A Molecular Model for Representing Protein Structures and its Application to Protein Folding

Fernanda Hembecker, Heitor Silvério Lopes
Bioinformatics Laboratory, Federal University of Technology Paraná
Av. 7 de setembro, 3165 – 80230-901 Curitiba, Brazil
fernanda@denes.com.br, *hslopes@utfpr.edu.br*

***Abstract*—The protein folding problem is a central issue in Bioinformatics. It still represents a challenge for both Biology and Computer Science. Proteins are composed by up to hundreds of amino acids, each one with tenths of atoms. In general, a full representation of such structure and its interacting elements precludes computational simulations. This work proposes a molecular model for representing protein structures. This new model preserves some physico-chemical properties of the original protein and is aimed at allowing efficient computational simulations. A simulation of the model using a multiagents system is reported. Results so far suggest the adequacy of the proposed model for representing protein structures and their folding process with a reasonable complexity and suitable expressiveness.**

I. Introduction

Proteins are complex macromolecules that perform vital functions in all living beings. They are created in the ribosomes and are a combination of variable-length of the 20 standardized amino acids. The biological function of a protein is directly defined by its sequence of amino acids and the way it is folded into a specific three-dimensional structure, known as native conformation. Understanding how proteins fold is of great importance to Biology, Biochemistry and Medicine. Considering the full analytic atomic model of a protein, it is still not possible to determine the exact three-dimensional structure of real-world proteins, even with the most powerful computational resources [1]. To reduce the computational complexity of the analytic model, many simplified models have been proposed. However, even the simplest one (the well-known 2D-HP model [2]), was proved to be algorithmically intractable due to its NP-completeness [3]. Therefore, the quest for feasible models of protein representation is a central issue to be solved in Bioinformatics, so as to enable further development of algorithms for simulating protein folding.

The current approach for studying the structure of proteins is the use of heuristic methods that, however, do not guarantee the optimal solution [4]. For instance, evolutionary computation techniques have been proved to be efficient for many engineering and computer science problems, and this is also the case of unveiling the structure of proteins using simple lattice models [1],[5].

The Protein Data Bank – PDB [6] is the largest repository of protein structures. Currently there are data from around 61,300 structures (as in November/2009) in PDB. However, there are hundreds of thousands proteins known, but whose three-dimensional structure is not known. This is an important motivation for developing computational methods for unveiling the structure of these proteins.

The computational simulation of the folding of proteins encompasses two main issues: (a) a model for representing the three-dimensional structure of a protein with a level of details compatible with the computational power; (b) an explicit method, based on biological knowledge, that dictates the interaction between elements along time, during the folding process towards the native conformation of the protein. This work is focused on the first issue, since it is the base for the second one. Therefore, the objective is to propose a computationally feasible model for representing globular proteins, based on real-world data from the PDB. By using the proposed model, a simulation was done using a multiagents system and preliminary results are reported.

II. Protein Models and the Protein Folding Problem

The structure of proteins at the atomic level is complex. It is a function of the chemical composition of the amino acids and the bounds between them, as well as physico-chemical conditions. Each amino acid has several properties, such as electric charge, mass, volume, level of affinity to water. Besides these properties, once close each other, the spatial disposition of their atoms is influenced by the formation of hydrogen bonds and other factors. Therefore, the atomic composition of the amino acids compounding a protein is the main factor that determines how it will be folded.

According the Anfinsen's thermodynamical hypothesis, a biomolecule will stand the most part of the time in a state of minimum free-energy [7]. Considering a protein, such hypothesis is applicable to its native conformation. Accordingly, artificial models that follows this principle and aims at simulating the behavior of proteins, need to have the following:

- A model of the protein, that is, an abstraction of the atoms of the protein and their interactions;
- A model of possible conformations attainable by a given protein. This is expressed as a set of rules that describe the correct conformations;
- An energy function capable of computing the free-energy of each valid conformation of a molecule.

Currently, the several proposed models for studying the structure of proteins can be roughly divided into two cate-

978-1-4244-4713-8/10/$25.00 ©2010 IEEE

gories: analytical and discrete. The former models, also known as “all-atoms" models, consider all available atomic information and take into account all possible physico-chemical interactions between atoms. The massive amount of information to be manipulated and the huge computation required makes the molecular dynamics using such models rather unfeasible, even for small molecules. On the other hand, discrete models tend to simplify in excess the representation of a molecule. The loss of information of discrete models can be very relevant and many biological aspects are disregarded. However, such category of models are the most studied from the computational point of view.

