What Can You Prove About Proteins?

Suchetan Dontha, Matthew Coudron

July, 2024

Abstract

Biologically realistic mathematical models of protein molecules are complex and seemingly messy objects which have long eluded predictability via the techniques of proof-based mathematics, despite concerted efforts **(Matt C.:** citation**)** . In contrast empirical and heuristic approaches to problems such as protein structure prediction have made enormous progress in recent years **(Matt C.:** citations**)** . In this work we re-examine, from a new angle motivated by recent progress, the old question: "What can one *prove*, mathematically, about proteins?".

We wish to emphasize the following features and contributions of our approach:

- In contrast to prior theoretical work on proteins, we abandon all "toy models" of proteins in favor of working exclusively with the industry standard mathematical representation of proteins used ubiquitously by biologists, the so-called PDB (protein database) **(Matt C.:** check terminology**)** file format. This makes our proven results directly compatible with the model most familiar to biologists, at the expense of making each result much more difficult to prove. We believe this tradeoff gives our research the best chance of providing value in the long term.
- We exhibit a Linear Program which, given a folded protein described in PDB format, can efficiently and *provably* certify whether the given fold is an energy local minimizer **(Matt C.:** spelling?**)** under a given, industry-standard, potential energy function.
- To demonstrate the relevance of our Linear Program, we use it to show that some industry-standard Monte Carlo heuristics for protein folding, which are implicitly assumed to converge to an local-energy-minimizing fold, in fact converge to a fold which is not a local minimum at all. **(Matt C.:** double check that this is true.**)** To address this, we contribute a simple alternative gradient descent algorithm which, we demonstrate, finds true local-energy-minimizing folds as provably certified by our Linear Program. We provide examples of these phenomena for proteins consisting of a handful of amino acids, and show that our algorithms can scale efficiently to larger proteins, given more compute.
- We demonstrate that our Linear Program can be extended from certifying local minima to certifying the a given protein fold, in PDB format, lies in a "rough funnel" in the energy landscape. This is a concept considered by biologists but never previously formally proven about any protein fold to our knowledge. **(Matt C.:** check whether we actually include this experiment.**)**
- Since heuristic Monte Carlo energy minimization algorithms form a subroutine in the protocol used to calibrate tuneable parameters in industry-standard potential energy functions for protein folding, we propose a completely new way of calibrating the same tuneable parameters based on a provable algorithm instead. We describe a **(Matt C.:** efficient?**)** method to compute the optimal values of these tuneable paramaters directly from real protein imaging data based on a linear program with provable guarantees, thereby eliminating the need to use a heuristic Monte Carlo algorithm to define a centrally important object in the field. **(Matt C.:** Make sure to write a section on this idea, and perform several experiment if we have time.**)**

Awesome Webpage for visualizing PDB proteins in high detail, with amino acid sequences listed: https://www.rcsb.org/sequence/1SMD?assemblyId=1

1 Introduction

Should we try to get access to Rosetta source code? Can we run Rosetta ourselves?

Potential Research Directions:

- What is the largest subset of protein instances that are foldable in classical polynomial time (say, in the HP model)?
- How robust are natural folding algorithms when subject to small perturbations (say $\log(n)$ bit flips in protein instance)? What is a good metric for determining the difference between two folds that is consistent with optimal number of contacts (total variation distance, comparing protein structure at different scales)?
- Can we empirically estimate the hardness of approximation boundary of the FCC HP model (Find optimal contacts and use natural folding algorithms to find approximate contacts)?
- Can we use quantum (or quantum inspired) algorithms to solve protein folding?
- If proteins are not converging to the minimum energy configurations, then how do they converge so consistently to the same configuration?! Is this a special property of only certain types of sequences?
- Can we close the gap between approximation algorithms and hardness of approximation in, say, the FCC side chain HP model?

