Python Implementation of a Monte Carlo algorithm for 2D protein folding in the HP model

Marwa Ghraizi

M2 Biologie Informatique - Université Paris Cité

September 13th, 2023

Introduction:

Ab-initio protein folding is a complex biological problem requiring heavy computational resources and optimized algorithms. One of those algorithms is the Monte Carlo algorithm which is a computational method that explores the conformational space of proteins by minimizing their energies while allowing some energetically unfavorable conformations for a better exploration of the conformational space. This project is an implementation of the classical Monte Carlo algorithm in Python to fold proteins in two dimensions.

The implementation and documentation of this algorithm can be found at https://github.com/marwaghraizi/MC_protein_folding.git

Methods:

This project was written in `Python 3.11.5` using two external libraries `tqdm` and `graphviz`.

The model used to represent protein sequences is a simplified Hydrophobic-Polar (HP) model to reduce the complexity of the protein by reducing its amino acids to their two major physico-chemical categories.

The program takes in fasta files with HP or standard sequences as well as sequences in the command line. The initial conformation of the protein can be linear or randomized depending on the user's choice. Moreover, the user has the option to change the number of search iterations as well as the search temperature and most importantly the search neighborhood which can either be VSHD only (Fig. 1) including the end, corner and crankshaft moves or can be based entirely on pull moves shown in Fig. 2 or a hybrid VSHD-pull search neighborhood.

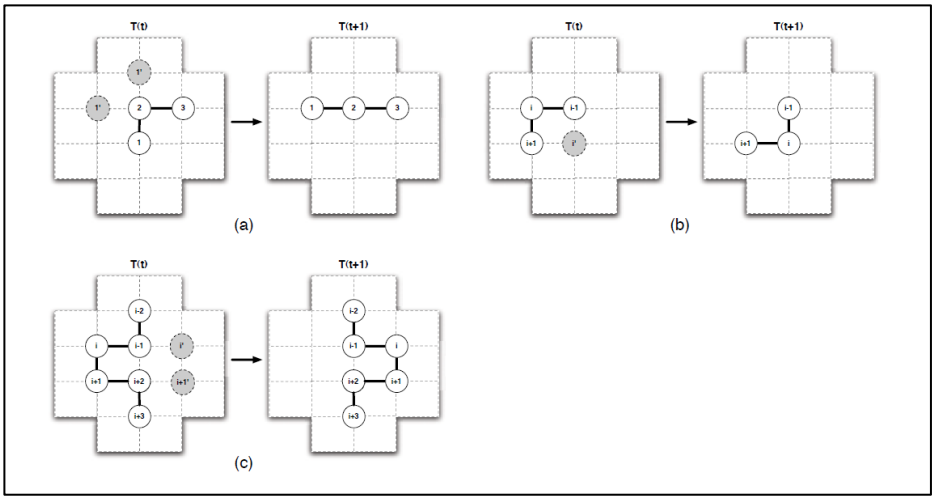


Fig.1: VSHD Moves. For more detailed explanations see [1]

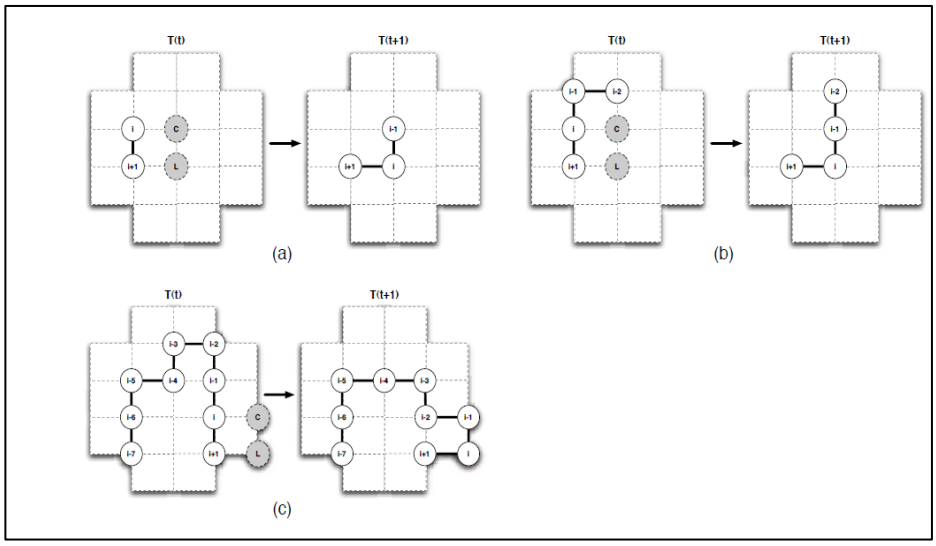


Fig. 2: Pull Moves. For more detailed explanations see [1] & [2]

