Physics 286 Final Report:

Protein Folding and an Examination of Metastable Tertiary Structures

Douglas McNally
Department of Physics
Miami University, Oxford, OH

Abstract

The dynamics of a simulated chain of amino acids (i.e. protein) are modeled. How the energy and length of the protein changes over time and as a function of temperature is discussed and results are compared with known simulations. From this data, information about the resolution of Levinthal's paradox and the energy landscape of a given protein as a function of the length of the protein is deduced. A novel algorithmic approach is introduced which allows for enhanced computation speed and therefore more potential calculations and more accurate results.

Introduction

Proteins are essential elements of biological life as it is known. Proteins play key roles in building muscle, storing energy, and the complicated ion flows that occur during neural activity.[1] Furthermore, many chemical reactions in the body are catalyzed by a special class of proteins called enzymes, and inter- and intra-cellular communication is handled by proteins. Truly these macromolecules are ubiquitous throughout biological processes of interest and therefore protein modeling is something of great utility and scientific importance.

On a very basic level, proteins are chains of amino acids that are covalently bonded together in some specific order. Amino acids are compounds containing a carboxyl group and an amine separated by a carbon with a side chain attached to it. A particular protein is composed of a particular sequence of amino acids, and that sequence alone with no regard to spatial configuration is known as the primary structure of the protein. In this scheme, two neighboring, covalently bonded proteins will always remain neighbors in this fashion.[2]

In nature there exist 20 well known amino acids (or monomers) that compose all proteins. Some proteins may only be comprised of 50 monomers while some may be as large as several thousand. A highly simplified, but still useful, model will be considered herein that confines a particular protein to a square grid that is n x n where n is the number of amino acids in the protein. In this model, the initial configuration will be either a straight line of proteins arranged sequentially with two adjacent amino acids being covalently bonded (figure 1), or a self-avoiding random walk (SAW) as is suggested on page 395 of ref. 2 (figure 2).

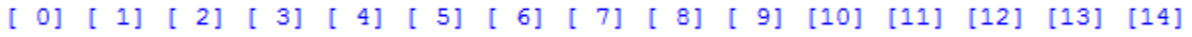

Figure 1 A simple linear arrangement of amino acids. The numbers indicate order.

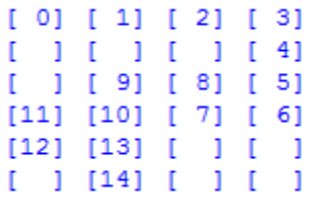

Figure 2 Amino acid arrangement generated by a self-avoiding walk (SAW). See Appendix A.

Real proteins are highly flexible molecules and their shape depends on many factors such as the temperature of the solution that they are in and the other chemicals in the solution (inside cells proteins are in a solution that is mostly water). As is intuitive, a large chain of amino acids can take on very many shapes, and some are more preferred than others. However only one very

particular orientation of the protein will be biologically active and this arrangement of the molecule is known nominally as its tertiary structure. Specifically this state has the lowest molecular energy configuration possible.[2] There are however other configurations that are so called "metastable" states which are at such low energies that when a protein reaches one of these configurations it is biologically inert but is "stuck" in this state which is analogous to a potential well.

From this information and interesting problem arises; that is that since there is only one biologically active state, the probability of this state occurring might be relatively low since the protein has $\sim 4^N$ possible configurations where N is the number of amino acids in the protein. And yet in nature, proteins are able to "find" this very specific biologically active state in a short amount of time (on the order of seconds), rather than what could be billions of years (assuming the protein spends $\sim 10^{-13}$ seconds in each folded state) for a large protein. This conundrum is known as Levinthal's paradox and is an interesting problem that is addressed in this modeling and in ref. 3 which will be considered in the analysis of the data obtained.

