Designing Proteins with Quantum and Neuromorphic Computing

Robert Strauss

April 6, 2022

New Mexico Supercomputing Challenge
Final Report
Team 27
Los Alamos High School

Team Members
- Robert Strauss
Teachers
- Dr. Vikram Mulligan
- Dr. Garrett Kenyon
Project Mentor
- Kyle Henke

Contents

1 Executive Summary

I acquired data of residue interaction energies for many protein-packing problems. I wrote python code transforming this data into Q matrices for each problem, involving creating an encoding scheme to represent the residue sequence in binary in such a way tabulated interaction energies can be translated into the Q matrix. I wrote code to upload this Q matrix to a quantum computer (the D-Wave 2000Q) and a neuromorphic computer (the Intel loihi), with instructions to draw several thousand samples by repeatedly annealing or simulating. I received the set of samples from each machine, and compared these solutions to solutions generated by a classical simulated annealing algorithm. I analyzed which hardware performed the best and wrote various tests to determine how my algorithm could be improved. I created a novel encoding scheme which, tested on the isomorphic map-coloring problem, vastly outperformed the algorithm currently used in the literature, producing 3x as many binary sequences that represented valid solutions.

2 Introduction

2.1 The Protein Design Problem

Biological proteins are composed of a chain of amino acids (the residues) which fold the protein into a definite structure due to the forces between the residues. This structure, which is completely defined by the sequences of amino acids and potential energy of each interaction, gives the protein some function, such as catalyzing a reaction. Protein structure prediction has the goal of calculating the final structure of a protein given its amino acid sequence. This was named breakthrough of the year by Science Magazine, going from being considered infeasible to a solved problem in a matter over the past decades. Protein design is the reverse, starting from a protein structure and going backwards to find the amino acid sequence that would fold to such a structure. This is perhaps more significant, since being able to find the sequence that would fold to a desired structure would allow scientists to create that protein in a lab, giving the ability to make designer proteins for any function. This could mean binding a neurotoxin or COVID molecule to stop it, being a highly precise targeted drug, or self-assembling into biological molecule-scale machines or computers. However, protein design is combinatorially complex. There is an exponential amount of possible sequences, making it infeasible to guess and check on a classical computer.

2.2 Quantum Annealing Computing

Quantum computers can theoretically find a global optimum solution to NP-complete problems in O(1) time, making them excellent for solving NP-complete problems. In reality, neither quantum annealing or quantum gate computing has shown this level of performance yet. An active area of research is finding what problems are best matched for these computers, so the performance they can reach currently is useful. Protein design, being an optimization-based, combinatorially complex problem with large utility, is an excellent candidate to try applying quantum computing to. Quantum annealing, rather than gate computing, is best matched to this scenario.

In a quantum annealer such as the D-Wave 2000Q there is a set of qubits which are either 1 or 0 when observed and collapsed, but can be in a quantum superposition of 1 and 0 before collapse. The values of these qubits (once collapsed) represent the answer to the optimization problem posed by the potential applied to them through quantum entanglement.

2.3 Neuromrophic Computing

Neuromorphic computers don't use the traditional Von-Neumman model of seperate memory and computation, but instead take inspiration from the biological brain to perform calculations. These systems are composed of an array of leaky integrate and fire neurons which accumulate voltage from spike signals recieved from other neurons, then fire their own spike signals through synapse transmission lines. This spike-based, memory-integrated model has the advantages of using thousands of times less energy than traditional computers for the same level of calculation [BCHE19] and can solve optimization problems, exploiting random noise as a resource to cross energy barriers [JHM16]. However, its difficult to exploit these capabilities since these computers can't be programmed with machine code or any programming language. Instead, parameter values have to be loaded onto the device, setting the connectivity of the neurons. This makes it difficult to encode just any problem for this device, but they work especially well with quadratic unconstrained binary optimization.