Reading List:

- Hart and Istrail paper of FCC side chain 0.86 approximation algorithm
- Approximate protein folding in the HP side chain model on extended cubic lattices, Volker Heun
- **(Matt C.:** What is the hardness of approximation ratio obtained by the following paper?:**)** Spatial Codes and the Hardness of String Folding Problems ASHWIN NAYAK, ALISTAIR SINCLAIR, and URI ZWICK
- R. Backofen. An upper bound for number of contacts in the HP-model on the facecentered-cubic lattice (FCC). In R. Giancarlo and D. Sankoff, editors, Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching, number 1848 in LNCS, pages 277–292, 2000.
- R. Backofen and S. Will. Optimally compact finite sphere packings hydrophobic cores in the FCC. Proc. of the 12th Annual Symposium on Combinatorial Pattern Matching (CPM2001), volume 2089 of Lecture Notes in Computer Science, Berlin, 2001.
- Papers on the inverse protein folding problem, such as: https://www.researchgate.net/publication/26288532_Inverse_Protein_Folding_in_3D_Hexagonal_Prism_Lattice_under_HPC_Model

Here are the papers that cited the Istrail-Lam Survey paper, and look interesting

- Here's the full list of papers citing Istrail-Lam: https://www.researchgate.net/publication/228853142_Combinatorial_Algorithms_for_Protein_Folding_in_Lattice_Models_A_Survey_of_Mathematical_Results
- Here are some of the most interesting looking titles:
- https://www.researchgate.net/publication/321738982_Combinatorics_of_Contacts_in_Protein_Contact_Maps
- https://www.researchgate.net/publication/359612492_On_the_Number_of_Saturated_and_Optimal_Extended_2-Regular_Simple_Stacks_in_the_Nussinov-Jacobson_Energy_Model
- https://www.researchgate.net/publication/286238228_Exact_methods_for_lattice_protein_models
- https://www.researchgate.net/publication/279825295_Computational_Methods_for_Lattice_Protein_Models
- https://www.researchgate.net/publication/262878215_Zigzag_Stacks_and_m-Regular_Linear_Stacks
- https://www.researchgate.net/publication/265603424_Are_there_Unfoldable_Proteins_in_Dimension_Three
- https://www.researchgate.net/publication/262786934_How_Good_Are_Simplified_Models_for_Protein_Structure_Prediction

- https://www.researchgate.net/publication/230763895_Producing_High-Accuracy_Lattice_Models_from_Protein_Atomic_Coordinates_Including_Side_Chains

Results on the PCLF (Protein Chain Lattice Fitting?) problem

- https://www.researchgate.net/publication/230763895_Producing_High-Accuracy_Lattice_Models_from_Protein_Atomic_Coordinates_Including_Side_Chains
- https://pubmed.ncbi.nlm.nih.gov/18324748/
- LocalMove: computing on-lattice fits for biopolymers.
- Producing high-accuracy lattice models from protein atomic coordinates including side chains.
- https://pubmed.ncbi.nlm.nih.gov/24876837/
- https://pubmed.ncbi.nlm.nih.gov/24876837/

Notes on existing software for protein folding:

- Rosetta Commons: https://www.rosettacommons.org/software
- Phyre and Phyre2: http://www.sbg.bio.ic.ac.uk/ phyre2/html/page.cgi?id=index
- https://github.com/OkkeVanEck/prospr

Refuting Random Satisfiabilty instances using Semidefinite Programming: https://www.tu-chemnitz.de/informatik/TI

The following paper contains the "rough funnel" idea, which is a really intutive notion of "good" protein sequences. It also helps at lot with understanding terminology and history: https://www.pnas.org/doi/10.1073/pnas.95.11.5921

Matt's to do list

- Read and understand the hardness of the PCLF problem: https://www.cs.ubc.ca/ jmanuch/journal/papers/JBCB200
- Read van Heun paper
- Look up whether there has been any more work on the Istrail-Lam Self-Assembly Conjecture.

Ideas for proving results about continuous space protein folding!!!

