Link to the project in GitHub: https://github.com/AnisMerabet/Monte-Carlo-based-peptide-folding-simulation.git

Anis Merabet, Master 2 BI, Université Paris Cité **13/09/2023**

Folding of a simplified protein model by Monte Carlo algorithm

1 Introduction

Studying protein folding is a complex problem. For that, simplification models were generated to tackle specific aspect of protein folding. One of the models there is the Dill's HP model which is also knows as the two-dimensional hydrophobic-hydrophilic model. This model is used to study the basic principles of protein folding and how hydrophobic and hydrophilic interactions influence the native structure of proteins. In this model, hydrophobic interactions tend to drive the folding of the protein, as hydrophobic amino acids cluster together, while polar amino acids are more exposed to the solvent [1]. Monte Carlo simulations are a powerful computational technique used to explore the energy landscape of protein folding in Dill's HP model. They provide insights into the thermodynamics and kinetics of protein folding [2]. THACHUK C. and al. implemented the replica exchange Monte Carlo (REMC) method for the HP model.

Monte Carlo algorithm follows a gradient descent but can, with a certain probability, move in the opposite direction. This mechanism allows for escaping local minima, thereby increasing the likelihood of discovering the global minimum. The probability of applaying the moves is given by the following formula:

$$\begin{cases} 1 & \text{if } \Delta E < 0 \\ e^{\frac{-\Delta E}{T \times K_B}} & \text{else} \end{cases}$$

ΔE = energy variation between two conformations, T is the temperature and K_B stands for Boltzmann's constant. This formula not only ensure decrease of conformation energy through Monte Carlo steps, but also avoiding local minima. Nevertheless, it depends on the temperature that has to be set before the simulation.

In this study we implement Monte Carlo algorithm to simulate 2D protein folding according to the HP model.

2 Materials and methods

2.1 Implemented set of moves

In order to perform Monte Carlo simulation, we implemented VSHD moves and pull moves as illustrated in figure 1. VSHD moves encompass end moves, corner moves and Crankshaft moves as described previously [2]. At each step of Monte Carlo simulation, our algorithm calculates all possible VSHD moves for the current conformation of the peptide, then randomly select one of the moves and apply it. In our study, we implemented an enhanced version of the already known pull move. A pull move according to the previous version, requires the position C to be empty or occupied by the amino acid i-1 (figure 1.A). This condition limits amino acid mobility within the peptide. Our pull move implementation considers all possible moves for all amino acids of the peptide. Thus, as we present in figure 1.E, new positions are allowed for certain amino acids. At each step of Monte Carlo simulation, one of the amino acids is randomly chosen and is moved according to pull move mode.

2.2 Prot Fold software

In this study, we developed the software Prot Fold which provides graphical user interface, then facilitating Monte Carlo simulation. The button browse is used to load a fasta file containing a protein sequence. A random conformation is attributed to the sequence once loaded and the current energy of this conformation is calculated and displayed. The temperature, the number of Monte Carlo simulation steps and the move mode (VSHD or Pull move) are set by the user before starting the actual simulation. At the end of the simulation statistics are displayed including the current energy of the simulation and calculation time, with the possibility to display a graph indicating the evolution of the energy according the Monte Carlo steps In figure 2, we show an example usage of the software.

2.3 Implementation

All scripts of Prot Fold software were coded in python 3.11.4. The scripts were written in object-oriented programming and in respect of all programming good practices. Conda environnement included Biopython, Matplotlib and Tkinter packages.

3 Results

3.1 Optimal temperature might depend on protein size

To study the influence of the temperature on the efficiently of our algorithm on folding peptides and escaping local minima, we considered two benchmark proteins used in the literature. One is a small protein of 32 amino acids and has the particularity to present only one optimal conformation with energy equal to -12, and the other is a bigger protein with 100 amino acids. The number of MC steps was set to 50000 and move mode to pull moves. We varied the

temperature according to three different values: 40, 120, 190 and 300 which ensures to cover a large range of temperature (figure 3). At low temperature, the results show that local it ends at local minima whereas at high temperature it shows difficulty in stabilizing the conformation with low energy. The best results were obtained with intermediate temperatures. Nevertheless, the optimal temperature was 120 for the small protein whereas it was 190 for the bigger protein. This suggests that the optimal temperature depends on the size of the protein in a way that bigger proteins require higher temperature to be stabilized at minimal energy. VSHD move mode show similar results (data not shown).

