoding to represent the states of the protein structure in the input and output data (HATTORI *et al.*, 2018). This scheme dealing with geometrical constraints of the fixed unit-length bonds between amino acids (r), when using the 3D-AB *off-lattice* model.

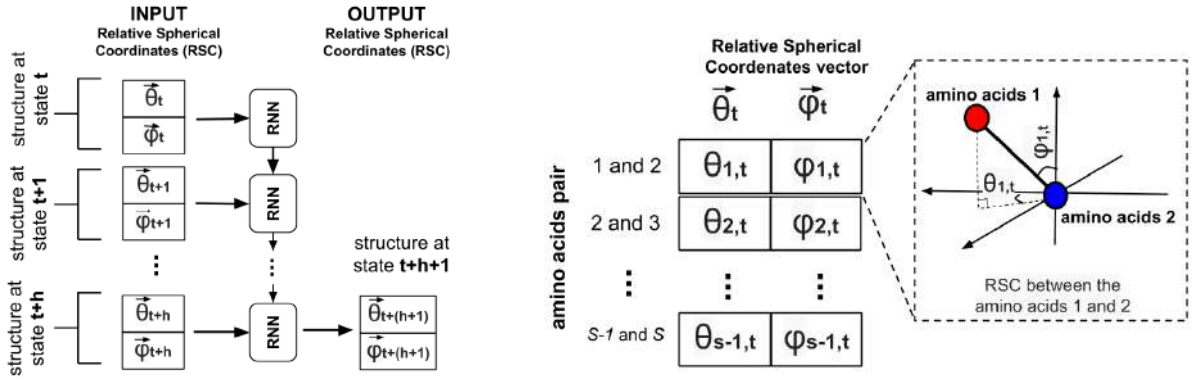
Due to the PathMolD-AB generating and storing protein structures data in the Cartesian coordinate encoding (x, y, z), it was necessary to perform a pre-processing step to convert into the RSC encoding (θ, φ). This procedure is presented in the Algorithm 6. Considering the folding of a protein with S amino acids, this conversion is performed in subsequent amino acid pair, such as, between ith and $(i+1)$th, $(i+1)$th and $(i+2)$th, until the $(S-1)$th and Sth pair, as presented in Figure 23(b). Thus, an RSC encoding has $2S-2$ variables, and r is equal to one as the unit length bonds between amino acids in the 3D-AB off-lattice model. Finally, RSC vectors are normalized of the range $[0:\pi]$ and $[-\pi:\pi]$ to the range $[0:1]$.

Algorithm 6 – Conversion procedure of the Cartesian coordinate to the RSC, as proposed in (HATTORI *et al.*, 2018).

Start
Let S be the protein size (number of amino acids)
Let p be the input Cartesian coordinates ($\overrightarrow{x}_i$, $\overrightarrow{y}_i$, $\overrightarrow{z}_i$)
Let $\overrightarrow{RSC}$ be the relative Spherical coordinates of the output
Let r be the unit length bond between i and $(i+1)$ amino acids
for $i = 1 \rightarrow S - 1$ **do**
  for $j = 0 \rightarrow S$ **do**
    $x_j = x_j - x_{i-1}$
    $y_j = y_j - y_{i-1}$
    $z_j = z_j - z_{i-1}$
  end for
  $RSC.r_{i-1} = sqrt(x_j^2 + y_j^2 + z_j^2)$
  $RSC.\theta_{i-1} = acos(z_j / RSC.r_{i-1})$
  $a.\varphi_{i-1} = atan2(y_j / x_j)$
  $normalize(RSC)$
end for
return a
End

Source: Hattori *et al.* (2018)

Figure 23 – (a) **LSTM for the protein folding prediction based on the many-to-one model.** (b) **Sample of the relative spherical coordinate vector of the state** χ_0 **from a protein with** s **amino acids.**



(a) Many-to-one model. **(b) Relative Spherical Coordenates.**

Source: own authorship

4.2.2 RNN Many-to-one Encoding

As presented in the previous section, the data is encoded using RSC, the input data is composted by a sequence of folding states, and the target data is composed by the next folding state. Then, we modelled the RNN with the many-to-one model to receive the input data and to compare the target data.

Considering the folding of a protein with S amino acids, we represent the set of relative spherical coordinates of the amino acids as a one-dimensional feature vector, as shown in Figure 23(a) (**step 3**). The first amino acid of the sequence is located at the origin. Thus, a feature vector has $(2S - 2)$ variables, these positions kth and $k + 1$th represent the spherical coordinates θ_k and

φ_k of the second to S amino acid of the sequence. The input and output vectors are normalized in the range of $[0:1]$.

4.2.3 RNN Setup and Architecture

In this study, the Gradient Optimizer RMSProp (TIELEMAN; HINTON, 2012) is used to optimize the gradient descent of the RNNs network. The RMSProp was selected among others based on a previous analysis of optimizers (HATTORI *et al.*, 2017b).

The RNN methods used in this work are sRNN, GRU, or LSTM, which are commonly explored in the literature (see Section 3.1). In the benchmark analysis using these RNN, 800 neurons were used in the networks, based on our previous work (HATTORI *et al.*, 2018). The output layer added a fully-connected layer with the number of RSC used to represent the protein structure (see Figure 23(a)), such as 24 neurons for 13FIBO protein, 110 for 2GB2, and 196 for the 1PLC. The activation function of the output layer is a sigmoid function to obtain output values in the range $[0,1]$.

As proposed by Hattori *et al.* (2018), the cost function applied in this study is the Mean Absolute Error (MAE), shown in Equation 39, which is used to evaluate the prediction of RSC concerning target data.

$$MAE = \frac{\sum_{i=1}^{S-1} |target - output|}{S-1}, \tag{39}$$

this measure is the absolute difference between the predicted ($output$) and the $target$. Here, both $target$ and $output$ are in the range $[0:1]$ and S is the sequence length.

In this work, it was also used the trained model to compare the predict results of the test subset with the target data generated by the MD approach in terms of radii of gyration (see Section 4.1.2) and energies (see Section 2.4.3). To perform this analysis the output vector is returned to the real range, and the RSC encoding is represented in the Cartesian coordinates. Then, we calculated the radii of gyration and energies of the output and the target structures.

5 RESULTS AND ANALYSIS

As presented in Chapter 4, we propose a framework to generate a dataset of protein folding pathways to train DL methods (HATTORI *et al.*, 2020a; HATTORI *et al.*, 2020b). In our preliminary study, we observe indications that this approach is viable and could improve (HATTORI *et al.*, 2018). In this regard, we consolidate and enhance this methodology. In the next sections, we will present these results stating from the results of the generation of the in silico dataset, experiments to improve the acceleration of the dataset generation, and the DL methods benchmark with a new model.

5.1 GENERATION OF THE PROTEIN FOLDING DATASET

Experiments were run in a workstation running Ubuntu 18.04 LTS operating system, composed of an Intel i7-8700 processor at 3.2GHz, 32 GBytes RAM, and an Nvidia Titan-Xp GPU (12 GBytes RAM DDR5 and 3,840 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.

5.1.1 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 the pure sequential approach, previously introduced by (BENÍTEZ; LOPES, 2012).

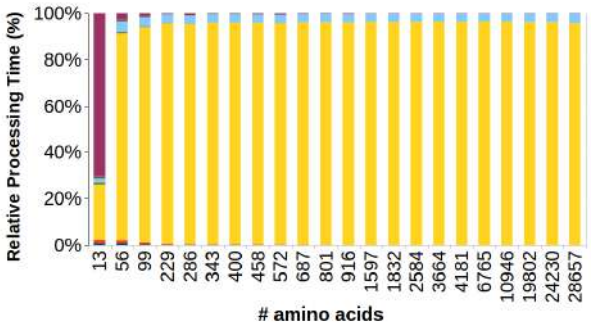
The sequences used to evaluate the performance of the proposed parallel MD were the four proteins shown in Section 4.1.2. Other synthetic sequences, ranging from 286 to 28,657 amino acids, were also used specifically to assess the scalability of the parallel approach.

The experiments performed were based on 3,000 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 is divided by the corresponding processing time of the parallel approach. Figures 24(a) and 24(b) show 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. The most

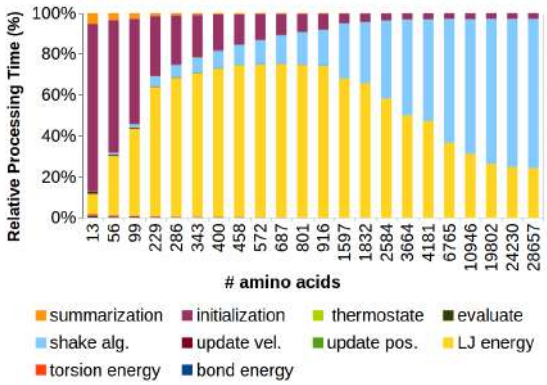
time-consuming part is the computation of the LJ function, considering the sequential approach. Only for the smallest protein (with 13 amino acids), the processing time of the initialization function exceeded the other functions. On the other hand, for the parallel approach, the processing time of the 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. 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.

Figure 24 – Processing time of the PathMolD-AB functions, for both, sequential and parallel approaches.



(a) Sequential approach



(b) Parallel approach

Source: own authorship

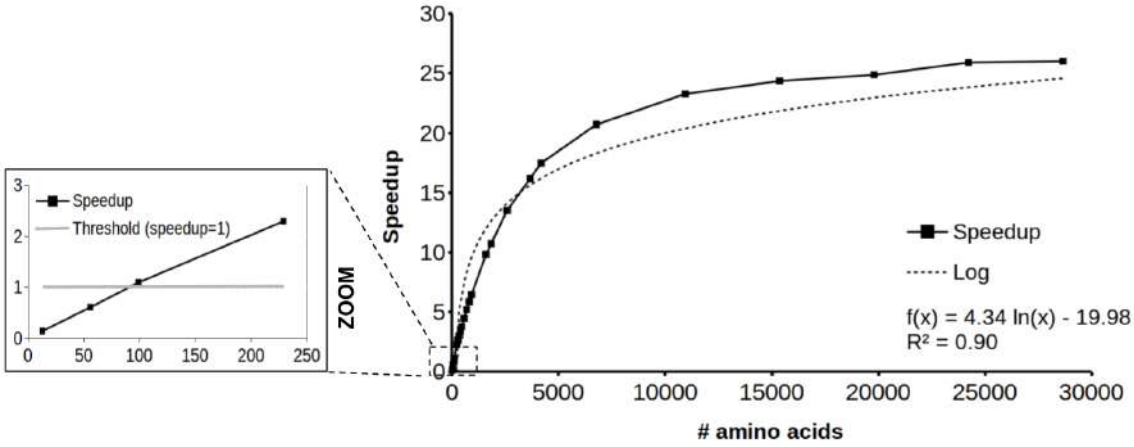
The speedup of the parallel model relative to the sequential model was evaluated using synthetic sequences of different growing sizes (see Figure 25). Surprisingly, a speedup lower than one (the sequential approach was faster than the parallel version) was observed for sequences smaller than 99 amino acids, such as 13FIBO and 2GB1. This behavior happened due to the time required for the communication between GPU-CPU and, more specifically, by the initialization function. 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 suggests that the parallel approach has high scalability for large sequences than the sequential approach. Regardless of the sequential or parallel approach, the processing time tends to follow a logarithmic curve, as shown in Figure 24(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.

Figure 26 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. 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 also be observed for large sequences. Overall, the parallelization of these two functions also helped to increase the speedup value of the approach.

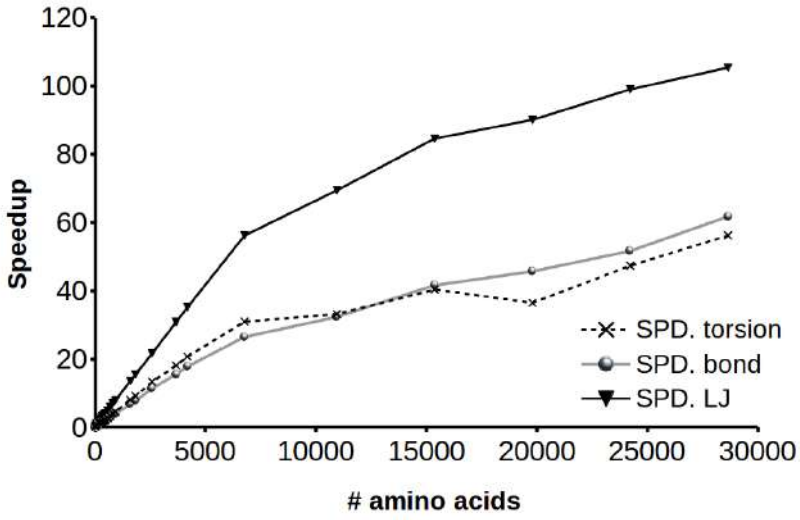
Despite the speedup decay of the parallel approach for the large protein sequences, most of the real biological proteins are quite below that upper bound (BROCCHIERI; KARLIN, 2005; TIESSEN *et al.*, 2012). The statistical information extracted from PDB, as shown in Figure 27, corroborates that this improvement covers more than 92 % of proteins currently deposited in PDB.

Figure 25 – Overall speedup for the simulation of a single pathway, considering the sequential and parallel approaches.



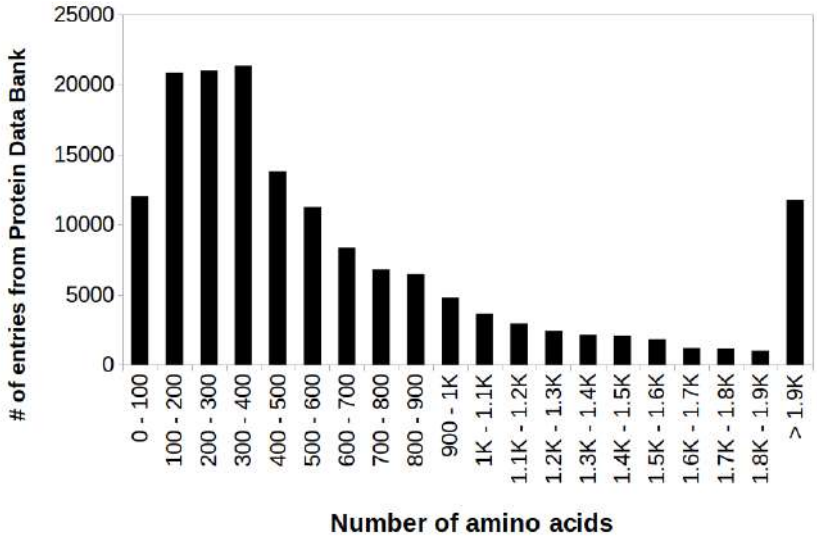
Source: own authorship

Figure 26 – Energy functions speedup for the simulation of a single pathway, considering the sequential and parallel approaches.



Source: own authorship

Figure 27 – Number of entries per protein size range



Source: www.rcsb.org/. Acessed in October 2019).

5.1.2 Data analysis of the case study

As proposed in Section 4.1.3, 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 the structures will evolve towards their native structure, starting from any initial spatial position. Therefore, the initial structures were randomly initialized before running the PathMolD-AB simulation. In such a 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 $1{,}000$ 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 4.1.4). Results were normalized in the range $[0..1]$ and plotted in the heatmaps shown in Figure 28, for the initial and final structures.

Each point of the horizontal and vertical axes of the heatmaps represents 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 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 towards the native structure of the protein. Figure 29 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 with the iterations and tends to stabilize at the end of the simulation.

Figure 30 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.

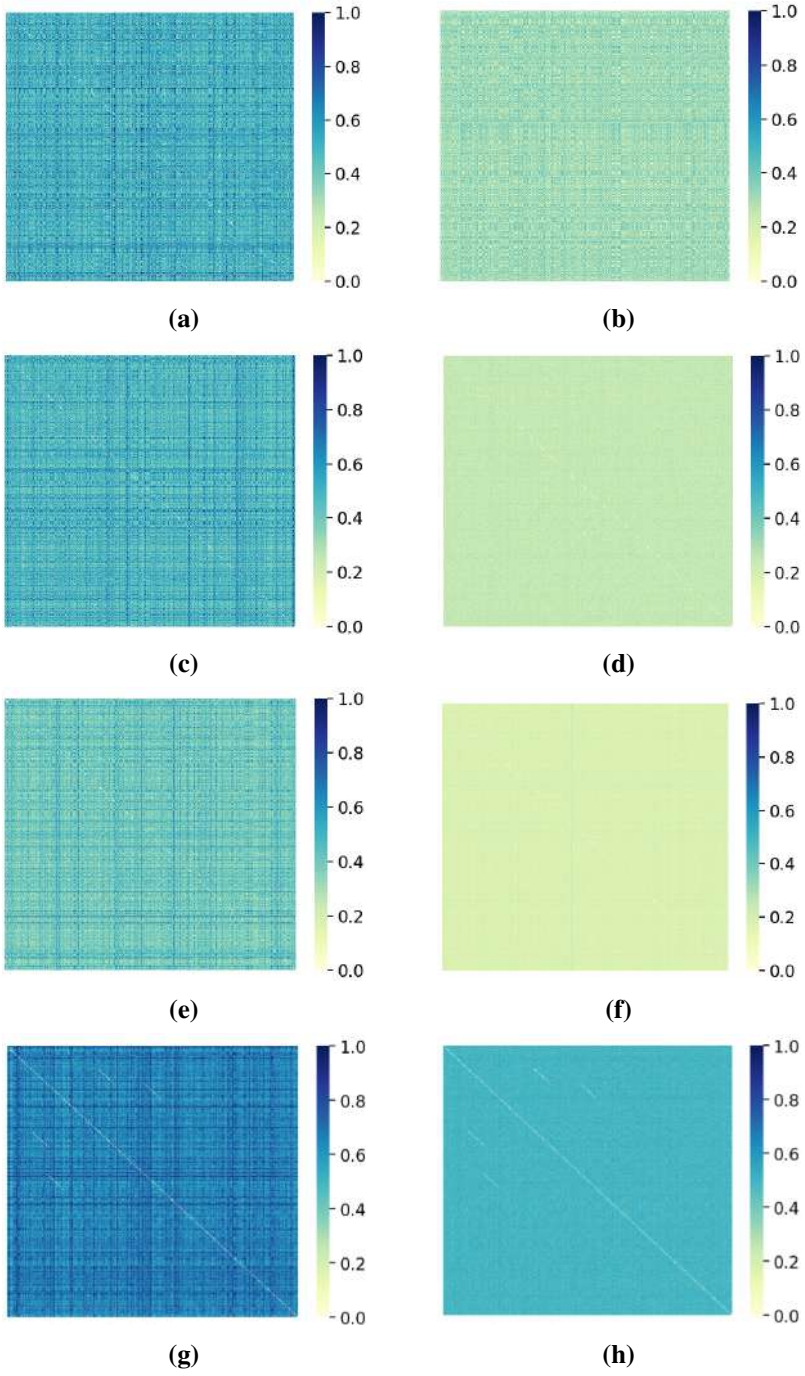
Additional information is provided in Table 8. It presents the average and standard deviation values of the energy and the radii of gyration, computed at the final step of the pathways. Final values of RgH are lower than RgP, suggesting the formation of a hydrophobic core (DILL; MacCallum, 2012). Notice that the standard deviations are small 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.