- http://www.math.uwaterloo.ca/ aghodsib/courses/f10stat946/notes/lec10-11.pdf
- https://arxiv.org/pdf/0903.1476.pdf
- https://link.springer.com/article/10.1007/s10208-009-9045-5
- http://proceedings.mlr.press/v35/hardt14b.pdf

2 Convex Optimization for continuous-space HP model protein folding

This section outlines an idea for a convex optimization problem whose solution gives an approximation to the (non-convex) continuous space protein folding problem in the HP model. While the convex program is unlikely to return the correct answer in every case, it may perform well for the "natural".

Link to shared code in colab: https://colab.research.google.com/drive/1zFle5KVBnrF-LDF-fg0jYMCcF1oMYlAB?usp=sharing

Matt's local copy of colab code: https://colab.research.google.com/drive/13GAt4ExHnLc-RBbWMVwvu_t9EGQrlrve

2.1 Proposed Semidefinite Program (SDP) for continuous-space, HP-model protein folding

Given a protein, in the HP model, which is a chain of n amino acids, uniquely specified by the subset $H \subset [n]$ of indices of the amino acids in the protein which are hydrophobic (all other amino acids are hydrophilic, according to the HP model), and given a positive real number (energy target) E, consider the following SDP:

$$\begin{array}{ll} \text{minimize} & \|K\|_* = \mathbf{tr}(K) \\ \text{subject to} & K \succeq 0 \\ & 3.8 \leq 2K_{ij} - K_{ii} - K_{jj} - (4 - 2/n)tr(K) \quad \forall i, j \text{ such that } i \neq j \\ & 3.8 = 2K_{i,i+1} - K_{ii} - K_{i+1,i+1} - (4 - 2/n)tr(K) \quad i = 1, \ldots, n-1 \\ & \sum_{i,j \in H, i \neq j}(2K_{ij} - K_{ii} - K_{jj} - (4 - 2/n)tr(K) - 3.8) \leq E \end{array} \tag{1}$$

The intuition here is that we hope that the optimal solution to this SDP we end up having low rank (rank 3 or less). For large enough value of E there should be a low rank solution, and once we find a value of E for which there is a low-rank solution, we can binary search on E to find the minimal E such that the optimal solution is rank 3.

2.2 Explanation of SDP intuition

$K = -(1/2)CDC$ where $C = I - 1/nJ$, and J is an $n \times n$ matrix of all ones. Therefore, $K = -(1/2)(D - 1/nDJ - 1/nJD + 1/n^2JDJ)$

$$K_{ij} = -(1/2) * (d_{ij} - 1/n(\sum_k d_{ik} + d_{kj}) + 1/n^2 \sum_{kl} d_{kl}) \tag{2}$$

Note that:

$$K_{ii} = -(1/2) * (-1/n(\sum_k d_{ik} + d_{ki}) + 1/n^2 \sum_{kl} d_{kl}) = 1/n \sum_k d_{ik} - 1/(2n^2) \sum_{kl} d_{kl} \tag{3}$$

3 Hardness Result for Continuous-Space Protein Folding:

Idea: To begin with, consider the continuous space HP-model in two dimensions. Consider a protein sequence which has long consecutive subsequences of hydrophobic cores, interspersed with length-1 sequences of hydrophilic cores.

Question: Can such a protein sequence fold in order to achieve the absolute maximum number of contacts between hydrophobic cores.

Note: When the total number of hydrophobic cores is the correct integer, a YES answer to the above question should require that the hydrophobic cores lie perfectly in a ball on the triangular lattice, which also implies that the single-spacing hydrophilic cores with lie perfectly around the boundary of that ball on the triangular lattice.

It is note clear whether there is an efficient algorithm to determine when this is possible for a given protein sequence. Could it be NP-hard?

Next step: Look at the NP-hardness proof for the Hamiltonian path problem. Also look again at Berger-Leighton and consider bin-packing.

