Computer Simulation of Protein Folding Using 3D Hydrophobic-Polar Lattice Model

Richard Wang, Dr. Siqian He National Center for Biotechnology Information, NIH, Bethesda, MD, 20892

Background

- The protein folding problem is still a grand puzzle yet to be solved by researchers
- Hydrophobic interactions is the most important factor that governs how a protein folds
- Accurate simulations of real proteins are extremely computationally expensive (NP-complete)
- Simplified models are used to help understand the dynamics of how the native structure of a protein is formed and how to predict the native structure given the amino acid sequence
- A commonly used model for simulating how proteins fold is the Hydrophobic-Polar (HP) model
- In the HP model, each amino acid (bead) of the protein (chain) is classified as either hydrophobic or polar
- When folding, hydrophobic beads try to get as close to other hydrophobic beads as possible and away from water
- Lattice models use a coordinate system to place beads in a grid
- Monte Carlo method use statistical randomness to simulate water molecule bombardment on the protein structure

Goal

- To develop a protein folding simulation using the HP lattice model and Monte Carlo dynamics
- To compare contact-based and distance-based energy function folding capability

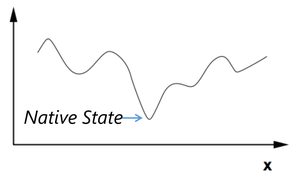
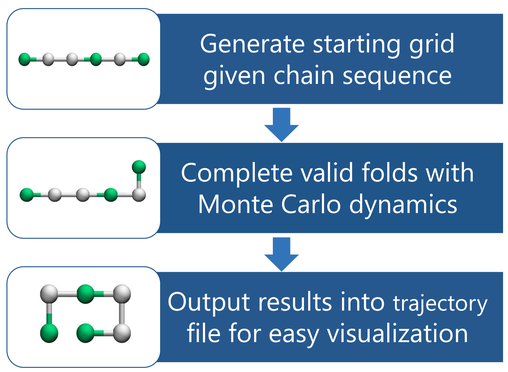


Figure 1: Sample energy landscape (2D)

Methods

- Written in Python 2.7.2 using VIM operated on Linux CentOS 7.2.1511
- Graphical output viewable through Visual Molecular Dynamics 1.9.2



Results

Chain: PHHPPPPPPHHPPPHHHPHPPHPHHPPHPPHPPHPPHHHHHHHHPPHH

Length: 48 Total steps: 7079 Minimum Found: -41 Native: -46

Key:
Green = Hydrophobic
White = Polar

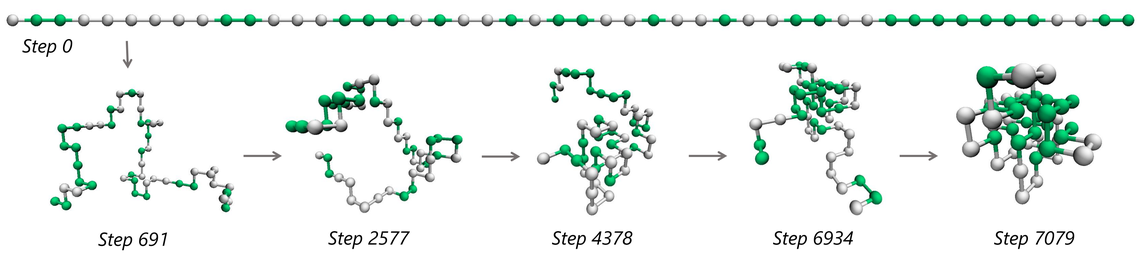


Figure 2: Folding progression of a 48 length sequence. Note "risky" folds near beginning, core formation and retention near end.

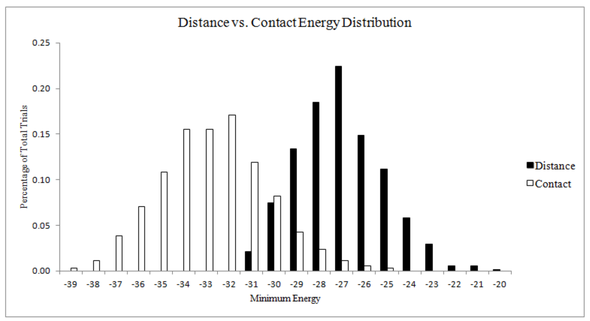


Figure 3: Comparison of distribution of minimum energies found with both energy functions

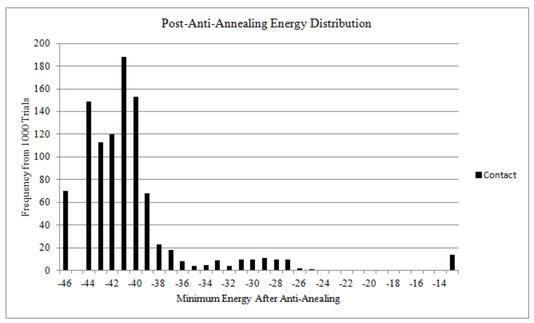


Figure 4: Histogram of minimum energy after anti-annealing using contact energy function. Notable points: -45, -38, -13.