3.2 Replica exchange Monte Carlo algorithm show better results than our Mont Carlo simulation

To compare our algorithm to Replica exchange Monte Carlo algorithm the 4 most long amino acid sequences used in [2] were selected. Because the best energy was 190 for long sequences in our previous results, we set it for all this analysis. The number of MC steps was set to 50000 and we analyzed pull moves and VSHD moves separately. Replica exchange Monte Carlo algorithm performed better for all the selected benchmark sequences and whatever is the move mode. It worth to mention that in our simulation, pull move performed better than VSHD move for all the cases. Even though the calculation time was always higher for VSHD moves compared to pull moves (figure 4 and 5).

4 Discussion

Prot Fold software correctly implemented VSHD moves and pull moves. The algorithm of VSHD was characterized by the calculation of all possible moves before randomly selecting one move to apply. This provides the advantage avoid having MC steps with no moves, then reducing the number of steps to reach optimal conformation. It also reduces calculation time compared to the case where and amino acid is randomly selected and checked if a VSHD move is possible. In the latter case, the probability of finding a movable amino acid become very low when approaching the optimal conformation.

The enhanced version of pull move presented in this study confirmed to properly work and the calculation time was lower compared to VSHD move, which could be explained by the fact that in pull move, the algorithm does not calculate all possible moves, but randomly select an amino acid. If the latter fail to present a possible move, the algorithm selects another amino acid within the same MC step ensuring that at each MC step, a pull move is applied.

Our results suggest that the optimal temperature could depend on the size of the protein, hence the interest to try different temperature each time a protein is studied. Otherwise, this explains why replica exchange algorithm outperformed ours.

Absence of computational assessment, if fact there should be many runs, each one with a unique random seed

In our study we did not not conduced a computational assessment which is required for checking significance in differences and obtain reproductible results.

Even though, the calculation time was well incorporated in our software, it does not allow a proper comparison with effectiveness of algorithms from other studies as we did not take into consideration CPU time.

5 Conclusion

In a nutshell, Prot Fold software is a reliable tool to analyze protein folding according to the 2D HP model. It incorporates two sets of moves, VSHD and pull moves and provides user graphical user interface facilitating loading of protein in fasta file and analyzing protein folding. The performance of our algorithm is limited comparing to already existing algorithm for the same model. Extending our model to the replica exchange Monte Carlo algorithm by keeping the enhanced version of pull moves and the characteristics of our VSHD moves is warranted.

6 Appendix

6.1 Bibliography

[1] N. Lesh, M. Mitzenmacher, and S. Whitesides, “A complete and effective move set for simplified protein folding,” in *Proceedings of the seventh annual international conference on Research in computational molecular biology*, in RECOMB ’03. New York, NY, USA: Association for Computing Machinery, Apr. 2003, pp. 188–195. doi: 10.1145/640075.640099.

[2] C. Thachuk, A. Shmygelska, and H. H. Hoos, “A replica exchange Monte Carlo algorithm for protein folding in the HP model,” *BMC Bioinformatics*, vol. 8, no. 1, p. 342, Sep. 2007, doi: 10.1186/1471-2105-8-342.