Check Satisfiability of the following constraints:

$$\begin{array}{l} K \succeq 0 \\ rank(K) = 3 \\ 3.8 \leq 2K_{ij} - K_{ii} - K_{jj} - (4 - 2/n)tr(K) \quad \forall i, j \text{ such that } i \neq j \\ 3.8 = 2K_{i,i+1} - K_{ii} - K_{i+1,i+1} - (4 - 2/n)tr(K) \quad i = 1, \ldots, n-1 \\ \sum_{i,j \in H, i \neq j}(2K_{ij} - K_{ii} - K_{jj} - (4 - 2/n)tr(K) - 3.8) \leq E \end{array} \tag{4}$$

Take another look at Appendix C, and proof of Theorem 16 in http://proceedings.mlr.press/v35/hardt14b.pdf (hardness of low rank PSD matrix completion).

4 ϵ-Convex-Charts

Idea about ϵ-convex-charts of size 1.

Definition 1 (S_x). *For each point $x \in X_{convex}$ we define the set of all convex descriptions of the point X, as*

$$S_x \equiv \{\{x_i, p_i\}|_i : \sum_i p_i \cdot x_i = x \text{ and } p_i \geq 0 \text{ and } \sum_i p_i = 1\}.$$

Definition 2 (Convex Envelope). *Given a function $E : X_{convex} \to \mathbb{R}$, we define the convex envelope of E to be the function $C(x) \equiv \inf_{\{x_i,p_i\}|_i \in S_x} \sum_i p_i \cdot E(x_i)$.*

Lemma 3. *If an energy landscape function, $E : X_{convex} \to \mathbb{R}$, has an ϵ-convex-chart of size 1, the the convex envelope $C(x)$ (see Definition 2) is a 2*ϵ-convex-chart of size 1 for $E(x)$.*

Proof. Let $E(x)$ be the energy function, which, by assumption, has an ϵ-convex-chart of size 1. This means that there exists a convex function $F(x)$ such that $|F(x) - E(x)| \leq \epsilon$ everywhere.

We wish to prove that the convex envelope $C(x)$ (see Definition 2) is a 2ϵ-convex-chart for $E(x)$.

We know that $C(x) \leq E(x)$ at every point x because $\{x, 1\} \in S_x$, and therefore, $C(x) = \min_{\{x_i,p_i\}|_i \in S_x} \sum_i p_i \cdot E(x_i) \leq E(x)$.

We now prove that $C(x)$ is convex. That is, for every two points $y_1, y_2 \in X_{convex}$, and two weights $p_1, p_2 \geq 0$ with $p_1 + p_2 = 1$, we will show that $C(p_1y_1 + p_2y_2) \leq p_1C(y_1) + p_2C(y_2)$. To see this, note that, by definition of C, there exists convex descriptions $\{x_{1i}, q_{1i}\}|_i$ (resp. $\{x_{2i}, q_{2i}\}|_i$), such that $y_j = \sum_i q_{ji}x_{ji}$ for $j \in \{1, 2\}$, and $C(y_j) = \sum_i q_{ji}E(x_{ji})$ **(Matt C.:** ACtually I changed the definition to be an infimum, so there are only sequences of convex combinations approaching these limits. However, I believe the same proof strategy below will still work.**)** . Now, note that the point $p_1y_1 + p_2y_2$ has the following valid convex description: $\{x_{1i}, p_1 \cdot q_{1i}\}|_i \cup \{x_{2i}, p_2 \cdot q_{2i}\}|_i$. Therefore, by the definition of $C(x)$ we have that:

$$C(p_1y_1 + p_2y_2) \leq p_1 \cdot \sum_i q_{1i} \cdot E(x_{1i}) + p_2 \cdot \sum_i q_{2i} \cdot E(x_{2i}) = p_1C(y_1) + p_2C(y_2). \tag{5}$$

This shows that $C(x)$ is convex, and $C(x) \leq E(x)$ everywhere. It remains to prove that $|C(x) - E(x)| \leq 2\epsilon$ everywhere. To show this, first recall that we know there exists a convex function $F(x)$ that is at most ϵ away from $E(x)$ at any point. We will now show that $|C(x) - F(x)| \leq \epsilon$ everywhere.

