LSSPRO: a Local Small Scale PROtein folding engine

Markus Svedenheim
markus@svedenheim.se

Malcolm Alencar
malcolm@alencar.se

Kungsholmens Gymnasium
February 13, 2023
Under the guidance of Anders Häggqvist

Abstract

Understanding the three dimensional structure of a protein is an important part of molecular biology as it helps us understand both the functional properties of the proteins. There are multiple accurate implementations for calculating the three dimensional structure of a protein that employ deep learning. In order to better understand the challenges that have prevented traditional computed simulations from achieving an acceptable level of accuracy, this project simulated the folding of the mini-protein TRP-cage *in-silico* through traditional computation by approximating relevant protein energy functions. The simulations show a significant discrepancy between the reference and simulated protein and demonstrate various challenges of traditional computation. Further development was out of scope for this project but various approaches could be implemented to improve the results of the simulation, however it is unlikely if not computationally unfeasible to achieve the same level of accuracy as state-of-the-art deep learning approaches.

Contents

1 Introduction

Proteins are the basic workhorse of any cell. Comprised of chained together amino acids, proteins play vital roles in our bodies taking the form of enzymes catalyzing various chemical processes, hormones acting as neurotransmitters and key parts of our immune system, to mention a few. The vast variance in function is in turn dictated by the three-dimensional shape of the protein, also referred to as its structural conformation. The conformation of a protein is regulated by two cardinal variables: the combination of amino-acids, as well as the biological environment in which the protein is present. Changes of the latter causes proteins to regulate their shape throughout their life cycle and enables proteins to react to various stimuli. It can be said that the environment primarily dictates the state of the protein, for example when an arbitrary protein A binds to a presence signal substance θ, enabling it to inhibit certain functions of the cell. The combinations of amino acids, in turn, rather dictates the overall function of the protein. [Fürst, 2009]

1.1 A brief history of the protein folding problem

Because the shape of the protein is directly correlated with its function, the order of amino acids and subsequent structural arrangement of a protein is of value for a variety of biomedicinal and biochemical applications [Dill and MacCallum, 2012]. While the method of analysis for the primary structure of a protein is well established [Luo et al., 2011], the analysis of any three-dimensional conformation is far more complex and time consuming requiring x-ray crystallography for larger proteins [Uzman, 2003].

This has caused the research of the past decades to increasingly focus on structural prediction of proteins, instead of actual hands-on analysis, denominating the goal of rapid, high-resolution structural prediction of proteins as "the protein folding problem" [Dill and MacCallum, 2012].

The relevance and viability of a simulatory approach has increased bilaterally with the improvements of computational hardware, prompting rapid development of new com-

putational methodologies [Kuhlman and Bradley, 2019]. In recent years, the focus of the field has gravitated towards employing machine learning engines, with deep learning approaches being the most successful [Torrisi et al., 2020]. Deep learning is a subcategory of machine learning requiring large computational resources as well as sufficiently large and qualitative datasets[Torrisi et al., 2020]. Datasets would in the case of proteins be the amino acid chains as well as the folded three dimensional structure. In that way, a deep learning engine could predict the three dimensional structure of a protein using patterns observed between amino acid sequence and folded conformation in the dataset.

The current state of the art regarding deep learning applications to the protein folding problem is that of AlphaFold[Kuhlman and Bradley, 2019]. Developed by OpenAI, AlphaFold predicts the three dimensional structure of proteins based on their primary structure with an average root mean square deviation (RMSD) of 0.32 Å. Considering that the diameter of a carbon atom is approximately 1.5 Å its predicitons are considered accurate for usage in further biomedical research [Jumper et al., 2021].

1.2 The biology behind protein folding

As proteins are transcribed in the ribosome they first assume a primary structure and thereafter fold themselves to assume their three dimensional minimum energy structure [Fürst, 2009]. The folding is commenced by the formation of a molten globule, an intermediary stage between the de-natured and natured state of the protein [Eaton, 2021]. Molten globules are characterized by an overall tight packing with exception for amino acid side chains who have yet to stabilize the protein[Eaton, 2021]. The formation of the molten globule in human cells is primarily dependent on the attraction and repulsion between the hydrophobic and hydrophilic regions of across the polypeptide chain [Eaton, 2021]. As the protein is transcribed into the cytoplasm of the cell, hydrophobic regions gather in the center while hydrophilic areas protrude outwards toward the polar cytoplasm[Eaton, 2021]. This results in a tightly packed core of the protein, as the hydrophobic ends align internally and give way to favourable Van der Waal interactions