5.1.3 Comparison with biological structures

As mentioned in Section 4.1.4, we proposed a procedure for comparing the structures predicted by the MD method with structures re-scaled from the PDB.

Figure 31 shows the results for the 2GB1 and 1PLC proteins in terms of $RgAll$, RgH, and RgP (see Equation 36). 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

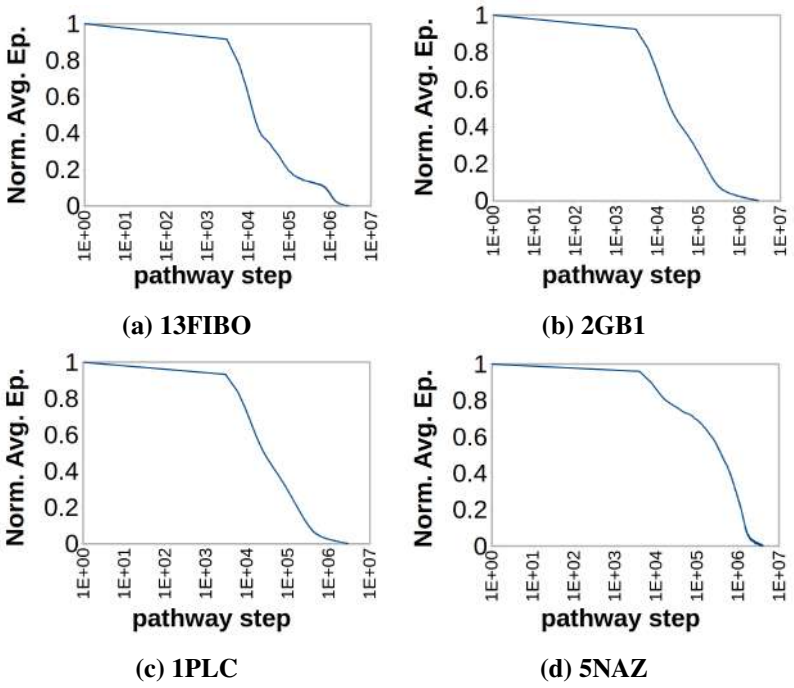
Figure 28 – (a,c,e,g) Normalized Kabsch RMSD between the 1,000 initial structures of the four datasets, and the final structures similarity (b,d,f,h).



Source: own authorship

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 Equation 13. 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.

Figure 29 – Average potential energy (E_p) per iteration.

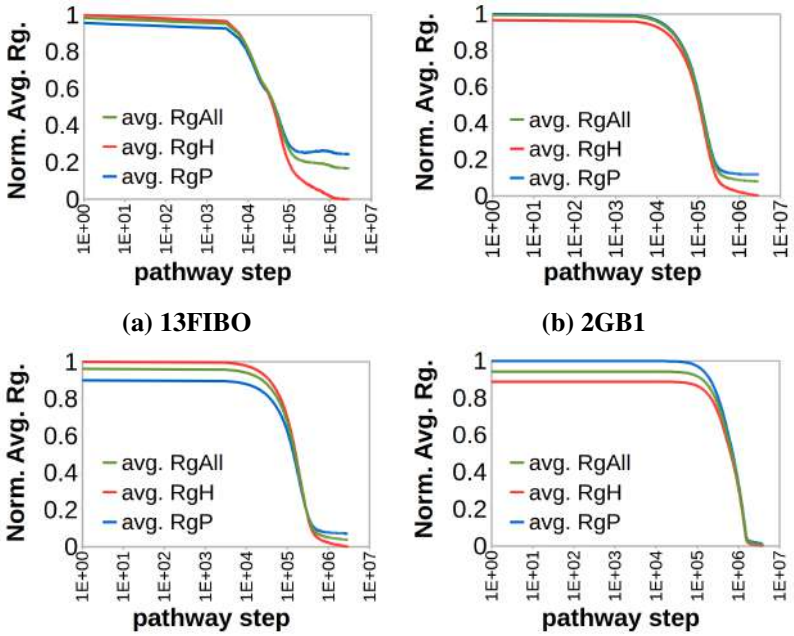


(a) 13FIBO **(b) 2GB1**

(c) 1PLC **(d) 5NAZ**

Source: own authorship

Figure 30 – Average radii of gyration ($RgAll$, RgP and RgH) per iteration.



(a) 13FIBO **(b) 2GB1**

(c) 1PLC **(d) 5NAZ**

Source: own authorship

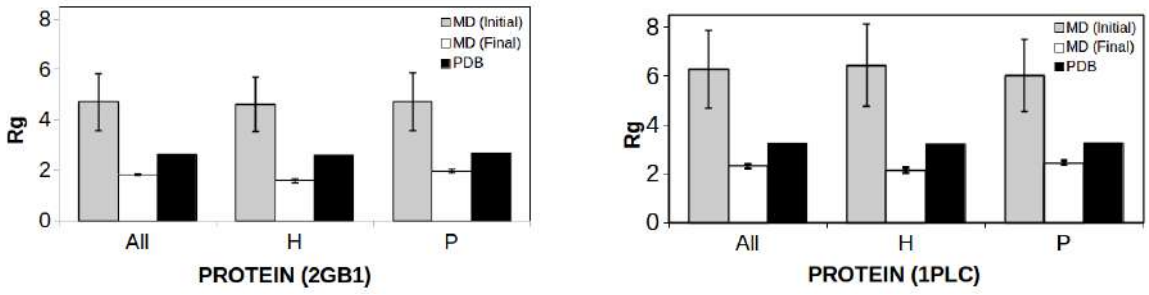
The predicted and the re-scaled biological structures were directly compared using the Kabsch-RMSD method (see Section 4.1.4), as shown in Figure 32. Kabsch-RMSD values were observed to be more distinct (6 for the 1PLC and 4.5 for the 2GB1) in the initial iterations than in the final ones (4.5 for the 1PLC and 3.5 for the 2GB1). Similarly, the standard deviation

Table 8 – 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. $\pm$ σ)	
	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

Source: own authorship

Figure 31 – 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.

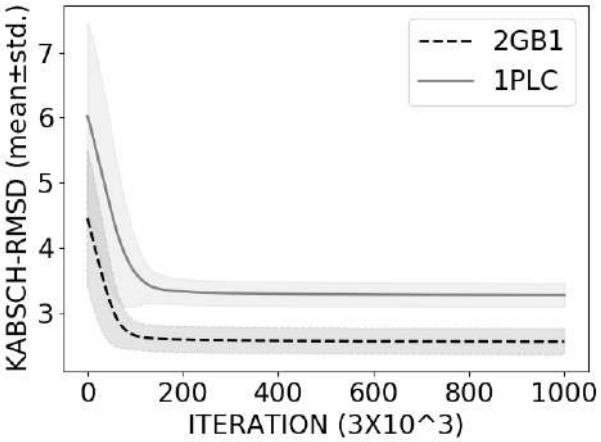


(a) 2GB1 **(b) 1PLC**

Source: own authorship

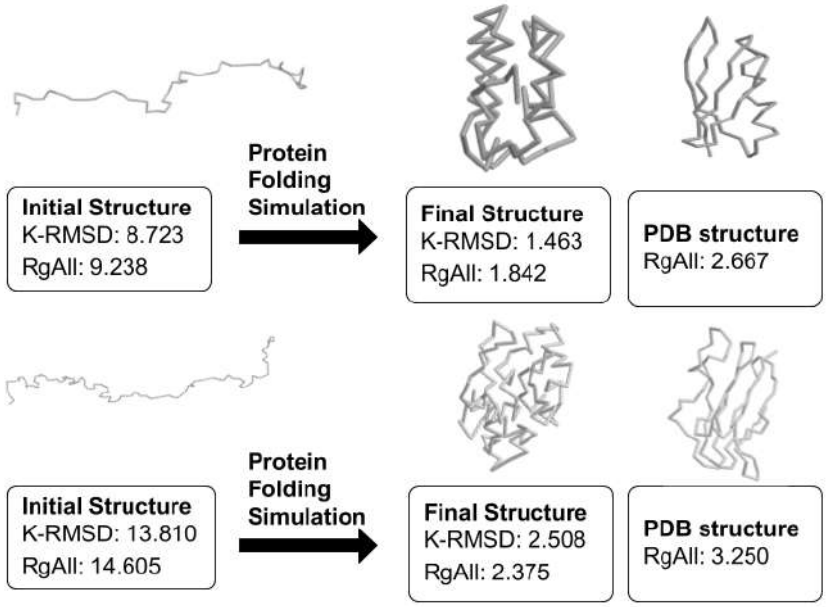
is higher in the initial iterations than in the final ones. These results reinforce the analysis of compactness presented before and the conclusion that the simulation produces results structurally similar to the biological structure (see, also, the diagram of Figure 33).

Figure 32 – Kabsch-RMSD (Mean and standard deviation) between the biological sequence and the predicted structure along of the iterations.



Source: own authorship

Figure 33 – Sample of a folding pathway simulation of 2GB1 and 1PLC proteins compared with the re-scaled biological structures from the PDB.



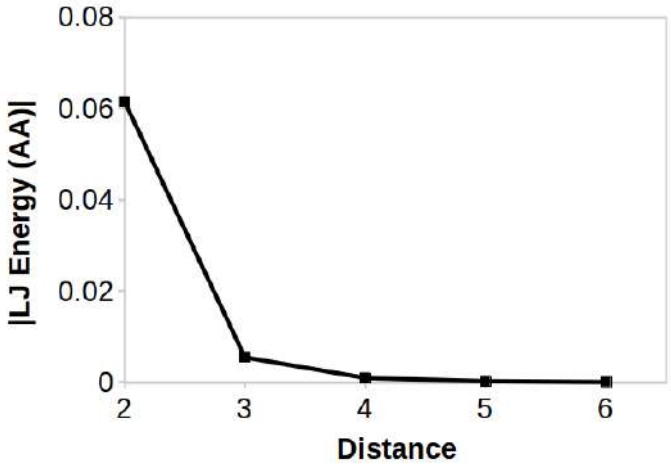
Source: own authorship

5.2 PARALLEL MOLECULAR DYNAMICS WITH NEIGHBOURHOOD LIST

This Section aims at analyzing the feasibility of the proposed Neighbourhood List with a parallel Molecular Dynamic method (NL) presented in Section 4.1.5. Our approach was compared with results obtained with the pure sequential MD (seq) and parallel MD (par) approaches (previously presented by (BENÍTEZ; LOPES, 2012) and (HATTORI *et al.*, 2020b)). This analysis includes experiments of the relative processing time assessment and speedup evaluation. We also investigated the energy over time to observe the impact of the NL application.

The cut-off radius that defines the neighbors of each protein was based on the analysis of the Lennard-Jones energy (LJ) values in different distances, as shown in Figure 34. The energy was based on the value generated by two hydrophobic residues in five different distances between two to six. Results indicated a high energy decay for a longer distance between the residues, as expected, given the Equation 12. For example, the energy value for a distance equal to four ($-9.76x10^{-4}$) between two hydrophobic residues represents only 1.58% when compared to the energy in the distance equal to two ($-6.15x10^{-2}$). Then, residues with distances higher than four have even fewer values and are near to zero. Given that this work´s main objective is not to test different cut-off radius values, we defined empirically this value equal to four.

Figure 34 – The absolute Lennard-Jones energy value is generated by interacting two hydrophobic residues (AA) at different distances.



Source: own authorship

5.2.1 Speedup Performance evaluation

The experiments were executed in a GPU Titan XP and 12GB of global memory, CPU processor Intel(R) Core(TM) i7-8700 3.20GHz with 31GB of RAM with Linux Ubuntu server 18.04. To compile the GPU program, we used CUDA 9.0.

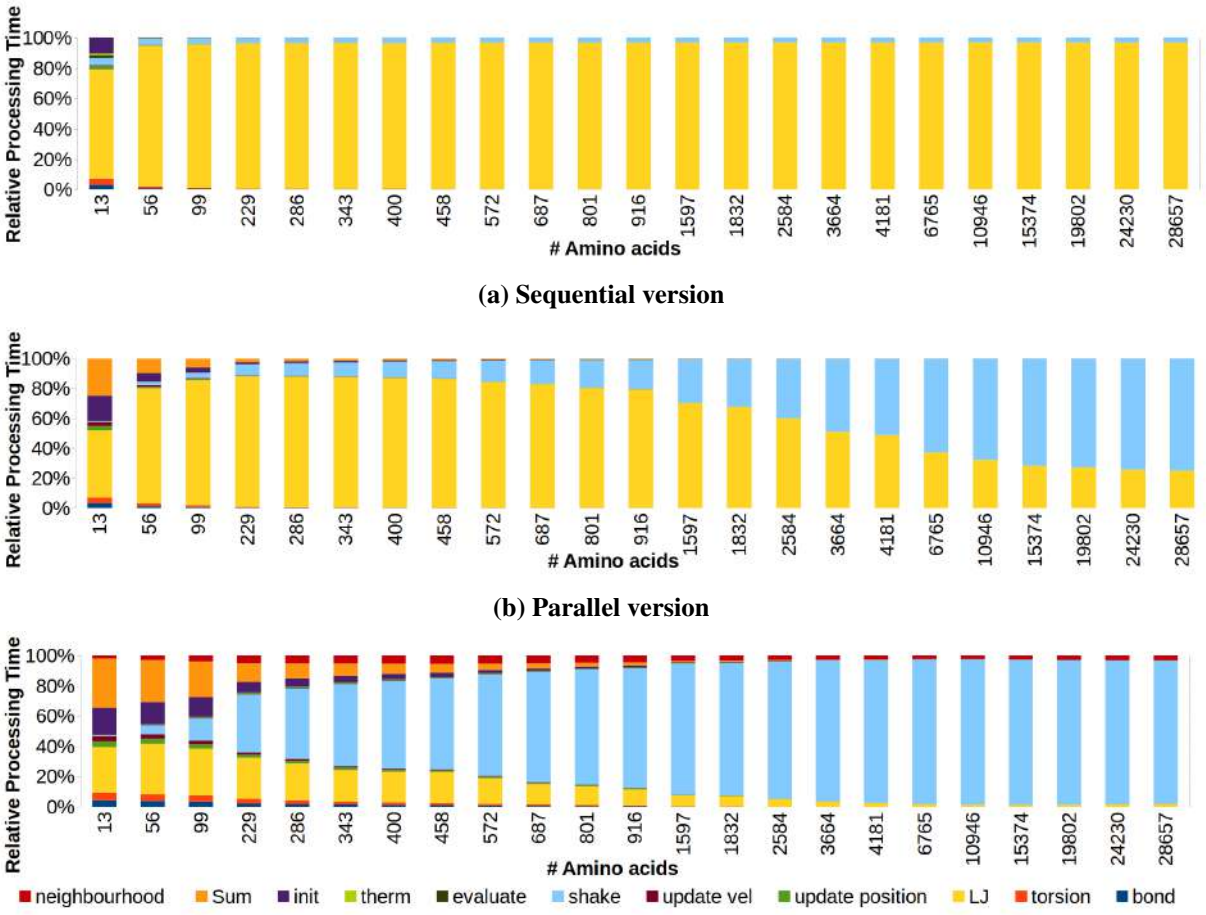
Synthetic sequences were used to evaluate the speedup performance of the NL method (see Section 4.1.5). The performed experiments were based on 3,000 iterations of the MD method.

Figures 35(a), 35(b), and 35(c) show the relative processing time of each function of the MD method for the sequential, parallel, and NL approaches, respectively. In the pure sequential and parallel models, we observed the same behavior of the previous works presented by (HATTORI *et al.*, 2020b). In the sequential model, it was shown that most of the used processing time is spent by the LJ function. Also, it was observed in the parallel model that the proportion of LJ function time-consumption decreased when compared to the sequential model.

In the NL model, we observed that the time-consumption of the LJ calculation decreased even more when compared to the other two models. Furthermore, for larger protein sequences, the relative processing time almost vanishes from the plot. On the other hand, we observed that the Shake algorithm is a new issue for larger protein sequences (see Algorithm 1). As previously presented by (HATTORI *et al.*, 2020b), this algorithm is a non-parallel process given the dependence of the variables that are updated.

The speedup performance of the sequential and parallel models was evaluated with

Figure 35 – Time-consuming of the MD functions in the sequential, parallel, and parallel with neighbourhood list.



(a) Sequential version

(b) Parallel version

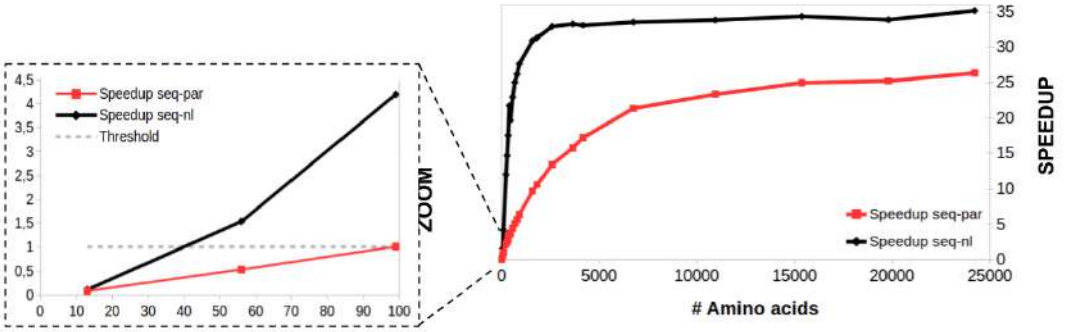
(c) Parallel with neighbourhood list

Source: own authorship

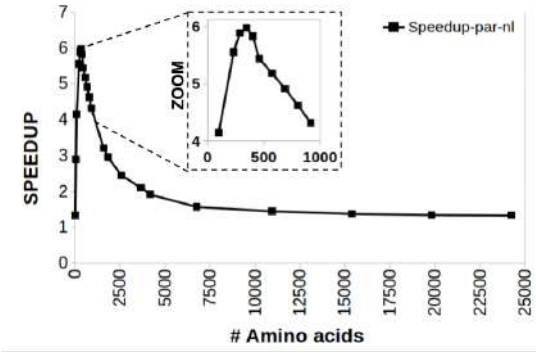
chains of different amounts of amino acids, as showed in Figure 36. The speedup metric consists of the division of the computational time of one model by the other model. For example: the time generated by the sequential model divided by the pure parallel version. Figure 36(a) shows the overall speedup between the sequential model and the parallel MD ($seq - par$), as well as the sequential model and parallel MD with the NL technique ($seq - nl$). We observed that the NL approach obtained performance improvement when compared to the parallel model speedup for all different proteins' sizes. As shown in the zoom, the NL method overcomes the performance of the sequential model (speedup > 1) for protein sequences larger than 56 amino acids, while the pure parallel model overcomes the sequential approach for proteins higher than 99 amino acids. Despite the performance improvement, the speedup curve maintained the same logarithmic behavior.

