Quantum Computing Algorithms for Protein Structure Prediction

Gino Prasad
Department of Computer Science
UC San Diego, giprasad@ucsd.edu

Introduction

Motivation Protein folding is an important and well-known problem in computational biology. To give a bit of background, proteins are biological molecules that can be represented as sequences, where each character in the sequence is one of 20 amino acids (AAs). Many human diseases such as Alzheimer's and Amyotrophic Lateral Sclerosis (ALS) are believed to be caused by the misfolding of proteins (i.e., altered protein sequences) [1]. Computational methods attempt to find the 3D structure of a protein given its string of amino acid sequences.

With that biology background out of the way, I will describe current methods for simulating protein folding with quantum computing. This survey will step through the quantum algorithm from Perdomo et al. for Hydrophobic-Polar protein folding [2]. This algorithm utilizes the quantum adiabatic evolution algorithm from Farhi et al. [3].

Problem Formulation While more complex and biologically rooted models for protein folding have been proposed [4], we will restrict all of our analysis to the simplest model: the Hydrophobic-Polar (HP) Lattice Model. Here, a protein sequence P is defined as a bitstring of size N.

$$P = \{0,1\}^N$$

Where

$P_i = 0$ when the ith AA in the protein sequence is Polar.
$P_i = 1$ when the ith AA in the sequence is Hydrophobic.

Our goal is to find the optimal *self-avoiding* walk in a grid of size N^3. This corresponds to a path in a 3D lattice where no vertex is traversed more than once.

The position of the ith AA in the protein sequence is the position after $(i-1)$ moves in the walk. For example, in Fig 2C from Perdomo et al. below, the x-position of the 3rd AA is 2 (10 in binary), and the y-position is 1 (01) [2].

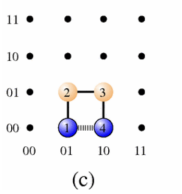

Figure 1: Example protein folding in 2D grid (Perdomo et al.) [2]. Here, the beads represent numbered AAs, with blue and beige AAs representing hydrophobic ($P_i = 1$) and polar ($P_i = 0$) beads, respectively.

A self-avoiding walk is considered optimal if it contains the maximum number of non-adjacent hydrophobic interactions (dashed lines in Fig. 2C). An interaction is where 2 non-neighboring

Preprint. Under review.

hydrophobic AAs (blue beads) are next to each other (L1 distance = 1) in the lattice. We will define the interaction score $S_{i,j}$ for each AA pair $i, j \in [N]$:

$$S_{i,j} = -1 \text{ if } ||position_i - position_j||_1 = 1, |i-j| \neq 1, P_i = P_j = 1$$

$$S_{i,j} = 0 \text{ otherwise}$$

Let $position_i, position_j$ represent the position of the ith and jth AA, respectively, in the lattice.

A self-avoiding walk is *optimal* if it minimizes $\sum_{i,j} S_{i,j}$.

Encoding

Our algorithm's goal is to output the optimal 3D structure of a protein. We can define the output encoding as a list of size N, where the ith element contains the 3D position of the ith amino acid.

Since each axis of the $3D$ grid has N tickmarks, encoding the position of an amino acid in a single dimension requires a binary encoding of size $\lceil log_2(N) \rceil$. For simplicity, we will assume that N is a power of 2 for the remaining analysis. This means the algorithm's output for N amino acids in 3 dimensions is a binary string of size $3Nlog_2(N)$.

Our algorithm will take as input the protein sequence $P \in \{0,1\}^N$, and output the 3D structure $q \in \{0,1\}^{3Nlog_2(N)}$:

Goal: Adiabatic Quantum Algorithm for the HP Problem (Perdomo et al.)

Let the equation below from Farhi et al. [3] represent the change of our state $|\psi(t)\rangle$ in time.

