Validating Protein Structure Models Using Internal Energy

By: Kaustubh Deshpande
ECS 129 – Final Project - Option 5

Abstract

This project seeks to determine which conformation of a protein is more likely to be found by minimizing the internal energy of proposed topologies. Internal energy is calculated by making approximations of Van der Waals forces, electrostatic energy, and solvation energy. The project is run in standard python 3 with protein data files that were already preprocessed. The python script uses these files to compute energy values for two potential protein structures and predicts which structure is more likely to be found based off thermodynamic reasoning. We have further analyzed the energy differences between the two protein structures in an attempt to understand were these differences originate. Lastly, we have also dissected our algorithm and conducted speed analysis.

Introduction and Background

Proteins are extremely important to help the human body function. Right now, we have a basic idea of the general structures that a protein may have, such as a variety of secondary and tertiary structures that make a protein. In addition, many quaternary structures have been determined for a wide variety of proteins with functions ranging from regulatory to structural purposes. However, given a polypeptide sequence, we would like to accurately identify the resulting structure, which includes widespread alpha helices and beta sheets in addition to more rare structures like the beta sheet hairpins.

Intermolecular forces such as Van der Waals, electrostatic interactions, and solvation energy have a profound effect on protein folding. Gibbs free energy and spontaneity dictates that a molecule wants to reduce its internal energy, therefore staying in a lower energy conformation. By predicting internal energy of multiple possibilities for 3D protein structures, one can identify the most probable folded structure by finding the structure that minimizes internal energy. For a molecule that has N atoms, its total energy is approximately the sum of Van der Waals energy, its electrostatics energy and its solvation energy (ignoring the bonded interactions). This sum can be expressed by the following equation. This equation will be dissected, and each component will be analyzed

in detail. In this equation flag(i,j) = 0 if i and j are atoms involved in a chemical bond, or connected through two chemical bonds, and flag(i,j)=1 otherwise.

$$U = \sum_{i=1}^{N} \sum_{j=i+1}^{N} flag(i,j) \left[\epsilon_{ij} \left[\left(\frac{s_{ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{s_{ij}}{r_{ij}} \right)^{6} \right] + \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{r_{ij}} \right] + \sum_{i=1}^{N} ASP(i) ASA(i)$$

The approximation for Van der Waals energy is done using the equation for Lennard-Jones-Potential as shown below. The Lennard-Jones potential is a mathematical model that approximates the interaction between two neutral atoms or molecules. ε is the depth of the potential well, s_{ij} is the distance at which the potential reaches minimum value, and r is the distance present between the particles. This equation accounts for the attraction and repulsive forces that an atom may experience depending on its distance relative to other atoms within the peptide.

Lennard-Jones-Potential = $$\epsilon_{ij} \left[\left(\frac{s_{ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{s_{ij}}{r_{ij}} \right)^{6} \right]$$

The approximations for electrostatic interactions are obtained by calculating Coulomb potential. Coulomb potential, also called electric potential, is essentially “the work needed to move a unit positive charge from a reference point to a specific point inside the field without producing any acceleration”. This potential can be calculated by the equation given below. In this equation, for two atoms I and j, rij is the distance between them. While qi and qj are the charges of atoms i and j, respectively. Lastly, εr is the dielectric constant for water and has a value of 4.

Coulomb potential = $$\frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{r_{ij}}$$

By chemical definition, solvation energy is the amount of energy associated with dissolving a solute in a solvent. The structure of a protein is heavily influenced by this value. During computation of solvation energy for a protein structure, the contribution of each protein atom is approximated as the product of the accessibility of the atom to solvent and its atomic solvation parameter. In the formula given below for solvation energy, ASP(i) is the atomic solvation parameter for atom i. While ASA(i) is the accessible surface area of i. The accessible surface area can be roughly approximated as

$$ASA_i = 0.2 * 4 * \pi * \left(r_i + R_{H2o} \right)^2$$

Solvation Energy = $\sum_{i=1}^{N} ASP(i)ASA(i)$

While computer models have aided researchers to predict protein models, algorithms built to analyze sequences are not perfect and constantly change over time. Currently, scientists in this field have been able to detect variations of the same protein to determine the most likely structure that the protein may have with very little variability. However, not much research has been done about finding accurate sequences from two different proteins and observing how they aligned with each other. We are not yet able to observe the variability from the sequence and the model compared to the protein in vivo. We plan to take a sequence that one may have to code for amino acids in a protein and accurately determine which of the two forms given is more energetically favorable.