6.2 Figures

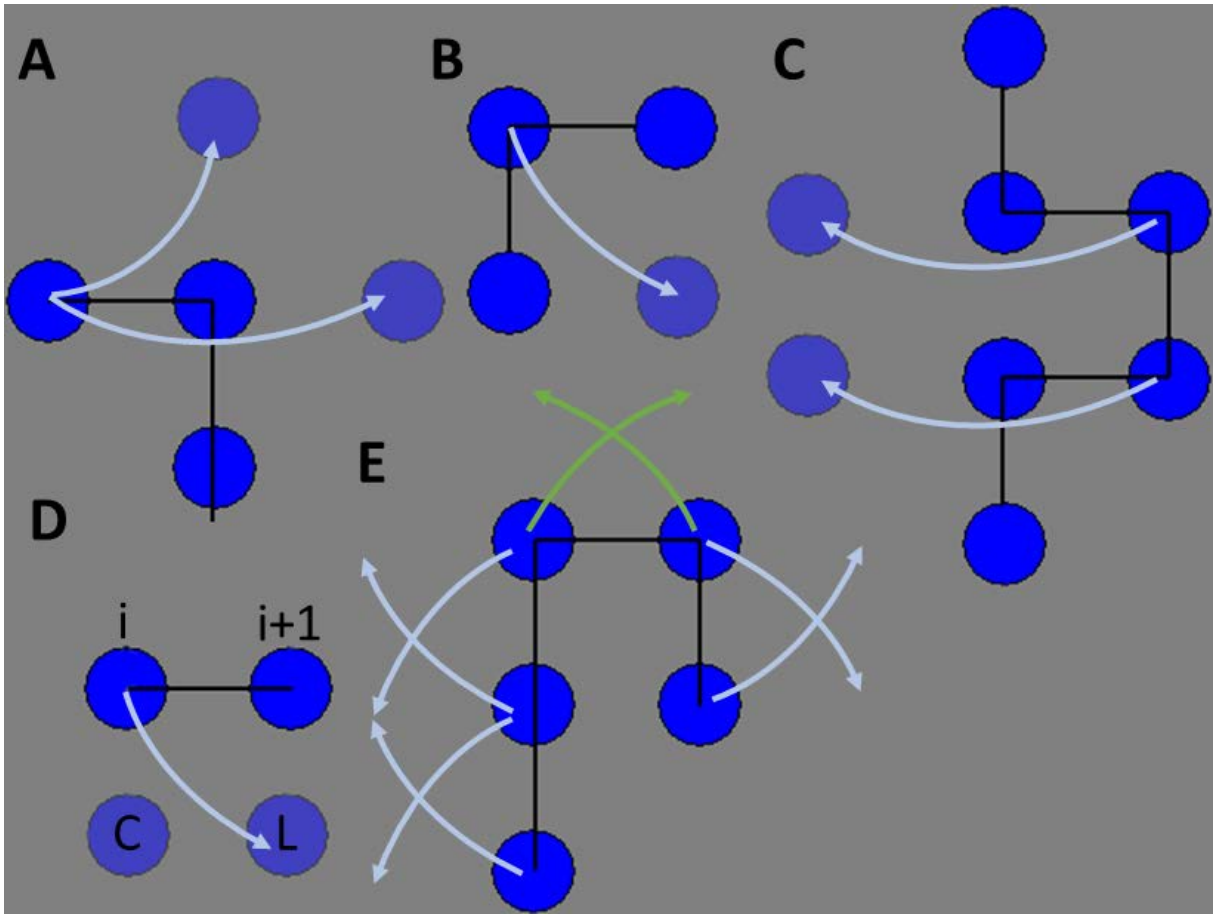


Figure 1: VSHD and pull move modes.
A. End move concerns the first and the last amino acid of a peptide. B. Corner move concerns the internal amino acids when an angle of 90° is formed and the position completing the square is empty. C. Crankshaft move concerns two amino acids and requires two free position. D. Pull move could concern any amino acid if the position L is free and the position C is not occupied by an amino acid other than the amino acid i-1. E. All possible pull moves are indicated by arrows. Green arrows indicate the pull moves newly implemented in this study and where never considered before, to our knowledge.

