# Quantum Annealing with Turn Ancilla Encoding and Simulated Protein Folding for Drug Discovery Implications

Alice Liu

# Abstract

The Protein Folding Problem is characterized by the question on how a protein's amino acid sequence can determine its three-dimensional atomic structure. Increased understanding and accurate predicting of a protein's structure-function will allow for the accelerated formation of new protein-based drugs. The computational complexity of this is large, and conventional ways of using classical machine learning techniques are extremely restricted due to computational size limits run on current computers. With the rise of quantum computing relying on quantum mechanical properties such as superposition, entanglement and tunneling, this will allow for speedups in optimization methods. This specific study utilizes quantum annealing in order to simulate 2D lattice protein folding. The process tests the efficiency in the folding of amino acids in the varying lengths of 6, 9, and 12 residues with three different approaches: 1) simulated annealing with the conventional Monte Carlo method as the control variable 2) quantum annealing with turn ancilla encoding on a regular central processing unit (CPU) and 3) quantum annealing with turn ancilla encoding on a quantum processing unit (QPU) using D'Wave's hardware. An annealing algorithm is run where the lowest energy path for each sequence is determined by conducting random walks until the temperature cools to its minimum energy state. The method that leverages quantum annealing with turn ancilla encoding on a regular CPU yielded the highest number of instances that occurred in the lowest energy conformation when folding while quantum annealing with a QPU yielded the lowest runtime when reaching the minimum energy state.

# I. Introduction

*The Protein Folding Problem* has been an ongoing question since the 1960s on how a protein's amino acid sequence is able to dictate to its three-dimensional atomic structure [1]. Proteins naturally fold into their native structures by minimizing their free energy during the process. The protein achieving its native state or free energy minimum indicates that it is in its properly folded or assembled form. When proteins don't fold correctly to their lowest energy state, they are considered to be misfolded or denatured. A misfolded protein has a contorted resulting shape making it unfavorable to its cellular environment, which in turn may negatively impact the health of the cell regardless of the protein's function. Accumulation of these misfolded proteins can cause various degenerative and neurodegenerative disorders including Parkinson's disease, Alzheimer's disease, Huntington's disease, and cystic fibrosis [2]. Knowing how proteins fold, in terms of their folding sequences and patterns when reaching its minimum energy state, allows for a better understanding in their 3D structure-function relationship and enzymes, in order to accelerate the development of treatments for these misfolded-protein diseases.

Current computational methods to model the folding of these proteins are limited to constraints in both time and efficiency. Even with enhanced machine learning models demonstrated with programs such as DeepMind's Alphafold [3], which were tested to have a median prediction score of 92.4 in the Global Distance Test (GDT) during the tested folding of the proteins with an RMSD of 1.6 Angstroms, there are still limitations to the size of the protein that is able to be modeled. Current classical computers lack the computational power to effectively model the folding of larger proteins, where proteins larger than 150 residues are unable to be computed classically. No classical algorithm

currently exists that is able to find the lowest energy state of a lattice protein within polynomial time [4].

Within The Protein Folding Problem, individual questions include:

1. How can an amino acid sequence determine the 3D native structure of a protein?
2. How can a protein fold so quickly despite a vast number of possible conformations (Levinthal's Paradox)?
3. How does the protein know what conformations not to search? Is it possible to create an algorithm to predict a protein's native structure based on its amino acid sequence alone?

The third question is addressed in this study, where computational advantages within methods in quantum computing are leveraged, specifically with speedups and optimization. This approach attempts to find the lowest free-energy configuration of the minimum energy state (the native state of the protein) given its amino acid sequence. Lattice protein folding is taken to use, which is able to show a coarse-grained description of the protein folding problem. Lattice models are based on the hydrophobic-polar model (only models the hydrophobic interactions), which allow for the better modeling of protein conformations while defining the energetic properties of the amino acid [5].

In this study, experimental implementations of quantum annealing within lattice protein models were tested and ran on D'Wave's quantum annealer and compared to today's conventional methods of simulated annealing. The questions addressed are, how does the quantum annealing approach that is run on quantum hardware-based gates and circuits with varying lengths of amino acid sequences affect the number of successful instances and time to reach the lowest energy conformations? How do the number of successful instances as well as the time to reach the minimum energy state compare between the conventional simulated annealing with Monte Carlo and quantum annealing on a CPU (central processing unit) and QPU (quantum processing unit)? This approach specifically takes an amino acid sequence (6, 12, 18 residues) and performs 1)

conventional simulated annealing based on the Monte Carlo method 2) quantum annealing with turn ancilla encoding on a central processing unit (CPU) and 3) quantum annealing on D'Wave's annealer or quantum processing unit (QPU).