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = \hat{H}(t) |\psi(t)\rangle$$

We will set $\hat{H}(t) = (1 - \frac{t}{\tau})\hat{H}(0) + (\frac{t}{\tau})H_f$

By the Adiabatic Theorem [3]: If the initial configuration is set to the ground state (eigenvector with smallest eigenvalue and multiplicity 1) of $\hat{H}(0)$, then if we discretize the above Schrodinger equation in many many steps, the final configuration will be in the ground state of $\hat{H}(\tau)$, the optimal solution.

So when τ is large, then the distance between $|\psi(t)\rangle$ and the ground state of $\hat{H}(t)$ will be small for all $t \leq \tau$. One caveat is that the gap between the two lowest energy levels of the Hamiltonian $\hat{H}(t)$ must be greater than 0 for all $0 \leq t \leq \tau$.

Initial Hamiltonian matrix

$$\hat{H}(0) = \sum_{i=1}^{n} J_i$$

Here J_i is given below, where the i th 2x2 matrix in the Kronecker product equation is $J = \frac{1}{2}(I - \sigma_X)$

$$J_i = I \otimes ... \otimes I \otimes J \otimes I \otimes ... \otimes I$$

$\hat{H}(0)$ has a ground state $|\psi_g\rangle = H^{\otimes n} |0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle$ that is nondegenerate (unique).

Proof:

This means the eigenvector corresponding to the smallest eigenvalue of $\hat{H}(0)$ is the uniform superposition $|\psi_g\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle$ and the multiplicity of this eigenvalue is 0.

First, I will prove that $\hat{H}(0)$ has real and non-negative eigenvalues, and real eigenvectors. Since J is symmetric, J must have real eigenvectors. Let $\mathbf{x} \in \mathbb{R}^2$

$$\mathbf{x}^T J\mathbf{x} = \mathbf{x}^T \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \mathbf{x} = \frac{1}{2}(x_1^2 - 2x_1x_2 + x_2^2) = \frac{1}{2}(x_1 - x_2)^2 \geq 0$$

$$\mathbf{x}^T I\mathbf{x} = \mathbf{x}^T\mathbf{x} = ||\mathbf{x}||_2 \geq 0$$

Therefore J, I are positive semidefinite. Since the Kronecker product preserves positive semi-definiteness,

$J_i = I \otimes ... \otimes I \otimes J \otimes I \otimes ... \otimes I$ is also positive semidefinite.

Since the sum of positive semidefinite matrices is also positive semidefinite, $\hat{H}(0) = \sum_{i=1}^{n} J_i$ is positive semidefinite. Since $\hat{H}(0)$ is a positive semidefinite matrice, it must have **real eigenvectors and both non-negative and real eigenvalues**.

Since $JH|0\rangle = \frac{1}{2}\begin{bmatrix} \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \end{bmatrix} = \mathbf{0}$, then $J_i H^{\otimes n}|0\rangle = ... \otimes \mathbf{0} \otimes ... = \mathbf{0}$

Therefore $\hat{H}_0|\psi_g\rangle = \sum_{i=1}^{n} J_i|\psi_g\rangle = \mathbf{0}$, so $|\psi_g\rangle$ is an eigenvector of $\hat{H}$ with eigenvalue 0. Since $\hat{H}(0)$ has non-negative real eigenvalues, then $|\psi_g\rangle$, with eigenvalue 0 is the **smallest eigenvalue**.

Finally, I will show that $|\psi_g\rangle$ is non-degenerate: Let us assume there is a real $|y\rangle \in \mathbb{R}^{2^n}$ where $\hat{H}(0)|y\rangle = \mathbf{0}$ and therefore $\langle y|\hat{H}(0)|y\rangle = 0$

$$\langle y|\hat{H}(0)|y\rangle = \langle y|J_1|y\rangle + ... + \langle y|J_n|y\rangle \text{ and } \langle y|J_i|y\rangle \geq 0$$

$$\text{Therefore } \hat{H}(0)|y\rangle = \mathbf{0} \implies \forall i, \langle y|J_i|y\rangle = \mathbf{0}$$

If $|x\rangle \in \mathbb{R}^2$, then $J|x\rangle = \mathbf{0} \implies |x\rangle = \pm\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \pm|+\rangle$.

