Protein folding
Assignment 3, Computational Physics

Federico Rossi

April 2024

Abstract

This report presents a study of ...

1 Introduction

Proteins are a large biomolecules that are responsible of many functions such as catalysing biochemical reactions, providing structural support and DNA replication. They are a long chain of amino acids, linked together in a specific way. The bonds between the monomers are flexible and this allows the chain to fold into itself. This process is usually very favourable since the interactions between not adjacent monomers that appears after the folding are usually giving a substantial stabilization, lowering the total energy. After the folding is completed, the final 3D structure is what makes the protein truly active, thanks to pockets that contains not adjacent monomers held in specific near positions that are responsible for the catalytic behaviour. The final shape is strongly connected to the sequence of amino acids, as the interactions between them are different and eventually drive the folding towards a specific structure.
In this work, the process of protein folding is simulated using Monte Carlo methods, starting from an initial 2D representation of a protein and then extending the method to a 3D representation.

2 Methodology

The protein is initially represented as chain of 2D vectors, with a type, to indicate the amino acid in each position, numbered from 1 to 20. The initialization of the protein will always be random, assigning random amino acid types at each position.

$$[T_1, [x_1, y_1]], [T_2, [x_2, y_2]], \ldots, [T_N, [x_N, y_N]] \quad (1)$$

where T_i is the type of the amino acid and $[x_i, y_i]$ its coordinates. To evaluate the energy chages due to the forming interactions among not covalently bound monomers an interaction matrix J was defined as a 20 x 20 matrix containing uniformly distributed values between $-2k_b$ and $-4k_b$. Since each element corresponds to the energy due to the interaction between monomers, the matrix is generated symmetric since the contribution of i interacting with j is the same as j interacting with i. The matrix is represented in Fig. 1. The values are all negative, which means that all the interactions are stabilizing the structure. In reality, it is not uncommon to have repulsion between specific amino acids, especially when the lateral chains carry charges of the same sign.

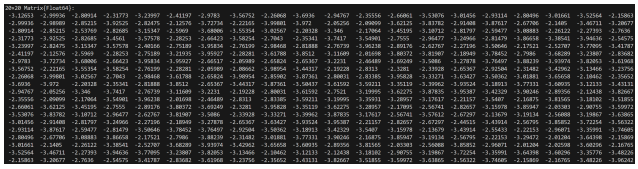

Figure 1: Symmetric interaction matrix generated with uniformly distributed values between -2 and -4, used for all the simulations in this work unless specified differently. The values as expressed in k_b units.

The energy associated with a given structure is given by summing the interaction contributions of not bonded monomer pairs.

$$E = \sum_{n,m}^{\text{neighbours}} J[T_n, T_m] \quad (2)$$

where the sum only runs for (n, m) neighbours and adds the term of J corresponding to their types.
For the Monte Carlo simulation, the Metropolis–Hastings algorithm was used as following:

1. Initialize a protein
2. Draw a random amino acid in the protein and look for possible transitions. If none are possible, draw another one. If more than one is possible, pick a random one.
3. Evaluate the ΔE associated to the transition, in k_b units.
4. Define $p = \min(1, e^{-\Delta E/T})$
5. Generate a random number $\alpha \in [0, 1]$
6. If $\alpha > p$ reject the step and revert to the initial protein, otherwise accept it.

Steps 2 to 6 are repeated N times, with N the number of amino acids in the protein, to complete a full MC sweep. A logger is created to store the energy, end-to-end distance and radius of gyration during the simulation.

3 First simulation

A random protein is initialized linearly and the results of the simulation of 100 sweeps at $T = 10$ are reported in Fig. 2 after 1, 10 and 100 sweeps.