# II. Methods

The overall objective of this study is to leverage the optimization method of quantum annealing run on actual quantum hardware (on D'Wave's Annealer) to explore its benefits in terms of speedups and increased accuracy over current methods run on classical computers by simulating 2D lattice protein folding. The efficiency in the folding of amino acids in the varying residue lengths of 1) 6 residues 2) 9 residues and 3) 12 residues are tested in three different parts: 1) simulated annealing with the conventional Monte Carlo method 2) quantum annealing with turn ancilla encoding on a CPU and 3) quantum annealing on D'wave's QPU. The method of simulated annealing with conventional Monte Carlo functioned as the control variable being tested. A total of 9 simulated experiments were conducted where each experiment measured 1) the number of lowest energy conformations achieved and 2) the time it took for the amino acid sequence to reach its minimum energy state.

It was predicted that the method that leverages quantum annealing with turn ancilla encoding on a regular CPU would yield the most effective results in terms of the highest number of instances that occured in the lowest energy conformations in the least amount of time. This method is a hybrid of both classical simulated annealing and quantum annealing, where it is able to leverage the "speedups" within quantum annealing but at the same time not be susceptible to noise and decoherence as experienced with quantum hardware when running on a CPU.

The entirety of the experiment was conducted solely on Python with the assistance of a few other quantum-based software packages. The full list of the materials used are as follows:

- Modern Operating System consisting of x86 64-bit CPU (Intel / AMD architecture) with at least 4 GB RAM and 5 GB free disk space. For this experiment, the Mac OS Mojave Version 10.14.5 with a 1.1 GHz Intel Core m3 was used.
- Interactive web-based computational environment. Jupyter Notebook was used in this experiment.
- D'wave's Leap API Integration for 2000-Qubit Annealer
- D'wave's Composites Library, D'wave's Samplers Library
- Python Version 3.7.4

## Simulated Annealing Algorithm with Monte Carlo

Simulated annealing is defined as a probabilistic technique for approximating the global optimum of a function within a large search space, which in this study was leveraged by finding the lowest energy conformation for a protein given its amino acid sequence. Annealing itself is the heating and controlled cooling of a material to increase the size of its crystals and reduce their defects, which affects both the temperature and the thermodynamic free energy. When applied to simulated annealing, the slow cooling is represented by the slow decrease in the probability of accepting worse solutions as the solution space is explored. By allowing the acceptance of worse solutions, this allows for a more extensive search for the global optimal solution when finding the lowest energy conformation [6].

The algorithm accepts "worse" solutions compared to the current one as the solution space is explored. This allows for a more extensive search for the global optimum solution. For each step, the simulated annealing algorithm considers a neighboring state *S* to the current state *s* and decides probabilistically to either move the system to state S or stay in s. A *move*, or the way the states are changed in order to produce the neighboring states, results in small changes in the previous state in order to

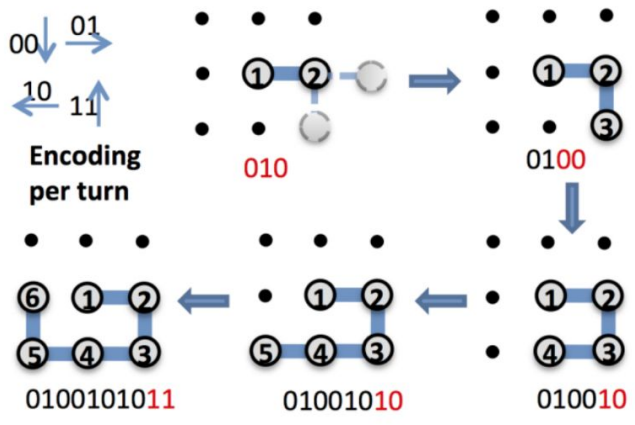
improve the solution by constantly improving its parts. The requirement to make the move is if it increases the state energy, ie. a slightly worse solution. All the "worse solutions" can be identified with temperature exponentially decreasing as the algorithm progresses in order to reach the optimal. The move is according to the temperature-dependent probabilities of selecting better/worse solutions, where the energy of the new state is assessed using an objective function. The energy level is then compared to the previous state and it is decided whether to accept the new solution or reject it based on the current temperature. The probabilities lead the system to move to states of lower energy and this is repeated until it reaches its minimum when the temperature slowly cools. The goal is bringing the system from an initial state to a state with the *minimum possible energy*.