[Kuhlman and Bradley, 2019]. Side chains generally rotate freely around their own axis. However, in protein folding and especially in the core of a formed globule, side chains conform into an available set of rotations, referred to as rotameres [Jumper et al., 2021]. Rotamere prediction generally requiring high-accuracy prediction due to their intricacy, variability and dependence on already simulated superordinate variables such as backbone conformation[Torrisi et al., 2020]. Globule formation and backbone conformation demand a lower resolution when processing as they are not dependent on superordinate variables to the same extent as rotamere predicition[Eaton, 2021]. Comparatively, hydrophobic interactions are generally of the high importance in determining the final structure, as they are cardinal in deciding the backbone structure. Electrostatic interactions allow for the formation of hydrogen bonds and are core to stabilization and rotamere adjustment[Eaton, 2021].

1.3 Methods of prediction

Folding mechanics are complex in nature and dependent on a variety of factors that are often best approximated by a biophysical approach, through Newtonian physics predicting how the amino acids conform. These approaches can generally be divided into to two distinct categories: template-based prediction, where already known structures of certain fragments and similar proteins are combined to construct the protein; as well as template-free modeling where the protein is constructed solely based on the amino acid sequence. The second approach in theory allows for simulation of any protein, even if the structure and composition varies vastly from other analyzed proteins. By step-wise bringing the model nearer to the native structure according to the laws of Newtonian physics it would be possible to create a novel protein only using the amino acid sequence. This strategy is commonly referred to as molecular dynamics-based refinement and works by simulating the target protein sequence in water,, using protein energy functions and analyzing the minimum-energy outcome. A protein energy function is a combination of algorithms for determining both the potential energy of a protein, as well as how that en-

ergy might be reduced. The attraction of oppositely charged atoms would be an example of this.[Kuhlman and Bradley, 2019]

An issue of molecular dynamics-based refinement is that proteins often arrange into conformations that hold a local energy minima. A local energy minima is a structural conformation in which no adjustments decrease the potential energy of the protein. Even though there may exist other conformations that are lower in energy, there are no viable folding pathways to those minima. The volatile *in-vivo* environment in which proteins are folded prevents the structure from locking into local energy minima formations, resulting in the protein structure eventually reaching the global energy minima conformation. When computing the structure using molecular dynamics-based refinement, there is significant risk that the refinement converges toward a local energy minima, instead of a global minima, as the energy environment of the actual protein is not replicated beyond the protein energy function. [Kuhlman and Bradley, 2019]

1.3.1 Template-free prediction through traditional computation

Protein energy functions can be simulated using traditionally written code, placing a straight sequence of amino acids. By then modeling appropriate protein energy functions, for example a physics based model for electrostatic forces depending on the electrostatic charges on the atoms of the amino acids [Kuhlman and Bradley, 2019], it is possible to gradually refine the protein conformation toward a lower energy state[Singharoy et al., 2016]. There are two major concerns with this approach, the first being that developing accurate protein energy functions that replicate the energy environment of an arbitrary protein is complex, something that researchers has resembled to "finding a needle in a haystack" [Dill, 1993]. The second being that the accuracy required in simulation, with millions or even billions of simulatory steps required, makes the approach computationally expensive [Kuhlman and Bradley, 2019].

1.3.2 Template-free prediction through deep learning

The inherent need to generalize the predictive process in order lower computational strain has lead the field toward employment of artificial intelligence and deep learning [Torrisi et al., 2020]. These various deep learning implementations rely on taking in large amounts of data from the Protein Database (PDB) on which a prediction model is trained with a leviathan computational effort [Kuhlman and Bradley, 2019]. The prediction model is then able to, with relative computational ease, predict the shape of a novel protein given its primary structure. The approach has proven to be extremely fruitful, with implementations such as AlphaFold and RoseTTAfold achieving an accuracy of 0.4 Å [Baek et al., 2021]. This is, considering the size of a carbon atom being 1.5 Å, a more than acceptable result.

