Pichon Julien Rambaud Opale - M2 Bioinformatique 2020/2021

Project report

FOLDING OF AN HP MODEL OF PROTEIN BY A MONTE CARLO ALGORITHM

(Subject 1)

Introduction

Predicting protein tertiary structure problematic

One of the largest problematic in biochemistry is to predicting protein tertiary structure from a given amino acid sequence while at the same time minimizing the energy function in order to find the optimal final folding.
Current structure prediction techniques are time consuming and not really effective. Experimental structure determination techniques like X-ray crystallography and nuclear magnetic resonance are no longer sufficient in the face of the growing amount of data.

Stochastic methods

Stochastic refers to a randomly determined process. In the case of a folding problem, one of the possible solutions is to inject randomness in prediction. Several examples of stochastic method exist like Bernoulli process, Random Walk, Wiener process, Poisson process or Monte Carlo algorithm.

Monte Carlo algorithm

In principle, Monte Carlo methods can be used to solve any problem having a probabilistic interpretation. In our case, the Monte Carlo Algorithm is used to choose randomly the movement of amino acids among several possible moves.

The HP model

In this model, amino acids are classified as either H (hydrophobic) or P (polar). Informally, a sequence of H's and P's is embedded into a lattice structure.
A valid conformation of the sequence corresponds to a self-avoiding walk on the lattice.

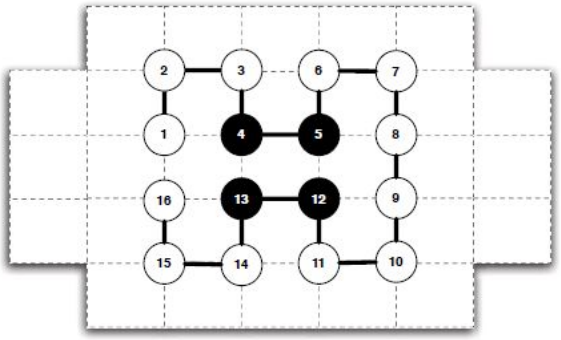

Figure 1 : A ground-state conformation in the 2D HP model.

The figure 1 show the hydrophobic contact between residues 4 and for and between residues 5 and 12.

The HP model is our input. Then we applied moves by applying the Monte Carlo algorithm to recreate the folding. The algorithm choose among this four moves :

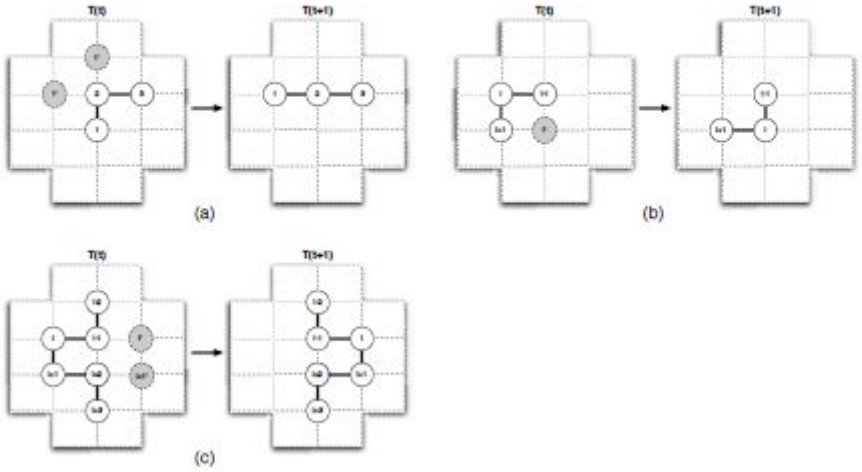

Figure 2 : schematic representation of VSHD, End, Corner and Crankshaft moves

- VHSD move (a)

The residue is pivoted relative to its connected neighbour to a free position adjacent to that neighbour.
Each position is checked in random order for availability. If a position is found to be free, the residue is moved.

- End moves (a)

For a chain of length n, an end move can be performed on residue 1 or residue n. This is the same mechanism as the VSHD move.

- Corner moves (b)

A corner move can potentially be performed on any residue excluding the end residues. For a corner move to be possible, the two connected neighbours of some residue i must be mutually adjacent to another, unoccupied position on the lattice.