The steps of the algorithm used in this study's implementation are as follows:

1. Define a schedule for the annealing temperature T
2. Randomly choose residue $r_i$
3. Perform a random walk with respect to $r_{(i-1)}$
4. Compute energy change in energy $\Delta E = E - E'$
5. Accept step if exp(-ΔE/T) expresses the probability of a state of energy E relative to the probability of a state of zero energy > random.uniform(0,1).

   If $\Delta E \leq 0$, always accept. The probability is defined as: $P = \frac{1}{1+ e^{\frac{-\Delta E}{T}}}$

## Turn Ancilla Encoding Method




This visual denotes the binary representation of a 6-residue lattice protein within turn ancilla encoding [7].

Two qubits are needed per bond and the turn directions are denoted by 00 (downards), 01 (rightwards), 10 (leftwards) 11 (upwards)

Turn ancilla encoding is able to simulate random walks within a 2D lattice by adding constraints and requirements on where the protein folds during the process of minimizing the objective function when finding the lowest energy conformations. By applying turn ancilla encoding with quantum annealing, it introduces ancillary qubits into the Hamiltonian energy function in order to encode information about the interactions between the amino acids [8]. These constraints enhance the accuracy and reduce redundancies of the folding in the amino acid sequence, decreasing the computational barriers in the case when a protein misfolds in the process. As a result, by incorporating this method with annealing, it will be more efficient than the conventional one-by-one "test and take" process of simulated annealing.

The energy function constructed with the turn ancilla scheme consists of four subcomponents E-back, E-overlap and E-pair as defined:

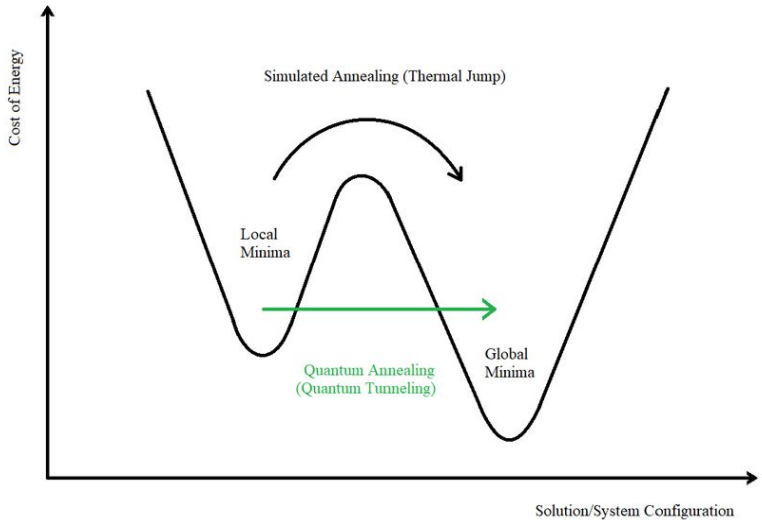
$$E(q) = E_{back}(q) + E_{overlap}(q) + E_{pair}(q)$$

The E-back component penalizes protein fold and marks the protein as invalid if back-to-back when 2 consecutive edges go in between the same pair of vertices (to ensure the ground state does not have these properties). For example, if an *n* edge is represented by vertices ($v_2$, $v_3$) and is followed by another edge ($v_3$, $v_1$), this is invalid as the protein will go back on itself. The E-overlap penalizes the protein fold if the lattice protein folds over itself, with the intention in reducing the number of ancillary qubits required. The E-pair marks the interaction between non-bonded acids that are adjacent on a lattice.

# Quantum Annealing Algorithm

Quantum computing itself is characterized by the use of quantum phenomena, such as superposition, entanglement and tunneling to perform computations. Due to superposition and entanglement, the main advantage for quantum computers comes from how they can process a vast number of calculations simultaneously. While our current computers that we use leverage bit strings of 1s and 0s, which are evaluated one by one, quantum computers leverage qubit strings of 1s, 0s, and superpositions of 1s and 0s (10, 01). This allows for $2^n$ simulations to be completed all at the same time. For example, a 100 qubit computer can run $2^{100}$ calculations simultaneously. This speedup is what makes quantum computing methods ideal for optimization.

Quantum annealing is a technique that is able to find the global minimum of a given objective function over a set of candidate states by using quantum computation methods (quantum tunneling), which allows for speedups over its classical counterpart [9]. The concept of quantum tunneling can be seen below:




*Quantum tunneling* is an effect during quantum annealing that allows the passage during the search through an energy barrier. As opposed to going up and around the barrier which is seen with the simulated annealing method, it reduces both time and computational power when taking this step [10].

This quantum mechanical tunneling effect allows for more efficient exploration of potential energy landscapes. Quantum fluctuations (tunneling) occurs between states representing different model protein conformations or folds, which makes it extremely effective for optimization problems by searching for the best configurations out of many different possible combinations in order to find the minimum energy state.