1.3.3 Template-free small scale prediction engine

While deep learning algorithms and traditional computation can be viable alternatives for predicting the shape of a protein, they are often limited by the available computational power and time required to model a proper environment to simulate an accurate folding process. When creating a small scale prediction engine, deep learning requires too much hardware and data, instead it is mainly preferable to simulate to folding of a protein by creating a small scale physic model with only the most paramount physics forces [Kuhlman and Bradley, 2019]. Python is language commonly used for this purpose with regard to its vast library of computational molecular biology implementations. However, a lack of computational speed has been found to hinder the language when performing larger calculations [Fourment and Gillings, 2008].

The speed of the core algorithms is heavily dependent on the language in which it has been developed but by creating a fast algorithm with an acceptable big O notation the computational complexity can be improved. Big O notation refers to the amount of iterations as a function of the cardinality of the input set. As such, an algorithm with a linear O-notation such that $O(n) = n$ would execute faster than an algorithm

implementing a quadratic O-notation such that $O(n) = n^2$. With the difference becoming greater as the cardinality of the set increases. [Rubinstein-Salzedo, 2018]

Considering the field of research surrounding *in-silico* protein folding models has converged towards a deep-learning approach, the purpose of this study is to examine the viability of a traditionally computed protein folding simulation utilizing newtonian-based protein energy functions. Considering the demanding nature of traditional computation, the protein selected to examine the output of the simulation is tryptophan-cage (TRP-cage), a 20 amino acid residue short protein significantly more stable than alternative mini proteins[Neidigh et al., 2002]. TRP-cage is not a functional protein in the body, though it is thought to be the smallest protein-like folding motif which is beneficial for computational reasons.

1.4 Aim

The study aims to, through simulation with Python on consumer-level hardware, predict the structure of the protein TRP-cage and evaluate the attainable level of accuracy and whether or not the protein energy functions stabilize the protein.

1.5 Theory

Apart from the molecular dynamics of folding a protein, this research is heavily dependent on a basic understanding of linear algebra: A vector ($\vec{v}$) is a variable containing a magnitude and a direction and can be used to represent either a position or a force. The magnitude of a vector is denoted as $\|\vec{v}\|$ and its direction as $\hat{v}$. Torque is calculated according to:

$$\vec{\tau} = \sum_i \vec{r_i} \times \vec{F_i} \tag{1}$$

where:

$\vec{\tau}$ = torque vector on the segment calculated as a sum of cross products

$\vec{F_i}$ = net force with its origin in a position

$\vec{r_i}$ = radius vector of the position

2 Method

The amino acid sequence of Trp-cage is loaded in its primary form. Each atom on the protein is associated with three variables: Its element ($E \in \{C, H, N, S\}$), its position (P) as a vector, and its partial charge (Q) calculated in accordance with Biotite's [Kunzmann and Hamacher, 2018] methods for calculating *partial equalization of orbital electronegativity* (PEOE) of a molecule. The protein is also structured into segments divided by each peptide bond, the protein is folded at these peptide bonds. The folding process is divided into several iterations where folding is done in a step-wise process: A torque vector (τ) is calculated according to formula 1, where the net force is calculated as a sum of $\vec{F_e}$, $\vec{F_h}$, and $\vec{F_b}$ (formula 2, 3, 4); The segments for which the torque has been calculated are then rotated in the direction of the torque, where $\theta = step \cdot \ |\tau\|$ with *step* representing the step and θ being the resulting angle of rotation. A shorter step length means that τ is calculated at more points resulting in a higher resolution of the simulation.

The angle of the bond is limited to 90 deg and in the case of a rotation causing atoms to collide, the rotation is undone. After each iteration, the average distance between each corresponding atom in the simulated model and the reference model [Neidigh et al., 2002] is calculated, following transformations to align their translation and rotation, this is also done for the previous and current iteration of the simulated model.

The electrostatic force is calculated according to:

$$\vec{F_e} = k_e \cdot \frac{Q_1 \cdot Q_2}{\|\vec{d}\|^2} \cdot \hat{d} \tag{2}$$

where:

$\vec{F_e}$ = electrostatic force vector on the atom

Q = partial charge of the atoms

$\vec{d}$ = distance vector from $P_2 \rightarrow P_1 = \Delta P$

k_e = constant to scale the weight of the electrostatic force

The hydrophobic force is calculated according to:

$$\vec{F_h} = k_h \cdot (\overline{|Q|} - |Q|) \cdot \vec{r} \tag{3}$$

where:

$\vec{F_h}$ = hydrophobic force vector on the atom

Q = partial charge of the atom

$\vec{r}$ = distance vector from P to the molecules center

k_h = constant to scale the weight of the hydrophobic force

Hydrogen bonds occur when E_1 and E_2 are either hydrogen and oxygen or hydrogen and nitrogen, it is then calculated as:

$$\vec{F_b} = \begin{cases} k_b \cdot \hat{d} & \text{if } \|\vec{d}\| < D_b \\ 0 & \text{if } \|\vec{d}\| \geq D_b \end{cases} \tag{4}$$

where:

$\vec{F_b}$ = force vector of the hydrogen bond

E = the element of the atom

D_b = minimum distance for hydrogen bond to form

$\vec{d}$ = distance vector between atoms

k_b = constant to scale the weight of hydrogen bonds

During the simulation the following constant were defined as:

$k_e = 1.0 \cdot 10^{-4}$

$k_h = 1.0 \cdot 10^{-1}$

$k_b = 1.0 \cdot 10^{-2}$

3 Results

During the folding process the three dimensional structure of the protein follows an initial curve succeeded by a more stable linear shape where only the ends are curved. This state is reached after approximately 300 iterations whereas collected data shows a decrease in movement between iterations and movement relative to the reference model after 150 and respectively 250 iterations.

3.1 Folding Process

Performing 500 iterations of folding with $step = 8 \cdot 10^{-4}$ on the simulated model yields a molecule with general linear shape, the ends of the protein are more twisted and fold in on themselves (figure 1). This general shape is achieved following 300 iterations of the simulation and then consists throughout the remaining 200 iterations, with the exception of general rotation of the whole molecule. Prior to 200 iterations the shape of the model is more curved and more closely resembling of the reference models general shape, this curve is achieved around 150 iterations, following which the model starts to twist around itself and finally reach its final state. The process of this folding is available in the attached animation.

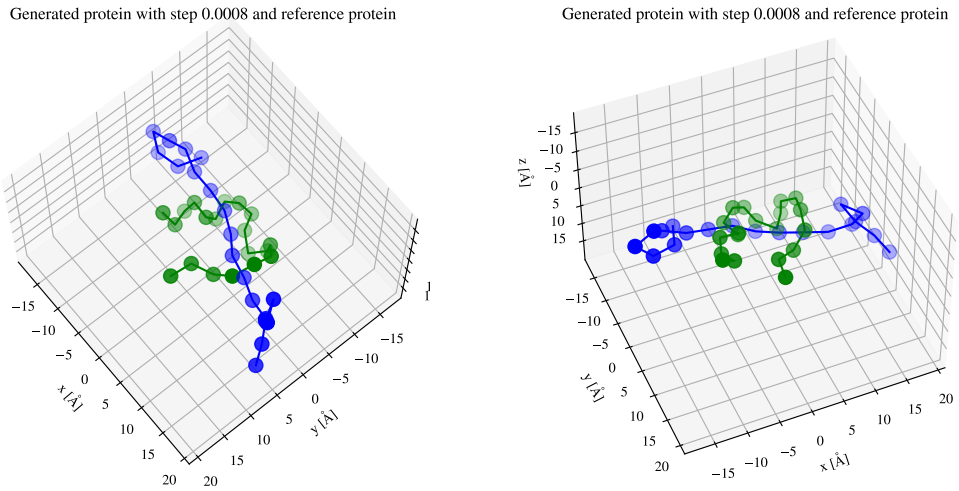
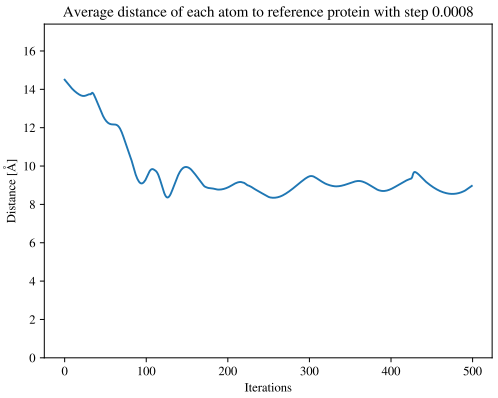


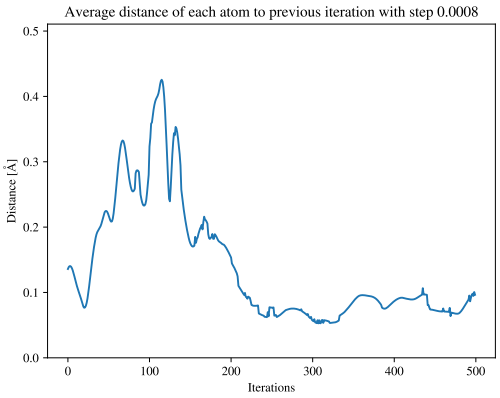
Figure 1: The figure depicts the final three dimensional position of each nitrogen atom following a peptide bond in the generated (blue) and reference (green) models. The models do not closely overlap and the generated model is a mainly linear with curved ends. The two models are captured from different angels with the same end remaining at the same side.