By predicting accurate models, researchers would be able to identify minute differences between distantly related models. We could use this data to observe how the structure of a protein changes due to mutations in the sequence. Once a library of models is made, professionals in the medical field can access this database and see which specific variations of protein they may need to target. Medical professionals may also use this data to determine if a person is able to react to certain viruses or bacteria that may enter the immune system. They may also use this information to target enzymes that regulate or are a part of many metabolic pathways.

Methods

An internal energy calculator was designed with python using the formulas mentioned above. The script opens a preprocessed protein file that contains a tabularized list of atoms in the protein with their associated numerically defined properties. The atoms are stored as a python dictionary and are looped through to calculate internal energy based on the atomic interactions. The Tabularized list contains properties for each atom. The properties available are:

Property	Unit	Description
X	Angstroms (Å)	3D space coordinate
Y	Angstroms (Å)	3D space coordinate
Z	Angstroms (Å)	3D space coordinate
R	Angstroms (Å)	Van der Waals radius
Epsilon	kcal/mol	Depth of the potential well (Wikipedia)
Sigma	Angstroms (Å)	Distance at which the potential reaches its minimum (Wikipedia)
Charge	Coulomb Fraction	Electric Potential
ASP	$Å_2$	Atomic solvation parameter
ASA	kcal/mol/$Å_2$	accessible surface area

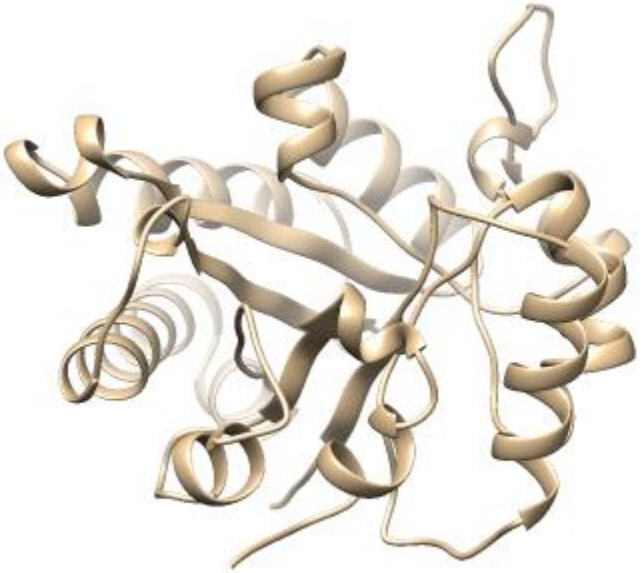
Input: CRD file that includes various crucial pieces of information such as x, y, z coordinates, radius, epsilon, sigma, charge, ASP, atom name and residue name for individual atoms.
Output: Energy levels of each protein model after calculations. Also, printing of statement that tells the user which protein model works best depending on energy levels.

Since our algorithm makes use of nested loops, we estimate the time complexity of the algorithm as O(N^2). The time complexity of our algorithm has been analyzed in detail in results and analysis section of the report.

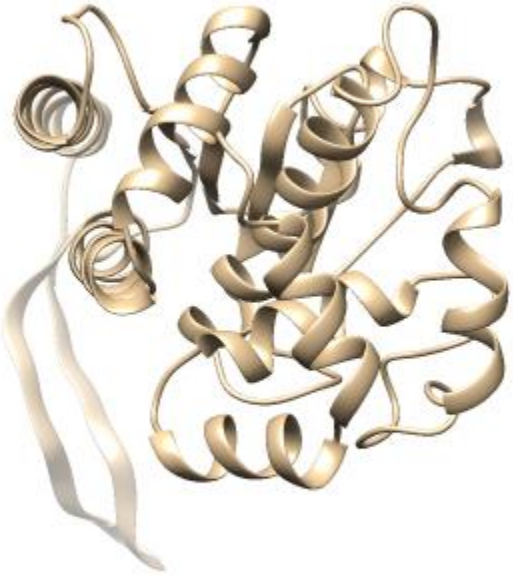
Results

This section contains 3D conformations of both proteins, numerical results of our program and a detailed analysis. This section will also analyze the time complexity of our algorithm in detail.