Within the hardware in the quantum processor, each qubit goes from a superposition state to its classical states of 0 or 1. An energy landscape is defined as a double well potential. At first, the qubit falling in the 0 and 1 state are even. What state the qubit falls in can be controlled by applying an external magnetic field to the qubit with what is called a *bias*. This in turn tilts the double well potential and increases the probability of the qubit ending up in the lower well (closer to reaching the minimum state), where it is able to automatically minimize its energy in the presence of an external magnetic field. In addition to applying biases, *coupling* is also used within the quantum processor buildup, where it is able to start linking the qubits together, which in turn will start influencing each other. Couplers make it energetically favorable for qubits to end up in the same state (either 00 or 11) or opposite states (either 01 or 10). There are now 4 states: 00, 01, 10, 11 where relative energies of these states depend on the biases of each qubit and the coupling between them. If the coupler wants the 2 qubits to be the same (00 or 11), it will lower the energy of those 2 states in comparison to the other states. Quantum computing essentially chooses a whole set of biases and couplers (determines direction and strength) that defines an energy landscape while quantum annealing in this approach helps find the minimum energy of that energy landscape.

Compared to simulated annealing, quantum annealing is more adaptive, being able to work with a gradually decreasing parameter (instead of a fixed parameter) while not being limited by the barrier width and height. The qubits with couplers and biases also allow for a more efficient search as testing with the algorithm is able to be completed simultaneously due to superposition.

The steps of the quantum algorithm used in this study's implementation are as follows:

1. Define a schedule for annealing temperature T
2. Randomly choose $i^{th}$ qubit $q_i$
3. Perform a qubit flip
4. Compute energy change in energy $\Delta E = E - E'$
5. Accept step if exp(-ΔE/T) expresses the probability of a state of energy E relative to the probability of a state of zero energy > random.uniform(0,1).

   If ΔE <=0, always accept.

   The probability is defined as: $P = \frac{1}{1+ e^{\frac{-\Delta E}{T}}}$

## Preprocessing and Coded Implementation

A set of preprocessing was held before the implementation of the algorithms. This includes defining the variables to hold the combinations of amino acids in the chain for the interaction of the energy components, defining the possible directions during the random walk, and defining the energy evaluations including measuring and outputting current energies, while evaluating probabilities with neighboring states.

The workflow process for the simulated annealing implementation includes first defining the sequence to fold: YYC-PET-GTWY-AGT, defining the annealing schedule, running the simulated annealing algorithm, and outputting the total number of lowest energy conformations.

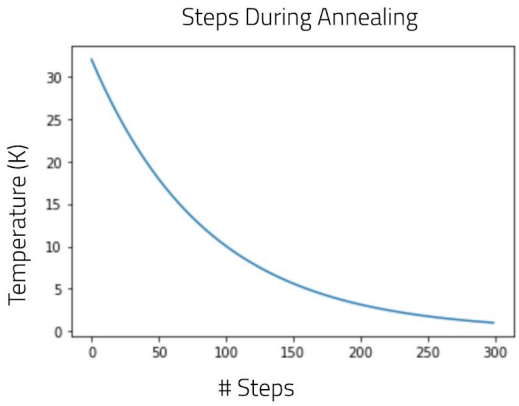
The workflow process for the quantum annealing with the turn ancilla encoding implementation includes defining the sequence and annealing schedule, defining the energy functions of E-overlap, E-back and E-pair, while adjusting the walk based on these components. The quantum annealing algorithm is then inputted and ran where the total number of lowest energy conformations is outputted.

The workflow process for the quantum annealing on D'Wave architecture also includes defining the sequence to fold and the annealing schedule, as well as importing the library packages of D'wave's Sampler and Embedding Composites and integrating with D'Wave 2000-qubit annealer API with the local machine. After the integration is complete, the solver and corresponding adjacent graph for embedding is defined as well as the adaption of qubits into the energy functions and scaling the energy landscape.