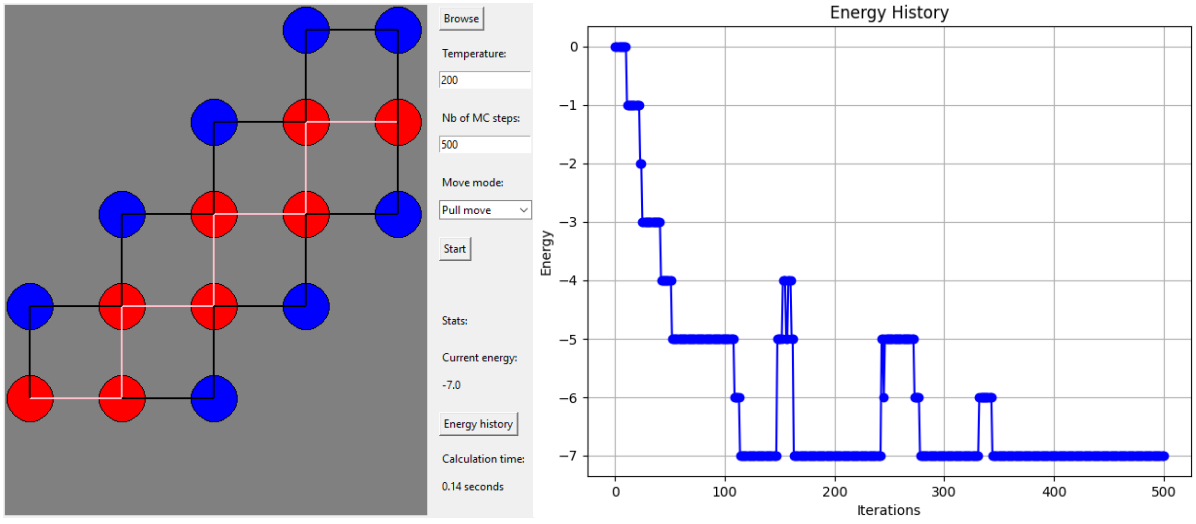


Figure 2: Prot Fold software.

In the left panel, a protein sequence was loaded. Hydrophobic amino acids are shown in red whereas hydrophilic amino acids are shown in blue. After setting the temperature to 200, the number of iterations to 500 and the move mode to Pull move, the simulation was starting by cliquing on the button Start. The current energy of the sequence after the simulation is displayed and also the calculation time consumed by the simulation. By cliquing on Energy history, the right panel is displayed. It shows the evolution of the energy during the steps of MC simulation.