The amount of details of the structure modeled is the main factor that contributes for the complexity when simulating computationally the folding. For instance, a protein model could have the full spatial representation of all its atoms, all its atoms but hydrogen, only the backbone without the side-chains, or as simple hydrophobic-polar elements embedded in a lattice.

The Protein Structure Prediction (PSP) problem can be defined as determining the final three-dimensional structure of a protein by using the information about its primary structure. On the other hand, the Protein Folding Problem (PFP) is understood as being the discovery of the pathways by which a protein is folded into its natural conformation, during its synthesis [8]. It should be noted that in the current literature those two terms are frequently misused. A computational approach for predicting the structure of a protein as well its folding pathways demands, in first place, a model that represents it abstractly, in a given level of details. This is the starting point of this work.

Several methods were proposed for both PSP and PFP. Most of them are based on heuristic methods, since the computational complexity of the underlying models makes their complete computational simulation is very expensive and, sometimes, unfeasible.

III. Methodology

A protein can be viewed as a collection of atoms connected each other. An analytical model having a detailed description of the three-dimensional structure of a protein includes the information about all its individual atoms [9]. Therefore, to specify the tertiary structure of a protein, it is possible to establish values for angles, lengths and torsions of the connections among atoms in the structure. Such analytical model is too complex to be treated efficiently with current computational resources. Therefore, to reduce the inherent complexity of this model, some atoms could be disregarded or even grouped into larger elements, and treated as equivalent single atoms by the model. Obviously, such reduction can decrease the visual equivalence between the model and a real protein, for a given conformation. On the other hand, such approach decreases the number of degrees of freedom of the structure, allowing a more convenient modeling.

The format of a PDB file is text and contains many fields for the several different information recorded for each protein.

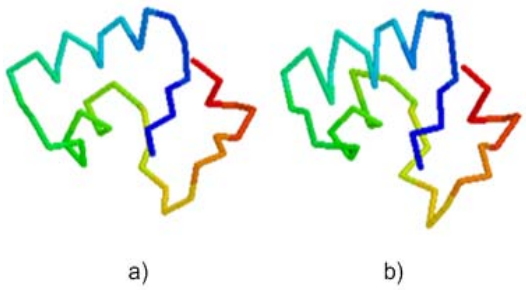

Fig. 1. Backbone of the protein 1CRN using original PDB data (a), the molecular model using the alpha carbon as the center of atoms.

Amongst these information, it can be cited: bibliographic data, atomic coordinates, primary and secondary structures, and crystallographic data. To process the information in the PDB files in a reasonable way it is necessary to filter out information that is not relevant for the computational model to be used in the protein folding process. The PDB format has 12 sections, in which 46 different fields are represented. The following fields are relevant for our model: SEQRES (defines the amino acids sequence of the protein), HELIX and SHEET (identify the amino acids that form these secondary structures), and ATOM (represents the spatial distribution of the atoms of the protein in its native conformation).

Considering that, in average, each amino acid has eight atoms, we propose that each amino acid of a real protein be represented by a single element, as if it was a “big molecule" (thus the model was named). This is done to decrease the amount of information to be dealt by the model. Also, knowing that the alpha carbon of the amino acid is a representative element of the amino acid and part of the backbone of the protein, its position was set as the center of the big molecule.

Fig. 1 shows the backbone of the native conformation of protein 1CRN, using the original PDB data as well as the proposed representation. It can be observed that, using such approach, the structural representation of the protein is not significantly changed, preserving the visual characteristics of the protein. In fact, the simplified molecules are equivalent to the original three-dimensional atom groups.

Furthermore, the model also includes some chemical and physical properties of the original structure, such as:

- Volume and diameter of the amino acids: all the molecules representing the amino acids have the same average volume and average diameter of the original amino acid. Each molecule is represented as a sphere whose diameter is a function of average spatial distribution of the atoms compounding a given amino acid.
- Minimum and maximum distance for peptidic bonds: Based on the analysis of a large amount of PDB data, two 20 x 20 matrices were constructed. Each cell of the first matrix is the average minimum distance (in Angstroms) between each pair of amino acids. Conversely, the other matrix is the average maximum distance. Such distances have to be respected during the folding process, whatever

TABLE I
AVERAGE DIAMETER (IN ANGSTROMS) AND HYDROPHOBICITY OF STANDARD AMINO ACIDS.