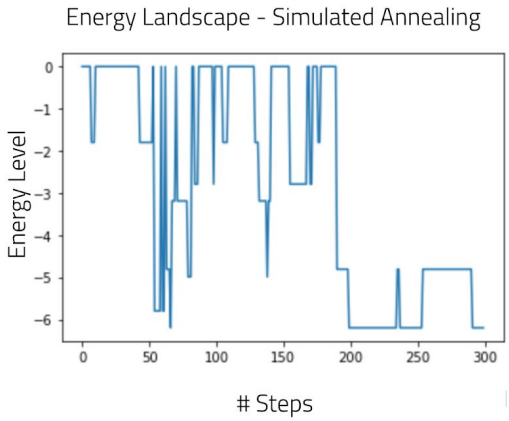
# III. Results

## Annealing Schedule and Energy Landscape

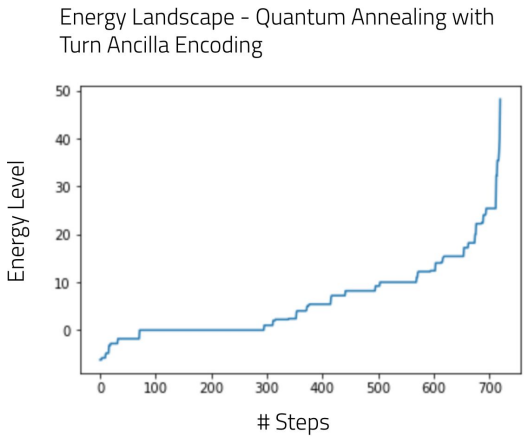
**A.**



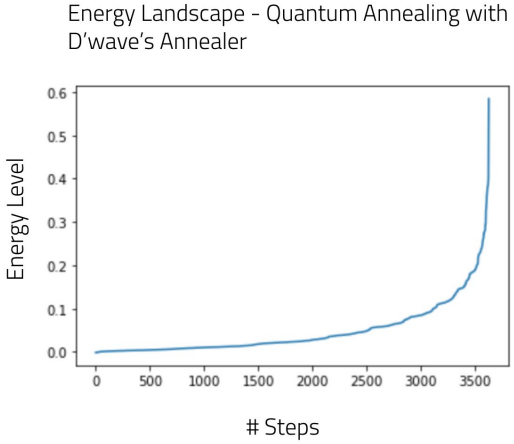

**B.**




**C.**




**D.**



The annealing schedule for all three implementations is defined (A) with the total number of steps being defined as 300 as the temperature starts from 30 K° and anneals to 0 K°. The energy landscape for each of the three implementations are also visually shown. The simulated annealing approach (B) upholds a fixed parameter which results in varying depths in wells during each step as the temperature anneals and the energy levels decrease. The quantum annealing approach with turn ancilla encoding (C) and on D'Wave's hardware (D) have a gradually decreasing parameter which results in less varying depths within the energy potential wells, instead increasing the energy levels as the number of steps increase. This is due to how quantum annealing finds the "worst solutions," ie. neighboring solutions that increase the state energy, which are eliminated in order to find the optimal.

## Simulated Annealing with Monte Carlo

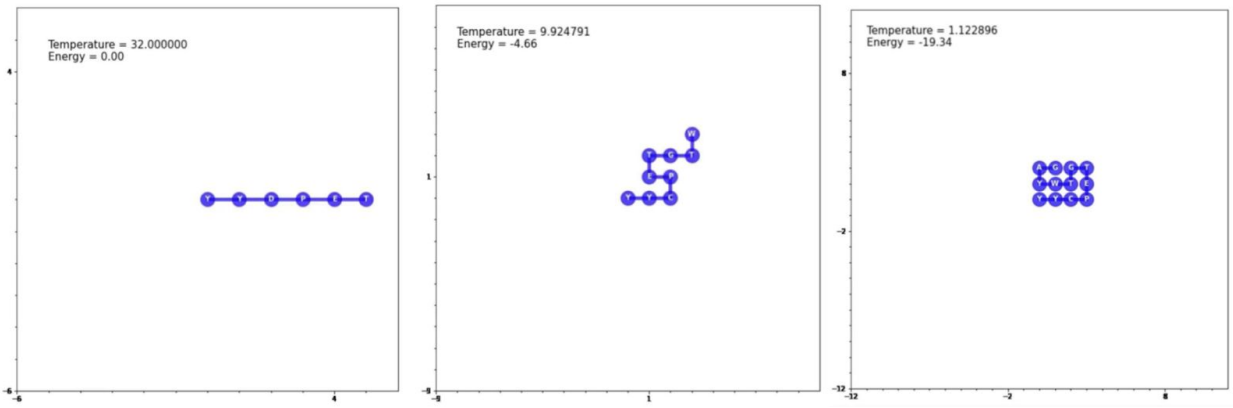



Figure 1: visual representation of the amino acid sequence in each iteration (6 residues, 9 residues, 12 residues) reaching its minimum energy and native state during simulated annealing. The final annealing temperatures as well as the energy levels are indicated.

| | % Instances In Lowest Energy Conformation | Time (s) To Reach Minimum Energy State |
|---|---|---|
| 6 Residues | 18.6% | 12.2 |
| 9 Residues | 13.3% | 25.0 |
| 12 Residues | 6.1% | 38.0 |

Figure 2: the number of instances divided by the total number of steps in each amino acid is outputted to get the percentage of instances in the sequence, as well as the time it took to reach the final minimum energy state. As the number of residues increase, the percentage of instances within the lowest energy conformation decrease and the time to reach the minimum energy state increases.