2.4 Quadratic Unconstrained Binary Optimization

In Quadratic Unconstrained Binary Optimization (QUBO), the values being optimized over are a set of bits. The criterion for optimization is the weighted sum of the products of each pair of bits:

$$E(x) = \sum_i Q_{i,i} x_i + \sum_{i,j} Q_{i,j} x_i x_j \min_x E(X) \tag{1}$$

where x is a binary vector of n bits, Q is the n by n matrix defining the problem, and E(x) will be called the energy. Both the D-wave 2000Q and Intel loihi are set up with a given Q matrix specific to

the problem, and run a thousand times or more to sample solutions with low energy E (although they may not necessarily find the global minimum).

In the D-Wave, the $Q_{i,j}$ represents the strength of the entanglement made between qubits i and j. If both are 1, this weight contributes to the total energy of the system. Since it starts in the lowest-energy state, the system aims to stay in the global minimum of this function, so when measured the values of the qubits probabalistically collapse, often to a solution minimizing this equation.

Neuromorphic computers also manage to minimize this form of the energy function [JHM16]. If neurons i and j are connected together symmetrically with a synapse of weight proportional to $-Q_{i,j}$, one's spiking will inhibit the other's. With the addition of random noise to make spiking more probabilistic, this leads to the system relaxing into a low energy configuration, where whether a neuron is spiking or not represents a bit in the solution.

3 Research Question

The goal of this work was to test the protein design problem on quantum and neuromorphic hardware to see how well these computers were suited for the problem. Simulated annealing, a classical computing algorithm, was applied to the same cases for a comparison. What hardware is better for the problem of protein design: quantum, neurmorphic, or classical computing?

4 Prior Work

Dr. Mulligan *et al.* designed and tested their QPacker algorithm which successfully created a peptide running on the D-Wave 2000Q quantum annealer [MMM+19]. This is the only prior example of quantum computing being applied to the protein design problem. Neuromorphic computing has never been applied to the problem before now.

5 The Challenge: Encoding Protein Design as a QUBO

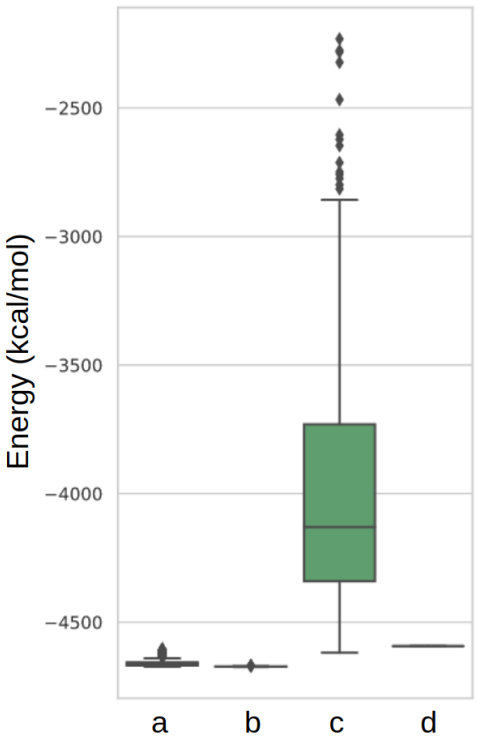
To use quantum or neuromorphic computers for protein design, the protein design problem simply has to be encoded as a Q-matrix according to equation 1, and the optimization parameter x should somehow represent the amino acid sequence in binary. Protein design already has some very convenient parallels with QUBO. Proteins fold to minimize their total potential energy, meaning the sequence of a fold will minimize the sum of one- and two-body interaction energies of contacting residues. In other words, if amino acid A is contacting amino acid B, the interaction energy of A and B is added to the total: $E+ = U(A, B)x_A x_B$ where x_A is a bit representing whether amino acid A is in the first position, and x_B is a bit representing whether amino acid B is in the other position. So by having a bit for every type of amino acid in every position, the Q-matrix is constructed of copies of a table of the pre-computed interaction energies of every pair of amino acids. This system is called one-hot, where there as many bits per position as there are amino acids, but only one is equal to 1. However, requiring only one bit have the value 1 is a constraint, which QUBO doesn't allow. This constraint can't be programmed into the computers, which only take the input of a Q-matrix. Instead, a soft constrain can be used, adding a high energy "penalty" to solutions with two bits in one sequence position equal to 1. This is a simple addition to the Q-matrix, and should decrease the probability of finding solutions which are invalidly encoded, with more than one non-zero bit in a position.