- Crankshaft moves (c)

A crankshaft move can occur if some residue i is part of a u-shaped bend in the chai. The crankshaft move can be performed in 2D if positions i' and i + 1' are empty. Crankshaft moves in 2D always involve a 180° rotation of a u-shaped structure consisting of four connected neighbours on the chain.
In our programm the Crankshaft move does not work optimally and we obtain some errors.

- Pull moves

Unfortunately we were not able to implement the pull move. It is described by the diagonal movement of two residues, n and n + 1. If the residue n - 1 is still in contact with the residue

n, therefore the movement is over. However, if not, n - 1 residue takes the previous coordinates of n + 1 residue. N - 2 takes the last coordinates on the way of n - 1 to previous n + 1. This process is performed until residue are correctly linked together.

Materials and methods

We developed the script run on python3, without external modules. We chose to use HP files as input (see data/ repository) and to produce a pdb file as output, with the different conformation taken by the protein each step. Keeping each conformation in the pdb file is essential to visualize the different conformation taken by the protein during all the steps.

During a step, a random residue is chosen. This residue can perform one of the movements described earlier if there aren't other proteins at the targeted coordinates.
The energy from the new conformation is calculated using the hydrophobic bond characteristic of the HP model. If energy is inferior to the previous conformation, the new conformation is adopted. Otherwise, Monte-Carlo approach allows the probability that the new conformation is still adopted depending on the energy difference and the temperature.

We create a residue class, in order to have an easy access to the main characteristic of the residue and perform the different movements.

Results

We didn’t manage to incorporate the “pull move” in the time given. Moreover, our crankshaft move has an unidentified error, with a frequency according to the length of the protein and the number of steps, causing a sys.exit. Thus, we decided to incorporate this move as an argument.
In spite of that, our script runs correctly on the dataset, taking less than 10 seconds when the number of steps < 10k and the protein length < 25 residues.
We didn’t manage to correctly visualyze the protein using the .pdb on pymol. Indeed, we see only the primary sequence. However, when the .pdb contains only one conformation (we run the script with step = 1), we can visualize the protein on pymol. It means that the issue could be certainly solved playing with some settings on pymol.
We tried our script several times on HP1, HP2, HP3 files at a constant temperature T = 300K and reported the mean energy on table 1.

Min E	Real	n = 1000	n = 10 000	n = 100 000
HP1	-9	-1	-5	-2
HP2	-9	-2	-4	-3
HP3	-8	-2	-3	-4

Table 1. Real minimal energy of the protein, and the minimal energy found by our script for 1000, 10 000 and 100 000 steps starting with a linear protein. Tchachuk et al. obtained the real minimal energy with the Replica exchange Monte-Carlo.

With its default parameters (n = 10 000) our script doesn't manage to find the correct folding and calculate the minimal energy. Surprisingly, we obtain worse results when we increase the number of steps. Running the script a lot of time could help to deal with the random part of Monte-Carlo approach to rule out an error in the script.
In order to get closer to the real minimal energy value, we made the hypothesis that starting from a non-linear protein could help to explore easier other conformations. We edited HP files in order to have a starting protein with each residue forming an elbow with its neighbours. Results are presented in table 2.

Min E	Real	n = 1000	n = 10 000	n = 100 000
HP1	-9	-2	-4	-5
HP2	-9	-2	-4	-4
HP3	-8	0	-3	-3

Table 2. Real minimal energy of the protein, and the minimal energy found by our script for 1000, 10 000 and 100 000 steps starting with an elbowed protein. Tchachuk et al. obtained the real minimal energy with the Replica exchange Monte-Carlo.

Elbowed protein didn't perform significantly better than the linear one. The causes could be explained by several hypotheses. We could not have performed enough replicates to compensate for the randomness of the Monte-Carlo approach. More probably, there is an error in the script, or, the crankshaft move and the pull move is essential to find the correct protein folding.

Conclusion

Compared to the article, we did not succeed in obtaining minimal energies. This means that we did not manage to find the ideal foldings for these proteins. This is due to the fact that we did not manage to implement all the movements.
So we could not prove that the use of the MC algorithm helps to find the ideal folding.

In addition, a graphical representation of the folds obtained could have helped to visualize the final tertiary structure obtained by our programm.

We encountered difficulties implementing the movements as in the article, this does not allow us to obtain significant results. The future step is to implement the other movements to compare well our results with those in the article.