## Quantum Annealing

### Turn Ancilla Encoding

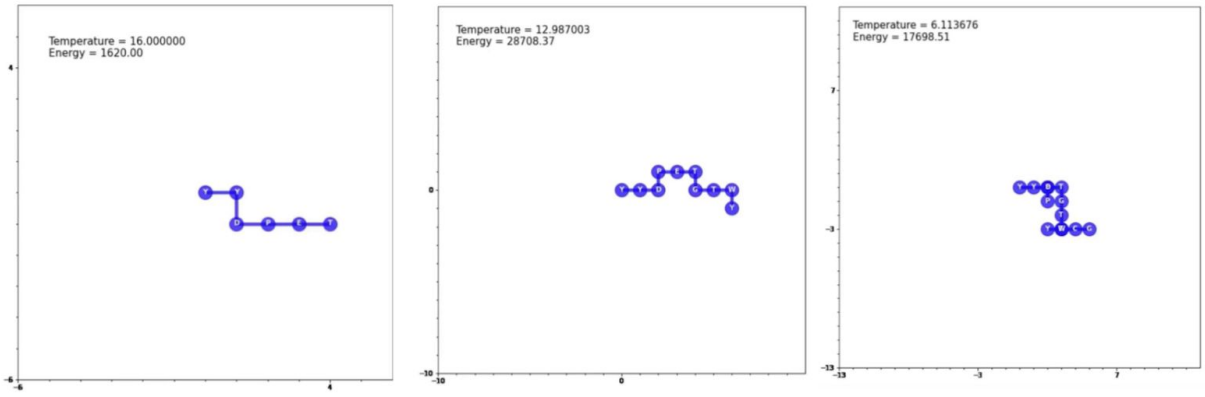
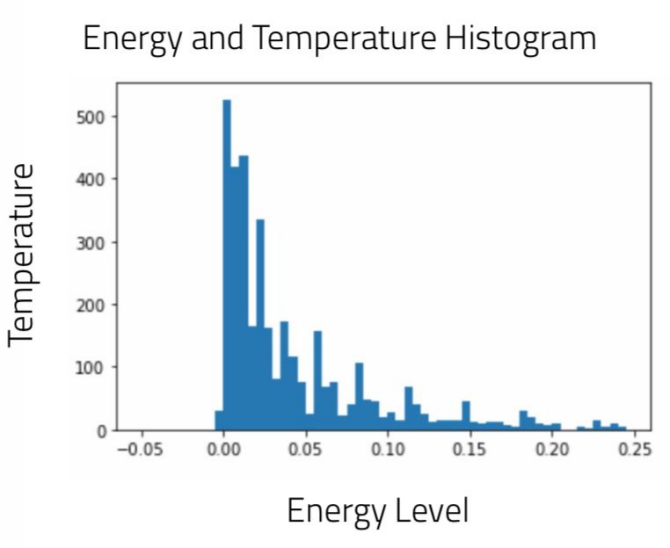



Figure 3: visual representation of the amino acid sequence in each iteration (6 residues, 9 residues, 12 residues) reaching its minimum energy and native state during quantum annealing with turn ancilla encoding. The final annealing temperatures as well as the energy levels are outputted and indicated.

**E.**

| | state | isvalid | energy | Hb | Ho | Hi | count |
|---|---|---|---|---|---|---|---|
| 0 | 1011011100111000110 | True | -1.391382e-03 | 0 | 0 | -6.20 | 3 |
| 1 | 1111011101111110000 | True | -1.301616e-03 | 0 | 0 | -5.80 | 8 |
| 2 | 0111011101111100000 | True | -1.301616e-03 | 0 | 0 | -5.80 | 1 |
| 3 | 1011011101100000010 | True | -7.158887e-04 | 0 | 0 | -3.19 | 1 |
| 4 | 1111011101111010000 | True | -6.754937e-04 | 0 | 0 | -3.01 | 3 |
| 5 | 0110011100110100000 | True | -6.261221e-04 | 0 | 0 | -2.79 | 1 |
| 6 | 0010001100110111100 | True | -6.261221e-04 | 0 | 0 | -2.79 | 2 |
| 7 | 1111011100110110100 | True | -6.261221e-04 | 0 | 0 | -2.79 | 2 |
| 8 | 1111011101110110000 | True | -6.261221e-04 | 0 | 0 | -2.79 | 4 |
| 9 | 0001000101110001101 | True | -4.039497e-04 | 0 | 0 | -1.80 | 1 |
| 10 | 0111011001110011001 | True | -4.039497e-04 | 0 | 0 | -1.80 | 1 |
| 11 | 1111011101111110001 | True | -8.976661e-05 | 0 | 0 | -0.40 | 1 |
| 12 | 0001000101110001100 | True | -2.081668e-16 | 0 | 0 | 0.00 | 1 |
| 13 | 0000011000000000000 | True | -1.387779e-17 | 0 | 0 | 0.00 | 1 |