As previously demonstrated in Figure 35(a), the LJ function is the most time-consumption function in the sequential model. As shown in Figure 36(c), the NL model showed

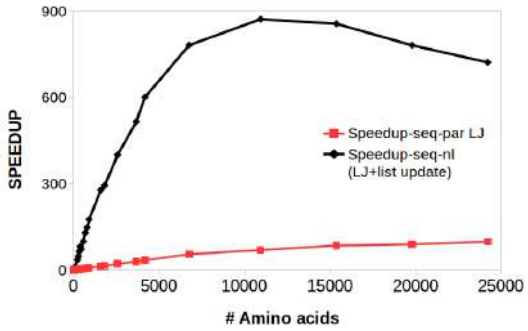
Figure 36 – Speedup analysis of the parallel and NL models.



(a) Speedup between the sequential and parallel model ($seq - par$), and speedup between the sequential and parallel model with neighborhood list ($seq - nl$).



(b) Speedup between the parallel model and parallel model with neighborhood list ($par - nl$).



(c) Speedup of the Lennard-Jones function of the sequential and parallel models ($seq - par\ LJ$), and the speedup of the Lennard-Jones function of the sequential and NL model, including the time of the neighborhood update list, ($seq - nl\ LJ + listupdate$).

Source: own authorship

a higher performance in terms of LJ speedup when compared to the parallel model, even when the time required by the updated list is added, this approach shows a higher speedup. We also compared the speedup between the parallel model and the NL method ($par - nl$), as shown in Figure 36(b). The results suggested that the improvement of the NL approach performance was between 99 and 916 amino acids. The NL achieves performance at least four times faster than the pure parallel model. This result is a crucial improvement given that many biological proteins contain this range of amino acids in the chain (BROCCHIERI; KARLIN, 2005; TIESSEN *et al.*, 2012). Based on PDB statistical information, this improvement covers more than 80% of the proteins from PDB, as presented in Figure 27.

5.2.2 Case Study

In this section, an analysis is presented of the behavior of energy and the compactness along the folding process. These analyses were performed with the sequential model and with the NL approach in four case studies (13FIBO, 2GB1, 1PLC, and 2QHT), see Sections 4.1.3 and 4.1.5 for more details. Also, the sequential MD as well as the MD with NL simulations were based on an average of 100 experiments.

Figures 37(a), 37(b), 37(c), and 37(d) show the normalized potential energy between zero and one along the iteration for 13FIBO, 2GB1, 1PLC, and 2QHT. Overall, we observed that the NL technique does not change the energy decay behavior, even for the longer protein used in this case study (2QHT). Surprisingly, the potential energy for the 13FIBO obtained smaller values in the NL simulation. Despite producing a lower energy value, the results suggest that applying NL to smaller proteins may be more sensitive. On the other hand, for proteins larger than 56, just a slight difference in behavior was observed, even for 2QHT protein, where the cut-off radius represents 1.9% of protein size.

Figures 38(a), 38(b), 38(c), and 38(d) show the $RgALL$ normalized along the folding simulation. We also observed that the structure compactness obtained similar results between the NL and the sequential approach. In the simulation of the 13FIBO protein, we observed in the NL simulation that the structure stabilized in the lower compactness values, while the sequential structures increased, possibly caused by the interactions between the elements of the structure.

5.3 RECURRENT NEURAL NETWORK FOR THE PROTEIN FOLDING PROBLEM

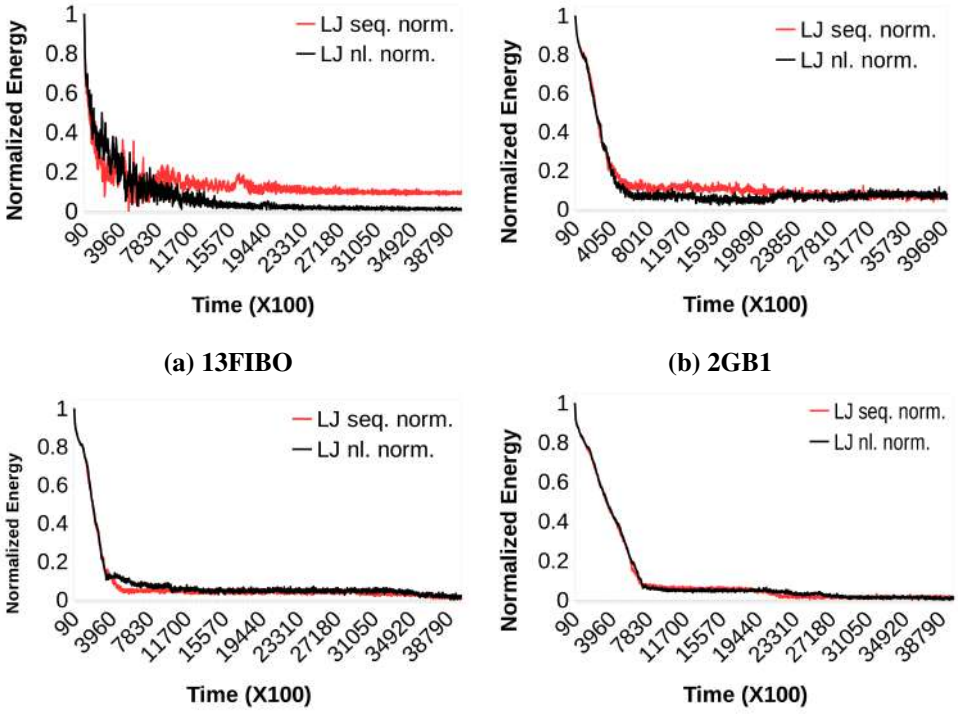
All experiments done in this section were run on a computer with an Intel Core i7 processor at 3.30GHz, two GPU Nvidia Titan X, and minimal installation of Ubuntu 18.04 LTS [1]. The software was developed using the Python programming language, the Keras 2.4, and Tensorflow 1.3 frameworks [2].

5.3.1 Dataset of protein folding trajectories

This experiment aims at analyzing the protein structure differences along the folding trajectories. The protein structure was collected along the MD simulation using different spacings.

[1] Available in: www.ubuntu.com
[2] Available in: https://keras.io/

Figure 37 – Energy normalized along the simulation for the 13FIBO, 2GB1, 1PLC, and 2QHT proteins.

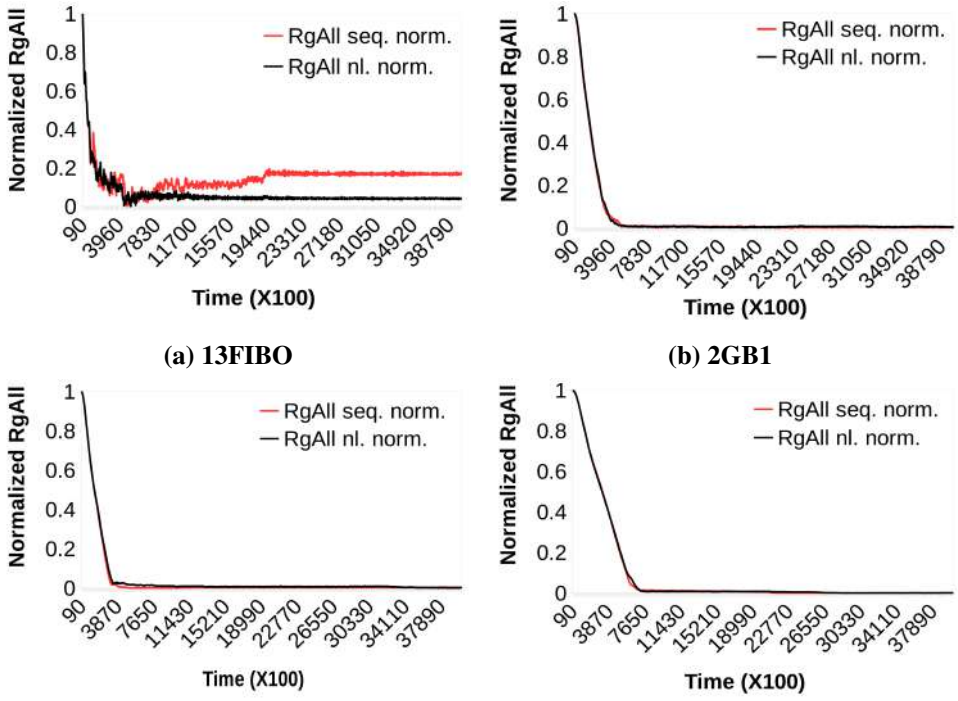


(a) 13FIBO **(b) 2GB1**

(c) 1PLC **(d) 2QHT**

Source: own authorship

Figure 38 – Compactness normalized along the simulation for the 13FIBO, 2GB1, 1PLC, and 2QHT proteins.



(a) 13FIBO **(b) 2GB1**

(c) 1PLC **(d) 2QHT**

Source: own authorship

The $step\_size$ (see Section 4.2) used in this work was at each $3{,}000$, $6{,}000$, $15{,}000$, and $30{,}000$ iteration of the MD. After that, the subsequence structures were analyzed using the Kabsch RMSD algorithm (see Section 4.2.1.1) to get the structural differences in distinct $step\_size$ values.

Figure 39 shows the average Kabsch RMSD (see Section 4.2.1.1) between the consecutive folding states along the trajectories. For all proteins datasets, in the three higher $step\_size$ values, we observed more abrupt changes in the structures along the folding process. The $step\_size$ equal to $3{,}000$ demonstrated smoother changes in the structure than the other $step\_size$ values. We also noted differences in structural changes in the 2GB1 and 1PLC datasets for each $step\_size$. They were indicating that even at the end of the simulation, the bigger the structure, the higher the changes in the $step\_size$ values. These abrupt or higher modifications could make it difficult to predict the tendency of the next folding state. Thus, we adopted the $step\_size$ equal to $3{,}000$ for the following experiments with RNN.

5.3.2 Recurrent Neural Networks analysis

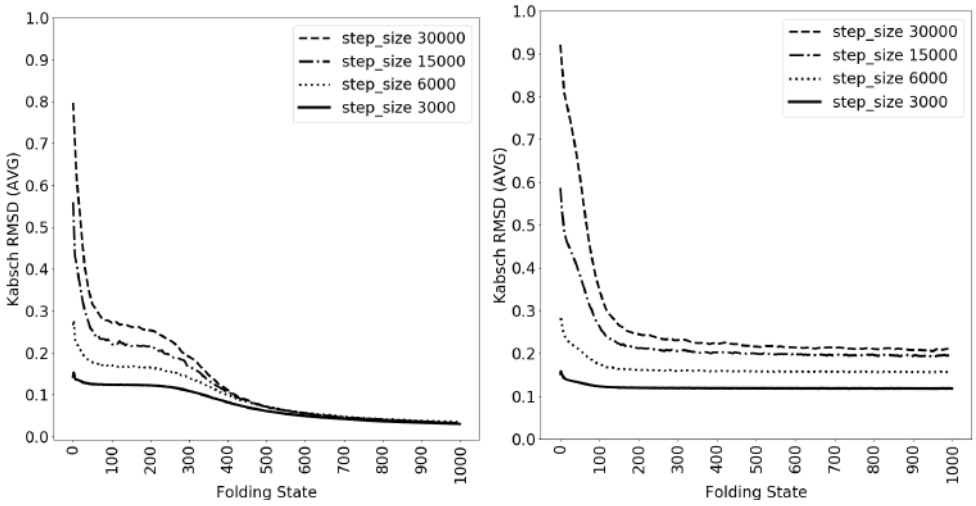
In this experiment creates a comparison between the RNN architectures (sRNN, GRU, and LSTM) and analyzes the number of folding trajectories on network results.

The RNNs setup used the same parameters, described in Section 4.2.3. In this experiment, 100 prior folding states were used to predict the next one. 13FIBO, 2GB1, and 1PLC datasets (see Section 5.3.1) with 1,000 folding trajectories data were used for the benchmark. A subset of these folding trajectories with 100 samples was also used to analyze the influence of the amount of trajectories data in the results.

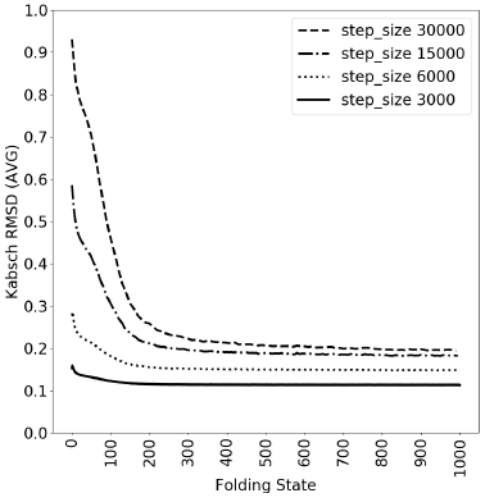
Table 9 presents the results of the RNN architectures. The RNN with the gate system (GRU and LSTM) showed smaller prediction errors and more similar results than the sRNN. Among these gate system networks, the LSTM got the smallest errors between the RNN architectures in all datasets. We also observed that a longer protein sequence produces higher differences between predicted and target structures. When the amount of the protein folding data was analyzed, we achieved a lower difference between the prediction and target using more data. The sRNN was the architecture with the most positive impact with the increased data amount. Finally, the LSTM learning curve showed that the model could generalize the predictions, as shown in Figure 40.

The second experiment aims to analyze the impact of different amounts of previous

Figure 39 – The subsequent structural change of the protein over the simulation, considering different $steps\_size$ (3000, 6000, 15000, 30000).



(a) 13FIBO **(b) 2GB1**



(c) 1PLC

Source: own authorship

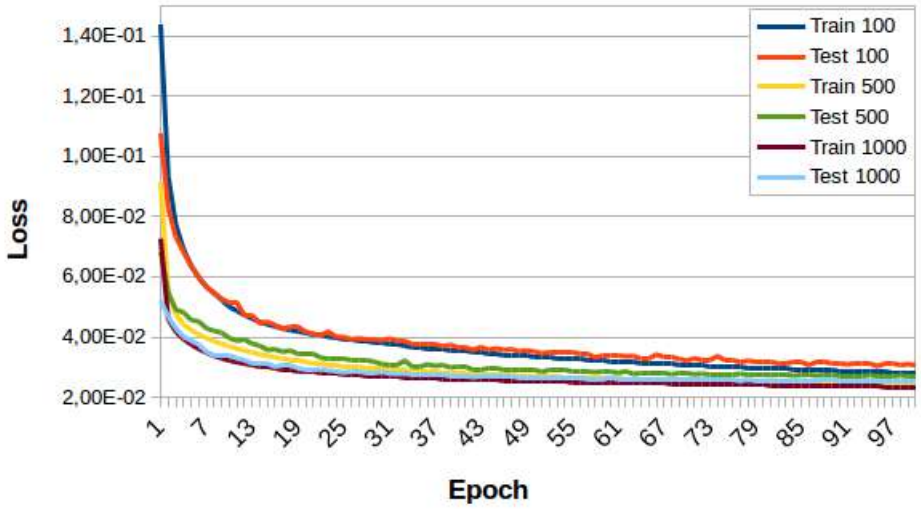
Table 9 – Test loss performance of the sRNN. GRU and LSTM in 13FIBO. 2GB1 and 1PLC datasets using 100 and 1,000 pathways data.

	Test Loss (x10^-2)					
	13FIBO		2GB1		1PLC	
network / #pathway	100	1000	100	1000	100	1000
sRNN	7.95	5.60	5.15	4.25	11.58	5.21
GRU	3.17	2.63	4.84	3.85	6.09	4.81
LSTM	3.03	2.53	4.77	3.59	5.97	4.42

Source: own authorship

states to predict the next one. According to the best setup of the previous experiment, we used the LSTM architecture. The result of the second experiment is present in Figure 41. The LSTM was tested using 2, 20, 40, 60, 80, and 100 previous folding states. Among these different amounts of such conditions, we observed an improvement in the prediction using over 20 previous folding

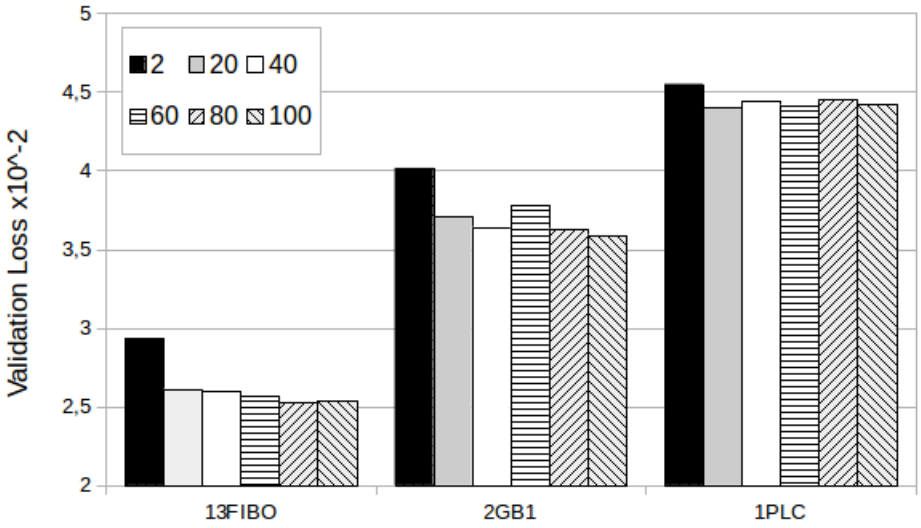
Figure 40 – LSTM learning curve for the train and test sets using different # pathways.



Source: own authorship

states. However, the use of 20, 40, 60, 80, and 100 of them did not reveal notable improvements. It was observed that even with the increase of the previous folding states amount, the results for longer proteins were less effective.

Figure 41 – Test MAE loss for different amounts of previous folding states (2, 20,40,60, 80, and 100) to predict the next state in 13FIBO, 2GB1, and 1PLC datasets.