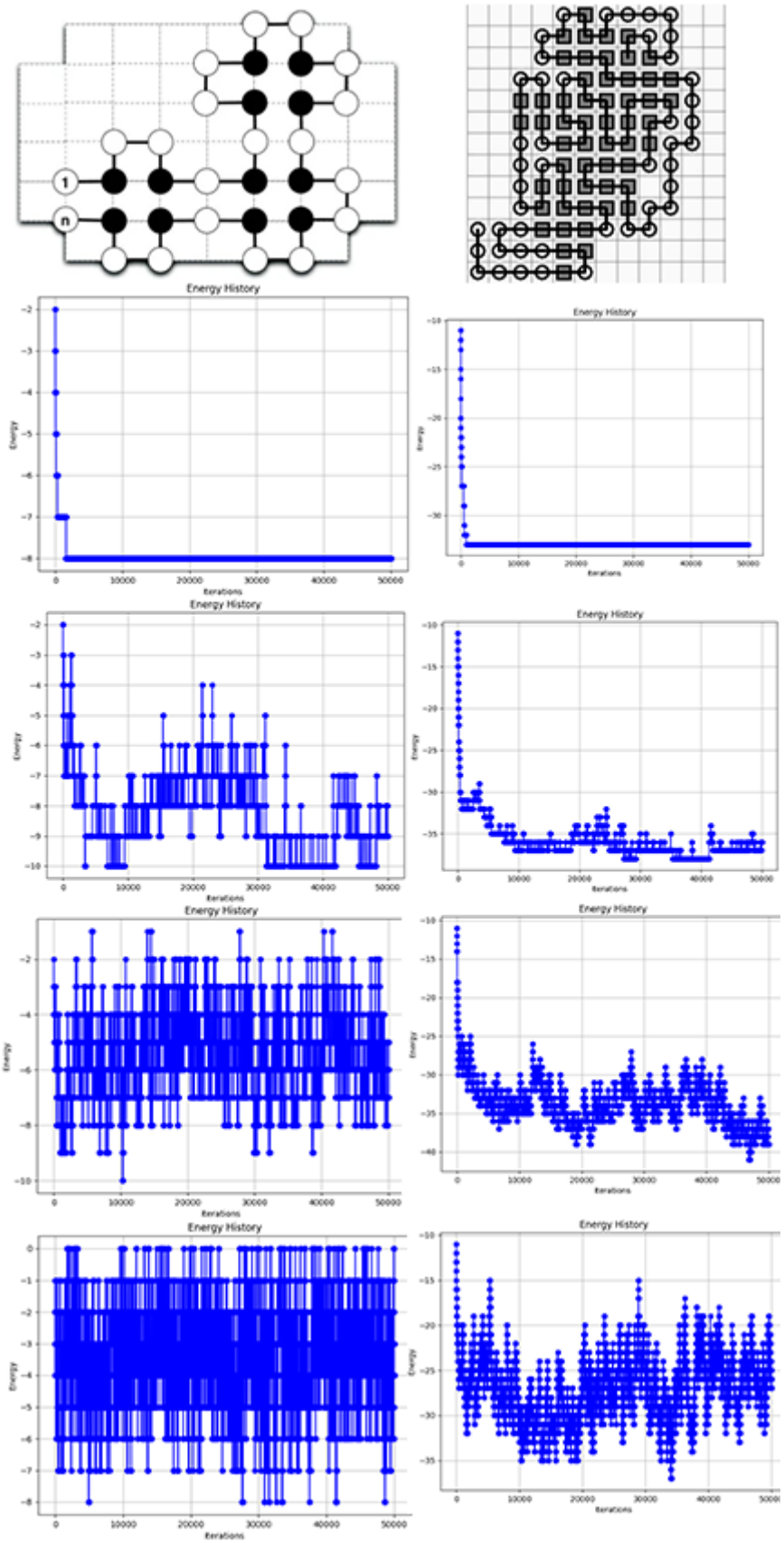
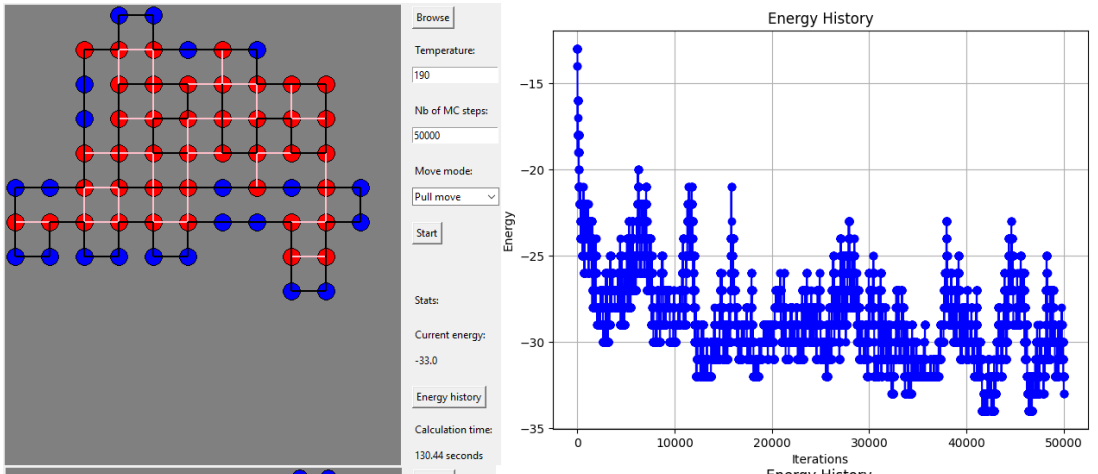