**F.**




The iteration process (E) during annealing for the individual energy components is outputted for the turn ancilla encoding method. This includes the state of each fold encoded in binary representation, the energy state, and the values in the energy components of E-back, E-overlap and E-pair. A histogram (F) representing the energy level in relation to the temperature for the turn ancilla encoding method is shown.

| | % Instances In Lowest Energy Conformation | Time (s) To Reach Minimum Energy State |
|---|---|---|
| 6 Residues | 86.6% | 8.0 |
| 9 Residues | 70.8% | 12.0 |
| 12 Residues | 61.0% | 24.0 |

Figure 4: the number of instances divided by the total number of steps in each amino acid is outputted to get the percentage of instances in the sequence, as well as the time it took to reach the final minimum energy state. Similar to the results from the simulated annealing approach, as the number of residues increase, the percentage of instances

within the lowest energy conformation decrease and the time to reach the minimum energy state increases.

## Quantum Annealer - D'wave's QPU

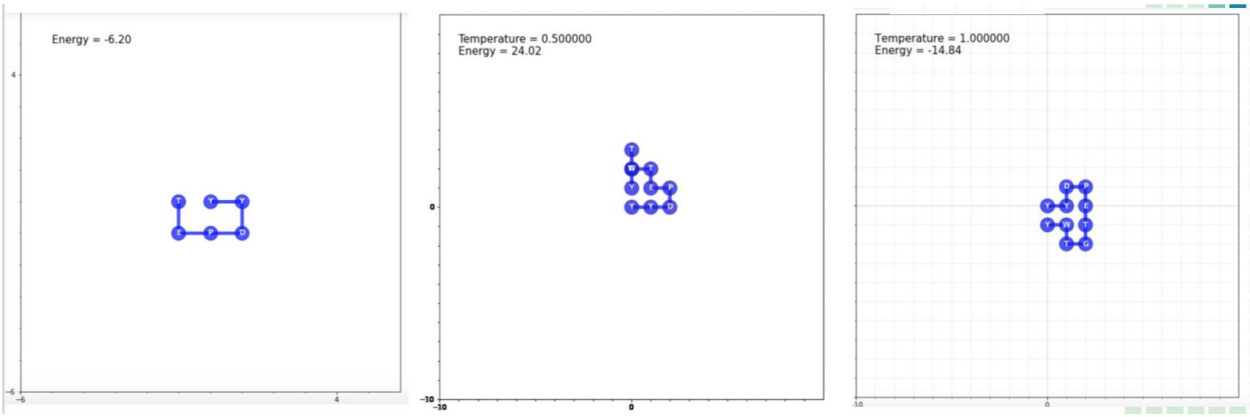



Figure 5: visual representation of the amino acid sequence in each iteration (6 residues, 9 residues, 12 residues) reaching its minimum energy and native state during quantum annealing with D'Wave's QPU. The final annealing temperatures as well as the energy levels are outputted and indicated.

| | % Instances In Lowest Energy Conformation | Time (s) To Reach Minimum Energy State |
|---|---|---|
| 6 Residues | 74.7% | 2.6 |
| 9 Residues | 62.3% | 8.0 |
| 12 Residues | 51.4% | 12.0 |

Figure 6: the number of instances divided by the total number of steps in each amino acid is outputted to get the percentage of instances in the sequence, as well as the time it took to reach the final minimum energy state. Similar to the results from the simulated

annealing approach and quantum annealing with turn ancilla encoding, as the number of residues increase, the percentage of instances within the lowest energy conformation decrease and the time to reach the minimum energy state increases.

# IV. Discussion

The quantum annealing method yielded more efficient results compared to simulated annealing, in terms of a higher percentage in the number of instances achieving the lowest energy conformations and a lower runtime when reaching the minimum energy state. The highest number of instances in successful energy conformations was yielded by the second approach of quantum annealing with turn ancilla encoding. The lowest time to reach the minimum energy state was yielded by the third approach of quantum annealing on D'Wave's quantum hardware. The conventional simulated annealing (control variable) had the smallest yield with the lowest percentage in the number of total successful energy conformations as well as the longest time to reach the minimum energy state across all residue lengths. A quantum speedup is demonstrated from this experiment.