3.2 Gathered Data

Data provided by the simulations show the shape changing less as more iterations are performed, reaching a notable low point after approximately 200 iterations (figure 2b). This minimum occurs at roughly the same iteration as the distance to the reference protein initializes its stabilization (figure 2a). If the simulation is instead run with $step = 8 \cdot 10^{-3}$ it provides contrasting results (figure 3a and 3b). Instead of the smoother movement of the model referenced by figure 2a and 2b the movement of the former is more abrupt and never manages to reach a state of relative stability. It does, however, provide a closer overlap with the reference model in comparison with the former *step* while also needing notably fewer iterations to reach this state (figure 3a), although the model provided is not as stable (figure 3b) as the former (figure 2b).

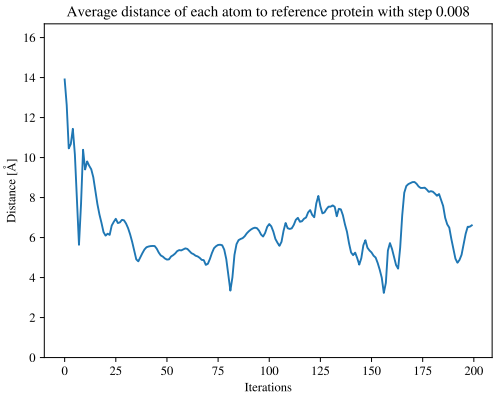


(a) Shows the average distance of each atom compared to the reference model of Trp-cage over a total of 400 iterations. The graph shows a rapid initial decline in distance, approximately 6 Å, over 100 iterations. Thereafter, the distance stabilizes around 9 Å from the reference protein and fluctuates between 8-10 Å for remaining iterations.

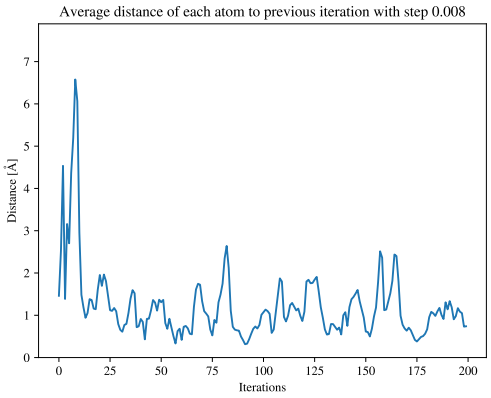


(b) Shows the average distance between each atom compared to the previous iteration. The graph implies an initial spike in movement, correlating with the rapid decrease in distance shown in figure 2a, followed by a drop to around 0.1 Å following iteration 200. The average distance is at most slightly above 0.4 Å and at its lowest around 0.05 Å.

Figure 2: Illustrates the folding process of the generated model, with figure 2a being the average distance to the reference model respectively figure 2b as the movement of the model as iterations increase. Figure 2b shows an approximate derivative of figure 2b.



(a) Shows the average distance between each atom compared to the reference model. The graph implies an early decline in distance, reaching a general minimum of 5.0 Å around iteration 60, followed by an increase to around 8.0 Å near iteration 120. The average distance reaches its global minimum at iteration 160 where it is around 3.5 Å.



(b) Shows the average distance between each atom compared to the previous iteration. The graph implies an initial spike in movement, correlating with the rapid decrease in distance shown in figure 3a, followed by a decrease to an average of 1.0 Å but with a high deviation after only 15 iterations, this pattern persists for the remaining iterations. The average distance is at most around above 6.5 Å and at its lowest around 0.5 Å.

Figure 3: Illustrates the folding process of the generated model, with figure 3a being the average distance to the reference model respectively figure 3b as the movement of the model as iterations increase. Figure 3b shows an approximate derivative of figure 3b.