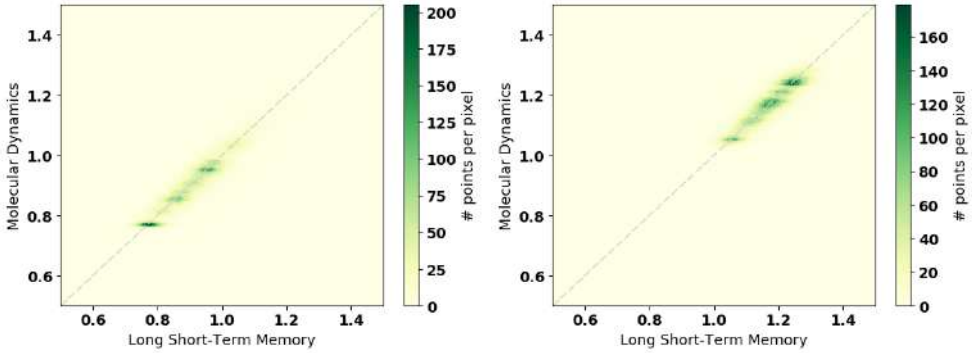
Source: own authorship

When compared to the results of the literature, the loss results reduced ten times the error prediction for the protein 13FIBO (0,33 to 0,03), see Appendix A, showing that the new network model and dataset proposed in this work contributed to the improved performance of the result.

In the third experiment, we analyzed the LSTM predictions in the test subset with the trained model using other Bioinformatics metrics, such as radii of gyration and energies. These experiments were performed using the heatmap plot. In this representation, more similar values between predicted (LSTM) and target (MD) are more concentrated and placed closer to the diagonal of the heatmap. It is also possible to identify the scattering of these values in this plot since a considerable density value in one point of the plot increases the color intensity of the pixel.

The hydrophobic and polar radius of gyration of the LSTM and MD method is presented in Figure 42. It was observed that the prediction results of the radius of gyration of the hydrophobic elements (RgH) generated similar values from the prediction (see Figure 42(a)). It was also detected that this behavior was reproduced at different stages of folding, given that the distribution of points changes along the diagonal of the plot. The radius of gyration of polar elements (RgP) also showed a similar behavior of the hydrophobic results, as presented in Figure 42(b). Also, it was noticed that RgH results are lower than the RgP ones, the expected behavior of the globular representation of proteins, where the hydrophobic drives the folding process to generate a core of these elements.

Figure 42 – Heatmap plots of the angle, torsion, and Lennard-Jones energies of the predicted structure (LSTM) and the target (MD).



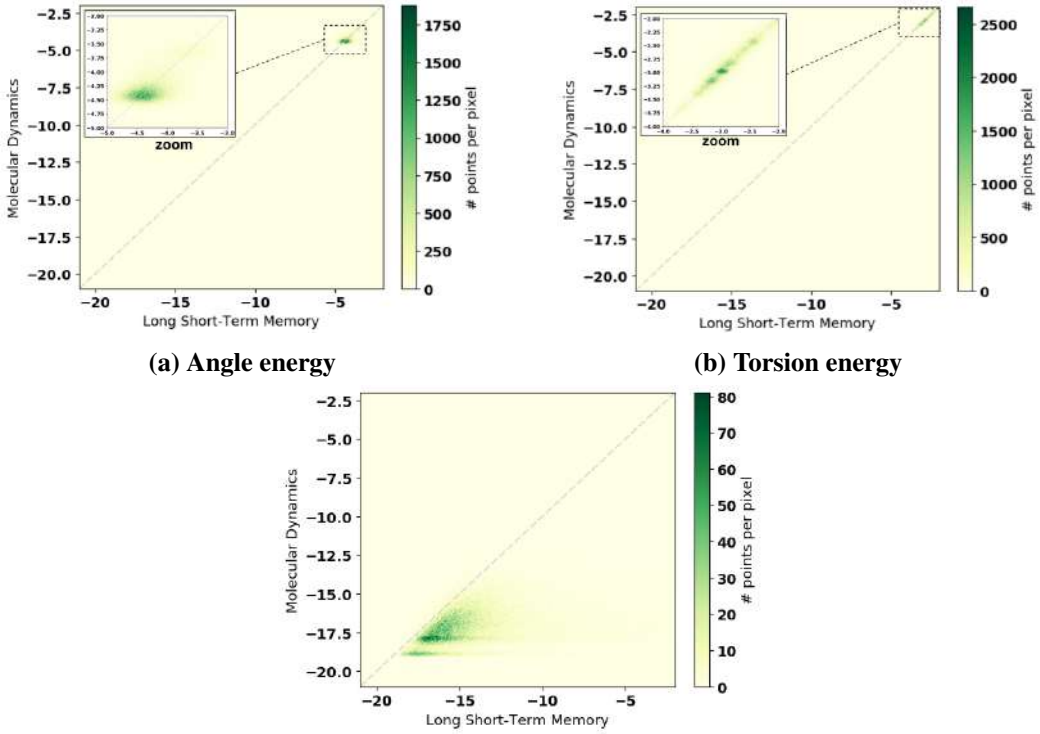
(a) Hydrophobic radius of gyration (RgH) **(b) Polar radius of gyration (RgP)**

Source: own authorship

The angle, torsion, and Lennard-Jones energies of the LSTM and MD method are presented in Figure 43. The angle and torsion energies of the predicted model demonstrated a more similar behavior when compared with the target, as showed in Figures 43(a) and 43(b), indicating that the relative spherical coordinates can be favorable to represent these two energy features. We observed that the Lennard-Jones´s energy obtained by the LSTM was higher than the MD results (see Figure 43(a)), showing a distribution under the diagonal line of the

plot. The high sensitivity of the LJ energy to near distances of two residues can generate this difference in the behavior. Even with a small error prediction, the LJ energy might produce higher values depending on the interaction and the distance of elements. This result shows that our representation model may not be enough to predict the structure in this energy term.

Figure 43 – Heatmap of angle, torsion, and Lennard-Jones energy of the predicted structure (LSTM) and the target (MD).



(a) Angle energy

(b) Torsion energy

(c) Lennard-Jones energy

Source: own authorship

6 CONCLUSION

The Protein Folding Problem (PFP) is an open challenge in the Computational Biology area. Recent studies indicate that Deep Learning approaches have overcome some traditional methods in many proteomics problems. However, computational intelligence has been poorly explored for the PFP. In this theses, we proposed a novel approach based on Recurrent Neural Networks focused on one-step-ahead prediction of protein folding pathways. (1) a framework to generate protein folding datasets using sequential and parallel MD; (2) an NL approach to the GPU-parallel MD; and, then, (3) RNNs approaches for the PFP.

MD is an approach widely used 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 of the simulated protein increases. This fact justifies developing the first step of this thesis, more efficient methods, such as the PathMolD-AB package. This software package uses MD with the canonical ensemble that deals with the Newtonian evolution of protein models. This software also uses a coarse-grained model to represent proteins and a parallel master-slave computing architecture that enables experiments for tracking the Spatio-temporal pathways of protein folding. Such pathways can help analyze the structure changes over the folding and visualize possible abnormal events during this process, such as misfolding and structural instability, typical of intrinsically disordered structures.

The performance of parallel and sequential MD approaches showed that the parallel version is faster than the sequential version for protein sequences larger than 99 amino acids, and speedup increases significantly for the parallel version. We showed that, among several functions of the PathMolD-AB package, the LJ function is the most computationally expensive. Notwithstanding, we achieved the highest speedup in this function with the parallel version, decreasing the bottleneck of sequential MD method.

The speedup measured in different protein sequences sizes suggested a logarithmic trend. The decay of speedup for large sequences could result from the massive concurrency between processing threads in the CUDA cores of the GPU and the high processing demand of the Shake algorithm. However, this is not a drawback, since the distribution of the protein sequences sizes deposited in the PDB shows that the proposed software can simulate the folding of most biological proteins in the PDB.

PathMolD-AB generated a large amount of simulation data when applied to the case

studies. Their analyses indicated that at the final state of the Spatio-temporal trajectories lead to similar conformations, starting from distinct initial structures, as suggested by the energy funnel theory. The thermodynamic characteristics are coherent with the energy curve that 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 coarse-grained model employment, as "biological-like" validation is a step for more realistic simulations.

The main drawback of the PathMolD-AB simulations are that the compactness of predicted structures were smaller than the re-scaled biological structures. These results suggest the need to optimize the hydrophobicity interaction weights between the residues proposed by Irbäck *et al.* (1997).

In the second step of this thesis, the NL approach to the GPU-parallel MD was proposed to decrease the computational time of the purely sequential and parallel MD versions. Then, the time-consumption performance of the NL approach was compared to the purely sequential and GPU-parallel MD methods (presented in the PathMolD-AB package). Results revealed that the NL approach reduces the time-consumption of the LJ function compared to the purely sequential and parallel versions.

The NL approach was faster in almost all cases compared to the purely sequential and parallel MD versions, only with protein sequences smaller than 56 amino acids, there was a faster simulation result to the sequential version. The higher improvements of the NL compared to the purely parallel MD were in simulations with protein sequences between 99 to 1,000, covering 80% of the sequences from Protein Data Bank. This improvement indicates that the NL approach can be promising for many Bioinformatics and Biophysics studies simulating large sequences.

Since the NL decreases time-consumption omitting long-range pairwise interactions, the energy curve of simulations were analyzed to observe whether this approach generates anomalies in the curve. For proteins larger than 13 amino acids, few differences were observed between the simulation performed were identified by the sequential MD, and the NL approaches. These results indicate that this approach can be more appropriate for lager protein sequences and conclusions presented in the speedup experiment.

In the third step of this thesis, RNNs using the many-to-one model were proposed for the PFP, and protein folding datasets using Relative Spherical Coordinates (RSC) were presented. Analysis of protein folding datasets indicated that the low frequency of data collection during the

MD simulation ($step\_size$) generates more sudden changes in the structure. At the beginning of the simulation, the structure tends to change suddenly, where the energy curve tends to decrease abruptly. This fact indicates that trajectory data with sudden changes (higher $step\_size$) can make it more difficult to predict the movement and to achieve the next folding state.

Regarding the comparative analysis between LSTM, GRU, and sRNN approaches for the PFP, network architectures with a gated system (GRU and LSTM) obtained the smaller prediction errors, indicating a more suitable for this problem. LSTM achieved the best results compared to the GRU and sRNN in all datasets, using less or more protein trajectory data. Thus, results indicated that LSTM is a more consistent approach to the PFP study.

Regarding the number of previous folding steps for predicting the step ahead, 20 previous folding states generate a higher positive impact on predictions with the LSTM. It was observed that the addition of previous folding states in the input data helped to decrease the error prediction, indicating that the many-to-one with the RNN model is an appropriate model to explore this problem. Concerning the number of previous folding states, we detected a higher improvement when it was used over 20 previous times. However, we do not observe a significant difference for longer than 40 previous folding states.

The radii of gyration and energies analysis showed new insights of the predicted structures. For example, we observed that the compactness of the hydrophobic and the polar elements between the target and predicted structures obtained similar values. And so the bond and torsion energies. This shows that the relative spherical representation is suitable for this problem. However, we observed more dissimilar values between the predicted and the target for the Lennard-Jones features. A possible reason for this result is the sensibility energy. Additional investigations regarding the LJ energy shall be done seeking to keep reducing the foreseeability gap.

The PathMolD-AB package can be applied for several computational studies related to the PFP and generate new insight about the Spatio-temporal feature of the data. The NL approach using parallel MD with GPU showed promising features to studies applied to a massive amount of protein folding data, making this type of research more feasible to be performed. Finally, our study related to the LSTM obtained new research directions that suggest the continuity of this work, the initial objectives were achieved satisfactorily.

6.1 FUTURE WORKS

Future works will address the MD refining, 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).

In the NL technique, the future work may include an approach to decrease the frequency at which the Shake algorithm is computed, since the geometric constraint function causes the highest computing overhead. We also suggest applying other NL techniques, such as Bounding Volume Hierarchy (BVH) (ERICSON, 2004), and Bounding Volume Compressed (BVC) (HOWARD *et al.*, 2019b). In addition, another distance metrics will be considered, such as Mahalanobis distance (MARTOS *et al.*, 2013).

Other works may include Transfer Learning analysis between RNN models trained in different datasets (13FIBO, 2GB1, 1PLC, and 5ANZ). It is suggested to perform the protein folding prediction with other types of RNN, such as the Neural Turing Machines and Memory Networks. Other features can be considered to decrease the protein folding prediction error, such as the hydrophobicity of the protein sequence. In addition, experiments applying Transfer Learning techniques are proposed as future work (WEN *et al.*, 2020).

REFERENCES

ABRAHAM, Mark James; MURTOLA, Teemu; SCHULZ, Roland; PÁLL, Szilárd; SMITH, Jeremy C; HESS, Berk; LINDAHL, Erik. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. **SoftwareX**, v. 1, p. 19–25, 2015.

ALBERTS, B; JOHNSON, A; LEWIS, J; RAFF, M; ROBERTS, K; WALTER, P (Ed.). **Molecular Biology of the Cell**. 5. ed. New York: Garland Science, 2002. v. 1. 1392 p.

ALEMI, Omid; FRANÇOISE, Jules; PASQUIER, Philippe. GrooveNet: Real-time music-driven dance movement generation using artificial neural networks. **Networks**, v. 8, n. 17, p. 1–26, 2017.

ALQURAISHI, Mohammed. End-to-end differentiable learning of protein structure. **Cell Systems**, v. 8, n. 4, p. 292 – 301.e3, 2019.

ANFINSEN, Christian B. Studies on the principles that govern the folding of protein chains. **American Association for the Advancement of Science**, v. 181, n. 4096, p. 223–230, 1973.

ANGERMUELLER, Christof; PÄRNAMAA, Tanel; PARTS, Leopold; STEGLE, Oliver. Deep learning for computational biology. **Molecular Systems Biology**, v. 12, n. 7, p. 878, 2016.

AQUINO, Marcelo R.; GUTOSKI, Matheus; HATTORI, Leandro T.; LOPES, Heitor S. Soft Biometrics Classification Using Denoising Convolutional Autoencoders and Support Vector Machines. *In*: **Proceedings of 4th IEEE Latin American Conference on Computational Intelligence LA-CCI**. Piscataway, NJ: IEEE Press, 2017. p. 1–6.

AQUINO, Marcelo R.; GUTOSKI, Matheus; HATTORI, Leandro T.; LOPES, Heitor S. The Effect of Data Augmentation on the Performance of Convolutional Neural Networks. *In*: **Proceedings of 4th IEEE Latin American Conference on Computational Intelligence LA-CCI**. Piscataway, NJ: IEEE Press, 2017. p. 1–6.

ATKINS, J.; HART, W.E. On the intractability of protein folding with a finite alphabet. **Algorithmica**, v. 25, n. 2, p. 279–294, 1999.

BAILEY, Nicholas P; INGEBRIGTSEN, Trond S; HANSEN, Jesper Schmidt; VELDHORST, Arno A; BØHLING, Lasse; LEMARCHAND, Claire A; OLSEN, Andreas E; BACHER, Andreas K; COSTIGLIOLA, Lorenzo; PEDERSEN, Ulf R *et al.* Rumd: A general purpose molecular dynamics package optimized to utilize gpu hardware down to a few thousand particles. **SciPost Phys**, v. 3, p. 038, 2017.

BARRETO-OJEDA, Estefania; CORRADI, Valentina; GU, Ruo-Xu; TIELEMAN, D. Peter. Coarse-grained molecular dynamics simulations reveal lipid access pathways in p-glycoprotein. **The Journal of General Physiology**, v. 150, n. 10, p. 1–13, 2018.

BENÍTEZ, César Manuel Vargas. **Contributions to the Study of the Protein Folding Problem using Bioinspired and Molecular Dynamics**. 2015. 191 p. Phd Thesis (PhD Thesis) — Universidade Tecnológica Federal do Paraná, 2015.

BENíTEZ, César Manuel Vargas; LOPES, Heitor Silvério. Parallel artificial bee colony algorithm approaches for protein structure prediction using the 3dhp-sc model. *In*: ESSAAIDI, Mohammad; MALGERI, Michele; BADICA, Costin (Ed.). **Intelligent Distributed Computing IV**. Berlin, Heidelberg: Springer, 2010. p. 255–264.

BENÍTEZ, César Manuel Vargas; LOPES, Heitor Silvério. Molecular Dynamics for Simulating the Protein Folding Process Using the 3D AB Off-Lattice Model. *In*: **7th Brazilian Symposium on Bioinformatics, BSB**. Heidelberg: Springer, 2012. p. 61–72.

BENÍTEZ, César Manuel Vargas; LOPES, Heitor S. Ab-initio protein folding using molecular dynamics and a simplified off-lattice model. **Journal of Bionanoscience**, v. 7, n. 4, p. 391–402, 2013.

BENÍTEZ, César Manuel Vargas; WEINERT, Wagner; LOPES, Heitor Silvério. Gene expression programming for evolving two-dimensional cellular automata in a distributed environment. *In*: **Intelligent Distributed Computing VIII**. Heidelberg: Springer, 2015. p. 107–117.

BERENDSEN, Herman JC; POSTMA, JPM van; GUNSTEREN, Wilfred F van; DINOLA, ARHJ; HAAK, JR. Molecular dynamics with coupling to an external bath. **The Journal of Chemical Physics**, v. 81, n. 8, p. 3684–3690, 1984.

BERNO, Brenda S.; GABARDO, Cristiano A.; GUTOSKI, Matheus; HATTORI, Leandro T.; LOPES, Heitor S. A Framework for Analyzing Book Covers and Co-purchases using Object Detection and Data Mining Methods. *In*: **Proceedings of 7th IEEE Latin American Conference on Computational Intelligence LA-CCI**. Piscataway, NJ: IEEE Press, 2019. p. 1–6.

BEZKOROVAYNAYA, Olga. **Coarse-grained peptide models: conformational sampling, peptide association and dynamical properties for peptides**. 2011. Phd Thesis (PhD Thesis) — Johannes Gutenberg-Universität Mainz, 2011.

BOHNUUD, Tanggis; LUO, Lingqi; WODAK, Shoshana J; BONVIN, Alexandre MJJ; WENG, Zhiping; VAJDA, Sandor; SCHUELER-FURMAN, Ora; KOZAKOV, Dima. A benchmark testing ground for integrating homology modeling and protein docking. **Proteins: Structure, Function, and Bioinformatics**, v. 85, n. 1, p. 10–16, 2017.

BOIANI, Mateus; PARPINELLI, Rafael Stubs. A gpu-based hybrid jde algorithm applied to the 3d-ab protein structure prediction. **Swarm and Evolutionary Computation**, v. 58, p. 100711, 2020.

BONETTI, Daniel Rodirgo Ferraz. **Algoritmos de estimação de distribuição para predição ab initio de estruturas de proteínas**. 2015. Phd Thesis (PhD Thesis) — Universidade de São Paulo, 2015.

BREUEL, Thomas M. Benchmarking of LSTM Networks. ArXiv. 2015.

BRILHADOR, Anderson; GUTOSKI, Matheus; HATTORI, Leandro T.; LOPES, Heitor S. Classification of Weeds and Crops at the Pixel-Level Using Convolutional Neural Neural Networks and Data Augmentation . *In*: **Proceedings of 7th IEEE Latin American Conference on Computational Intelligence LA-CCI**. Piscataway, NJ: IEEE Press, 2019. p. 1–6.