Explanations for the outputted results are due to the quantum mechanical tunneling effect that is held in quantum annealing. The quantum annealing algorithm is able to work with a gradually decreasing parameter and is not limited by barrier heights or widths within the energy landscape, allowing for more efficient exploration, as opposed to simulated annealing that works with a fixed parameter and is therefore inefficient when barriers are high. In addition, the QPU from D'Wave is still a state-of-the-art system that can sometimes be subject to noise when testing, and would therefore be somewhat less efficient and in-depth in processing compared to the CPU.

The implications for leveraging protein folding in computational design and drug discovery are wide. The shape of the protein accompanied by its folding process is able to dictate its specific function in the body. By being able to predict a protein's structure,

this will enable the synthesizing of new protein-based drugs to treat different kinds of diseases. Identifying different states explored by a protein and the protein landscape allows us to identify allosteric sites and newer targets during protein misfolding and aggregation[11]. Knowing which proteins to target and how the protein is folded will help make molecules with complimentary shapes to different parts of the protein in order to influence that protein's actions. New computational design for drugs can now be generated and created.

# V. Conclusion

This study developed an approach to using quantum annealing methods to simulate the 2D lattice protein folding while comparing the experimental results to today's conventional simulated annealing method. This was completed by using three main methods of implementation. Within each method, three types of amino acid sequences were tested: 1) 6-residue amino acid sequence 2) 9-residue amino acid sequence 3) 12-residue amino acid sequence. The three methods include the conventional simulated annealing with Monte Carlo, quantum annealing with turn ancilla encoding on a regular CPU and quantum annealing on the D'Wave annealer QPU. The total number of successful energy conformations as well as the time it took to reach the minimum energy state were determined within each trial in order to determine the general efficiency during the simulations.

A quantum speedup was demonstrated based on the experimental results. The quantum annealing methods yielded the most efficient results, holding the highest number of successful energy conformations and the least amount of time to reach the minimum energy state across all residue lengths.

# VI. Bibliography


1. Dill, K. A., Ozkan, S. B., Shell, M. S., & Weikl, T. R. (2008). The protein folding problem. *Annual review of biophysics*, *37*, 289–316. https://doi.org/10.1146/annurev.biophys.37.092707.153558
2. Chaudhuri TK, Paul S. Protein-misfolding diseases and chaperone-based therapeutic approaches. FEBS J. 2006 Apr;273(7):1331-49. doi: 10.1111/j.1742-4658.2006.05181.x. PMID: 16689923.
3. Senior, Andrew W., et al. "Improved Protein Structure Prediction Using Potentials from Deep Learning." *Nature News*, Nature Publishing Group, 15 Jan. 2020, www.nature.com/articles/s41586-019-1923-7.
4. Outeiral, Carlos, et al. "The Prospects of Quantum Computing in Computational Molecular Biology." *ArXiv.org*, 26 May 2020, arxiv.org/abs/2005.12792.
5. "Estimation of Effective Interresidue Contact Energies from Protein Crystal Structures: Quasi-Chemical Approximation." *Macromolecules*, pubs.acs.org/doi/10.1021/ma00145a039.
6. Dimitris Bertsimas. John Tsitsiklis. "Simulated Annealing." Statist. Sci. 8 (1) 10 - 15, February, 1993. https://doi.org/10.1214/ss/1177011077
7. Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M. *et al.* Finding low-energy conformations of lattice protein models by quantum annealing. *Sci Rep* **2,** 571 (2012). https://doi.org/10.1038/srep00571
8. Robert, A., Barkoutsos, P.K., Woerner, S. *et al.* Resource-efficient quantum algorithm for protein folding. *npj Quantum Inf* **7,** 38 (2021). https://doi.org/10.1038/s41534-021-00368-4
9. Babej, Tomáš, et al. "Coarse-Grained Lattice Protein Folding on a Quantum Annealer." *ArXiv.org*, 2 Nov. 2018, arxiv.org/abs/1811.00713.
10. Babbush, Ryan, et al. "Construction of Energy Functions for Lattice Heteropolymer Models: A Case Study in Constraint Satisfaction Programming

and Adiabatic Quantum Optimization." *ArXiv.org*, 11 June 2013, arxiv.org/abs/1211.3422.

11. Nero TL, Parker MW, Morton CJ. Protein structure and computational drug discovery. Biochem Soc Trans. 2018 Oct 19;46(5):1367-1379. doi: 10.1042/BST20180202. Epub 2018 Sep 20. PMID: 30242117.