Amino acid	average diameter	average volume	K&D hydrophobicity
ALA	3.522	88.6	1,8
ARG	7.740	173.4	-4,5
ASN	4.915	114.1	-3.5
ASP	5.246	111.1	-3.5
CYS	0.176	108.5	2.5
GLN	6.007	143.8	-3.5
GLU	6.201	138.4	-3.5
GLY	3.454	60.1	-0.4
HIS	5.310	153.2	-3.2
ILE	5.429	166.7	4.5
LEU	5.322	166.7	3.8
LYS	6.917	168.6	-3.9
MET	6.069	162.9	1.9
PHE	6.856	189.9	2.8
PRO	4.428	112.7	-1.6
SER	3.946	89.0	-0.8
THR	4.530	116.1	-0.7
TYR	7.922	193.6	-1.3
TRP	6.994	227.8	-0.9
VAL	4.440	140.0	4.2

method will be used for.

- Hydropathicity: defines the level of affinity to water according to the Kyte-Doolitle scale [10]. Hydrophobic amino acids tend to converge towards the inner part of the protein during folding, so as to be protected from the solvent by the polar (or hydrophilic) amino acids. Therefore the proposed molecular model was developed primarily for representing globular proteins (with a well-defined hydrophobic center) and with a single domain (a single polypeptidic chain) [11].

A sample of 200 proteins was used to compute the average diameter (in Angstroms) and volume of each amino acid, as well as the average minimum and maximum distance between every pair of amino acids. We used a specific group of proteins (mioglobins), although this study could be done for other group or family of proteins. Table I summarizes these information, as well as the hydrophobicity value according the Kyte-Doolitle scale (for the hydrophobic amino acids, the value is positive, and for the polar ones, the value is negative) [10]. The matrices for average minimum and maximum distances are not shown here due to space restrictions.

As mentioned before, the original PDB files need to be filtered and a new simplified format is obtained (*PDBLight*). This format includes only the minimal information needed for folding the protein. This new format will be the input for a folding algorithm later developed, and the output of them. Furthermore, converting real-world protein data to the *PDBLight* format it is possible to compare the real native conformation of a protein and that generated by a folding algorithm.

Besides defining the structure of the molecular model, we also define a number of general rules for conducting the folding procedure using the model. Such rules define how the molecules can move in the space during the folding, as follows:

1) Progressive folding: in order to be biologically plausible, the folding should consider one molecule at a time (as it happens in real organisms). As the molecules come up in the chain, interactions between them takes place.
2) Hydrophobic center: hydrophobic molecules move preferentially towards the center of the protein being folded. Recall that the geometric center of the protein is always changing according to the spatial distribution of the molecules.
3) Polar chain movement: in principle, the polar amino acids do not move by themselves. Instead, they are pushed or pulled by the hydrophobic amino acids to which they are immediately connected. Eventually, if three or more polar amino acids are connected each other, a given (polar) amino acid will be pushed or pulled by the most distant polar amino acid to which it is connected. Therefore, a chain reaction takes place when a hydrophobic amino acid pushes/pulls a polar amino acid, and this one will push/pull other polar amino acid. As the folding progress, chains of connected polar amino acids will tend to keep an average distance between each other.
4) No overlaps: molecules occupy a space defined by their volume and the physical law that two bodies cannot occupy the same space holds here.
5) No collisions: molecules move respecting the minimum and maximum distances between them, so as to avoid collisions. Further experiments suggested that such limits should be flexible at the beginning of the folding process, and then, decreasing the tolerance.
6) Balance of forces: the position of a molecule in the space must consider all the forces actuating over it. A vectorial sum of forces will point the direction and intensity of the movement of the molecule, provided the previous rules are not broken. If this situation happens, the molecule is moved as little as possible without broking rules.

Recall that the above-mentioned rules does not take into account the formation of secondary structures, such as alpha-helices and beta-sheets. These rules are generic and, for secondary structures, specific formation and movement rules have to be devised. For proteins having alpha-helices the following additional procedures are done:

- A secondary structure recognition system have to be used in order to identify potential helices (or beta-sheets) in the linear sequence of amino acids. There are currently several valuable methods for predicting secondary structures that can be used in tandem with the sequential synthesis of the protein in the simulated ribosome.
- Once an alpha-helix is known to occur in the incoming sequence of amino acids (during folding), such sequence should be distributed helicoidally in the space. Once established the helix, its shape will be maintained during folding, but subject to rotation and translation in space.
- The movement of an alpha-helix will be driven by the

hydrophobicity value of its compounding amino acids. For instance, if the helix contains strongly hydrophobic amino acids, they will push the helix towards the center of the protein.