4 Discussion

The core of an analysis of the results must take its start in an evaluation on attained accuracy as well as the perceived amount of stabilization. If the protein cannot be regarded as stable, the efficacy of the simulatory engine is in question. This can be deduced by observing whether or not the protein converges to any given conformation, regardless of the accuracy. Nevertheless, accuracy is an expected central question which can be tackled through examining the difference between the native state of the protein.

4.1 Stabilization of the protein

From figure 2a the amount of movement compared to the previous iteration immediately builds to a spike and subsequently mellows out for the last 300 iterations. An entirely static and stabilized protein would clearly have zero movement or close-to-zero when compared to the previous iteration, despite this figure 2a clearly shows and average movement of 0.05-0.1 Å for over 300 iterations. This could be explained with the lack of friction in the simulation allowing a set repeating conformations to toggle between each other in a repeating pattern in a kinematic loop. To conclude this, the simulation would have to be carried out for a larger amount of iterations.

4.2 Similarity to the reference protein

As can be seen in figure 2b the average distance of the atomic positions comparing the simulated conformation with the x-ray crystallography-analyzed conformation decreases over a span of 500 steps. This metric alone does not imply that the simulated molecule has begun to stabilize, however, it can be used as a measurement of whether the approximation is converging toward a conformation with improved similarity to the reference protein. The initial large changes in average distance is consistent with how a protein would fold *in-vivo*, where folds of the backbone structure cause large conformational changes, essentially formation of the molten globule state. The flattening of the graph

between 200-500 iterations would correlate with minor adjustments that stabilize the protein to its native state. As can be observed in the graph, the average distance from the reference protein is approximately 9 Å, compared to the 1.5 Å diameter of a carbon atom, the discrepancy is significant which is also clearly illustrated in figure1.

4.3 Optimal scale of step

As the step length was arbitrarily selected, the simulation was conducted in the same manner for 200 iterations, however the step length was increased by one order of magnitude as this granted a more stable final conformation while preventing an early lock-in into a local energy minima. The overall trends observed in figure 3a and 3b were similar to those in 2a and 2b respectively, however an increase in fluctuation is observed in the simulation with a higher step length. This indicates that a step length is inversely correlated with the accuracy of the simulation. This would also affect the amount of kinematic looping as the protein stabilizes, as described in **section 4.1**. It naturally follows that a change step length would require an inversely correlated change in computational effort and as such the determination of the step length must be weighted in accordance to available computational power.

4.4 Cause of inaccuracies

As observed in all conducted simulations, the simulated protein never achieves a similarity closer than 8 Å while stabilization is also lackluster, with perceived noise movement below converging around 1 Å. The primary cause of this is believed to be the protein energy function for calculating hydrophobic interactions, as it for the sake of the scope of the study, specifically making the simulation feasible on consumer-level hardware, was simplified to a large degree. Instead of simulating individual water molecules, force vectors aimed at the center of the protein were applied depending on the level of hydrophobicity.

4.5 Possible improvements

As the level of accuracy and stabilization were both sub-standard there is an interest to examine elements of the method that can be improved. This mainly pertains to the lackluster calculation of hydrophobicity, but also resource-saving measures such as program rewrites in computationally efficient programming languages.

4.5.1 Language speed

If the scope of consumer-level hardware is to be maintained, a restructuring of the simulation engine should be considered in order to achieve an improved O-notation, in laymans terms to improve the scalability of the engine to larger proteins. Further, rewriting the program in the C/C++ programming language would provide faster calculations when compared to Python which was used in this project. Although this also holds true for almost all other compiled languages as well the main issue is the time investment required to produce a similar algorithm without high-level libraries provided. An alternative solution providing some middle ground is also Cython, a compiled language built for integration with C and Python, which also provides adequate library integration to simplify the development phase.

4.5.2 Hydrophobicity

Since the biggest factor in deciding the structure of the protein is hydrophobic interaction, it may be of interest to evaluate new protein energy functions for simulating amino acid interactions with surrounding water molecules. Directly simulating individual molecules would regardless of efficiency improvements mentioned under subsection 4.5.1 likely prove unfeasible with regards to the associated computational cost. The LSSPRO implementation carries a remarkably linear O-notation whit regard to calculation of hydrophobic forces. However, it has proven necessary to reevaluate this approach as the current method does not consider a volatile point of hydrophobic attraction and instead focuses mainly on the algorithms time complexity.