Features

- Monte Carlo calculations

$$p = e^{-\frac{\Delta E}{kT}} \qquad r = random(0,1)$$

$$r < p\ ?$$

  - Program will accept a move if condition r<p is met
  - Replicates water molecule bombardment to escape local energy minimums in search for global energy minimum
- Contact-based and distance-based energy function
  - Contact: energy calculated using frequency of adjacent H-H contacts
  - Distance: energy calculated using the sum of inverse distances between H beads
- Simulated annealing: encourages risky folds near beginning and safer folds near end
  - Temperature of simulation decreases over time
- Anti-annealing: used to examine energy landscape

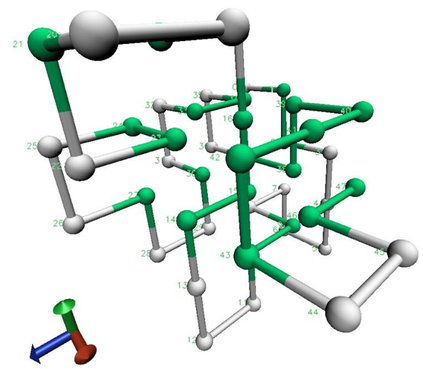


Figure 5: Lowest energy structure found using algorithm (-41)

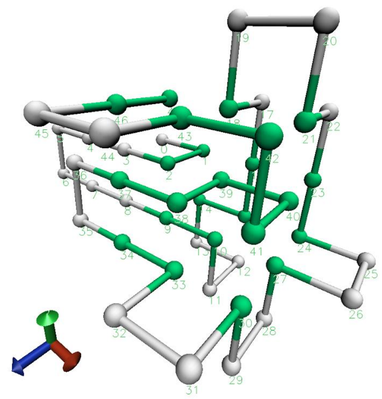


Figure 6: Verified native state structure (-46)

Advantages

- Monte Carlo method provides a significant increase in computational efficiency compared to enumeration "exhaustive search" style algorithms, which can take years to fold the chain shown on the left.
- Monte Carlo dynamics and statistical randomness allows the simulation to jump out of potential energy wells and local energy minimums
- Simulated annealing allows full exploration of potential energy wells

Conclusions

- Contact-based energy function superior over distance-based energy function
- Jagged energy landscape observed with "golf - hole" shaped global energy minimum
- Program successfully folds protein chain to near-optimal state in an efficient manner
  - Greater efficiency with shorter sequences
- An independent exhaustive search algorithm was used to verify that HP algorithm is capable of reaching the native folding state
- Interactive graphical output allows user to follow folding step-by-step
- Configurable parameters allow user to specify protein sequence, simulation temperature, trial length, number of repetitions

Future Directions

- Addition of heuristic folding patterns
- Incorporation of advanced algorithms that more accurately mimic actual protein folding
- Usage of parallel threading to conduct multiple folds simultaneously
- Take simulation off lattice
- Optimize code to further increase computational efficiency

Applications

- Structure-based drug design
- Precursor to development of new folding algorithms

References

Dill, K. A., Bromberg, S., Yue, K., Chan, H. S., Ftebig, K. M., Yee, D. P., & Thomas, P. D. (2008). Principles of protein folding - A perspective from simple exact models. Protein Science, 4(4), 561–602. doi:10.1002/pro.5560040401

Istrail, S., & Lam, F. (2009). Combinatorial algorithms for protein folding in lattice models: A survey of mathematical results. Communications in Information and Systems, 9(4), 303-346. doi:10.4310/CIS.2009.v9.n4.a2

Traykov, M., Angelov, S., & Yanev, N. (2016). A New Heuristic Algorithm for Protein Folding in the HP Model. Journal of Computational Biology, 23(8), 662–668. doi:10.1089/cmb.2016.0015

Yue, K., Fiebig, K. M., Thomas, P. D., Chan, H. S., Shakhnovich, E. I., & Dill, K. A. (1995). A test of lattice protein folding algorithms. PNAS, 92(1), 325-329. doi:10.1073/pnas.92.1.325