BROCCHIERI, Luciano; KARLIN, Samuel. Protein length in eukaryotic and prokaryotic proteomes. **Nucleic Acids Research**, v. 33, n. 10, p. 3390–3400, 2005.

BROWN, Scott; FAWZI, Nicolas J; HEAD-GORDON, Teresa. Coarse-grained sequences for protein folding and design. **Proceedings of the National Academy of Sciences**, v. 100, n. 19, p. 10712–10717, 2003.

BYRNE, Michael P; MANUEL, R Lee; LOWE, Laura G; STITES, Wesley E. Energetic contribution of side chain hydrogen bonding to the stability of staphylococcal nuclease. **Biochemistry**, v. 34, n. 42, p. 13949–13960, 1995.

CASINO, P.; GOZALBO-ROVIRA, R.; RODRIGUEZ-DIAZ, J.; BANERJEE, S.; BOUTAUD, A.; RUBIO, V.; HUDSON, B. G.; SAUS, J.; CERVERA, J.; MARINA, A. Structures of collagen IV globular domains: insight into associated pathologies, folding and network assembly. **International Union of Crystallography Journal**, v. 5, n. 6, p. 765–779, 2018.

CHAN, Hue Sun; DILL, Ken A. Origins of structure in globular proteins. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 87, n. 16, p. 6388–6392, 1990.

CHEBARO, Yassmine; PASQUALI, Samuela; DERREUMAUX, Philippe. The coarse-grained OPEP force field for non-amyloid and amyloid proteins. **The Journal of Physical Chemistry B**, v. 116, n. 30, p. 8741–8752, 2012.

CHEN, Gang. A gentle tutorial of recurrent neural network with error backpropagation. ArXiv preprint arXiv:1610.02583. 2016.

CHENG, Bo; WU, Shaogui; LIU, Shixin; RODRIGUEZ-ALIAGA, Piere; YU, Jin; CUI, Shuxun. Protein denaturation at a single-molecule level: the effect of nonpolar environments and its

implications on the unfolding mechanism by proteases. **Nanoscale**, v. 7, n. 7, p. 2970–2977, 2015.

CHEON, Mookyung; CHANG, Iksoo; HALL, Carol K. Extending the PRIME model for protein aggregation to all 20 amino acids. **Proteins: Structure, Function, and Bioinformatics**, v. 78, n. 14, p. 2950–2960, 2010.

CHO, Kyunghyun; MERRIËNBOER, Bart Van; GULCEHRE, Caglar; BAHDANAU, Dzmitry; BOUGARES, Fethi; SCHWENK, Holger; BENGIO, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. ArXiv. 2014.

CIEPLAK, Marek; HOANG, Trinh Xuan. Scaling of folding properties in go models of proteins. **Journal of Biological Physics**, v. 26, n. 4, p. 273–294, 2000.

CLEMENTI, Cecilia; NYMEYER, Hugh; ONUCHIC, José Nelson. Topological and energetic factors: what determines the structural details of the transition state ensemble and "en-route" intermediates for protein folding? an investigation for small globular proteins1. **Journal of Molecular Biology**, v. 298, n. 5, p. 937–953, 2000.

CRICK, Francis H. On protein synthesis. **Society for Experimental Biology**, v. 12, n. 1, p. 138–163, 1958.

CUFF, James A.; CLAMP, Michele E.; SIDDIQUI, Asim S.; FINLAY, Matt; BARTON, Geoffrey J. JPred: a consensus secondary structure prediction server. **Bioinformatics**, v. 14, n. 10, p. 892–893, 1998.

DELLAGO, Christoph; BOLHUIS, Peter G; CSAJKA, Félix S; CHANDLER, David. Transition path sampling and the calculation of rate constants. **The Journal of Chemical Physics**, v. 108, n. 5, p. 1964–1977, 1998.

DEWITTE, R. S.; SHAKHNOVICH, E. I. Pseudodihedrals: simplified protein backbone representation with knowledge-based energy. **Protein Science**, v. 3, n. 9, p. 1570–1581, 1994.

DICKSON, Alex; BROOKS, Charles L. Wexplore: hierarchical exploration of high-dimensional spaces using the weighted ensemble algorithm. **The Journal of Physical Chemistry B**, v. 118, n. 13, p. 3532–3542, 2014.

DILL, A.; MacCallum, J. L. The protein-folding problem, 50 years on. **Science**, v. 338, n. 6110, p. 1042–1046, 2012.

DILL, Ken A. Dominant forces in protein folding. **Biochemistry**, v. 29, n. 31, p. 7133–7155, 1990.

DILL, Ken A. Polymer principles and protein folding. **Protein Science**, v. 8, n. 6, p. 1166–1180, 1999.

DILL, Ken A; BROMBERG, Sarina; YUE, Kaizhi; CHAN, Hue Sun; FTEBIG, Klaus M; YEE, David P; THOMAS, Paul D. Principles of protein folding–a perspective from simple exact models. **Protein Science**, v. 4, n. 4, p. 561–602, 1995.

DILL, Ken A; OZKAN, S Banu; SHELL, M Scott; WEIKL, Thomas R. The protein folding problem. **Annual Review of Biophysics**, v. 37, n. 1, p. 289–321, 2008.

DING, Chris HQ; DUBCHAK, Inna. Multi-class protein fold recognition using support vector machines and neural networks. **Bioinformatics**, v. 17, n. 4, p. 349–358, 2001.

DOBSON, Christopher M. Experimental investigation of protein folding and misfolding. **Methods**, v. 34, n. 1, p. 4–14, 2004.

DOR, Ofer; ZHOU, Yaoqi. Real-SPINE: An integrated system of neural networks for real-value prediction of protein structural properties. **Proteins: Structure, Function, and Bioinformatics**, v. 68, n. 1, p. 76–81, 2007.

DROZDETSKIY, Alexey; COLE, Christian; PROCTER, James; BARTON, Geoffrey J. JPred4: a protein secondary structure prediction server. **Nucleic Acids Research**, v. 43, n. 1, p. W389–W394, 2015.

DUCHI, John; HAZAN, Elad; SINGER, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. **Journal of Machine Learning Research**, v. 12, n. 5, p. 2121–2159, 2011.

EASTMAN, Peter; PANDE, Vijay S. Efficient nonbonded interactions for molecular dynamics on a graphics processing unit. **Journal of Computational Chemistry**, Wiley Online Library, v. 31, n. 6, p. 1268–1272, 2010.

EASTMAN, Peter; SWAILS, Jason; CHODERA, John D; MCGIBBON, Robert T; ZHAO, Yutong; BEAUCHAMP, Kyle A; WANG, Lee-Ping; SIMMONETT, Andrew C; HARRIGAN, Matthew P; STERN, Chaya D *et al.* OpenMM 7: rapid development of high performance algorithms for molecular dynamics. **PLoS Computational Biology**, v. 13, n. 7, p. e1005659–e1005676, 2017.

EISENBERG, D; SCHWARZ, E; KOMARONY, M; WALL, R. Amino acid scale: Normalized consensus hydrophobicity scale. **Journal of Molecular Biology**, v. 179, n. 1, p. 125–142, 1984.

ELMAN, Jeffrey L. Finding structure in time. **Cognitive Science**, v. 14, n. 2, p. 179–211, 1990.

ENGLANDER, S Walter. Protein folding intermediates and pathways studied by hydrogen exchange. **Annual Review of Biophysics and Biomolecular Structure**, Palo Alto, CA, USA, v. 29, n. 1, p. 213–238, 2000.

ERICSON, Christer. **Real-time collision detection**. Ohio, EUA: CRC Press, 2004.

ESSAID, Mokhtar; IDOUMGHAR, Lhassane; LEPAGNOT, Julien; BRéVILLIERS, Mathieu. Gpu parallelization strategies for metaheuristics: a survey. **International Journal of Parallel, Emergent and Distributed Systems**, v. 34, n. 5, p. 497–522, 2019.

FANG, C.; SHANG, Y.; XU, D. Prediction of Protein Backbone Torsion Angles Using Deep Residual Inception Neural Networks. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 15, n. 5, p. 1–11, 2018.

FARAGGI, Eshel; ZHANG, Tuo; YANG, Yuedong; KURGAN, Lukasz; ZHOU, Yaoqi. SPINE X: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. **Journal of Computational Chemistry**, v. 33, n. 3, p. 259–267, 2012.

FARRIS, Alfred; LI, Ying Wai; SEATON, Daniel; LANDAU, David. Monte Carlo Simulations of Coarse-grained Protein Models for Crambin. **Bulletin of the American Physical Society**, v. 27, n. 3, p. 1–6, 2018.

FERSHT, Alan. **Structure and mechanism in protein science: A Guide to enzyme catalysis and protein folding**. Singapore: World Scientific, 2017. v. 9. 656 p.

FINKELSTEIN, AV. 50+ years of protein folding. **Biochemistry (Moscow)**, v. 83, n. 1, p. S3–S18, 2018.

FLEISCHMANN, Nico; ADAMI, Stefan; ADAMS, Nikolaus A. Numerical symmetry-preserving techniques for low-dissipation shock-capturing schemes. **Computers & Fluids**, v. 189, n. 1, p. 94 – 107, 2019.

FLEMING, Noah; KINSELLA, Benjamin; ING, Christopher. Predicting Protein Thermostability Upon Mutation Using Molecular Dynamics Timeseries Data. BioRxiv. 2016.

FRIGORI, Rafael B. Breakout character of islet amyloid polypeptide hydrophobic mutations at the onset of type-2 diabetes. **Physical Review E**, v. 90, n. 5, p. 052716–052724, 2014.

FRIGORI, Rafael B. PHAST: Protein-like heteropolymer analysis by statistical thermodynamics. **Computer Physics Communications**, v. 215, p. 165 – 172, 2017.

FRIGORI, Rafael B; RIZZI, Leandro G; ALVES, Nelson A. Microcanonical thermostatistics of coarse-grained proteins with amyloidogenic propensity. **The Journal of Chemical Physics**, v. 138, n. 1, p. 015102, 2013.

GABARDO, A.C.; HATTORI, L. T.; GUTOSKI, M.; BERNO, B. C. S.; AGOSTINHO, W. R. U.; LOPES, H. S. **Como Mensurar a Importância, Influência e a Relevância de Usuários do Twitter? Uma análise da interação dos candidatos á presidência do Brasil nas eleições de 2018**. Curitiba, 2018. 1–4 p.

GALVÃO, L. C.; NUNES, L. F.; LOPES, H. S.; MOSCATO, P. A new greedy heuristic for 3DHP protein struture prediction with side chain. *In*: **IEEE International Conference on Bioinformatics and Biomedicine Workshops**. New York, NY, USA: IEEE, 2012. p. 77–81.

GANESAN, Aravindhan; COOTE, Michelle L; BARAKAT, Khaled. Molecular dynamics-driven drug discovery: leaping forward with confidence. **Drug Discovery Today**, v. 22, n. 2, p. 249–269, 2017.

GERS, Felix A; SCHMIDHUBER, Jürgen. Recurrent nets that time and count. *In*: **Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks**. New York, NY, USA: IEEE, 2000. p. 189–194.

GHANTY, Pradip; PAL, Nikhil R. Prediction of protein folds: extraction of new features, dimensionality reduction, and fusion of heterogeneous classifiers. **IEEE Transactions on Nanobioscience**, v. 8, n. 1, p. 100–110, 2009.

GIZOPOULOS, Dimitris; PAPADIMITRIOU, George; CHATZIDIMITRIOU, Athanasios; REDDI, Vijay Janapa; SALAMI, Behzad; UNSAL, Osman S; KESTELMAN, Adrian Cristal; LENG, Jingwen. Modern Hardware Margins: CPUs, GPUs, FPGAs Recent System-Level Studies. *In*: IEEE. **2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)**. Piscataway, NJ, 2019. p. 129–134.

GO, Nobuhiro; TAKETOMI, Hiroshi. Respective roles of short-and long-range interactions in protein folding. **Proceedings of the National Academy of Sciences**, v. 75, n. 2, p. 559–563, 1978.

GOPAL, Srinivasa M; MUKHERJEE, Shayantani; CHENG, Yi-Ming; FEIG, Michael. PRIMO/PRIMONA: A coarse-grained model for proteins and nucleic acids that preserves near-atomistic accuracy. **Proteins: Structure, Function, and Bioinformatics**, v. 78, n. 5, p. 1266–1281, 2010.

GOWER, J. C.; DIJKSTERHUIS, G. B. **Procrustes Problems**. Oxford, UK: Oxford University Press, 2004.

GRAVES, Alex; FERNÁNDEZ, Santiago; SCHMIDHUBER, Jürgen. Bidirectional LSTM networks for improved phoneme classification and recognition. *In*: SPRINGER. **International Conference on Artificial Neural Networks**. Heidelberg, 2005. p. 799–804.

GRAVES, Alex; WAYNE, Greg; DANIHELKA, Ivo. Neural Turing Machines. ArXiv. 2014.

GREFF, Klaus; SRIVASTAVA, Rupesh K; KOUTNÍK, Jan; STEUNEBRINK, Bas R; SCHMIDHUBER, Jürgen. LSTM: A search space odyssey. **IEEE Transactions on Neural Networks and Learning Systems**, v. 28, n. 10, p. 2222–2232, 2017.

GRONENBORN, A. M.; FILPULA, D. R.; ESSIG, N. Z.; ACHARI, A.; WHITLOW, M.; WINGFIELD, P. T.; CLORE, G. M. A novel, highly stable fold of the immunoglobulin binding domain of streptococcal protein G. **Science**, v. 253, n. 5020, p. 657–611, 1991.

GU, Junfeng; LI, Honglin; JIANG, Hualiang; WANG, Xicheng. A simple Cα-SC potential with higher accuracy for protein fold recognition. **Biochemical and Biophysical Research Communications**, v. 379, n. 2, p. 610–615, 2009.

GUO, Yudong; ZHANG, Juyong; CAI, Jianfei; JIANG, Boyi; ZHENG, Jianmin. CNN-based Real-time Dense Face Reconstruction with Inverse-rendered Photo-realistic Face Images. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 20, n. 4, p. 985–996, 2018.

GUSS, J. M.; BARTUNIK, H. D.; FREEMAN, H. C. Accuracy and precision in protein structure analysis: restrained least-squares refinement of the structure of poplar plastocyanin at 1.33 a resolution. **Acta Crystallographica**, B48, n. 1, p. 790–811, 1992.

GUTOSKI, Matheus; HATTORI, Leandro T.; AQUINO, Marcelo R.; LOPES, Heitor S. Feature Selection using Differential Evolution for Unsupervised Image Clustering. *In*: **Proceedings of 17th International Conference on Artificial Intelligence and Soft Computing**. Piscataway, NJ: IEEE Press, 2017. p. 1–6.

GUTOSKI, Matheus; HATTORI, Leandro T.; AQUINO, Marcelo R.; LOPES, Heitor S. Qualitative Analysis of Deep Learning Frameworks. **Learning and Nonlinear Models**, v. 15, n. 1, p. 1–8, 2018.

GUTOSKI, Matheus; RIBEIRO, Manasses; HATTORI, Leandro T; AQUINO, Marcelo Romero; LAZZARETTI, Andre E; LOPES, Heitor S. A comparative study of transfer learning approaches for video anomaly detection. **International Journal of Pattern Recognition and Artificial Intelligence**, World Scientific, 2020.

HANSON, Jack; PALIWAL, Kuldip; LITFIN, Thomas; YANG, Yuedong; ZHOU, Yaoqi. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional

long short-term memory with convolutional neural networks. **Bioinformatics**, v. 28, n. 3, p. 481–496, 2018.

HANSON, Jack; YANG, Yuedong; PALIWAL, Kuldip; ZHOU, Yaoqi. Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks. **Bioinformatics**, v. 33, n. 5, p. 685–692, 2017.

HATTORI, Leandro T.; BENÍTEZ, César M.V.; LOPES, Heitor S. A Novel Approach to Protein Folding Prediction based on Long Short-Term Memory Networks: A Preliminary Investigation and Analysis. *In*: **IEEE World Congress on Computational Intelligence (IEEE WCCI)**. Piscataway, NJ: IEEE Press, 2018. p. 1–6.

HATTORI, Leandro Takeshi; BENÍTEZ, César Manuel Vargas; LOPES, Heitor Silvério. A Deep Bidirectional Long Short-Term Memory Approach Applied to the Protein Secondary Structure Prediction Problem. *In*: **Proceedings of the 4th IEEE Latin American Conference on Computational Intelligence LA-CCI**. Piscataway, NJ: IEEE, 2017. p. 1–6.

HATTORI, Leandro T.; BENíTEZ, César M.V.; LOPES, Heitor S. A deep bidirectional long short-term memory approach applied to the protein secondary structure prediction problem. *In*: **Proc. 4th IEEE Latin American Conference on Computational Intelligence**. Piscataway, NJ: IEEE Press, 2017. p. 1–6.

HATTORI, Leandro T.; GUTOSKI, Matheus; AQUINO, Marcelo R.; LOPES, Heitor S. Patch-Based Convolutional Neural Network for the Writer Classification Problem in Music Score Images. *In*: **Proceedings of XIII Brazilian Congress on Computational Intelligence**. Piscataway, NJ: IEEE Press, 2017. p. 1–6.

HATTORI, Leandro Takeshi; GUTOSKI, Matheus; BENíTEZ], César Manuel [Vargas; NUNES, Luiz Fernando; LOPES, Heitor Silvério. A benchmark of optimally folded protein structures using integer programming and the 3D-HP-SC model. **Computational Biology and Chemistry**, v. 84, p. 107192, 2020.

HATTORI, Leandro Takeshi; PINHEIRO, Bruna Araujo; FRIGORI, Rafael Bertolini; BENíTEZ, César Manuel Vargas; LOPES, Heitor Silvério. Pathmold-ab: Spatiotemporal pathways of protein folding using parallel molecular dynamics with a coarse-grained model. **Computational Biology and Chemistry**, v. 87, p. 107301, 2020.

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. *In*: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. Piscataway, NJ: IEEE Press, 2016. p. 770–778.

HE, Yi; MOZOLEWSKA, Magdalena A; KRUPA, Paweł; SIERADZAN, Adam K; WIRECKI, Tomasz K; LIWO, Adam; KACHLISHVILI, Khatuna; RACKOVSKY, Shalom; JAGIEŁA,

Dawid; ŚLUSARZ, Rafał *et al.* Lessons from application of the UNRES force field to predictions of structures of CASP10 targets. **Proceedings of the National Academy of Sciences**, v. 110, n. 37, p. 14936–14941, 2013.