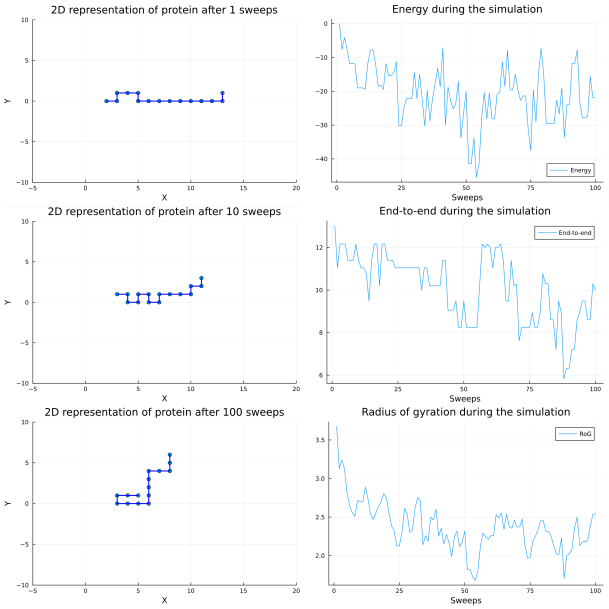

Figure 2: Simulation of a randomly generated protein with 15 monomers, initialized completely linear at $T = 10$. Geometries after 1, 10, 100 sweeps are presented, together with the energy, end-to-end distance and radius of gyration for each sweep.

From the figure, we can see the energy, the end-to-end distance and the radius of gyration seems to steadily decrease for about 20 sweeps and then start to oscillate quite a lot for the remaining sweeps. This is due to the initial structure that is evolving and folding as the simulation runs. To get a general idea of the average values, another simulation was run, letting the protein stabilize for 100 sweeps and then taking the running average of all the properties. In Fig. 3 the results are reported. After 100 sweeps, the values reported at each sweep i is the average of the first $i-100$ structures.

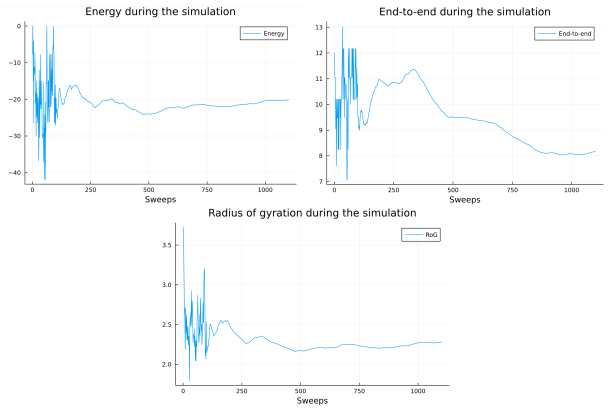

Figure 3: Another simulation of a randomly generated protein with 15 monomers, initialized completely linear at $T = 10$. Plots of energy, end-to-end distance and radius of gyration are shown, taking a moving average after 100 sweeps, for additional 1000 sweeps.

3.1 Lower temperature

Decreasing the temperature makes it more difficult to explore different geometries, as steps that make the energy increase are less likely to be accepted. This leads to a geometry that shifts very fast to a local minimum and is very stable in time, as shown in Fig. 4, where the results for a simulation on the same linear protein are reported at $T = 1$.

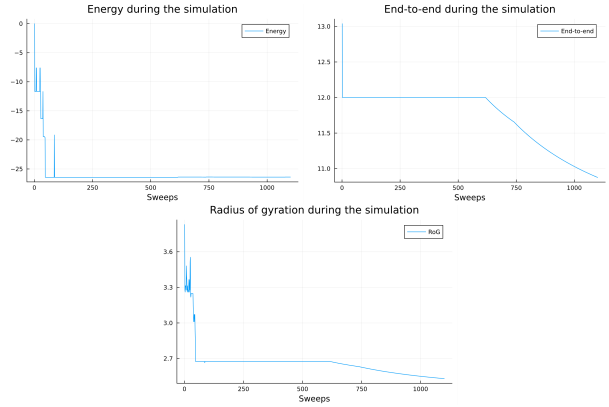

Figure 4: Another simulation of a randomly generated protein with 15 monomers, initialized completely linear at $T = 1$. Plots of energy, end-to-end distance and radius of gyration are shown, taking a moving average after 100 sweeps, for additional 1000 sweeps.

3.2 Increasing the size

Increasing the number of monomers from 15 to 100 increase the complexity of the system, that can now explore many more geometries. The dimensionality is changed and this is reflected in a longer time required to reach a steady state. The results are shown in Fig. 5.