With every iteration, an amino acid is randomly chosen followed by a random choice of a move; if the move was possible with this amino acid, it is added to the frames, if not another move is randomly selected until it can be successful. The resulting protein conformation is tested for its energy based on the following conditions:

$$Pr[c \to c'] := \begin{cases} 1 & \text{if } \Delta E \le 0, \\ e^{\frac{-\Delta E}{T}} & \text{otherwise.} \end{cases}$$

A move is considered successful if the energy of the new conformation is lower than that of the previous one. If not, a random probability to accept an energetically unfavorable conformation is calculated which is based on the search temperature. This is allowed in Monte Carlo algorithms to explore more search spaces and to avoid being stuck in a local minimum.

Finally, a log file is produced with a detailed description and tracing of each conformation of the protein as well as its starting energy, optimal energy and final energy for a better understanding of the convergence of the algorithm. Furthermore, a representation of the protein as a directed linear graph is produced to visualize the folding, the user can opt to see the final conformation or the optimal one with the lowest and most stable energy.

Results

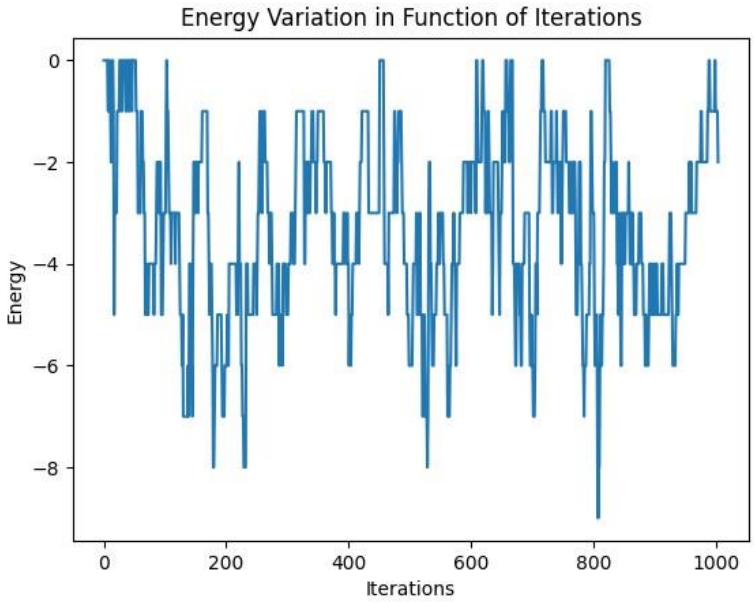
The results of the folding of 5 benchmark proteins can be found in Table. 1 which compares the optimal energy of the 3 different search neighborhoods against the results of the original publication's implementation. As expected, the energies obtained with this implementation are lower than those obtained in the original algorithm due to the optimization of their methods and the usage of the Replica Exchange Monte Carlo algorithm with the potential of yielding better results due to the simulation of the same system in parallel at different temperatures.

It can also be observed that the hybrid model of VSHD-pull moves is the most efficient model with the lowest energies followed by the pull moves only search and lastly the VSHD only search. This indicates that the pull move is extremely effective in comparison to the rest of the moves.

Protein	E* VSHD-pull	E* VSHD	E* pull	Benchmark E*
$(HP)_2PH_2PHP_2HPH_2P_2HPH$	-7	-5	-6	-9
$H_2(P_2H)_7H$	-7	-5	-6	-9
$P_2HP_2(H_2P_4)_3H_2$	-5	-3	-6	-8
$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$	-7	-4	-8	-14
$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$	-12	-5	-9	-23

Table. 1. Results of the energies obtained with the simulation of the folding of 5 benchmark proteins. This was done with 10000 iterations at a temperature of 100 with all 3 search neighborhoods.

Fig. 3 shows the energy variation of a VSHD-pull hybrid search of the 5th benchmark sequence for 1000 iterations. We can observe clear fluctuations of the energy with the lowest achieved around iteration 800. The lack of convergence of the energy can be due to a multitude of factors such as a low number of iterations, an unsuitable search space and a potentially unoptimized implementation of the algorithm.