6 Implementation

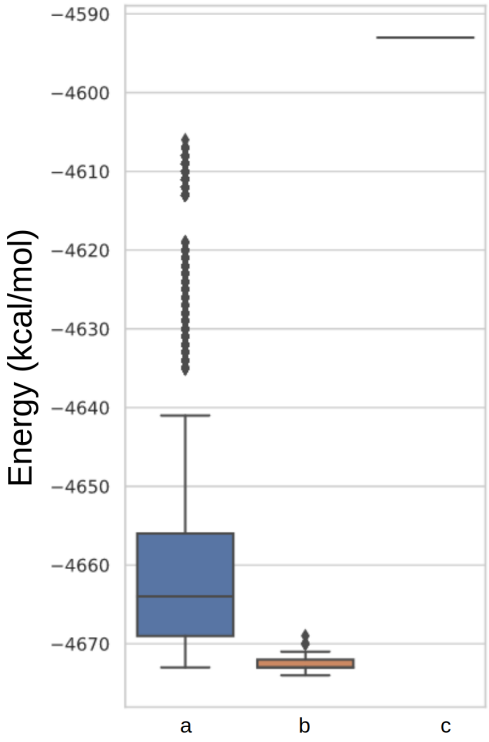
I wrote thousands of lines of python code reading in Rosetta amino acid interaction energy data, adding artificial encoding constraint terms, sending the results to quantum and neuromorphic computers, and analyzing solutions.

This included visualization method, file I/O methods, simulated annealing tests, one-hot constraint confirmation, energy calculations, and more.

To prevent confusion of the reader, it's important to emphasize **quantum and nueromorphic computers don't run code.** These devices are physics-based systems which evolve to find low-energy



(i) Box-and-whiskers plot showing the distribution of energy (y-axis) of solutions generated by (a) random sequences, (b) simulated annealing, (c) D-Wave 2000Q, and (d) Intel loihi.



(ii) Box-and-whiskers plot showing the distribution of energy (y-axis) of solutions generated by (a) random sequences, (b) simulated annealing, and (c) Intel loihi.

solutions to a QUBO problem. They are not programmed, they are set up with a Q-matrix, and run many times to generate a distribution of different samples.

7 Results of Hardware Comparison

On a particular small protein packing problem, I generated 10,000 solution samples from each of simulated annealing, D-Wave 2000Q, and Intel loihi. Due to an unknown technical issue, loihi would appear to fail after only a few samples, producing the same solution repeatedly. I calculated the energy of each solution and plotted the distribution of energies, shown in Figure 1i. Many of the solutions from D-Wave and loihi failed the soft constraint, producing solutions with multiple non-zero bits in a position, an invalid encoding which doesn't represent a sequence. These solutions are included in the energy distribution plot, with the energy penalty term for failing the constraint included. D-wave incurs so much from this penalty that the distribution dwarfs the others. To make the data more clear, I plotted the energy distributions excluding the D-Wave, shown in Figure 1ii. As a control, I generated random amino acid sequences and translated these to the one-hot binary representation of the problem, calculating these energies and plotting them. These random sequences would be very unlikely to fold in a lab, but they at least match the constraint, giving a reference for what kind of energy might be required for a good solution.