Computational Approach

In this simulation amino acids are assumed to be on a square grid as discussed above. This is first off a simplification because it restricts the system to two dimensions and ignores external variables acting on the protein aside from temperature. However the model still gives rise to useful results. The basic premise underlying the algorithms implemented here is that when the protein is in a given confirmation some amino acid in the chain is randomly selected. Upon being selected all of the 8 sites nearest to it in each direction to which it could potentially move (the so called "nearest neighbor" sites on the grid) are considered and one of these is randomly selected. Whether or not this move is possible is based on whether or not a covalent bond between amino acids is stretched, compressed, or broken. This model assumes that the covalent bond energy is so high that it cannot be overcome by molecular kinetics. If the movement is possible it is made if the change in energy of the molecule (ΔE_{move}) is negative or if the Boltzmann factor $\exp(-\Delta E_{move}/k_B T)$ where k_B is Boltzmann's constant and T is the temperature of the system is greater than a randomly generated number in the range [0,1). Note that T is treated in units of k_B for the sake of simplicity. This approach is used because the aim is for the molecule to reach lower energies and so movements that lower the energy will always occur and

there is some chance that disadvantageous moves will be made because of statistical (i.e. thermodynamic) fluctuation.

The approach as described up to this point gives rise to the obvious question of how exactly the energy of the molecule is determined and moreover what exactly it represents. The basic notion is that there will be some sort of energetic interaction between proteins that are adjacent to each other but not covalently bonded. This energy results from various forces such as Van der Waals forces, hydrogen bonding, and even interactions with the water molecules in the solution just to name some, however an actual calculation of this would be a large undertaking and in lieu of such values, random values were generated which turns out to still give reasonable behavior. In all cases these values were generated randomly in the range [-4, -2] and were stored in a symmetric 20 x 20 matrix (since there are 20 amino acids) where the (i, j) entry was equal to the (j, i) entry and the value at that entry corresponded to the interaction between a type i and type j monomer. In this same paradigm the amino acid types were simply represented by numbers 1-20 for the sake of simplicity, and in the language of ref. 2 the calculation for the interaction energy of the whole protein is given by:

$$E = \sum_{<m,n>} \delta_{m,n} J_{A(m),A(n)} \qquad (1)$$

where the summation is for all pairs of amino acids <m, n> and $\delta_{m,n}$ is 1 if amino acids m and n are nearest neighbors not connected by covalent bonds and 0 otherwise and $J_{A(m),\,A(n)}$ is the entry corresponding to amino acids of type A(m) and A(n) in the matrix containing the generated attraction energies values.[2]

Given this relationship and methodology, only the task of implementation remains; as it turns out the method of computation suggested in ref. 2 results in a dependence on an algorithm that is $O(n^2)$ or at best $O(n \log_2 n)$ because it requires carrying out the summation presented in eq. 1 which requires considering every amino acid relative to each other amino acid. Two key programmatic problems with this approach are that the algorithm is unsatisfyingly slow and it has a strong dependence on the length of the amino acid. And so given these problems, a superior technique was developed and implemented. The basic notion is that because of the way this system is being modeled (in Monte Carlo time steps), only one amino acid can move at a time. And so the obvious conclusion is that only other amino acids localized around the one that moved will affect any change in energy of the system and the rest will remain the same. This is the approach that was implemented by storing the total energy of the system and then adding or

subtracting from it the change introduced by the movement of just one amino acid. This algorithm turns out to be completely independent of the length of the protein and is essentially O(1) since at most 8 neighbor sites will need to be checked each time (4 for the previous position and 4 for the new position).

The last important computational method employed was that for calculating the average energy as a function of temperature. In ref. 2 it was suggested to average over $5x10^5$ Monte Carlo time steps to obtain the average energy for each of 20 temperatures (such as in figure 4 below). This gives the daunting number of 100 x 10^5 computations at minimum. If allowed to occur in a traditional sequential manner, and without the improved energy algorithm mentioned above, this task takes quite a long time to complete. However this is an embarrassingly parallel task and so the approach that was taken was to split each of these independent average energy calculations into subprocesses and therefore perform several groups of these calculations simultaneously. This improved the run speed greatly and allows for averaging beyond $5x10^5$ Monte Carlo steps with relatively little more time required. Obviously the speed gain from this will depend on the capabilities of a given CPU, but regardless the benefit is clear.