HECHT-NIELSEN, Robert. Theory of the backpropagation neural network. *In*: **Proocedings of International Joint Conference on Neural Networks**. Piscataway, NJ: IEEE Press, 1989. p. 65–93.

HEFFERNAN, Rhys; PALIWAL, Kuldip; LYONS, James; DEHZANGI, Abdollah; SHARMA, Alok; WANG, Jihua; SATTAR, Abdul; YANG, Yuedong; ZHOU, Yaoqi. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. **Scientific Reports**, v. 5, n. 3, p. 11476–11487, 2015.

HEFFERNAN, Rhys; YANG, Yuedong; PALIWAL, Kuldip; ZHOU, Yaoqi. Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. **Bioinformatics**, v. 33, n. 18, p. 2842–2849, 2017.

HENNESSY, John L; PATTERSON, David A. **Computer architecture: a quantitative approach**. New York, USA: Elsevier, 2011.

HESSA, Tara; KIM, Hyun; BIHLMAIER, Karl; LUNDIN, Carolina; BOEKEL, Jorrit; ANDERSSON, Helena; NILSSON, IngMarie; WHITE, Stephen H; von HEIJNE, Gunnar. Recognition of transmembrane helices by the endoplasmic reticulum translocon. **Nature**, v. 433, n. 7024, p. 377–381, 2005.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997.

HOPP, Thomas P; WOODS, Kenneth R. Prediction of protein antigenic determinants from amino acid sequences. **Proceedings of the National Academy of Sciences**, v. 78, n. 6, p. 3824–3828, 1981.

HOU, Jie; WU, Tianqi; CAO, Renzhi; CHENG, Jianlin. Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. BioRxiv. 2019.

HOWARD, Michael P; STATT, Antonia; MADUTSA, Felix; TRUSKETT, Thomas M; PANAGIOTOPOULOS, Athanassios Z. Quantized bounding volume hierarchies for neighbor search in molecular simulations on graphics processing units. **Computational Materials Science**, v. 164, p. 139–146, 2019.

HOWARD, Michael P.; STATT, Antonia; MADUTSA, Felix; TRUSKETT, Thomas M.; PANAGIOTOPOULOS, Athanassios Z. Quantized bounding volume hierarchies for neighbor

search in molecular simulations on graphics processing units. **Computational Materials Science**, v. 164, p. 139 – 146, 2019.

HSU, Hsiao-Ping; MEHRA, Vishal; GRASSBERGER, Peter. Structure optimization in an off-lattice protein model. **Physical Review E**, v. 68, n. 3, p. 037703, 2003.

IAKYMCHUK, Roman; BARREDA, Maria; WIESENBERGER, Matthias; ALIAGA, José I; QUINTANA-ORTÍ, Enrique S. Reproducibility strategies for parallel preconditioned conjugate gradient. **Journal of Computational and Applied Mathematics**, v. 371, n. 1, p. 112697, 2020.

INACIO, A. S.; HATTORI, L. T.; GUTOSKI, M.; LAZZARETTI, A. E.; LOPES, H. S. **Análise de espectros político-ideológicos dos partidos políticos utilizando métodos de agrupamento**. Curitiba, 2018. 1–4 p.

IRBÄCK, Anders; PETERSON, Carsten; POTTHAST, Frańk; SOMMELIUS, Ola. Local interactions and protein folding: A three-dimensional off-lattice approach. **The Journal of Chemical Physics**, v. 107, n. 1, p. 273–282, 1997.

IZVEKOV, Sergei; VOTH, Gregory A. A multiscale coarse-graining method for biomolecular systems. **The Journal of Physical Chemistry B**, v. 109, n. 7, p. 2469–2473, 2005.

JANIN, JOEL. Surface and inside volumes in globular proteins. **Nature**, v. 277, n. 5696, p. 491–492, 1979.

JONES, David T. Protein secondary structure prediction based on position-specific scoring matrices. **Journal of Molecular Biology**, v. 292, n. 2, p. 195–202, 1999.

JONES, David T; SINGH, Tanya; KOSCIOLEK, Tomasz; TETCHNER, Stuart. MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. **Bioinformatics**, v. 31, n. 7, p. 999–1006, 2014.

JONES, D. T.; TAYLOR, W. R.; THORNTON, J. M. A new approach to protein fold recognition. **Nature**, v. 358, n. 6381, p. 86–89, Jul 1992.

JORDAN, Michael I. Proceedings of Artificial Neural Networks. *In*: **Attractor Dynamics and Parallelism in a Connectionist Sequential Machine**. Piscataway, NJ: IEEE Press, 1990. p. 112–127.

JOZEFOWICZ, Rafal; ZAREMBA, Wojciech; SUTSKEVER, Ilya. An empirical exploration of recurrent network architectures. *In*: **Proceedings of International Conference on Machine Learning**. New York, USA: Proceedings of International Machine Learning Society (IMLS), 2015. p. 2342–2350.

KABSCH, W. A solution for the best rotation to relate two sets of vectors. **Acta Crystallographica**, A32, n. 5, p. 922–923, 1976.

KABSCH, Wolfgang. A discussion of the solution for the best rotation to relate two sets of vectors. **Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography**, v. 34, n. 5, p. 827–828, 1978.

KABSCH, Wolfgang; SANDER, Christian. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. **Biopolymers: Original Research on Biomolecules**, v. 22, n. 12, p. 2577–2637, 1983.

KALINOWSKA, Barbara; BANACH, Mateusz; WISNIOWSKI, Zdzislaw; KONIECZNY, Leszek; ROTERMAN, Irena. Is the hydrophobic core a universal structural element in proteins? **Journal of Molecular Modeling**, v. 23, n. 7, p. 205–205, 2017.

KARANICOLAS, John; BROOKS, Charles L. The origins of asymmetry in the folding transition states of protein L and protein G. **Protein Science**, v. 11, n. 10, p. 2351–2361, 2002.

KARCZYŃSKA, Agnieszka S; CZAPLEWSKI, Cezary; KRUPA, Paweł; MOZOLEWSKA, Magdalena A; JOO, Keehyoung; LEE, Jooyoung; LIWO, Adam. Ergodicity and model quality in template-restrained canonical and temperature/Hamiltonian replica exchange coarse-grained molecular dynamics simulations of proteins. **Journal of Computational Chemistry**, v. 38, n. 31, p. 2730–2746, 2017.

KARPLUS, Martin. The Levinthal paradox: yesterday and today. **Folding and Design**, v. 2, n. 4, p. 69–75, 1992.

KARPLUS, Martin; McCAMMON, J Andrew. Molecular dynamics simulations of biomolecules. **Nature Structural & Molecular Biology**, v. 9, n. 9, p. 646–652, 2002.

KAUSHIK, Aman Chandra; SAHI, Shakti. Biological complexity: ant colony meta-heuristic optimization algorithm for protein folding. **Neural Computing and Applications**, v. 28, n. 11, p. 3385–3391, 2017.

KHOKHLOV, Alexei R. **Statistical Physics of Macromolecules**. 1. ed. NY, USA: American Institute of Physics, 1994.

KIM, Jaegil; STRAUB, John E; KEYES, Tom. Replica exchange statistical temperature molecular dynamics algorithm. **The Journal of Physical Chemistry B**, ACS Publications, v. 116, n. 29, p. 8646–8653, 2012.

KINGMA, Diederik; BA, Jimmy. Adam: A Method for Stochastic Optimization. ArXiv. 2014.

KMIECIK, Sebastian; GRONT, Dominik; KOLINSKI, Michal; WIETESKA, Lukasz; DAWID, Aleksandra Elzbieta; KOLINSKI, Andrzej. Coarse-Grained Protein Models and Their Applications. **Chemical Reviews**, v. 116, n. 14, p. 7898–7936, 2016.

KMIECIK, Sebastian; WABIK, Jacek; KOLINSKI, Michal; KOUZA, Maksim; KOLINSKI, Andrzej. Protein dynamics simulations using coarse-grained models. *In*: **Computational Methods to Study the Structure and Dynamics of Biomolecules and Biomolecular Processes**. Heidelberg: Springer, 2019. p. 61–87.

KOLINSKI, Andrzej. Lattice polymers and protein models. **Multiscale Approaches to Protein Modeling: Structure Prediction, Dynamics, Thermodynamics and Macromolecular Assemblies**, v. 87, n. 16, p. 1–20, 2011.

KOLIŃSKI, Andrzej; SKOLNICK, Jeffrey. Assembly of protein structure from sparse experimental data: an efficient Monte Carlo model. **Proteins**, v. 32, n. 4, p. 475–494, 1998.

KRAVRAKI, E.L. **Geometric Methods in Structural Computational Biology**. 1th. ed. Texas, USA: Connexions, 2007.

KRYSHTAFOVYCH, Andriy; MONASTYRSKYY, Bohdan; FIDELIS, Krzysztof. Casp11 statistics and the prediction center evaluation system. **Proteins: Structure, Function, and Bioinformatics**, v. 84, n. S1, p. 15–19, 2016.

KYTE, Jack; DOOLITTLE, Russell F. A simple method for displaying the hydropathic character of a protein. **Journal of Molecular Biology**, v. 157, n. 1, p. 105–132, 1982.

LACAPÈRE, Jean-Jacques; PEBAY-PEYROULA, Eva; NEUMANN, Jean-Michel; ETCHEBEST, Catherine. Determining membrane protein structures: still a challenge! **Trends in Biochemical Sciences**, v. 32, n. 6, p. 259–270, 2018.

LAIO, Alessandro; PARRINELLO, Michele. Escaping free-energy minima. **Proceedings of the National Academy of Sciences**, v. 99, n. 20, p. 12562–12566, 2002.

LANG, Kevin J; WAIBEL, Alex H; HINTON, Geoffrey E. A time-delay neural network architecture for isolated word recognition. **Neural Networks**, v. 3, n. 1, p. 23–43, 1990.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, 2015.

LEE, Jooyoung; FREDDOLINO, Peter L; ZHANG, Yang. Ab initio protein structure prediction. *In*: **From Protein Structure to Function with Bioinformatics**. Heidelberg: Springer, 2017. p. 3–35.

LEHNINGER, AL; NELSON, DI; COX, MM. **Princípios de Bioquímica.** 3. ed. São Paulo: Roca, 2008. v. 1. 248 p.

LENA, Pietro Di; NAGATA, Ken; BALDI, Pierre. Deep architectures for protein contact map prediction. **Bioinformatics**, v. 28, n. 19, p. 2449–2457, 2012.

LENA, Pietro D.; NAGATA, Ken; BALDI, Pierre F. Deep Spatio-Temporal Architectures and Learning for Protein Structure Prediction. *In*: **Advances in Neural Information Processing Systems 25**. La Jolla, CA: Neural Information Processing Systems, 2012. p. 521–529.

LEVINTHAL, Cyrus. Are there pathways for protein folding? **Journal de Chimie Physique**, v. 65, n. 1, p. 44–45, 1968.

LEVITT, Michael. Protein folding by restrained energy minimization and molecular dynamics. **Journal of Molecular Biology**, v. 170, n. 3, p. 723–764, 1983.

LEWIS, Peter N; MOMANY, Frank A; SCHERAGA, Harold A. Chain reversals in proteins. **Biochimica et Biophysica Acta**, v. 303, n. 2, p. 211–229, 1973.

LI, B.; LIN, M.; LIU, Q.; LI, Y.; ZHOU, C. Protein folding optimization based on 3D off-lattice model via an improved artificial bee colony algorithm. **Journal Molecular Model**, v. 21, n. 10, p. 261–269, 2015.

LI, Haiou; HOU, Jie; ADHIKARI, Badri; LYU, Qiang; CHENG, Jianlin. Deep learning methods for protein torsion angle prediction. **BMC Bioinformatics**, v. 18, n. 1, p. 417–430, 2017.

LI, Jianan; LIANG, Xiaodan; SHEN, ShengMei; XU, Tingfa; FENG, Jiashi; YAN, Shuicheng. Scale-aware fast R-CNN for pedestrian detection. **IEEE Transactions on Multimedia**, v. 20, n. 4, p. 985–996, 2018.

LI, Min; LI, Wenkai; ZHENG, Ruiqing; LI, Xingyi; ZENG, Min. Network-based methods for predicting essential genes or proteins: a survey. bbz017, n. 1, 2019.

LI, Shumin; CHEN, Junjie; LIU, Bin. Protein remote homology detection based on bidirectional long short-term memory. **BMC Bioinformatics**, v. 18, n. 1, p. 443–451, 2017.

LI, Ting; ZHOU, Changjun; HU, Mandong. An Improved Artificial Bee Colony Algorithm for 3D Protein Structure Prediction. *In*: **Proceedings of the International Conference on Biometrics Engineering and Application**. New York, NY, USA: ACM, 2017. (ICBEA '17), p. 7–12.

LI, YiFei; CAO, Han. Prediction for Tourism Flow based on LSTM Neural Network. **Procedia Computer Science**, v. 129, n. 1, p. 277 – 283, 2018.

LI, Zhenqin; SCHERAGA, Harold A. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 84, n. 19, p. 6611–6615, 1987.

LI, Zhen; YU, Yizhou. Protein Secondary Structure Prediction Using Cascaded Convolutional and Recurrent Neural Networks. *In*: **Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence**. New York, USA: AAAI Press, 2016. p. 2560–2567.

LIN, Juan; ZHONG, Yiwen; LI, Ena; LIN, Xiaoyu; ZHANG, Hui. Multi-agent simulated annealing algorithm with parallel adaptive multiple sampling for protein structure prediction in ab off-lattice model. **Applied Soft Computing**, v. 62, p. 491–503, 2018.

Lindholm, E.; Nickolls, J.; Oberman, S.; Montrym, J. NVIDIA Tesla: A Unified Graphics and Computing Architecture. **IEEE Micro**, v. 28, n. 2, p. 39–55, 2008.

LIPTON, Zachary C; BERKOWITZ, John; ELKAN, Charles. A Critical Review of Recurrent Neural Networks for Sequence Learning. Arxiv. 2015.

LIU, Bin; LI, Chen-Chen; YAN, Ke. DeepSVM-fold: protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks. **Briefings in Bioinformatics**, v. 21, n. 5, p. 1733–1741, 2019.

LIU, B.; LI, S. ProtDet-CCH: Protein remote homology detection by combining Long Short-Term Memory and ranking methods. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 15, n. 1, p. 1–15, 2018.

LIU, Xueliang. Deep Recurrent Neural Network for Protein Function Prediction from Sequence. Arxiv. 2017.

LIWO, Adam; KHALILI, Mey; SCHERAGA, Harold A. 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 of the United States of America**, v. 102, n. 7, p. 2362–2367, 2005.

LIZAK, Christian; WORRALL, Liam J; BAUMANN, Lars; PFLEIDERER, Moritz M; VOLKERS, Gesa; SUN, Tianjun; SIM, Lyann; WAKARCHUK, Warren; WITHERS, Stephen G; STRYNADKA, Natalie CJ. X-ray crystallographic structure of a bacterial polysialyltransferase provides insight into the biosynthesis of capsular polysialic acid. **Scientific Reports**, v. 7, n. 1, p. 5842–5855, 2017.

LOPES, Heitor S. Evolutionary algorithms for the protein folding problem: A review and current trends. *In*: **Proceedings of Computational Intelligence in Biomedicine and Bioinformatics**. Heidelberg: Springer, 2008. p. 297–315.

MAGNAN, Christophe N; BALDI, Pierre. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. **Bioinformatics**, v. 30, n. 18, p. 2592–2597, 2014.

MALDONADO-NAVA, Fanny G; FRAUSTO-SOLÍS, Juan; SÁNCHEZ-HERNÁNDEZ, Juan Paulo; BARBOSA, Juan Javier González; LIÑÁN-GARCÍA, Ernesto. Comparative Study of Computational Strategies for Protein Structure Prediction. *In*: **Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications**. Heidelberg: Springer, 2018. p. 449–459.

MARAGLIANO, Luca; FISCHER, Alexander; VANDEN-EIJNDEN, Eric; CICCOTTI, Giovanni. String method in collective variables: Minimum free energy paths and isocommittor surfaces. **The Journal of Chemical Physics**, v. 125, n. 2, p. 24106–24122, 2006.

MARGARIT, Horia; SUBRAMANIAM, Raghav. A batch-normalized recurrent network for sentiment classification. *In*: **Advances in Neural Information Processing Systems**. CA, USA: Neural Information Processing Systems, 2016. p. 1–7.

MARTOS, Gabriel; MUÑOZ, Alberto; GONZÁLEZ, Javier. On the generalization of the mahalanobis distance. *In*: **Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 125–132.

MAUPETIT, Julien; TUFFERY, P; DERREUMAUX, Philippe. A coarse-grained protein force field for folding and structure prediction. **Proteins: Structure, Function, and Bioinformatics**, v. 69, n. 2, p. 394–408, 2007.

McCAMMON, J Andrew; GELIN, Bruce R; KARPLUS, Martin. Dynamics of folded proteins. **Nature**, v. 267, n. 5612, p. 585–590, 1977.

MCCULLOCH, Warren S; PITTS, Walter. Neurocomputing: foundations of research. **A Logical Calculus of the Ideas Immanent in Nervous Activity**, v. 28, n. 1, p. 15–27, 1988.

MCPHERSON, Alexander; GAVIRA, Jose A. Introduction to protein crystallization. **Acta Crystallographica Section F: Structural Biology Communications**, v. 70, n. 1, p. 2–20, 2014.

MERMELSTEIN, Daniel J.; LIN, Charles; NELSON, Gard; KRETSCH, Rachael; McCAMMON, J. Andrew; WALKER, Ross C. Fast and flexible gpu accelerated binding free energy calculations within the amber molecular dynamics package. **Journal of Computational Chemistry**, v. 39, n. 19, p. 1354–1358, 2018.

MIAO, Yinglong; FEIXAS, Ferran; EUN, Changsun; McCAMMON, J Andrew. Accelerated molecular dynamics simulations of protein folding. **Journal of Computational Chemistry**, v. 36, n. 20, p. 1536–1549, 2015.

MICHELARAKIS, Nicholas; SANDS, Zara A; SANSOM, Mark SP; STANSFELD, Phillip J. Towards Dynamic Pharmacophore Models by Coarse Grained Molecular Dynamics. **Biophysical Journal**, v. 114, n. 3, p. 558–572, 2018.

MICHELARAKIS, Nicholas; SANDS, Zara A.; SANSOM, Mark S. P.; STANSFELD, Phillip J. Towards Dynamic Pharmacophore Models by Coarse Grained Molecular Dynamics. **Biophysical Journal**, v. 114, n. 3, p. 558a, 2018.

MIKOLOV, Tomas; CHEN, Kai; CORRADO, Greg; DEAN, Jeffrey. Efficient estimation of word representations in vector space. ArXiv. 2013.

MIN, Seonwoo; LEE, Byunghan; YOON, Sungroh. Deep learning in bioinformatics. **Briefings in Bioinformatics**, v. 18, n. 5, p. 851–869, 2017.