During folding, it is necessary to evaluate continuously the quality of the conformation. Such evaluation allows a supervisory system to redirect the folding once detected, for instance, the premature stabilization of the protein (amino acids do not move and the folding stops). In this case it is possible to allow a temporary flexibilization of the distances (maximum and minimum) between amino acids, beyond the established limits. Besides, it is possible to make random moves in the position of amino acids (a kind of "noise"), so as to redefine the forces actuating over them and thus pushing the folding to a new path in the energy funnel [2].

To evaluate the quality of the folding in a given moment we propose a fast method. Since the three-dimensional comparison of the amino acids of two proteins is computationally expensive, we propose the use of a neighborhood measure. Based on the native conformation presented in the *PDBLight* model and a user-defined specific neighborhood radius, it is possible to identify which amino acids are the neighbors of a given amino acid of the chain. With such information, a given conformation can be evaluated taking into consideration the neighbors of each amino acid and comparing them with the native conformation. The more amino acids have the expected neighbors, the closer the current conformation is to the native conformation. To evaluate the proposed metrics, similar proteins were submitted to UCSF Chimera [12]. We compared the result of this public-domain software and our proposed metrics for two proteins 1A6M and 1A6K. Chimera and our metrics yielded, respectively, 98.68% and 98.40% of similarity.

IV. Computational Simulation

The main objective of this work is to propose a simplified, although expressive, model for representing the structure of proteins and their folding. To verify the utility and computational feasibility of the molecular model proposed, a multiagents system for protein folding was developed.

It is believed that the folding of real proteins is not governed by a central control. Possibly, it is the consequence of the interaction between amino acids and the surrounding environment where they are. The multiagents system proposed here was created following this principle. Three types of agents were devised, and their behavior is described in the next sessions.

A. Amino Acid Agent – AG-A

There is one independent amino acid agent of this type for each amino acid of the polypeptidic chain. The objective of this agent is to find its best position in the space, relative to the other agents, at each time step. To compute its own position, the AG-A agents take into account the balance of forces actuating on it as a function of the interaction with its neighborhood, and also its own chemical properties.

The generical steps that represents the behavior of AG-A's is as follows:

1) Consider attraction/repulsion to/from the center of the protein;
2) Consider attraction/repulsion to/from predecessor and successor amino acids, according to minimum and maximum allowable distances;
3) Consider repulsion from other amino acids that eventually are in its the neighborhood;
4) Compute the destination coordinates as function of the preceding rules;
5) If the above rules are not violated, move amino acid to the destination coordinates. Else, compute the smallest step towards the destination, such that rules are not violated.
6) Repeat all steps above until the AG-E detects that no more movement is possible.

B. Secondary Structure Agent – AG-S

This type of agent is created whenever a secondary structure is identified. A set of 200 protein data files was previously analyzed to identify sequences with alpha-helices. Based on these data, a parser was implemented. It is able to identify given sequences of amino acids, just inserted in the folding environment, potentially able to form secondary structures (and shall be controlled by an AG-S).

In the current version, only alpha-helices are detected, future versions will include beta strands and general turns. The amino acids belonging to a secondary structure will be tied together in such a way to limit (but not forbidding) their movements. When the AG-S takes control of a group of amino acids, their corresponding AG-As will get idle. This procedure assures that, once built a secondary structure, it will not be destroyed during folding, and other specific rules will control its movements. Therefore, possible rotations and translations represent movements of the whole secondary structure.

The behavior of AG-S is defined by the following generical steps:

1) Identify a sequence of AG-A's potentially capable of forming a secondary structure;
2) Inform the related AG-A's that, henceforth, their movement will be controlled by an AG-S;
3) Consider attraction/repulsion of the secondary structure to/from the center of the protein;
4) Consider repulsion from amino acids that eventually are in of the neighborhood of the secondary structure;
5) Compute the destination coordinates as function of the two preceding rules;
6) If the above rules are not violated, move amino acid to the destination coordinates. Else, compute the smallest step towards the destination, such that rules are not violated.
7) Repeat all steps above until the AG-E detects that no more movement is possible.