Results and Discussion

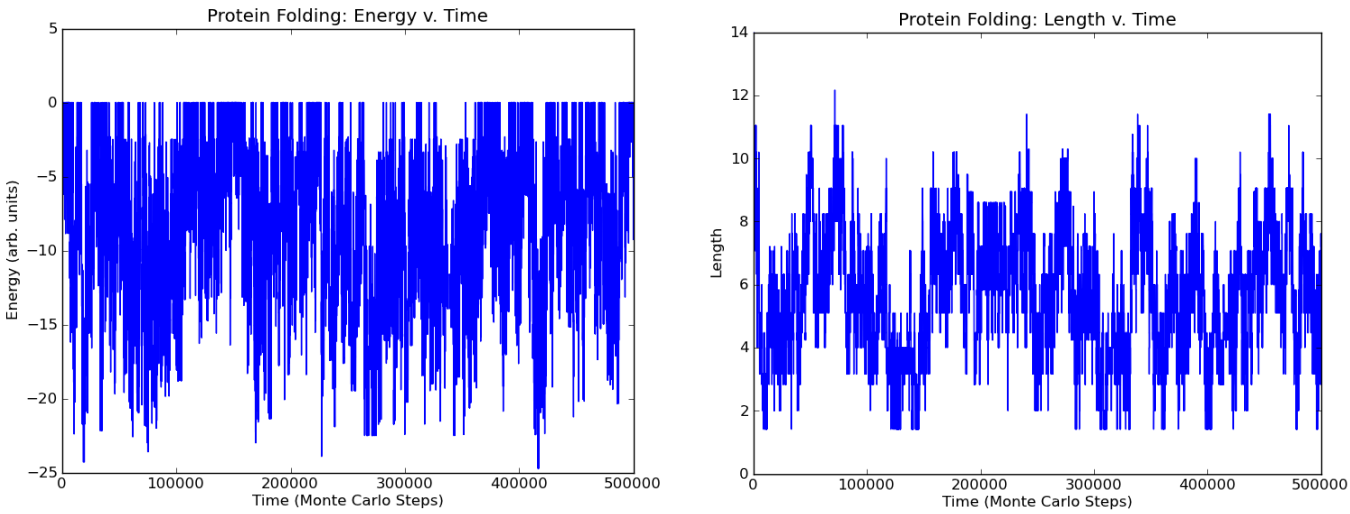

Figure 3 Plots of energy versus time (left) and length versus time (right) for a protein comprised of 15 amino acids at T=10 over $5x10^5$ Monte Carlo time steps.

Several simulations were conducted, and for the sake of comparison the set of amino acids and their matrix of attraction energies were held constant throughout the experiments so that the data

can be more accurately compared. Of particular interest is both how the energy of the protein and the length of the protein change in time and some sample data is plotted above in figure 3. Note that this system varies a lot in time and that no metastable state seems to be reached, i.e. the protein does not find a low energy value and stay there. This is because the temperature of the system is high and so trapping in a low energy state is very unlikely to occur since the Boltzmann factor is large for this temperature.

In general, it turns out that the average energy of the system increases as the temperature increases. This behavior is plotted in figure 4 which demonstrates a changing temperature and its effects on the average energy and length of the protein.