MIRABELLO, Claudio; POLLASTRI, Gianluca. Porter, paleale 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. **Bioinformatics**, v. 29, n. 16, p. 2056–2058, 2013.

MIRNICS, K.; MIDDLETON, F.A.; STANWOOD, G.D.; LEWIS, D.A.; LEVITT, P. Disease-specific changes in regulator of G-protein signaling 4 (RGS4) expression in schizophrenia. **Molecular Psychiatry**, v. 6, n. 3, p. 293–301, 2001.

MONTICELLI, Luca; KANDASAMY, Senthil K; PERIOLE, Xavier; LARSON, Ronald G; TIELEMAN, D Peter; MARRINK, Siewert-Jan. The MARTINI coarse-grained force field: extension to proteins. **Journal of Chemical Theory and Computation**, v. 4, n. 5, p. 819–834, 2008.

MOREL, Francois MM; HERING, Janet G. **Principles and Applications of Aquatic Chemistry**. 1. ed. Nova Jersey: Wiley-Interscience, 1993. v. 1. 608 p.

MORRISS-ANDREWS, Alex; SHEA, Joan-Emma. Computational studies of protein aggregation: methods and applications. **Annual Review of Physical Chemistry**, v. 66, n. 3, p. 643–666, 2015.

NG, Andrew Y. Feature selection, L1 vs L2 regularization, and rotational invariance. *In*: **Proceedings of the Twenty-First International Conference on Machine learning**. New York, NY, USA: ACM, 2004. p. 1–8.

NÖLTING, Bengt. **Methods in modern biophysics**. 2. ed. Heidelberg: Springer, 2006. v. 3. 257 p.

NUNES, Luiz Fernando; GALVÃO, Lauro Cesar; LOPES, Heitor Silvério; MOSCATO, Pablo; BERRETTA, Regina. An integer programming model for protein structure prediction using the 3D–HP side chain model. **Discrete Applied Mathematics**, v. 198, n. 1, p. 206–214, 2016.

ONOFRIO, Angelo; PARISI, Giovanni; PUNZI, Giuseppe; TODISCO, Simona; NOIA, Maria Antonietta Di; BOSSIS, Fabrizio; TURI, Antonio; GRASSI, Anna De; PIERRI, Ciro Leonardo. Distance-dependent hydrophobic–hydrophobic contacts in protein folding simulations. **Physical Chemistry Chemical Physics**, v. 16, n. 35, p. 18907–18917, 2014.

ORENGO, C.; JONES, D.; THORNTON, J. **Bioinformatics: Genes, Proteins and Computers**. Oxon: Taylor & Francis, 2003. 320 p.

PALIWAL, Kuldip; LYONS, James; HEFFERNAN, Rhys. A Short Review of Deep Learning Neural Networks in Protein Structure Prediction Problems. **Advanced Techniques in Biology and Medicine**, v. 5, n. 139, p. 1–2, 2015.

PARPINELLI, Rafael Stubs; LOPES, Heitor Silvério. A computational ecosystem for optimization: review and perspectives for future research. **Memetic Computing**, v. 7, n. 1, p. 29–41, 2015.

PAULING, Linus; COREY, Robert B. Configurations of polypeptide chains with favored orientations around single bonds two new pleated sheets. **Proceedings of the National Academy of Sciences**, v. 37, n. 11, p. 729–740, 1951.

PAULING, Linus; COREY, Robert B; BRANSON, Herman R. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. **Proceedings of the National Academy of Sciences**, v. 37, n. 4, p. 205–211, 1951.

PEDERSEN, C. **Algorithms in Computational Biology**. 2000. Phd Thesis (PhD Thesis) — University of Aarhus – Departament of Computer Science, Denmark, 2000.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher. Glove: Global vectors for word representation. *In*: **Proceedings of the Conference on Empirical Methods in Natural Language Processing**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014. p. 1532–1543.

PENNYCOOK, S. J.; HUGHES, C. J.; SMELYANSKIY, M.; JARVIS, S. A. Exploring simd for molecular dynamics, using intel xeon processors and intel xeon phi coprocessors. *In*: **2013 IEEE 27th International Symposium on Parallel and Distributed Processing**. Piscataway, NJ: IEEE Press, 2013. v. 1, n. 1, p. 1085–1097.

PEREZ, Alberto; MORRONE, Joseph A; SIMMERLING, Carlos; DILL, Ken A. Advances in free-energy-based simulations of protein folding and ligand binding. **Current Opinion in Structural Biology**, v. 36, n. 3, p. 25–31, 2016.

PHILLIPS, Carolyn L.; ANDERSON, Joshua A.; GLOTZER, Sharon C. Pseudo-random number generation for brownian dynamics and dissipative particle dynamics simulations on gpu devices. **Journal of Computational Physics**, v. 230, n. 19, p. 7191 – 7201, 2011.

PIERRI, Ciro Leonardo; GRASSI, Anna De; TURI, Antonio. Lattices for ab initio protein structure prediction. **Proteins: Structure, Function, and Bioinformatics**, Wiley Online Library, v. 73, n. 2, p. 351–361, 2008.

POBLETE, Simón; BOTTARO, Sandro; BUSSI, Giovanni. A nucleobase-centered coarse-grained representation for structure prediction of RNA motifs. **Nucleic Acids Research**, v. 46, n. 4, p. 1674–1683, 2018.

QIAN, Ning. On the momentum term in gradient descent learning algorithms. **Neural Networks**, v. 12, n. 1, p. 145–151, 1999.

RAMACHANDRAN, Gopalasamudram Narayana; RAMAKRISHNAN, Chandrasekharan; SASISEKHARAN, V. Stereochemistry of polypeptide chain configurations. **Journal of Molecular Biology**, v. 7, n. 1, p. 95–99, 1963.

RAPAPORT, Dennis C. **The Art of Molecular Dynamics Simulation**. 2. ed. Cambridge: Cambridge University Press, 2004. v. 1. 564 p.

REITH, Dirk; PÜTZ, Mathias; MÜLLER-PLATHE, Florian. Deriving effective mesoscale potentials from atomistic simulations. **Journal of Computational Chemistry**, v. 24, n. 13, p. 1624–1636, 2003.

ROBINSON, A. J.; FALLSIDE, Frank. **The Utility Driven Dynamic Error Propagation Network**. Cambridge, UK, 1987. Technical reports.

ROGAWSKI, Rivkah; MCDERMOTT, Ann. New NMR tools for protein structure and function: Spin tags for dynamic nuclear polarization solid state NMR. **Archives of Biochemistry and Biophysics**, v. 628, n. 3, p. 102–113, 2017.

ROJAS, Ana; LIWO, Adam; BROWNE, Dana; SCHERAGA, Harold A. Mechanism of fiber assembly: treatment of aβ peptide aggregation with a coarse-grained united-residue force field. **Journal of Molecular Biology**, v. 404, n. 3, p. 537–552, 2010.

ROLLINS, Geoffrey C.; DILL, Ken A. General Mechanism of Two-State Protein Folding Kinetics. **Journal of the American Chemical Society**, v. 136, n. 32, p. 11420–11427, 2014.

RUMELHART, David E; HINTON, Geoffrey E; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533, 1986.

RYCKBOSCH, Steven M; WENDER, Paul A; PANDE, Vijay S. Molecular dynamics simulations reveal ligand-controlled positioning of a peripheral protein complex in membranes. **Nature Communications**, v. 8, n. 1, p. 6–16, 2017.

SALOMON-FERRER, Romelia; CASE, David A; WALKER, Ross C. An overview of the Amber biomolecular simulation package. **WIREs Computational Molecular Science**, v. 3, n. 2, p. 198–210, 2013.

SCHAARSCHMIDT, Joerg; MONASTYRSKYY, Bohdan; KRYSHTAFOVYCH, Andriy; BONVIN, Alexandre M.J.J. Assessment of contact predictions in casp12: Co-evolution and deep learning coming of age. **Proteins: Structure, Function, and Bioinformatics**, v. 86, n. S1, p. 51–66, 2018.

SCHNEIRLA, TC; ROSENBLATT, Jay S; TOBACH, Ethel. Maternal behavior in the cat. **Maternal Behavior in Mammals**, v. 122, n. 43, p. 168, 1963.

SEEMAYER, Stefan; GRUBER, Markus; SÖDING, Johannes. CCMpred – fast and precise prediction of protein residue–residue contacts from correlated mutations. **Bioinformatics**, v. 30, n. 21, p. 3128–3130, 2014.

SHARMA, Alok; PALIWAL, Kuldip K; DEHZANGI, Abdollah; LYONS, James; IMOTO, Seiya; MIYANO, Satoru. A strategy to select suitable physicochemical attributes of amino acids for protein fold recognition. **BMC Bioinformatics**, v. 14, n. 1, p. 233–244, 2013.

SHELL, M Scott. The relative entropy is fundamental to multiscale and inverse thermodynamic problems. **The Journal of Chemical Physics**, v. 129, n. 14, p. 144108–144115, 2008.

SHEN, Zhen; ZHANG, Qinhu; HAN, Kyungsook; HUANG, De-shuang. A deep learning model for rna-protein binding preference prediction based on hierarchical lstm and attention network. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE, v. 1868, n. 11, 2020.

SIERADZAN, Adam K; LIWO, Adam; HANSMANN, Ulrich HE. Folding and self-assembly of a small protein complex. **Journal of Chemical Theory and Computation**, v. 8, n. 9, p. 3416–3422, 2012.

SKOLNICK, Jeffrey; KOLINSKI, Andrzej. Dynamic Monte Carlo simulations of a new lattice model of globular protein folding, structure and dynamics. **Journal of Molecular Biology**, Elsevier, v. 221, n. 2, p. 499–531, 1991.

SKOLNICK, Jeffrey; KOLINSKI, Andrzej; YARIS, Robert. Monte Carlo simulations of the folding of beta-barrel globular proteins. **Proceedings of the National Academy of Sciences**, v. 85, n. 14, p. 5057–5061, 1988.

SøNDERBY, Søren Kaae; WINTHER, Ole. Protein Secondary Structure Prediction with Long Short Term Memory Networks. ArXiv. 2015.

SONG, J.; TAN, H.; WANG, M.; WEBB, G. I.; AKUTSU, T. TANGLE: two-level support vector regression approach for protein backbone torsion angle prediction from primary sequences. **PLoS ONE**, v. 7, n. 2, p. e30361, 2012.

SPELLINGS, Matthew; MARSON, Ryan L.; ANDERSON, Joshua A.; GLOTZER, Sharon C. Gpu accelerated discrete element method (dem) molecular dynamics for conservative, faceted particle simulations. **Journal of Computational Physics**, v. 334, p. 460 – 467, 2017.

SPENCER, Matt; EICKHOLT, Jesse; CHENG, Jianlin. A deep learning network approach to ab initio protein secondary structure prediction. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 12, n. 1, p. 103–112, 2015.

SRIVASTAVA, Nitish; HINTON, Geoffrey E; KRIZHEVSKY, Alex; SUTSKEVER, Ilya; SALAKHUTDINOV, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, 2014.

Staples, M.; Chan, L.; Si, D.; Johnson, K.; Whyte, C.; Cao, R. Artificial intelligence for bioinformatics: Applications in protein folding prediction. *In*: **2019 IEEE Technology Engineering Management Conference (TEMSCON)**. Piscataway, NJ: IEEE Press, 2019. v. 14, n. 1, p. 1–8.

STILLINGER, Frank H; HEAD-GORDON, Teresa. Collective aspects of protein folding illustrated by a toy model. **Physical Review E**, v. 52, n. 3, p. 2872, 1995.

STONE, John E.; PERILLA, Juan R.; CASSIDY, C. Keith; SCHULTEN, Klaus. Gpu-accelerated molecular dynamics clustering analysis with openacc. *In*: **Proceedings of Parallel Programming with OpenACC**. Boston: Morgan Kaufmann, 2017. p. 215 – 240.

SUGITA, Yuji; KITAO, Akio; OKAMOTO, Yuko. Multidimensional replica-exchange method for free-energy calculations. **The Journal of Chemical Physics**, v. 113, n. 15, p. 6042–6051, 2000.

SUGITA, Yuji; OKAMOTO, Yuko. Replica-exchange molecular dynamics method for protein folding. **Chemical Physics Letters**, v. 314, n. 2, p. 141–151, 1999.

SUKHBAATAR, Sainbayar; WESTON, Jason; FERGUS, Rob. End-to-end memory networks. *In*: **Advances in Neural Information Processing Systems**. CA, USA: Neural Information Processing Systems, 2015. p. 2440–2448.

SUN, Daiwen; GONG, Xinqi. Tetramer protein complex interface residue pairs prediction with lstm combined with graph representations. **Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics**, Elsevier, v. 1868, n. 11, p. 140504, 2020.

SWENDSEN, Robert H; WANG, JIAN-SHENG. Replica Monte Carlo simulation of spin-glasses. **Physical Review Letters**, v. 57, n. 21, p. 2607, 1986.

SWOPE, William C; PITERA, Jed W; SUITS, Frank. Describing protein folding kinetics by molecular dynamics simulations. **The Journal of Physical Chemistry B**, v. 108, n. 21, p. 6571–6581, 2004.

SZEGEDY, Christian; IOFFE, Sergey; VANHOUCKE, Vincent; ALEMI, Alexander A. Inception-v4, inception-resnet and teh impact of residual connections on learning. *In*: **Proceedings of the Thirty-First Conference on Artificial Intelligence**. Palo Alto, California: AAAI Press, 2017. p. 1–12.

TAKIGUCHI, Lia A.; HATTORI, Leandro T.; BENÍTEZ, César M. V.; LOPES, Heitor S. Dinâmica Molecular para Predição de Pathways de Proteínas usando Neighbor Lists e o Modelo 3D-AB. *In*: **Seminário de Iniciação Científica e Tecnológica da UTFPR (SICITE)**. Curitiba: UTFPR, 2017. p. 1–4.

TANGHERLONI, Andrea; RUNDO, Leonardo; SPOLAOR, Simone; CAZZANIGA, Paolo; NOBILE, Marco S. GPU-powered multi-swarm parameter estimation of biological systems: A master-slave approach. *In*: IEEE. **2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)**. Piscataway, NJ, 2018. p. 698–705.

TIAN, Liqing; WU, Aiping; CAO, Yang; DONG, Xiaoxi; HU, Yun; JIANG, Taijiao. NCACO-score: An effective main-chain dependent scoring function for structure modeling. **BMC Bioinformatics**, v. 12, n. 1, p. 208, 2011.

TIELEMAN, T.; HINTON, G. Neural Networks for Machine Learning, Lecture 6.5 – rmsprop. Coursera. 2012.

TIESSEN, Axel; PÉREZ-RODRÍGUEZ, Paulino; DELAYE-ARREDONDO, Luis José. 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 Research Notes**, v. 5, n. 1, p. 85, 2012.

TIRION, Monique M. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. **Physical Review Letters**, v. 77, n. 9, p. 1905, 1996.

TOLSTRUP, N; TOFTGÅRD, J; ENGELBRECHT, J; BRUNAK, S. Neural network model of the genetic code is strongly correlated to the ges scale of amino acid transfer free energies. **Journal of Molecular Biology**, Elsevier, v. 243, n. 5, p. 816–820, 1994.

TORRIE, Glenn M; VALLEAU, John P. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. **Journal of Computational Physics**, v. 23, n. 2, p. 187–199, 1977.

TOZZINI, Valentina. Coarse-grained models for proteins. **Current Opinion in Structural Biology**, v. 15, n. 2, p. 144–150, 2005.

TOZZINI, Valentina; ROCCHIA, Walter; McCAMMON, J Andrew. Mapping all-atom models onto one-bead coarse-grained models: general properties and applications to a minimal polypeptide model. **Journal of Chemical Theory and Computation**, v. 2, n. 3, p. 667–673, 2006.

TSUBAKI, Masashi; SHIMBO, Masashi; MATSUMOTO, Yuji. Protein Fold Recognition with Representation Learning and Long Short-Term Memory. **Information Processing Society of Japan Transactions on Bioinformatics**, v. 10, n. 5, p. 2–8, 2017.

UEDA, Yuzo; TAKETOMI, Hiroshi; GŌ, Nobuhiro. Studies on protein folding, unfolding, and fluctuations by computer simulation. **Biopolymers**, v. 17, n. 6, p. 1531–1548, 1978.

VENDRUSCOLO, M; PACI, E; DOBSON, CM; KARPLUS, M. Three key residuees form a critical contact network in a transition state for protein folding. **Nature**, v. 409, n. 1, p. 641–645, 2001.

Villegas-Morcillo, A.; Gomez, A. M.; Morales Cordovilla, J. A.; Sanchez Calle, V. E. Protein fold recognition from sequences using convolutional and recurrent neural networks. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE Press, Piscataway, NJ, v. 14, n. 1, p. 1–8, 2020.

VIVO, Marco De; MASETTI, Matteo; BOTTEGONI, Giovanni; CAVALLI, Andrea. Role of molecular dynamics and related methods in drug discovery. **Journal of Medicinal Chemistry**, v. 59, n. 9, p. 4035–4061, 2016.

WANG, Sheng; PENG, Jian; MA, Jianzhu; XU, Jinbo. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. **Scientific Reports**, v. 6, n. 1, p. 1–11, 2016.

WANG, Sheng; SUN, Siqi; LI, Zhen; ZHANG, Renyu; XU, Jinbo. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. **PLOS Computational Biology**, Public Library of Science, v. 13, n. 1, p. 1–34, 2017.

WANG, Zhiyong; ZHAO, Feng; PENG, Jian; XU, Jinbo. Protein 8-class secondary structure prediction using conditional neural fields. **Proteomics**, v. 11, n. 19, p. 3786–3792, 2011.

WATABE, M.; NAKAKI, T. ATP depletion does not account for apoptosis induced by inhibition of mitochondrial electron transport chain in human dopaminergic cells. **Neuropharmacology**, v. 52, n. 2, p. 536 – 541, 2007.

WEN, Bo; ZENG, Wen-Feng; LIAO, Yuxing; SHI, Zhiao; SAVAGE, Sara R.; JIANG, Wen; ZHANG, Bing. Deep learning in proteomics. **PROTEOMICS**, v. 20, n. 21-22, p. 1900335, 2020.

WEN, Jingran; SCOLES, Daniel R; FACELLI, Julio C. Molecular dynamics analysis of the aggregation propensity of polyglutamine segments. **PloS One**, v. 12, n. 5, p. e0178333, 2017.

WERBOS, Paul J. Backpropagation through time: what it does and how to do it. **Proceedings of the IEEE**, v. 78, n. 10, p. 1550–1560, 1990.

WIMLEY, William C; WHITE, Stephen H. Experimentally determined hydrophobicity scale for proteins at membrane interfaces. **Nature Structural & Molecular Biology**, v. 3, n. 10, p. 842–848, 1996.