If $|y\rangle \in \mathbb{R}^{2^n}$, then $J_i|y\rangle = \mathbf{0} \implies |y\rangle = ... \otimes \pm|+\rangle \otimes ...$; the ith qubit is unentangled and is $\pm|+\rangle$

Since the above is true for all i, therefore $|y\rangle = \pm|+\rangle \otimes ... \otimes \pm|+\rangle = \pm\frac{1}{\sqrt{2^n}}\sum_{y\in\{0,1\}^n}|y\rangle$. Therefore $|\psi_g\rangle$ **is a non-degenerate ground state**. This is good news since $|\psi_g\rangle$ can be prepared as the initial configuration simply by applying a layer of all Hadamard gates.

Defining the Hamiltonian H_f

Ideally, we would want low energy eigenstates in the Hamiltonian H_f to translate to a self-avoiding walk with many hydrophobic interactions. We will use soft constraints to make sure a low-energy eigenstate satisfies:

1. The output state should translate to a valid walk (each AA position should have an L1 distance of 1 from the next AA in the sequence). We will define this soft constraint with the term H_w.

2. The translated walk should be self-avoiding (for all AAs the positions should be different). We will define this soft constraint with the term H_s.

3. We want to reward walks with more hydrophobic interactions (dashed lines in Fig. 2C) with an additional score of -1 for each interaction. This reward will be defined using the term H_r.

Our final Hamiltonian will be H_f, with each eigenstate's energy given by the sum $H_w + H_s + H_r$.

Defining H_s

Our goal is to penalize states that encode two AAs in the same position.

Using our encoding above, let $ind(i, k, r)$ represent the index corresponding to the rth bit of the ith AA in the kth dimension. Where $1 \leq r \leq \lceil log_2(N) \rceil$, $1 \leq i \leq N$, $1 \leq k \leq D$. Let q be the binary

representation of our state such that $q_{ind(i,k,r)}$ represents the rth bit of the kth dimension of the ith AA.

Let us define a function h which takes in two AA indices $1 \leq i, j \leq N$ and outputs a positive number when the ith and jth AA are in the same position in the output and 0 otherwise. We essentially want to perform an XNOR operation from each bit of i to the corresponding bit of j.

$$h(i,j) = \prod_{k=1}^{D} \prod_{r=1}^{log_2(N)} (1 - q_{ind(i,k,r)} - q_{ind(j,k,r)} + 2q_{ind(i,k,r)}q_{j,k,r})$$

The innermost expression of the product maps $00 \rightarrow 1, 11 \rightarrow 1, 01 \rightarrow 0$, and $10 \rightarrow 0$. This, therefore, matches the XNOR gate.

Now, we can search each pair of AAs and penalize with coefficient (N+1). This N+1 term will be clarified in the H_w section.

$$H_s = (N+1) \sum_{1}^{N-1} \sum_{j=i+1}^{N} h(i,j)$$

Defining H_w

We can define the squared L2 distance function between amino acids $1 \leq i, j \leq N$ using our binary encoding:

$$d_{i,j}^2 = \sum_{k=1}^{D} (\sum_{r=1}^{log_2(N)} 2^{r-1}(q_{ind(i,k,r)} - q_{ind(i,k,r)}))^2$$

Since we have integer grid points, $d^2(i,j)$ is only 1 when the L1 distance function is 1. Therefore, we can define H_w as:

$$H_w = N(1 - N + \sum_{i=1}^{N-1} d^2(i, i+1))$$

You may have noticed H_w gives a negative penalty (reward) to AAs with $d^2(i,j) = 0$ by 1. This is fixed by also considering H_s, which penalizes AAs in the same position by $N+1$ each pair. If any two AAs occupy the same position $H_s + H_w \geq N$

If no two AAs occupy the same position, but the L1 distance is greater than 1 for some neighboring AA pair, $H_w \geq N, H_s = 0$

If a configuration is valid, then $H_w = H_s = 0$.