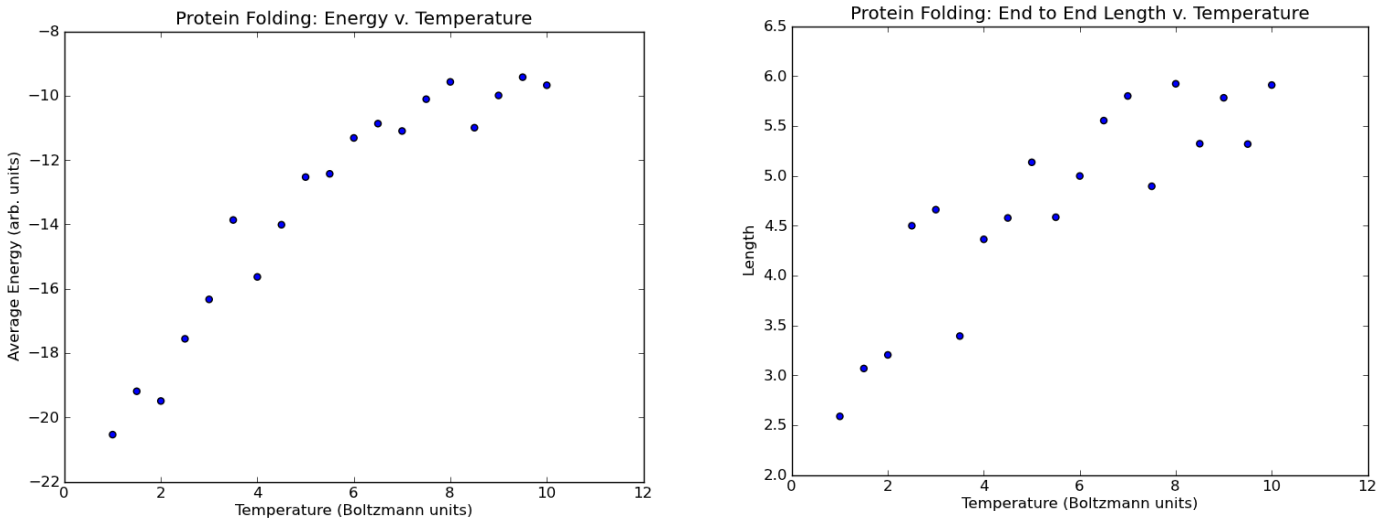

Figure 4 Energy versus temperature (left) and end to end length versus temperature (right). Each data point was averaged over $10x10^5$ time steps (an improvement on ref. 2) and the temperature was lowered in steps from T=10. There were 15 amino acids in the protein.

Note the correlation demonstrated in these plots and the clear trend toward higher energy (and length) for higher temperatures. From this data it can be concluded that finding a protein in a low energy metastable tertiary structure or even its optimal tertiary structure is much more likely in a low temperature system, and furthermore there will be drastically less variance in the system. And so the expectation is for the protein to search for some time and then fall to a low energy and more or less stay in this low energy configuration. This exact behavior is shown in figure 5 where the temperature is 1 instead of 10.