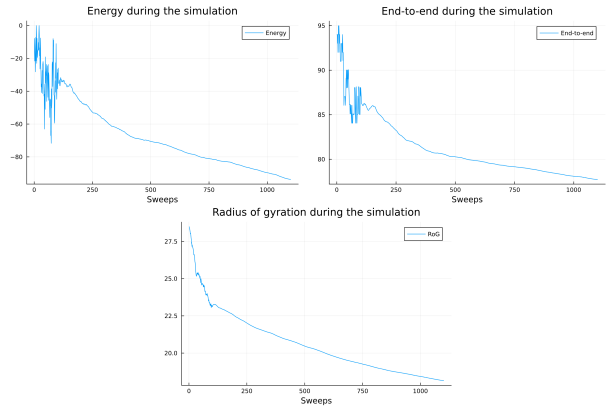

Figure 5: Simulation of a randomly generated protein with 100 monomers, initialized completely linear at $T = 10$. Plots of energy, end-to-end distance and radius of gyration are shown, taking a moving average after 100 sweeps, for additional 1000 sweeps.

4 Effect of temperature

A good practice is to start with an high temperature and decrease it slowly during the simulation. This allows for an initial exploration of many geometries that are progressively restricted, giving at the end a steady state. The results are shown in Fig. 6, where the simulation was run on linear proteins of 15 and 50 monomers (also 30 in the Appendix) decreasing the temperature at $T = 20, 10, 5, 3, 2, 1$ after 1000 sweeps.

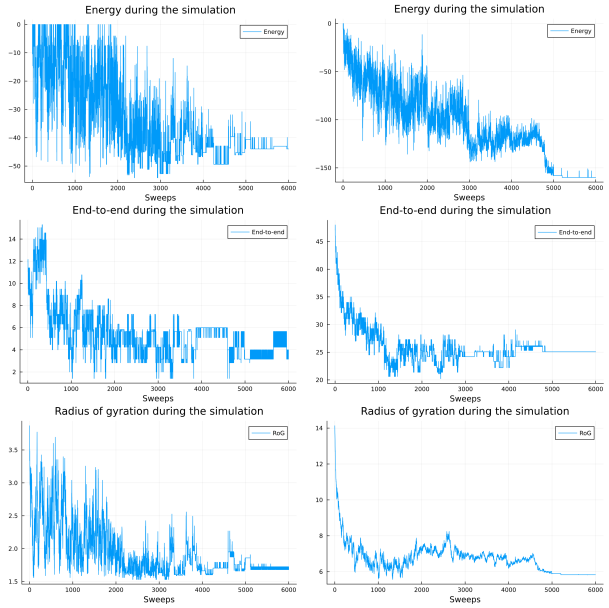

Figure 6: Simulation of a randomly generated protein with 15 monomers (left) and 50 monomers (right), initialized completely linear at $T = 20, 10, 5, 3, 2, 1$ changed after every 1000 sweeps. Plots of energy, end-to-end distance and radius of gyration are shown as a function of the sweeps. A simulation with 30 monomers was also run, with intermediates results. It will be added in the Appendix.

In order to keep the simulations comparable, the temperatures were kept constant for all the monomer dimensions. The oscillations are in fact decreasing a lot after each change in temperature, reaching a lower energy steady state. With a larger number of monomers, it looks like the average values are decreasing far more quickly even at the first temperature, whereas the oscillations of the protein with 15 monomers are very large.

To further investigate the effect of the temperature, simulations have been generated at different temperatures and let reach a steady state. The averaged properties are then collected in Fig. 7, for $N = 15, 50$ (and $N = 30$ in the Appendix). There appears to be a transition happening at about $T = 6$, where the slopes of the curve suddenly change, which is present for all the 3 chosen dimensions. For $N = 50$ it looks like the transitions happens at a slightly higher temperature, but it likely just a fabrication due to the fact that the values were varying a lot more and are not averaged as completely as the case of smaller sizes. To improve the description, more MC sweeps are needed. Another possibility is that, since the energies in play are larger with a bigger number of monomers, thanks to many more interactions that can be formed, at the same temperature conditions the effect of the temperature on the probability of acceptance is lower and higher temperatures are necessary to have the transition.