C. Environment Agent – AG-E

This agent starts the creation of the polypeptidic chain (represented by the community of AG-A's). It also is responsible for constantly computing the geometrical center of the protein under formation so as to guide the hydrophobic AG-A agents to build a hydrophobic core. AG-E agent is also responsible for informing the neighborhood of a given AG-A, so as to enable them to compute their own position relative to the corresponding neighborhood. Finally, an extra function is relied to AG-E: from time to time, it computes the quality of the folding, as mentioned in the previous section.

The behavior of AG-E is defined by the following generical steps:

1) Initialize all AG-A's according to the amino acids sequence of a *PDBLight* format;
2) Compute the geometrical center of the protein using current position of the amino acids;
3) Compute the free-energy of the folding;
4) Compute, for all AG-A's, if their position have stabilized;
5) When requested, inform AG-A's about the center of the protein and the position of neighboring amino acids;
6) Repeat the above steps (except the first one) until no more movements are possible.

When the AG-E detects that all AG-A's have stabilized their positions, the folding process finishes, and a file in the *PDBLight* format is generated. This file can later compared with the corresponding file of the real protein, using the procedure mentioned before. Therefore, it is possible to evaluate the whole methodology and results comparing results with real data.

V. Results and Conclusions

A result of the simulation process is shown in Fig. 2, according to the "spacefill" format where each sphere represents an amino acid molecule (yellow are the hydrophobic and red the polar). The resulting conformation has visual similarity with the original. This figure shows six snapshots of the conformation during the folding process. The protein simulated is the 2GN5, which data was taken from the PDB. This protein has 87 amino acids, therefore, only after the 88^{th} iteration all the amino acids are present and interacting each other. The figure shows the current folding at iterations 20, 50, 70, 110, 140 and 168. The simulation was stopped at iteration 168 since the folding was stabilized, that is, all amino acids were not able to move anymore.

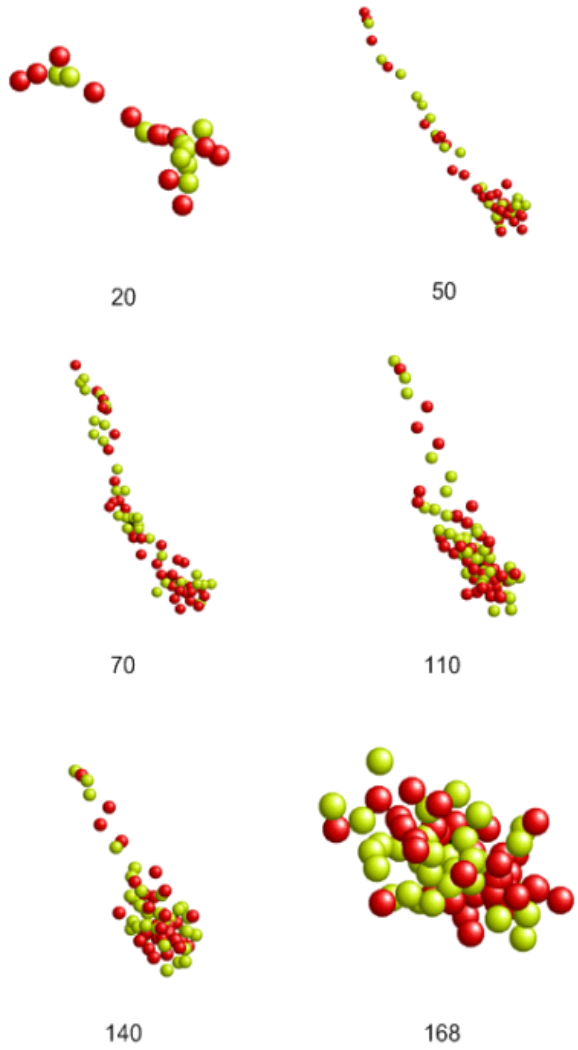

Fig. 2. Snapshots of the simulation of the folding of protein 2GN5 obtained with a multiagents system with the proposed model.

Although subfigures of Fig. 2 are not in the same scale, it is possible to observe how the folding is going on and the hydrophobic core is formed. It is important to recall that the rules of minimum/maximum distance between amino acids is respected, as well as their volume (there is no volumetric overlapping with neighboring amino acids in the chain).