Conclusion:

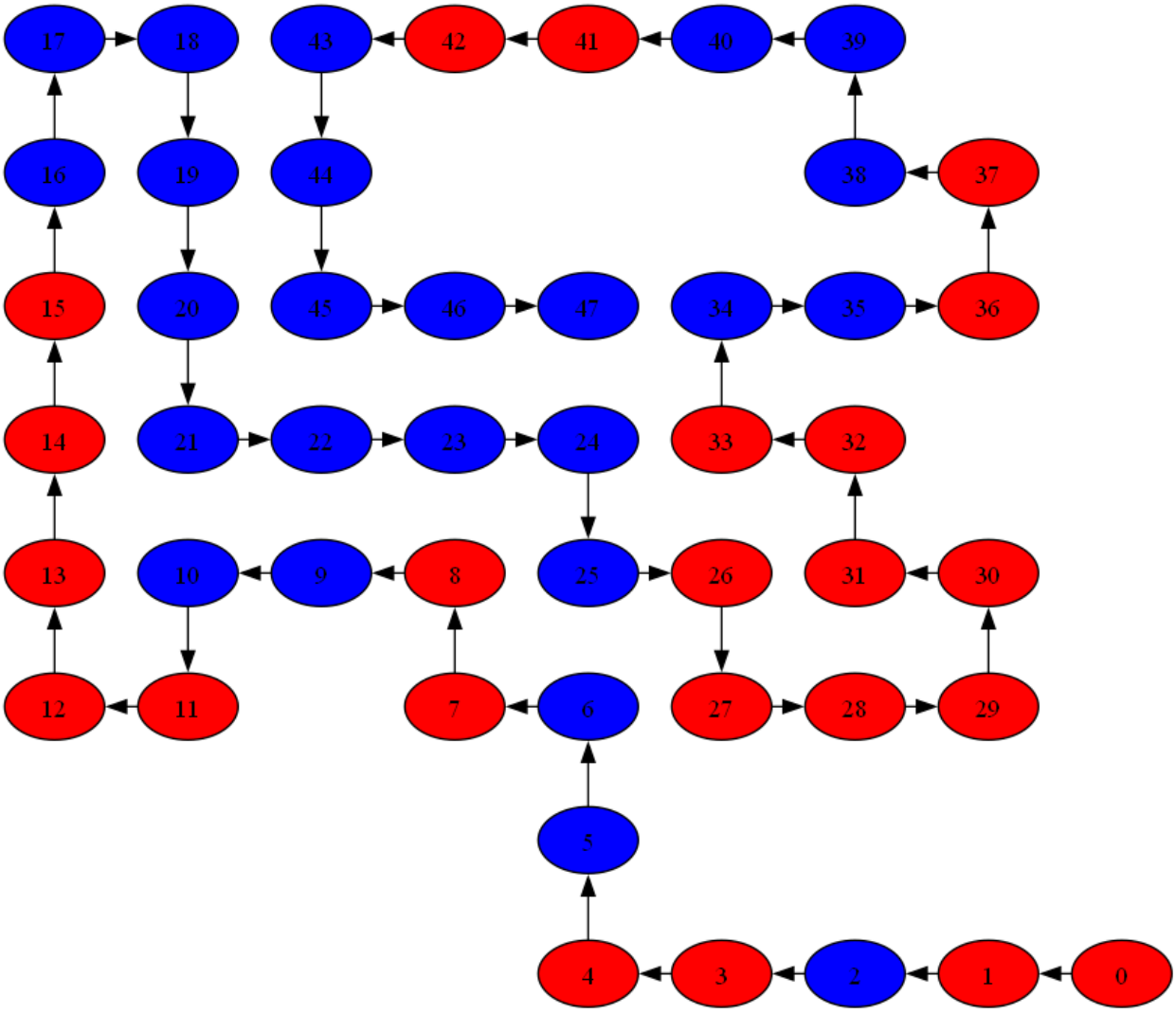
In conclusion, this project presented a Python implementation of the classic Monte Carlo protein folding algorithm. Overall, it was able to create relatively stable folded conformation of different proteins. This program can be optimized with parallelization and can be improved by implementing the Replica Exchange Monte Carlo Variation. Moreover, due to time constraints, the code does not include all good programming practices.

References:

[1] Thachuk, C., Shmygelska, A. & Hoos, H.H. A replica exchange Monte Carlo algorithm for protein folding in the HP model. BMC Bioinformatics 8, 342 (2007). https://doi.org/10.1186/1471-2105-8-342

[2] Lesh, N., Mitzenmacher, M., Whitesides, S. A Complete and Effective Move Set for Simplified Protein Folding. ACM Digital Library (2003). https://doi.org/10.1145/640075.640099

ANNEX:



Optimal conformation of the 5th benchmark sequence generated with 10 000 iterations, starting with a linear conformation and using the hybrid VSHD-pull search neighborhood at a temperature of 100. Polar residues are represented in red while hydrophobic residues are represented in blue.