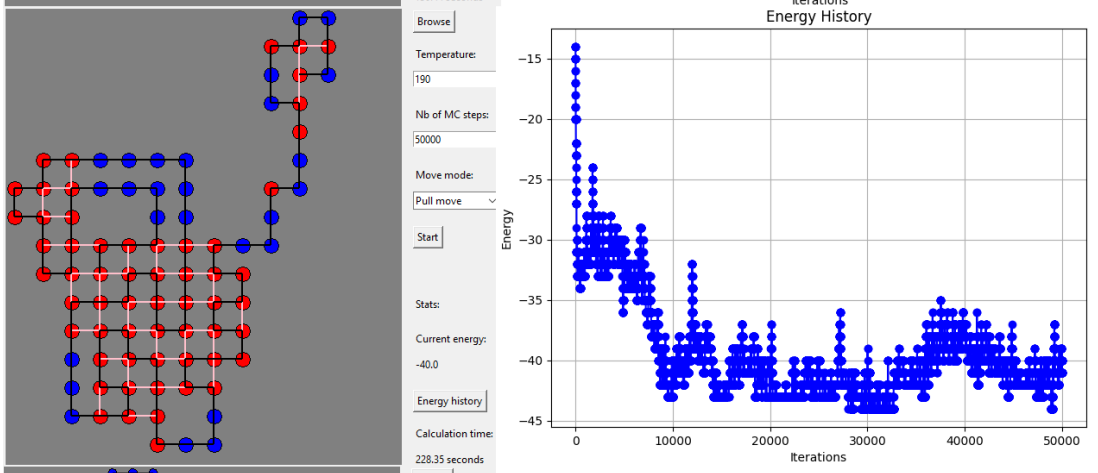
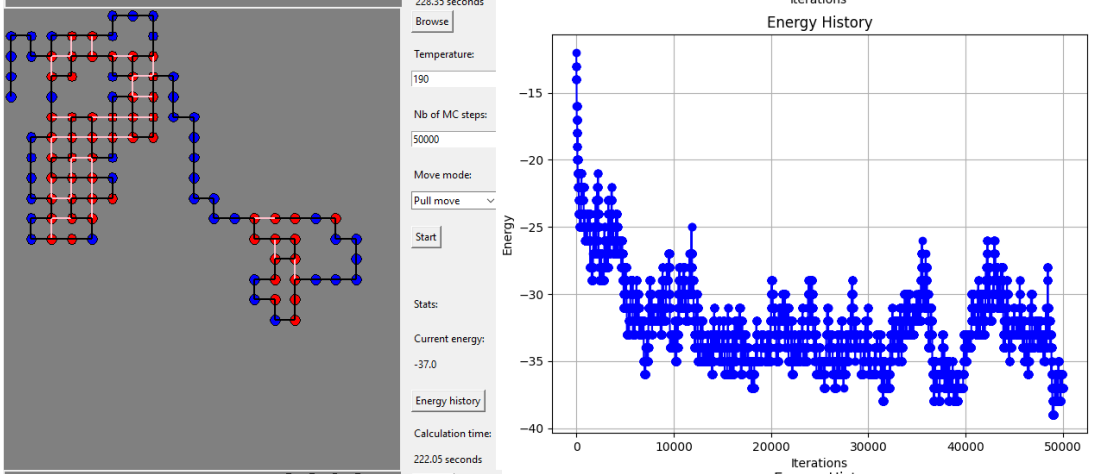