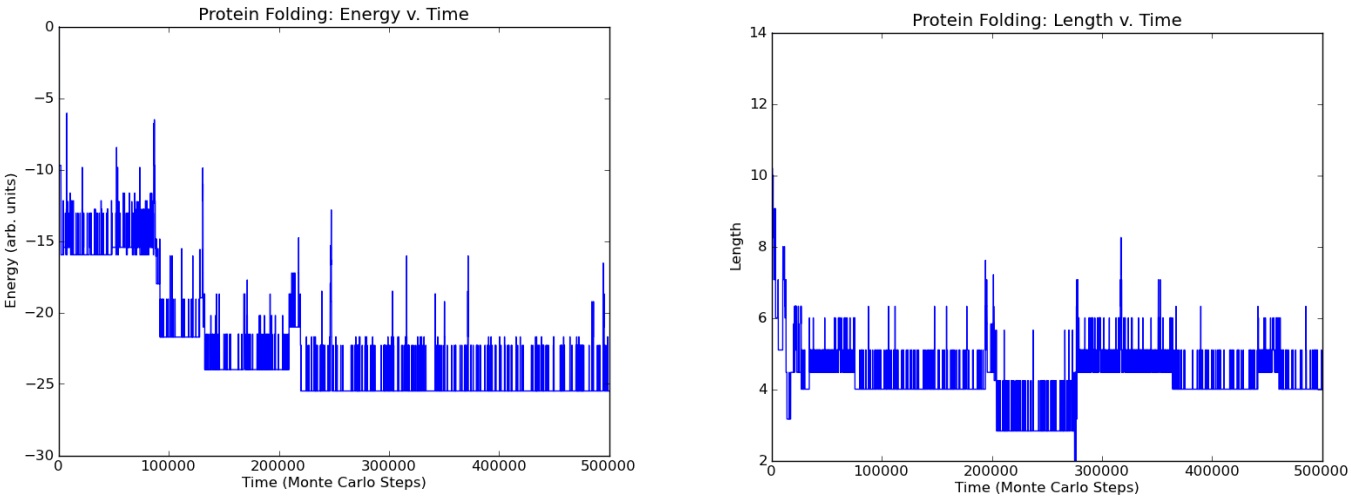

Figure 5 Plots of energy versus time (left) and length versus time (right) for a protein comprised of 15 amino acids at T=1 over $5x10^5$ Monte Carlo time steps.

And from this data, an interesting question that follows is how different are the metastable states reached by the protein. This was determined by allowing the simulation to run for $5x10^5$ Monte Carlo steps at T=1, and then comparing the final spatial configurations of the protein. Several such examples are shown in figure 6.

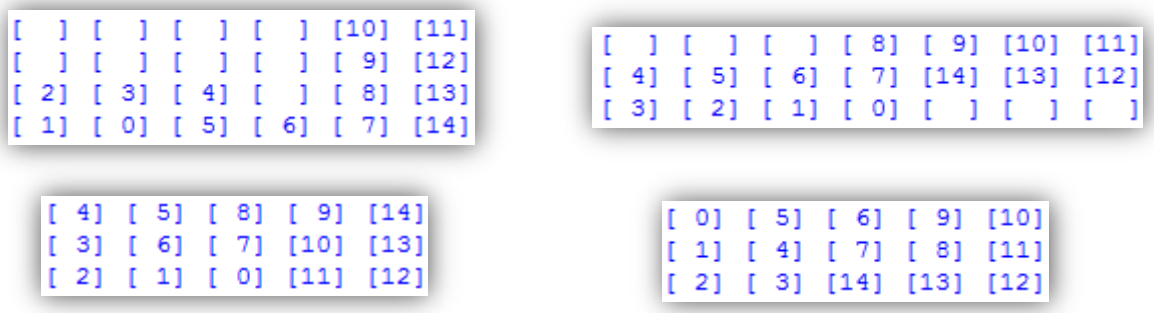

Figure 6 Four different metastable configurations for a protein with 15 amino acids.

From these few samples, it appears that metastable configurations can vary quite a bit, but they do tend toward a folded structure, which is not surprising since the energy of the protein decreases as the protein becomes more and more compacted. The energies of these metastable states varied between -28 and -23 (in arbitrary units of energy on the vertical axes of the energy plots).

Another simulation suggested in ref. 2 is a process called annealing which is essentially cooling the system slowly over time. The idea is that the protein will be able to sample a large number

of metastable states at high temperatures and find the correct state as the temperature goes low. This simulation is also considered here in figure 7. Notice that the energy of the final state of the protein in the T = 1 section of figure 7 is very low. In contrast with simulations that run for longer times at a constant lower temperature, this is a very optimal energy. And so it seems that the state obtained with this annealing process is better than those obtained at long times and so this annealing process is able to avoid some of the problems with metastability, however there is still a chance that the protein will get trapped in a metastable state. One important note though is that in a real physical situation a scenario similar to this annealing simulation is likely to occur because a given solution will cool gradually over time. It is possible that natural annealing experienced by a protein when a change is introduced to its environment may allow the protein to find its correct tertiary structure most of the time, and this may be sufficient because the natural world may be able to handle a small fraction of proteins failing to find the correct state.[2]