Protein 1:



Protein 2:

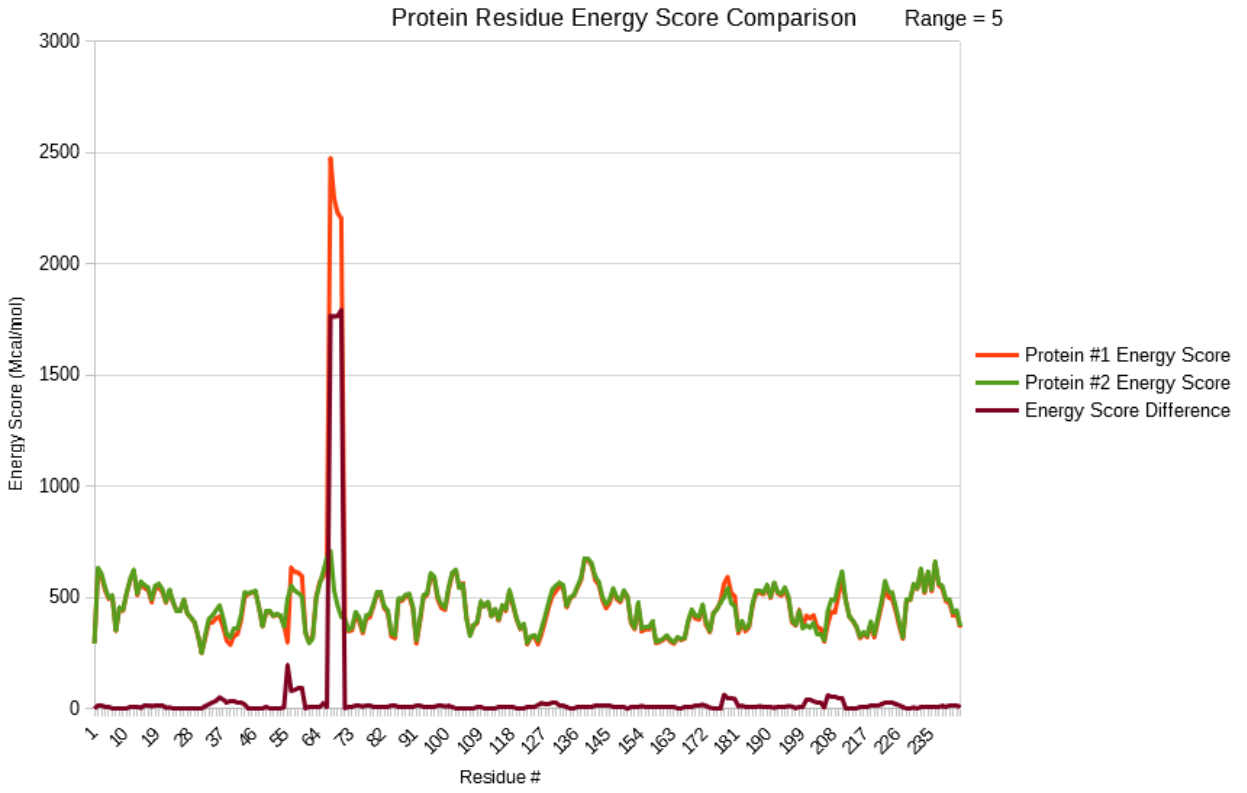


```
Kaustubhs-MacBook-Pro:CRD_File_py_src kaustubh$ python3 main.py
Protein #1 file processing time: 17 milliseconds
Protein #2 file processing time: 20 milliseconds
Protein #1 calculate internal energy processing time: 5807 milliseconds
The internal energy of the protein #1 is 81035209166 kcal/mol
Protein #2 calculate internal energy processing time: 5990 milliseconds
The internal energy of the protein #2 is 1723223 kcal/mol
The internal energy of the protein #2 is less, so it is more likely to occur
```

The energy score of conformation #1 was 8.1e9 kcal/mol, while conformation #2 was 1.7e6 kcal/mol. There was a significant difference in the energy scores between both protein conformations. Lower internal energy means more stability Conformation #2 energy score is lower and is thus more likely conformation because of thermodynamic reasons.