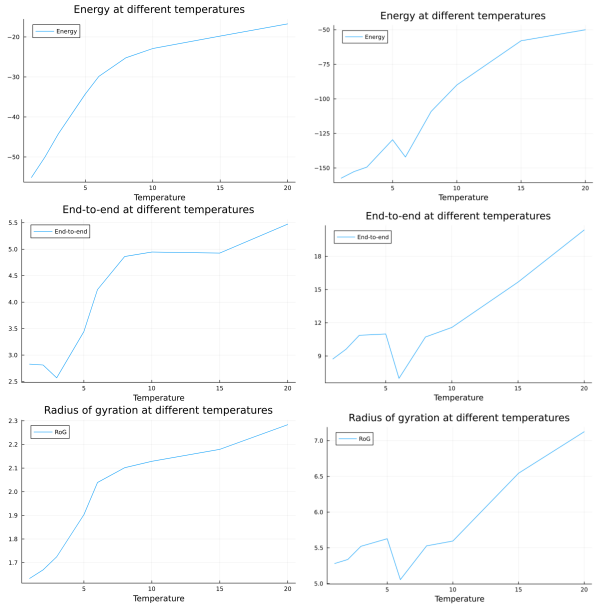

Figure 7: Simulation of a randomly generated protein with 15 monomers (left) and 50 monomers (right), initialized completely linear at $T = 20, 15, 10, 8, 6, 5, 3, 2, 1$, left for 100 sweeps and then averaged for other 3000 before moving to the next temperature. Plots of energy, end-to-end distance and radius of gyration are shown as a function of the temperature. A simulation with 30 monomers was also run, with intermediates results. It will be added in the Appendix.

4.1 Simulated annealing

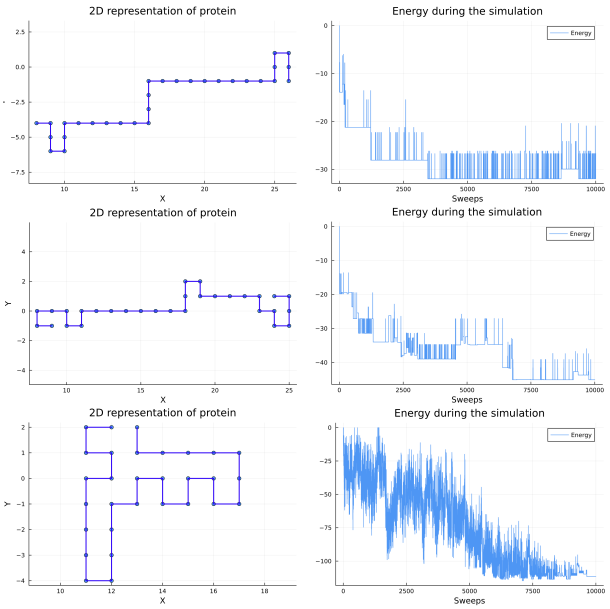

Figure 8: Simulation of a randomly generated linear protein with 30 monomers. The first two simulations are at a constant $T = 1$ for 10000 sweeps whereas the last one is done with simulated annealing, reducing linearly the temperature from $T = 10$ to $T = 1$ in 10000 sweeps. The energy profile during the simulations is also shown on each side.

In the simulated annealing simulation, the temperature is gradually reduced at each step. This is very important to let the protein explore different geometries and slowly reach a steady state. Looking at Fig. 8, we see that simulations kept at a constant temperature of 1 quickly reach a steady state, that is only slightly modified and especially at the free ends. With the simulating annealing we allow changes that might increase momentarily the energy but with an higher chance to reach a lower state eventually. In fact, the results of this process is a protein that is completely folded, in the sense that most of the monomers have at least one neighbouring interaction.

5 3D protein

Now the code is expanded to study 3D geometries. In Fig. 9, the results are presented for a linear protein of 10 amino acids at $T = 10$, taking the geometries after 1, 10 and 100 sweeps and their energies, end-to-end distances during the whole simulation.