Figure 3: Temperature impact analysis.
In the left panel, a benchmark protein from [2] (set at the optimal conformation) was analyzed by our algorithm. While the number of Monte Carlo steps was always set to 50000 and move mode to pull moves, the temperature was set to, 40, 120, 190 then 300 accordingly. In the

right panel, a protein benchmark from [1] (set at one of the optimal conformations) was analyzed similarly to the protein from the left panel.


Browse
Temperature:
190
Nb of MC steps:
50000
Move mode:
Pull move
Start
Stats:
Current energy:
-33.0
Energy history
Calculation time:
130.44 seconds
Energy History
Energy
Iterations


Browse
Temperature:
190
Nb of MC steps:
50000
Move mode:
Pull move
Start
Stats:
Current energy:
-40.0
Energy history
Calculation time:
228.35 seconds
Energy History
Energy
Iterations


Browse
Temperature:
190
Nb of MC steps:
50000
Move mode:
Pull move
Start
Stats:
Current energy:
-37.0
Energy history
Calculation time:
222.05 seconds
Energy History
Energy
Iterations

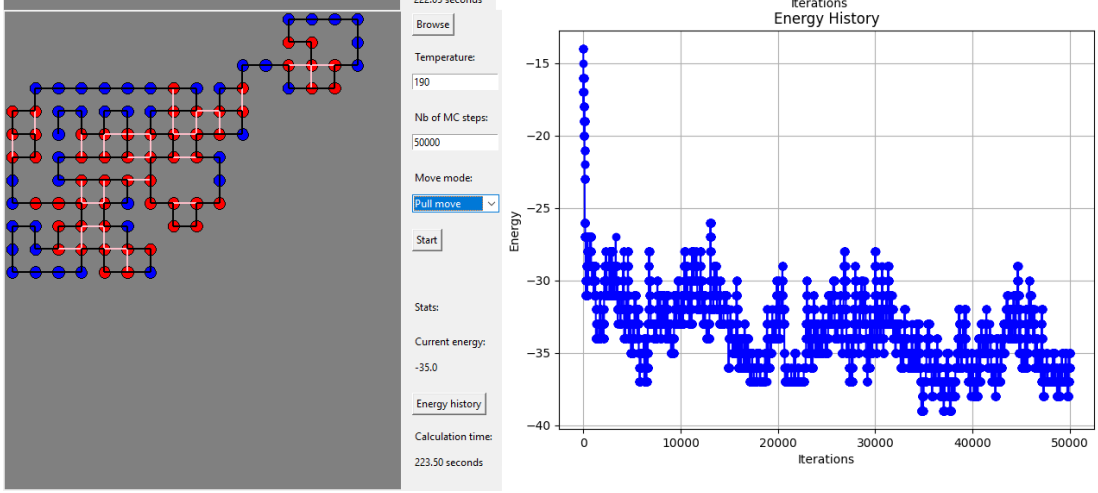

Browse
Temperature:
190
Nb of MC steps:
50000
Move mode:
Pull move
Start
Stats:
Current energy:
-35.0
Energy history
Calculation time:
223.50 seconds
Energy History
Energy
Iterations

Figure 4: MC simulation with pull move mode applied to benchmark proteins.
Our algorithm was applied to bench marks proteins S1-8, S1-9, S1-10 and S1-11 respectively. Temperature was set to 190, move mode to pull move and MC steps to 50000.