4.5.3 Time complexity

As mentioned to algorithm implements a $O(n) = n$ complexity for calculation of hydrophobic forces. Problematically the calculation of electrostatic forces and hydrogen bonds does not follow the same pattern and instead implements a $O(n) = n^2$ complexity, this is because all atoms depend on the attractive and repulsive forces of each other and each atom's net force must be calculated with all others in mind. This is a clear hindrance regarding the time complexity of the algorithm but could be improved by simplifying and grouping together atoms with similar properties prior to the folding process. By relocating the charges of the hydrogen atoms to their bonded atoms, a vast majority atoms can be ignored during the calculations and the time complexity improved significantly.

4.5.4 Further research

This project shows little potential in toppling the industry standard deep learning methods, even if necessary improvements mentioned under 4.5 were to be implemented. However, there may still be an interest in exploring meaningful improvements of the system for small scale applications and educational purposes.

4.6 Conclusion

In conclusion, despite the simulation showing trends toward both stabilization and the native conformation, the discrepancy in stability and accuracy of the prediction means that neither the desired nor implemented protein energy functions are viable for research purposes. While there is significant progress to be made to improve the engine, it remains that a traditionally computed protein folding engine may never replace deep learning approaches.

References

[Baek et al., 2021] Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G., Wang, J., Cong, Q., Kinch, L., Schaeffer, R., Millán, C., Park, H., Adams, C., Glassman, C., Degiovanni, A., Pereira, J., Rodrigues, A., Dijk, A., Ebrecht, A., and Baker, D. (2021). Accurate prediction of protein structures and interactions using a 3-track network.

[Dill, 1993] Dill, K. A. (1993). Folding proteins: finding a needle in a haystack. *Current Opinion in Structural Biology*, 3(1):99–103.

[Dill and MacCallum, 2012] Dill, K. A. and MacCallum, J. L. (2012). The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046.

[Eaton, 2021] Eaton, W. A. (2021). Modern kinetics and mechanism of protein folding: A retrospective. *The Journal of Physical Chemistry B*, 125(14):3452–3467. PMID: 33724035.

[Fourment and Gillings, 2008] Fourment, M. and Gillings, M. R. (2008). A comparison of common programming languages used in bioinformatics. *BMC bioinformatics*, 9:1–9.

[Fürst, 2009] Fürst, P. (2009). Basics in clinical nutrition: Proteins and amino acids. *e-SPEN, the European e-Journal of Clinical Nutrition and Metabolism*, 4(2):e62–e65.

[Jumper et al., 2021] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596:1–11.

[Kuhlman and Bradley, 2019] Kuhlman, B. and Bradley, P. (2019). Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, 20.

[Kunzmann and Hamacher, 2018] Kunzmann, P. and Hamacher, K. (2018). Biotite: a unifying open source computational biology framework in python. *BMC Bioinformatics*, 19(1):346.

[Luo et al., 2011] Luo, Y., Matejic, T., Ng, C.-K., Nunnally, B., Porter, T., Raso, S., Rouse, J., Shang, T., and Steckert, J. (2011). 8 - characterization and analysis of biopharmaceutical proteins. In Ahuja, S. and Scypinski, S., editors, *Handbook of Modern Pharmaceutical Analysis*, volume 10 of *Separation Science and Technology*, pages 283–359. Academic Press.

[Neidigh et al., 2002] Neidigh, J. W., Fesinmeyer, R. M., and Andersen, N. H. (2002). Designing a 20-residue protein. *Nature Structural Biology*, 9(6):425–430.

[Rubinstein-Salzedo, 2018] Rubinstein-Salzedo, S. (2018). *Big O Notation and Algorithm Efficiency*, pages 75–83. Springer International Publishing, Cham.

[Singharoy et al., 2016] Singharoy, A., Teo, I., McGreevy, R., Stone, J. E., Zhao, J., and Schulten, K. (2016). Molecular dynamics-based refinement and validation for sub-5 Å cryo-electron microscopy maps. *eLife*, 5:e16105.

[Torrisi et al., 2020] Torrisi, M., Pollastri, G., and Le, Q. (2020). Deep learning methods in protein structure prediction. *Computational and Structural Biotechnology Journal*, 18:1301–1310.

[Uzman, 2003] Uzman, A. (2003). Molecular biology of the cell (4th ed.): Alberts, b., johnson, a., lewis, j., raff, m., roberts, k., and walter, p. *Biochemistry and Molecular Biology Education*, 31:212 – 214.