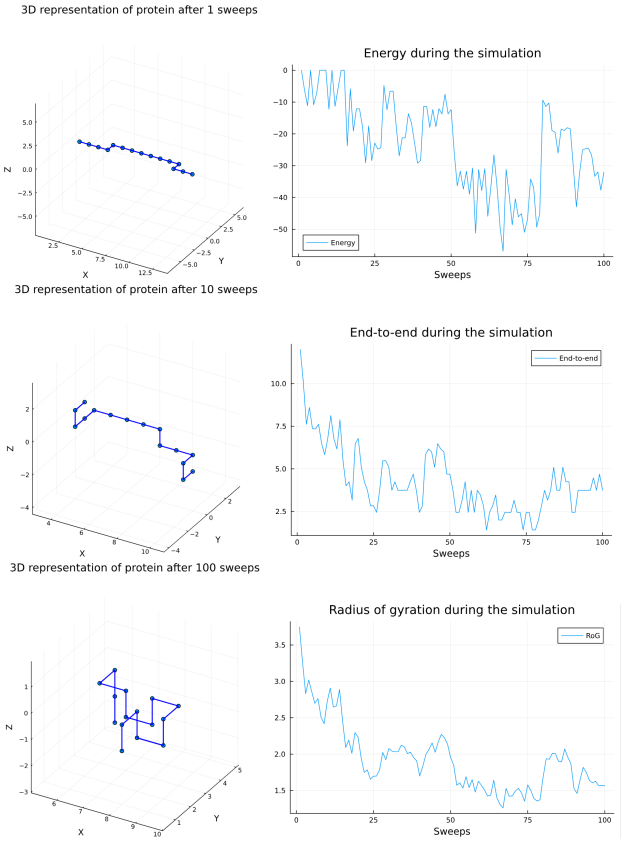

Figure 9: Simulation of a randomly generated 3D protein with 15 monomers, initialized completely linear at $T = 10$. Geometries after 1, 10, 100 sweeps are presented, together with the energy, end-to-end distance and radius of gyration for each sweep.

The study about the effect of the temperature was also repeated on a 3D linear protein of 15 monomers. The results are shown in Fig. 10, where the transition is again happening around $T = 6$.

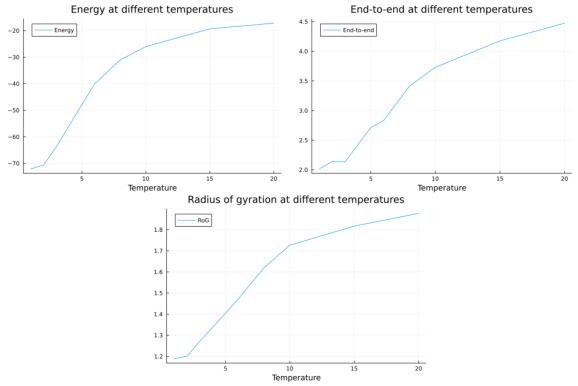

Figure 10: Simulation of a randomly generated 3D protein with 15 monomers, initialized completely linear at $T = 20, 15, 10, 8, 6, 5, 3, 2, 1$, left for 100 sweeps and then averaged for other 3000 before moving to the next temperature. Plots of energy, end-to-end distance and radius of gyration are shown as a function of the temperature.

6 Appendix

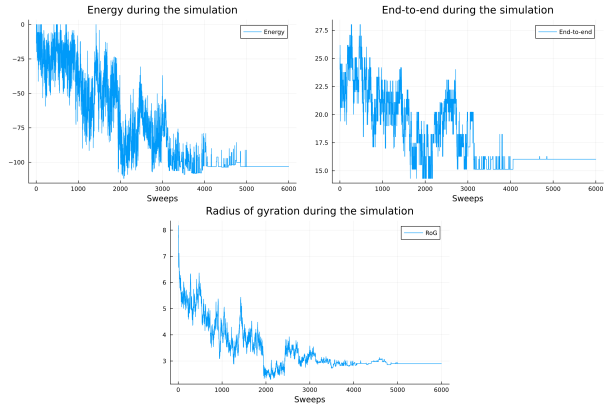

Figure 11: Simulation of a randomly generated protein with 30 monomers, initialized completely linear at $T = 20, 10, 5, 3, 2, 1$ changed after every 1000 sweeps. Plots of energy, end-to-end distance and radius of gyration are shown as a function of the sweeps.

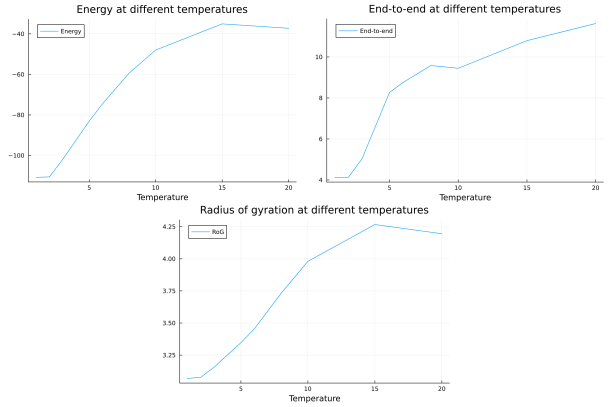

Figure 12: Simulation of a randomly generated protein with 30, initialized completely linear at $T = 20, 15, 10, 8, 6, 5, 3, 2, 1,$ left for 100 sweeps and then averaged for other 3000 before moving to the next temperature. Plots of energy, end-to-end distance and radius of gyration are shown as a function of the temperature.