Analysis

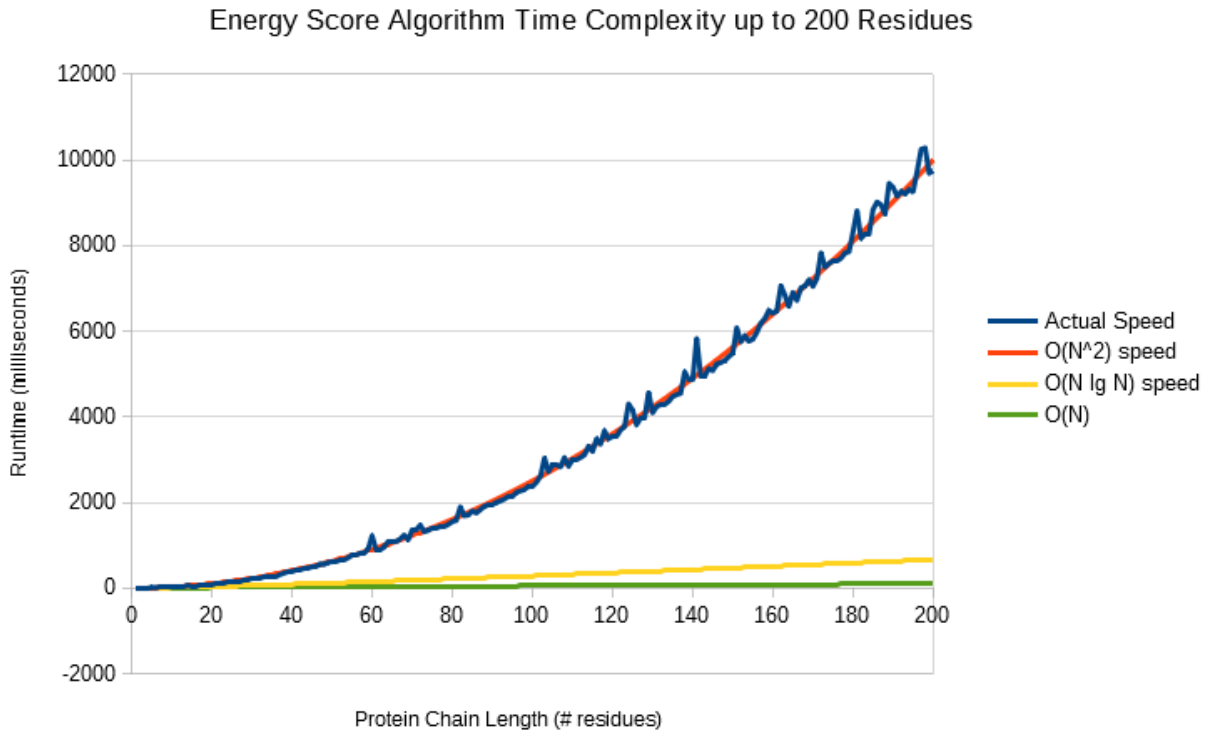
Our results displayed a drastic difference in energy score level. In this section we will analyze this difference and trace its origin with respect to protein structure. A python script was used to locally compute the score difference between the two proteins by creating arbitrarily defined substrings of atoms M length iterating atoms 0 to N. This python script generated a CSV file containing the obtained results. The CSV file was used to obtain the following plot. A subarray range of five residues was used to discover and plot the local energy scores. The use of amino acid position instead of atom position enables us to point to the anomaly on a 3D rendering of the atom.



The graph shows that both conformations of the protein had an average energy score difference of approximately 10 Mcal/mol. However, as expected, we observed an anomaly. This anomaly is the area of greatest local difference on the proteins. From amino acid 66 to 69, there was a significant difference. We can use the properties of the amino acids located at position 66-69 to further understand why there is such a huge energy difference.