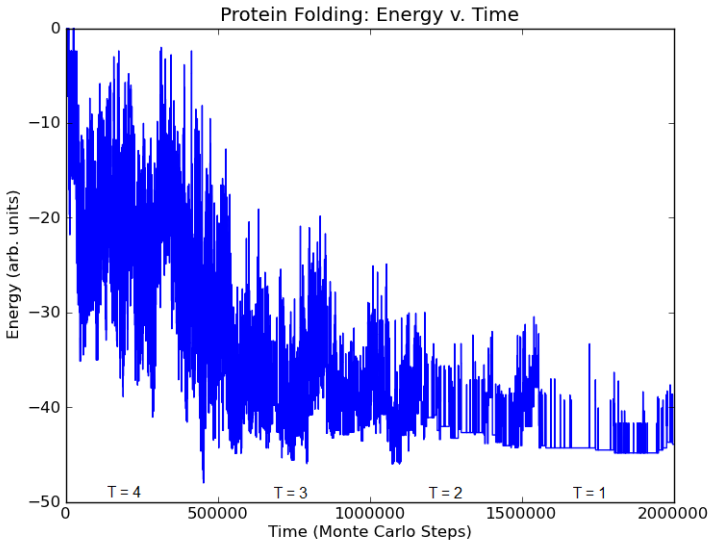

Figure 7 Annealing simulation with 30 amino acids in the chain. The temperature was reduced by 1 every $5x10^5$ Monte Carlo Steps

Conclusions

Some other interesting simulations that could be done with this same structure are to examine the effects of the number of amino acids in the protein on energy variance, the change in end to end length over time, and how the protein behaves with changing temperature – essentially repeating what is done here but change the size of the protein. This could be done programmatically with very little effort given the functioning program already implemented.

In general results obtained in *Computational Physics* (ref. 2) agreed well with those obtained here, and in some cases these results were better because of the computational precision granted by the improved algorithm discussed. Obviously the figures do not match up exactly because of the great degree of randomness introduced in these simulations, however the same general

behaviors were exhibited, and so with confidence it can be deduced that these simulations are accurate and in agreement with previous experiments.

As previously discussed, the problem of Levinthal's paradox is an intriguing phenomenon in the study of protein folding. There are some promising ideas for the resolution of this problem, and it is given treatment from an evolutionary standpoint in ref. 3. In the language used there, it is possible that early proteins were relatively short chains and they were able to find their tertiary structure through a mechanism known as three-stage random search which allows for the protein to fold on a reasonable time scale by the introduction of a transition state (this is discussed in more detail in ref. 3). The hypothesis goes on to say that since the early biotic environment was hot, unusually thermostable proteins were required such as those found in very primitive bacteria. From this point, proteins evolved into longer and longer chains and they had to fold on the same timescale as the small proteins. The suggested solution to this is that the evolved proteins had a large difference between the native and "non-native" (i.e. metastable) states and these proteins have a very pronounced global minimum (i.e. the lowest energy configuration – the biologically active tertiary structure) relative to the local minima. Allegedly a mechanism of transition from random-coil states to the tertiary state may be some intermediate state that varies with external conditions.[3]

With the simulations performed some key characteristics of protein folding were determined and the energy landscapes of various proteins were explored. The phenomena associated with metastability (so called local minima in the energy landscape) were studied. This concept may be of critical importance, and it is speculated in literature (ref. 3) that the future of structural prediction may lie in the derivation of a potential function rather than further simulations, although they may help in such a derivation. The applications of understanding what may be called protein physics are presently in being able to predict the structure of unknown proteins and then ultimately design new molecules with a desired shape.[2]

References

1. R. Callender, R. Gilmanshin, B. Dyer, and W. Woodruff, *Protein Physics*, Physics World, August 1994.
2. N. Giordano and H. Nakanishi, *Computational Physics.* Second edition. Prentice Hall, 2006.
3. A. Sali, E. Shakhnovich, and M. Karplus, *How Does a Protein Fold?*, Nature **369**, 248 (1994).

Special thanks to Hung Nguyen for some programming advice on how to optimize speed.