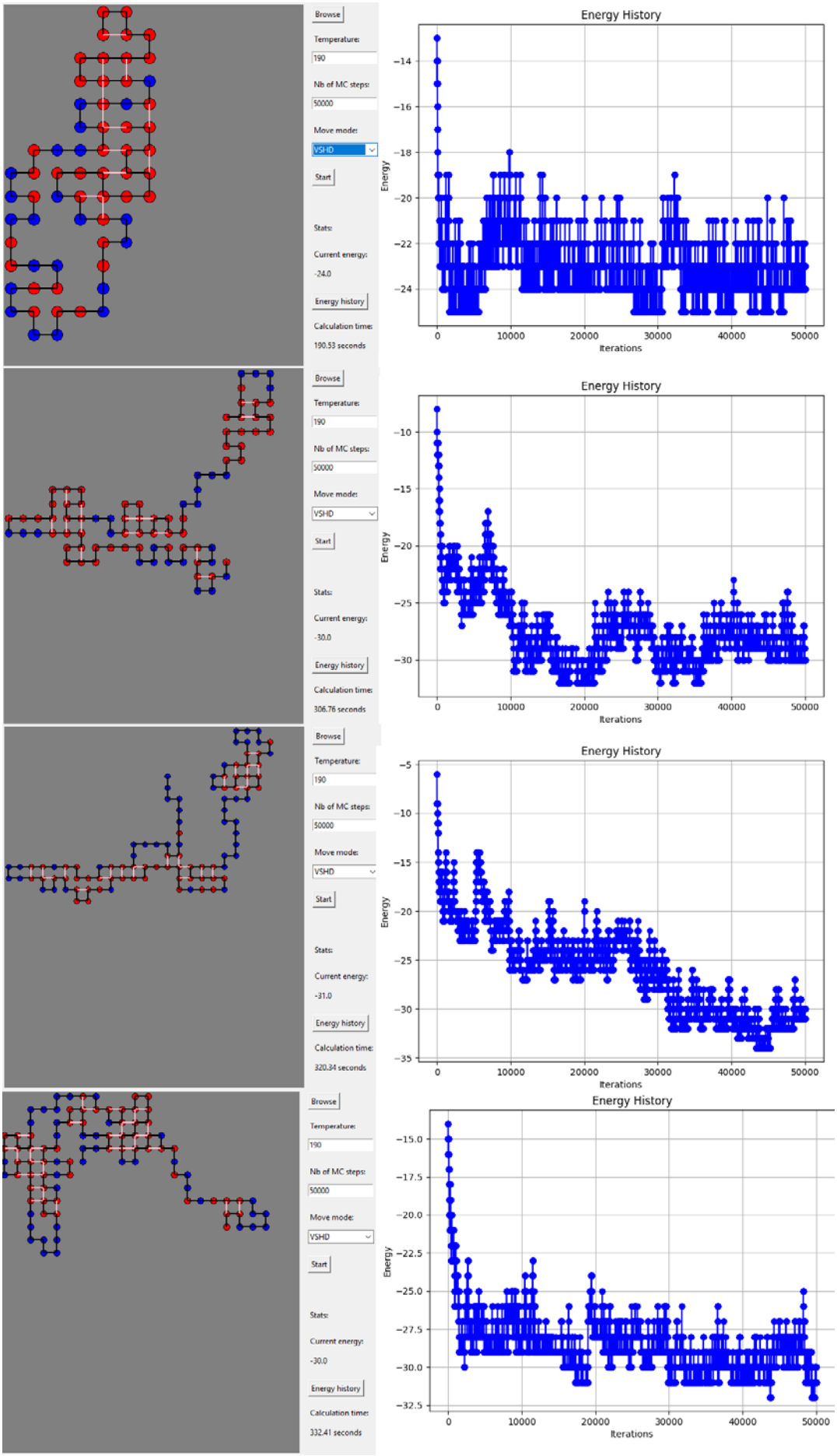

Browse
Temperature:
190
Nb of MC steps:
50000
Move mode:
VSHD
Start
Stats:
Current energy:
-24.0
Energy history
Calculation time:
190.53 seconds
Energy History
Energy
Iterations
Browse
Temperature:
190
Nb of MC steps:
50000
Move mode:
VSHD
Start
Stats:
Current energy:
-30.0
Energy history
Calculation time:
306.76 seconds
Energy History
Energy
Iterations
Browse
Temperature:
190
Nb of MC steps:
50000
Move mode:
VSHD
Start
Stats:
Current energy:
-31.0
Energy history
Calculation time:
320.34 seconds
Energy History
Energy
Iterations
Browse
Temperature:
190
Nb of MC steps:
50000
Move mode:
VSHD
Start
Stats:
Current energy:
-30.0
Energy history
Calculation time:
332.41 seconds
Energy History
Energy
Iterations

Figure 5: MC simulation with VSHD move mode applied to benchmark proteins.
Our algorithm was applied to bench marks proteins S1-8, S1-9, S1-10 and S1-11 respectively. Temperature was set to 190, move mode to VSHD move and MC steps to 50000.