Algorithm Analysis

As previously mentioned, the Lennard-Jones potential and electrostatic energy calculations are in a nested loop; thus, the time complexity of the algorithm is theorized as O(N^2). In order to verify this estimate, a python script was generated to test the time complexity with synthetic sequences. This python script obtains data for time complexity analysis by running the protein energy scoring algorithm on randomly generated protein chains of N length up to N=200. The obtained data confirms that the algorithm runs at an O(N^2) speed and has been graphed below.



Discussion

The program was successfully able to calculate an internal energy metric/score for both the protein conformations. Based on the results we can conclude that protein conformation #2 is more favorable as it had a lower score. Theoretically, this difference in energy can be explained by structural differences between the two conformations. For example, conformation #2 may exhibit no “clashes” (two atoms being too close to one another), hydrophobic residues being mostly buried, polar residues on the surface with no charges buried. Looking at the graph for local energy differences, we can conclude that one or more of the structural differences listed above was absent in positions 66-69 for conformation #1. This would explain the tremendous energy difference at this position.

While programing our algorithm we had considered the time complexity. Our initial understanding was that bonded interactions are local. Due to this reason, we would expect linear computational complexity (O(N)), where N is the number of atoms in the molecule considered. However, direct computation of the non-bonded interactions involves all pairs of atoms. Due to this reason, we would expect quadratic complexity (O(N^2)). In addition to this reasoning we also realized that out code for Lennard-Jones potential and electrostatic energy calculations was in a nested loop. As a result, our final estimate for time complexity was quadratic (O(N^2)). This estimate was confirmed when we tested our algorithm on synthetic sequences and graphed the results. For the case in consideration, the program takes around 7 seconds which is quite lengthy. With a

sufficiently large amount of possible protein topologies for a single protein that need to be validated, the script could take a large amount of time to finish.

A compiled language such as C or C++ would have been a better fit for this project. This can be explained by the fact that compiled languages have their source code translated through a compiler. This translated code can be executed numerous times. The overhead for the translation is a one-time cost during compilation. On the other hand, interpreted languages like python repeatedly require parsing, interpretation, and execution. Thus, making them very costly and less efficient in comparison to compiled languages.

For future applications of this project we were thinking of using our energy calculating algorithm to classify proteins as mis-folded or improbable if their internal energy calculation doesn't match the ideal one. Once we have identified a misfolded protein, we can use our local energy difference calculating algorithm to pinpoint at which position of amino acid the energy difference stems from. Additionally, further analysis could include juxtaposing the ideal 3D protein structures and misfolded 3D protein structure to pinpoint differences as well.

Bibliography

[1] Breda A, Valadares NF, Norberto de Souza O, et al. Protein Structure, Modelling and Applications. 2006 May 1 [Updated 2007 Sep 14]. In: Gruber A, Durham AM, Huynh C, et al., editors. Bioinformatics in Tropical Disease Research: A Practical and Case-Study Approach [Internet]. Bethesda (MD):

[2] National Center for Biotechnology Information (US); 2008. Chapter A06. Available from: https://www.ncbi.nlm.nih.gov/books/NBK6824/

[3] Maiorov and Abagyan, 1998 V. Maiorov, R. Abagyan Energy strain in three-dimensional protein structures Fold. Des., 3 (1998), pp. 259-269 https://www.sciencedirect.com/science/article/pii/S1359027898000376

[4] UCSF Chimera--a visualization system for exploratory research and analysis. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE. J Comput Chem. 2004 Oct;25(13):1605-12.

[5] Chang, Raymond. *Physical Chemistry for the Biosciences.* Sausalito, CA. University Science Books, 2005. (498-500)

[6] “Lennard-Jones potential,” Wikipedia, 12-Mar-2020. [Online]. Available: https://en.wikipedia.org/wiki/Lennard-Jones_potential. [Accessed: 13-Mar-2020].

[7] D. Eisenberg and A. D. McLachlan, “Solvation energy in protein folding and binding.,” Nature.[Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/3945310. [Accessed: 13-Mar-2020].