Theorem: The max number of interactions for a length N sequence of all hydrophobic AAs is less than N in a valid self-avoiding walk.

Therefore $-N < H_r \leq 0$. So, the total energy is $H_s + H_w + H_r > 0$ **for any invalid output** (not a self-avoiding walk). **For any valid self-avoiding walk**, the total energy is $H_s + H_w + H_r \leq 0$.

Defining H_r

$$H_r = -\sum_{i=1}^{N} \sum_{j=i+1}^{N} G_{i,j}M_{i,j}$$

$$G_{i,j} = P_iP_j \text{ if } |i-j| = 1$$

$$G_{i,j} = 0 \text{ otherwise}$$

Here, $G_{i,j} = 1$ only when the ith and jth AA are neighboring and both are hydrophobic. $G_{i,j}$ only needs to be computed once as it only depends on P.

$M_{i,j} = 1$ when the ith and jth AA are neighboring in the encoded walk, and 0 otherwise. Perdomo et al. defined $M_{i,j}$ using half and full adder circuits to compute L1 distance along each of the 6-axis directions. To keep this survey brief, I will avoid writing its explicit form.

Converting H_f to a 2-Local Hamiltonian Currently, this formulation of H_f requires $6log_2N$ locality. This is because the number of variables needed in each independent operation is the number of bits ($3log_2N$ each) for two amino acids.

To reduce the locality of H_f and make the algorithm more experimentally viable, Perdomo et al. use Boolean reduction methods. By creating a larger set of binary variables, the ancilla bits can be substituted into the original equation. A penalty term in the Hamiltonian is then used to create a soft constraint that the ancilla bits take the value of an intermediate computation. The one downside to this approach is that the decrease in locality comes at the cost of a much higher resource requirement from the ancilla qubits. After reducing H_f to a 2-local Hamiltonian, the number of qubits needed for the Hamiltonian computation is $(N-2)(N^D-1)$. (The N-2 term is from a slightly different output encoding strategy to reduce redundant self-avoiding walks).

Discussion

Protein folding simulation remains an important and biologically relevant problem. As shown by Pierce et al., the task of identifying the optimal protein structure is NP-hard [5]. The current state-of-the-art classical methods, AlphaFold and Phyre2, require massive labeled datasets of protein structures and protein sequences [6, 7].

Quantum annealing and Quantum Approximate Optimization Algorithm (QAOA) approaches enable biologically accurate structural predictions without needing to undertake massive experimental data collection [4, 8]. Additionally, quantum algorithms for protein folding go a step further, offering insight into how multiple proteins could interact and bind together. This could pave the way for new kinds of bioinformatic analysis of disease and validation for experimental drug testing through simulation.

References

[1] Enrique Reynaud. Protein misfolding and degenerative diseases | learn science at scitable. Cg_cat: Protein Misfolding and Degenerative Diseases Cg_level: MED Cg_topic: Protein Misfolding and Degenerative Diseases.

[2] Alejandro Perdomo, Colin Truncik, Ivan Tubert-Brohman, Geordie Rose, and Alán Aspuru-Guzik. Construction of model hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models. 78(1):012320. Publisher: American Physical Society.

[3] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution.

[4] Anton Robert, Panagiotis Kl Barkoutsos, Stefan Woerner, and Ivano Tavernelli. Resource-efficient quantum algorithm for protein folding. 7(1):1–5. Publisher: Nature Publishing Group.

[5] Niles A. Pierce and Erik Winfree. Protein design is NP-hard. 15(10):779–782.

[6] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. 596(7873):583–589. Publisher: Nature Publishing Group.

[7] Lawrence A. Kelley, Stefans Mezulis, Christopher M. Yates, Mark N. Wass, and Michael J. E. Sternberg. The phyre2 web portal for protein modeling, prediction and analysis. 10(6):845–858. Publisher: Nature Publishing Group.

[8] Mark Fingerhuth, Tomáš Babej, and Christopher Ing. A quantum alternating operator ansatz with hard and soft constraints for lattice protein folding.