WOLFF, Katrin; VENDRUSCOLO, Michele; PORTO, Markus. Coarse-grained model for protein folding based on structural profiles. **Physical Review E**, v. 84, n. 4, p. 041934, 2011.

WOLYNES, P G; ONUCHIC, J N; THIRUMALAI, D. Navigating the folding routes. **Science**, v. 267, n. 5, p. 1619–1620, 1995.

WOOD, Matthew J; HIRST, Jonathan D. Protein secondary structure prediction with dihedral angles. **PROTEINS: Structure, Function, and Bioinformatics**, v. 59, n. 3, p. 476–481, 2005.

WU, Sitao; ZHANG, Yang. ANGLOR: a composite machine-learning algorithm for protein backbone torsion angle prediction. **PloS One**, v. 3, n. 10, p. e3400, 2008.

XUE, Bin; DOR, Ofer; FARAGGI, Eshel; ZHOU, Yaoqi. Real-value prediction of backbone torsion angles. **Proteins: Structure, Function, and Bioinformatics**, v. 72, n. 1, p. 427–433, 2008.

YANG, Jae Shick; CHEN, William W.; SKOLNICK, Jeffrey; SHAKHNOVICH, Eugene I. All-atom ab initio folding of a diverse set of proteins. **Structure**, v. 15, n. 1, p. 53 – 63, 2007.

YANG, Lin; ZHANG, Feng; WANG, Cai-Zhuang; HO, Kai-Ming; TRAVESSET, Alex. Implementation of metal-friendly EAM/FS-type semi-empirical potentials in HOOMD-blue: A GPU-accelerated molecular dynamics software. **Journal of Computational Physics**, v. 359, p. 352 – 360, 2018.

YANG, Tao; KECMAN, Vojislav; CAO, Longbing; ZHANG, Chengqi; HUANG, Joshua Zhexue. Margin-based ensemble classifier for protein fold recognition. **Expert Systems with Applications**, v. 38, n. 12, p. 348–355, 2011.

YANG, Yuedong; HEFFERNAN, Rhys; PALIWAL, Kuldip; LYONS, James; DEHZANGI, Abdollah; SHARMA, Alok; WANG, Jihua; SATTAR, Abdul; ZHOU, Yaoqi. SPIDER2: A package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks. *In*: **Prediction of Protein Secondary Structure**. Heidelberg: Springer, 2017. p. 55–63.

YASEEN, Ashraf; LI, Yaohang. Context-based features enhance protein secondary structure prediction accuracy. **Journal of Chemical Information and Modeling**, v. 54, n. 3, p. 992–1002, 2014.

YASEEN, Ashraf; LI, Yaohang. Template-based C8-SCORPION: a protein 8-state secondary structure prediction method using structural information and context-based features. **BMC Bioinformatics**, v. 15, n. 8, p. 153–164, 2014.

ZACHARIAS, Martin. Protein–protein docking with a reduced protein model accounting for side-chain flexibility. **Protein Science**, v. 12, n. 6, p. 1271–1282, 2003.

ZAKI, Mohammed J; NADIMPALLY, Vinay; BARDHAN, Deb; BYSTROFF, Chris. Predicting protein folding pathways. **Bioinformatics**, v. 20, n. 5, p. i386–i393, 2004.

ZEILER, M. D. DADELTA: An Adaptive Learning Rate Method. ArXiv. 2012.

ZEYTUNI, Natalie; HONG, Chuan; FLANAGAN, Kelly A; WORRALL, Liam J; THEILTGES, Kate A; VUCKOVIC, Marija; HUANG, Rick K; MASSONI, Shawn C; CAMP, Amy H; YU, Zhiheng *et al.* Near-atomic resolution cryoelectron microscopy structure of the 30-fold homooligomeric SpoIIIAG channel essential to spore formation in Bacillus subtilis. **Proceedings of the National Academy of Sciences**, v. 114, n. 34, p. E7073–E7081, 2017.

ZHENG, Wenjun; WEN, Han. A survey of coarse-grained methods for modeling protein conformational transitions. **Current Opinion in Structural Biology**, v. 42, n. 1, p. 24–30, 2017.

ZHENG, Wei; WUYUN, Qiqige; LI, Yang; MORTUZA, S. M.; ZHANG, Chengxin; PEARCE, Robin; RUAN, Jishou; ZHANG, Yang. Detecting distant-homology protein structures by aligning deep neural-network based contact maps. **PLOS Computational Biology**, v. 15, p. 1–27, 10 2019.

ZHOU, Changjun; SUN, Chuan; WANG, Bin; WANG, Xiaojun. An improved stochastic fractal search algorithm for 3D protein structure prediction. **Journal of Molecular Modeling**, v. 24, n. 6, p. 125, 2018.

ZHOU, Jian; TROYANSKAYA, Olga G. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. ArXiv. 2014.

ZHOU, Ting; CAFLISCH, Amedeo. Free energy guided sampling. **Journal of Chemical Theory and Computation**, v. 8, n. 6, p. 2134–2140, 2012.

ZHU, Jianwei; ZHANG, Haicang; LI, Shuai Cheng; WANG, Chao; KONG, Lupeng; SUN, Shiwei; ZHENG, Wei-Mou; BU, Dongbo. Improving protein fold recognition by extracting fold-specific features from predicted residue–residue contacts. **Bioinformatics**, v. 33, n. 23, p. 3749–3757, 2017.

ANNEX

ANNEX A – PUBLICATIONS

It was developed projects in different Computational Intelligence areas that help, directly or indirectly, the development of this Thesis. These works are present bellow:

An approach to identifying writer based on songwriting images using Convolutional Neural Networks ((HATTORI *et al.*, 2017c), Data augmentation effects on CNN performance (AQUINO *et al.*, 2017b), soft biometrics classification using Convolutional Autoencoders ((AQUINO *et al.*, 2017a) Unsupervised image classification (GUTOSKI *et al.*, 2017), political parties using clustering methods (INACIO *et al.*, 2018), how to measure the importance, influence, and relevance of Twitter users? (GABARDO *et al.*, 2018). An analysis of the transfer learning technique for the anomaly detection (GUTOSKI *et al.*, 2020) A qualitative analysis of the Deep Learning *frameworks* (GUTOSKI *et al.*, 2018). A framework for analyzing book covers and co-purchases using object detection and data mining methods was proposed by Berno *et al.* (2019). A classification approach to identify weeds and crops at the pixel-level using CNNs was presented in Brilhador *et al.* (2019).

ANNEX B – ABSTRACTS

A Novel Approach to Protein Folding Prediction based on Long Short-Term Memory Networks: A Preliminary Investigation and Analysis

Leandro Takeshi Hattori*, César Manuel Vargas Benítez§, Matheus Gutoski †,
Nelson Marcelo Romero Aquino ††, Heitor Silvério Lopes‡
Bioinformatics and Computational Intelligence Laboratory
Federal University of Technology-Paraná
Curitiba, Brazil
Emails: *lthattori@gmail.com, §cesarbenitez@utfpr.edu.br, † matheusgutoski@gmail.com,
††nmarceloromero@gmail.com, ‡hslopes@utfpr.edu.br

***Abstract*—The Protein Folding Problem (PFP) is considered one of the most important open challenges in Biology and Bioinformatics. Long Short-Term Memory (LSTM) methods have risen recently, achieving the state-of-art performance for several Bioinformatics problems such as, protein secondary and tertiary protein structure prediction. This paper describes the application of a novel approach based on the LSTM networks to the PFP using a coarse-grained model of proteins. An specific encoding scheme for representing protein folding states is also presented. The proposed approach was evaluated by means of several experiments with a dataset of protein folding, which was obtained by Molecular Dynamics simulations. We also propose a novel method for evaluating the performance of the approach based on measures used in Bioinformatics. Furthermore, a new analysis method for protein folding pathways is presented. Results suggest that the proposed approach is able to learn the protein fold transitions. Also, it is promising for the research areas related to Bioinformatics and Computational Intelligence.**

I. Introduction

The Protein Folding Problem (PFP) is considered to be one of the most challenging open problems in science. Basically, a PFP consists in determining the sequence of folding events that leads from the primary structure of a protein to its native structure which, in turn, defines its specific biological function.

Notwithstanding, researchers have been focusing on the study of this process and, consequently, a large amount of information is currently available regarding this issue. This is mainly due to its importance for medicine, the several genome sequencing projects being conducted in the world and the development of computational models and approaches for the PFP. For instance, several diseases, known as proteinopathies, are believed to be the result of misfolded proteins (i.e. proteins structurally abnormal), such as Alzheimer's disease, cystic fibrosis and some types of cancer [1]. Here, it is important to know that therapeutic drugs for proteinopathies can be discovered from previous knowledge of polypeptide structures. Also, this problem rises three broad questions: (i) What is the physical code by which an amino acid sequence dictates a protein's native structure? (ii) How can proteins fold so fast? (iii) Can we devise a computer algorithm to predict polypeptide structures from their sequences? [2].

To the best of our knowledge, the Molecular Dynamics (MD) approach (including its variations) is the only computational method that really provides a time-dependent analysis of the folding mechanism [3]. Generally, it involves the three-dimensional coordinates of the particles that form the protein and numerical integration of the classical equations of motion. Despite the great advances in recent years, MD simulations have been limited mainly by their computationally expensive brute force calculation. Due to the lack of methods for solving such class of problems in a reasonable computing time, the need for alternative non-traditional mathematical approaches for reproducing the complex behavior of the folding process has risen.

For decades, Computational Intelligence (CI) has provided a large range of robust optimization methods, capable of successfully dealing with complex optimization problems, such as the Protein Structure Prediction (PSP) [4]. Furthermore, within the scope of CI, Deep Learning (DL) methods have yielded significant results on Bioinformatics [5], [6] during the recent years, including the torsion angles prediction methods proposed by [7], for instance. Among DL approaches, the Long Short-Term Memory (LSTM) networks have excelled results in sequential/temporal problems. Therefore, an alternative non-deterministic way to reduce the inherent complexity of the simulations with three-dimensional structures is proposed in this preliminary work, using a minimalist representation of proteins and a LSTM architecture.

The main highlights of this work are:

- a novel approach based on LSTM networks applied to the protein folding prediction;
- a novel method for evaluating the predictor performance based on measures commonly used in Bioinformatics;
- a novel encoding scheme for representing protein folding states and low-level input/output representation for Deep Learning approaches;

A Benchmark of Optimally Folded Protein Structures Using Integer Programming and the 3D-HP-SC Model

Leandro Takeshi Hattori[a], Matheus Gutoski[a], César Manuel Vargas Benítez[a], Luiz Fernando Nunes[a] and Heitor Silvério Lopes[a]

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

ARTICLE INFO

Keywords:
Biological sequences
Hydrophobic-Polar model
Integer Programming
Protein Structure Problem

ABSTRACT

The Protein Structure Prediction (PSP) problem comprises, among other issues, forecasting the three-dimensional native structure of proteins using only their primary structure information. Most computational studies in this area use synthetic data instead of real biological data. However, the closer to the real-world, the more the impact of results and their applicability. This work presents 17 real protein sequences extracted from the Protein Data Bank for a benchmark to the PSP problem using the tri-dimensional Hydrophobic-Polar with Side-Chains model (3D-HP-SC). The native structure of these proteins was found by maximizing the number of hydrophobic contacts between the side-chains of amino acids. T he problem was treated as an optimization problem and solved by means of an Integer Programming approach. Although the method optimally solves the problem, the processing time has an exponential trend. Therefore, due to computational limitations, the method is a proof-of-concept and it is not applicable to large sequences. For unknown sequences, an upper bound of the number of hydrophobic contacts (using this model) can be found, due to a linear relationship with the number of hydrophobic residues. The comparison between the predicted and the biological structures showed that the highest similarity between them was found with distance thresholds around 5.2 to 8.2 Å. Both the dataset and the programs developed will be freely available to foster further research in the area.

1. Introduction

The Protein Structure Prediction (PSP) problem is an active field of research in Bioinformatics. One of the many issues studied in this field comprises forecasting the three-dimensional native structure of proteins using only their primary structure information (Dill and MacCallum, 2012).

Proteins have key functions in the living cell, such as transmembrane receptors (Vinogradova et al., 2000; Chua et al., 2011), storage (Reinhard et al., 1999), cellular processes (Mortishire-Smith et al., 1995), and signaling (Grace et al., 2007). However, when some proteins fail to fold into their functional form, they are associated to some human diseases, such as Alzheimer (Benaki et al., 2005, 2006). They are also associated with viruses, like the Hepatitis C Virus (HCV) (Gouttenoire et al., 2009), Human Immunodeficiency Virus (HIV) (Amodeo et al., 2017), Bovine viral diarrhea virus (Sapay et al., 2006), and bacteria, e.g., *Escherichia coli* (Duarte et al., 2007) and *Acholeplasma laidlawii* (Lind et al.).

Experimental techniques are not trivial for unveiling protein structures. This is evidenced by the number of protein sequences that have been discovered along time (more than 160 million sequences[1]) compared to the number of known protein structures (155,618 structures[2]). This huge gap shows that it is still quite important to invest efforts in methods for unveiling protein structures. In this sense, computational approaches in Bioinformatics have been explored in the literature (Dorn et al., 2014a; Ovchinnikov et al., 2018) for the PSP problem.

According to the Levinthal's paradox, a protein can assume an astronomic number of possible conformations (Karplus, 1997). Therefore, if a protein was supposed to sequentially try all possible conformations sequentially until finding its native form, this possibly would take untold time. However, the paradox is that most proteins fold spontaneously on less than a millisecond time scale, To the computational point of view, finding the native structure of a protein is an open challenge. Atkins and Hart (1999), using the simplest HP model, demonstrated that an algorithm for exhaustive search of all conformations would take time that grows exponentially, as the size of the protein grows linearly. Therefore the protein structure prediction problem is reputed as NP-complete, that is, it cannot be solved in polynomial time. To overcome such a computational complexity, several approaches have been proposed, such as methods based on mathematical programming (Carr et al., 2003; Yanev et al., 2011, 2017) and those based on heuristic approaches (Parpinelli et al., 2014; Li et al., 2015; Kaushik and Sahi, 2017; Li et al., 2017). The Integer Programming (IP) optimization is a mathematical method that can present the optimal result given a set of constraints and an objective function (Nunes et al., 2016). This approach has been poorly explored in the literature when applied to the PSP problem (Yanev et al., 2011, 2017). However, since it produces optimal results, it can be very useful for establishing the ground truth for comparison with other heuristic approaches.

Due to the computational power required for the PSP

thumbnails/cas-email.jpg
lthattori@gmail.com (L.T. Hattori); nunes@utfpr.edu.br (L.F. Nunes); hslopes@utfpr.edu.br (H.S. Lopes)
ORCID(s): 0000-0003-3984-1432 (H.S. Lopes)

[1]as in September, 10th 2019 at the site https://www.uniprot.org/

[2]as in September, 10th 2019 at the site https://www.rcsb.org/

PathMolD-AB: Spatiotemporal Pathways of Protein Folding using Parallel Molecular Dynamics with a Coarse-Grained Model

Leandro Takeshi Hattori[a], Bruna Araujo Pinheiro[a], Rafael Bertolini Frigori[a], César Manuel Vargas Benítez[a] and Heitor Silvério Lopes[a]

[a]*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 3,500 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 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.,

ORCID(s): 0000-0002-1945-6855 (L.T. Hattori); 0000-0002-4861-7240 (R.B. Frigori); 0000-0002-5691-5432 (C.M.V. Benítez); 0000-0003-3984-1432 (H.S. Lopes)

ANNEX C – PATHMOLD-AB SOFTWARE

C.1 RUNNING PARAMETERS

In principle, PathMolD-AB 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 file 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 declaration of the routine (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 MD simulation contain all GPU communication, I/O functions, and ensemble control.

To improve the program's usability, we developed a script file for the program execution (`Makefile`). 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 the PDB file, extraction of the AB sequence, creation of the MD input file, compilation of 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.

C.2 INPUT AND OUTPUT FILES

It is necessary to configure only an input text file containing information about the simulation and the protein to be folded to simulate the protein folding trajectories using PathMolD-AB, as shown in Table 10. 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

Table 10 – Input file parameters for the protein folding simulation.

Parameter	Description	Example
sequence	hydrophobic-polar sequence of the protein	ABBABBABABBAB
ProtLen	number of amino acids	13
LV	box size of the simulation	26
stepLimit	maximum number of MD iterations	3000000
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

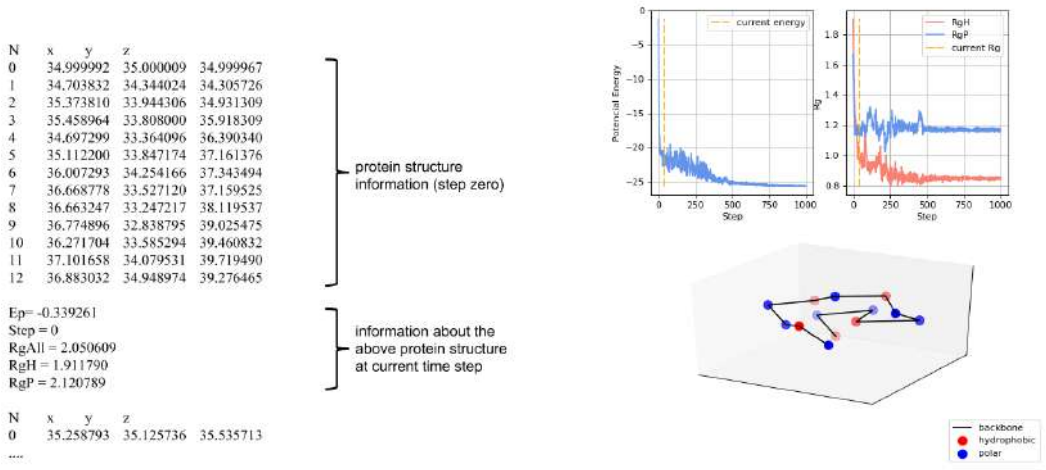
Source: own authorship

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 protein residues 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 with the overall E_p energy. The format of the records in the dataset is shown in Figure 44(a).

Figure 44 – (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.



(a) **(b)**

Source: own authorship

To make the pathway data generated in the simulations humanly interpretable, the PathMolD-AB software package provides a visualization tool (`pathway_print_multi-`

`subplot.py`). This program produces a video using the folding data's information showing the protein structure evolving along with many iterations. Other information is 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 Figure 44(b). This software was developed for the Linux operating system using the Python programming language.

ANNEX D – SOFTWARE-HARDWARE COMPATIBILITY

Table 11 presents the PathMolD-AB software compatibilities in terms of the CUDA toolkit version, programming language version, and compute capability[1], 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 11 – PathMolD-AB package compatibility

GPU model	Compute Capability	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		

Source: own authorship

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