Appendix A Self Avoiding Random Walk

To initialize the structure of the protein, one possible approach is to construct it using a self-avoiding random walk. This is suggested on page 395 of ref. 2 which states "We might instead let it [the initial structure] be a self-avoiding walk; we will leave such study to the interested reader." The basic idea is that a "walker" moves along the square grid randomly and every place it goes it leaves behind an amino acid. The movements made are all to nearest neighbor sites on the grid and the direction of the next move is always a random choice. If the walker randomly chooses a location that it has visited before then the entire process starts from the beginning again (initially at point (0,0) on the grid) and this repeats until a random path is found that does not intersect itself so that the probability of choosing one path over another is not skewed. And important feature of this approach though is that it will almost definitely not work well or at all with large proteins (say more than ~20 amino acids) since the algorithm is forced to restart each time an intersection occurs and the probability of this occurring grows very quickly as the protein gets longer and so finding an allowed random walk can take a very large number of attempts. For the scope of this experiment the SAW initialization approach is used exclusively in simulations with 15 amino acids in the protein.

The difference in the behavior of the SAW and linear approaches was determined by running simulations with the same set of amino acids and attraction energies, but changing the way the initial setup was done. Plotted below are two such tests for comparison.

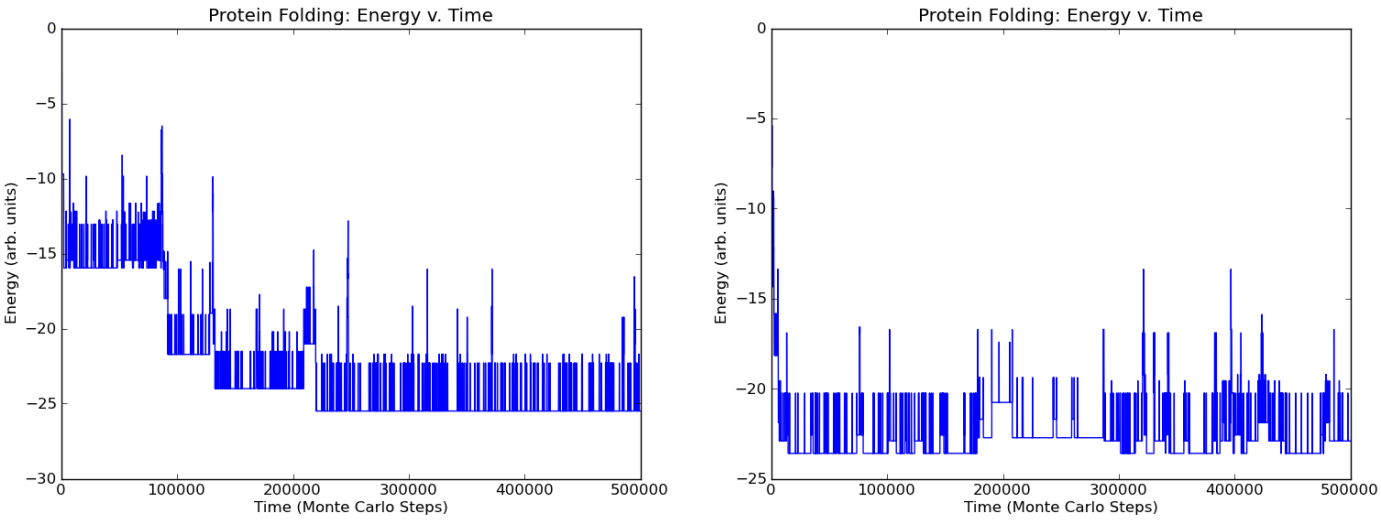

Figure A.1 Left: Linear initialization, T=1 and 15 amino acids in the chain. Right: SAW initialization, T=1 and 15 amino acids in the chain.

In all of the tests performed it was found that in general proteins initialized with the SAW method found a low energy state much faster than those initialized linearly. This same behavior can be seen in the above figures where the linearly initialized protein (left) takes more than 10^5 Monte Carlo steps to find a low energy state, but the SAW protein takes on the order of 10^4 or fewer Monte Carlo steps to do so. This makes a lot of sense because with the SAW method, the protein starts off in a fairly folded state and so it has more movement options initially and can even start out with a low, nonzero energy unlike the linear protein which will always have zero energy initially.

In the computational experiments discussed aside from this one, the SAW method was not used for the sake of consistency since experiments with large numbers of proteins cannot use it, but it does give some physical results and is probably more representative of a real protein.