To observe how the compactness of the protein is evolving throughout iterations, Fig. 3 shows the average distance of each molecule to the center of the folding. As expected, as the amino acids are being synthesized (until iteration 87), the average distance tends to increase. That is, the chain has a tendency to be straightened. However, at each iteration, the amino acids that are already present in the chain interact each other, and the balance of forces makes them to move and, consequently, change the average distance to the center. As a new amino acid is synthesized, the position of the center changes itself. Therefore, several fluctuations in the average distance to the center is expected throughout iterations. The largest average distance is reached at iteration 68, representing the moment that the conformation is most straightened (shown in Fig. 3 as an upward arrow). From this point forth, the average distance to the center tends to decrease thanks to the interactions between amino acids conducted by the multiagents system. In particular, after the 88^{th} iteration (shown in Fig. 3 as an downward arrow) the average distance tends to be monotonically decreasing.

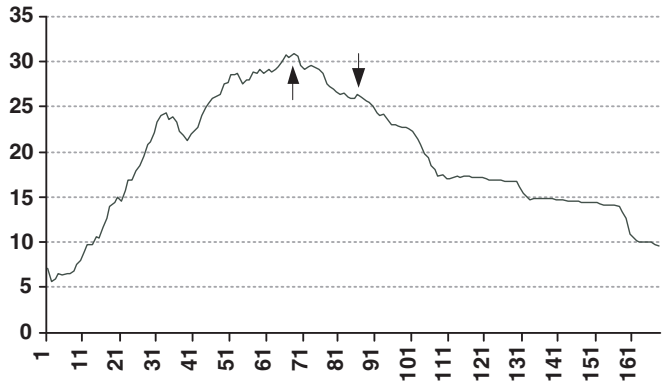

Fig. 3. Average distance of the amino acids to the center of the protein at each iteration.

The use of the physico-chemical properties of real proteins in the proposed molecular model demonstrated to be very useful. In the same way, the use of real-world proteins through their PDB files takes a degree of reality not seen in other simplified computational models for protein folding. However, more work have to be done towards improving the multiagents system, specially in what is related to the folding rules.

This is an ongoing work and the overall results obtained so far suggests the adequacy of the proposed model for representing protein structures with a reasonable complexity (from the computational point of view) and suitable expressiveness (from the biological point of view). Further work will focus on incorporating other biological information in the proposed model, as well as the development of more sophisticated folding algorithms.

ACKNOWLEDGMENT

This work was partially supported by grants from the Brazilian National Research Council – CNPq: 550977/2007-4, 481207/2007-0 and 309262/2007-0

REFERENCES

[1] H. Lopes, "Evolutionary algorithms for the protein folding problem: a review and current trends," in *Applications of Computational Intelligence in Bioinformatics and Biomedicine: Current Trends and Open Problems*, T. Smolinski et al., Ed. Springer, 2008, vol. I, pp. 297–315.

[2] K. Dill and H. Chan, "From Levinthal to pathways to funnels," *Nature Structural Biology*, vol. 4, pp. 10–19, 1997.

[3] P. Crescenzi, D. Goldman, C. Papadimitriou, and A. Piccolboni et al., "On the complexity of protein folding," *Journal of Computational Biology*, vol. 5, no. 3, pp. 423–465, 1998.

[4] S. Mitra and Y. Hayashi, "Bioinformatics with soft computing," *IEEE Transactions on Systems, Man and Cybernetics – Part C*, vol. 36, no. 5, 2006.

[5] S. Pal, S. Bandyopadhayay, and S. Ray, "Evolutionary computation in bioinformatics: a review," *IEEE Transactions on Systems, Man and Cybernetics – Part C*, vol. 36, no. 5, 2006.

[6] H. Berman, J. Westbrook, and Z. Feng et al., "The protein data bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, 2000.

[7] C. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, pp. 223–230, 1973.

[8] B. Honig, "Protein folding: from the levinthal paradox to structure prediction," *Journal of Molecular Biology*, vol. 293, pp. 283–293, 1999.

[9] C. Pedersen, "Algorithms in computational biology," PhD Thesis, Departament of Computer Science, University of Aarhus, Denmark, 2000.

[10] J. Kyte and R. Doolittle, "A simple method for displaying the hydropathic character of proteins," *Journal of Molecular Biology*, vol. 157, pp. 105–132, 1982.

[11] A. Lehninger, D. Nelson, and M. Cox, *Principles of Biochemistry*, 2nd ed., P. of Biochemistry, Ed. New York: Worth Publishers, 1998.

[12] E. Pettersen, T. Goddard, and C. Huang et al., "UCSF chimera - a visualization system for exploratory research and analysis," *Journal of Computational Chemistry*, vol. 25, no. 13, 2004.