8 Novel Encoding Approach

Seeing the clearly worse performance of the D-Wave system, with much higher-energy solutions on average than any other architecture or the randomized control group, I thought about how to improve this given the hardware. I scaled back to the isomorphic map-coloring problem, which has a much simpler Q-matrix and had been done numerous times in the literature [Dah13]. This problem uses the same one-hot encoding to represent the color of countries in a map. The approach in the literature for

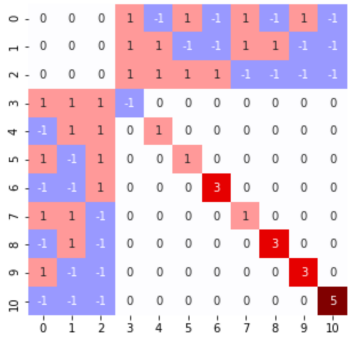


Figure 2: Q matrix of my novel one-hot/binary integer hybrid encoding scheme. The first 3 bits may encode any integer from 0-7 in binary. Of the next 8 (2^3) bits, only the one corresponding to the integer will be 1 to minimize energy. This can be scaled to any number of bits and used in map-coloring and similar QUBO problems.

enforcing one-hot encoding, which I used, implemented it as a soft constraint by assigning a very high energy penalty to solutions with a one-hot encoded with more than one non-zero bit. That is, $Q_{i,j}$ had a very high energy for all i and j representing mutually exclusive options (such as two amino acids in one position). This meant a large portion of values of the Q-matrix were non-zero, which should mean a large number of entanglement connections between each qubit in the D-Wave. However, the D-Wave only supports a finite number of connections between qubits, so when there are too many it creates a chain, connecting one qubit to many proxy qubits with a strong weights to give them all the same value. These chains weaken the efficiency of the annealer, wasting many samples due to chain breaks as qubits in a chain fail to have the same value. To decrease the amount of chains needed, I aimed to create a different way of encoding the sequence or penalty to make the Q matrix more sparse. I came up with a one-hot/binary hybrid encoding system. This had a bit for each option like the one-hot system, but also had a number (equal to the log base 2 of the number of options) of extra bits, which encoded an integer in binary corresponding to which of the bits should be 1. Rather than every bit being connected to all the others, in my novel encoding, each only need be connected to the group of integer-encoding bits. So rather than a dense block of values in the Q-matrix, my novel approach has the Q-matrix shown in Figure 2. This system still pushes towards one-hot solutions with only one amino acid chosen per position, and still has the one-hot representation so coupling the solution to amino acid interaction energies or any other data would be simple. Increasing the percent of validly encoded solutions from 9% to 32%, This innovation **tripled the success rate** of D-Wave, finding 3x as many validly-encoded solutions, **a huge improvement that other quantum researchers could make use of**.

9 Conclusion

Running the same protein design QUBO on classical, quantum, and neuromorphic hardware, I found classical vastly outperformed the others. However, quantum and neuromorphic technology are in their infancy compared to classical computing. As the scale of quantum computers grow and their ability to avoid decoherence improve, they may become more powerful tools for solving problems such as protein design. Neuromorphic and quantum computers alike struggle to adhere to a soft constraint imposed with an energy barrier in a QUBO. I developed a novel encoding scheme which triples the success rate of a quantum computer's solutions adhering to a soft constraint. I applied this to the case of the map-coloring problem, which can be fit to numerous useful problems with real significance such as the protein design problem.

References

[BCHE19] Peter Blouw, Xuan Choo, Eric Hunsberger, and Chris Eliasmith. Benchmarking keyword spotting efficiency on neuromorphic hardware. In *Proceedings of the 7th Annual Neuro-*

inspired Computational Elements Workshop on - NICE '19, page 1–8. ACM Press, 2019.

[Dah13] E. D. Dahl. Programming with d-wave: Map coloring problem, Nov 2013.

[JHM16] Zeno Jonke, Stefan Habenschuss, and Wolfgang Maass. Solving constraint satisfaction problems with networks of spiking neurons. *Frontiers in Neuroscience*, 10, 2016.

[MMM+19] Vikram Khipple Mulligan, Hans Melo, Haley Irene Merritt, Stewart Slocum, Brian D. Weitzner, Andrew M. Watkins, P. Douglas Renfrew, Craig Pelissier, Paramjit S. Arora, and Richard Bonneau. Designing peptides on a quantum computer. Sep 2019.