First, note that, at every x, $C(x) \leq E(x) \leq F(x) + \epsilon$.

Now, for the other direction, note that, by the definition of C, for every X there exists a convex description $\{x_i, p_i\}|_i \in S_x$, such that $C(x) = \sum_i p_i \cdot E(x_i)$. So,

$$C(x) = \sum_i p_i \cdot E(x_i) \geq \sum_i p_i \cdot (F(x_i) - \epsilon) = (p_i \cdot F(x_i)) - \epsilon \geq F(x) - \epsilon. \tag{6}$$

It follows that $|C(x) - F(x)| \leq \epsilon$ everywhere, and thus, that $C(x)$ constitutes a 2*ϵ-convex-chart for E.

(Matt C.: The intuition was that $C(x)$ is the largest convex function that is less than $E(x)$ everywhere. But, actually, maybe we don't need to prove this property? Is this property true, even if we don't need it? **)**

□

Definition 4 (Discrete Convex Envelope). *Given a list $\vec{w}$ of tuples $(x_i, v_i) \in X_{convex} \times \mathbb{R}$, we define a discrete version of S_x as:*

$$S_x^{\vec{w}} \equiv \{\{x_i, p_i\}|_i : \sum_i p_i \cdot x_i = x \text{ and } p_i \geq 0 \text{ and } \sum_i p_i = 1, \text{ and each } x_i \text{ appears in } \vec{w}\}. \tag{7}$$

the Discrete Convex Envelope of $\vec{w}$ is defined as:

$$C_{disc}(x) \equiv \inf_{\{x_i,p_i\}|_i \in S_x^{\vec{w}}} \sum_i p_i \cdot E(x_i). \tag{8}$$

Lemma 5. *Given $\vec{w}$ as in Definition 4, and a point $x \in X_{convex}$ we can compute $C_{disc}(x)$ in time $\mathsf{poly}(|\vec{w}|)$ using a linear program.*

Proof. To prove this statement we will exhibit the linear program and appeal to standard results about linear programming:

By Definition 4 we have that

$$C_{disc}(x) = \min \sum_i p_i v_i \tag{9}$$

$$\text{Subject to the constraints:} \tag{10}$$

$$\sum_i p_i x_i = x, \ \sum_i p_i = 1, \text{and } p_i \geq 0 \forall i \tag{11}$$

This is a linear program of size $\mathsf{poly}(|\vec{w}|)$ and can therefore be solved **(Matt C.:** under standard assumptions**)** in $\mathsf{poly}(|\vec{w}|)$ time **(Matt C.:** cite convex programming**)** . □

(Matt C.: SPARSEST SOLUTION USING COMPRESSED SENSING: I think that finding the sparsest solution to the above linear program is something we would like to be able to do in order to efficiently an naturally represent each point x as a convex combination of points in the mesh. This seems like something that could be done in practice using the compressed sensing techniques we discussed!!**)**

$$x_i = r \cdot \sum_{j=1}^{i} sin(\phi_j) \cdot cos(_j) \tag{12}$$

$$y_i = r \cdot \sum_{j=1}^{i} sin(\phi_j) \cdot sin(\theta_j) \tag{13}$$

$$z_i = r \cdot \sum_{j=1}^{i} cos(\phi_j) \tag{14}$$

$$d^x_{i,j}(\epsilon,\delta) = d^x_{i,j}(0,0) + \sum_{k=i}^{j} r * (\epsilon * cos(\phi_k)cos(\theta_k) + \delta * sin(\theta_k)sin(\phi_k) + \epsilon * \delta * sin(\theta_k) * cos(\phi_k)) \tag{15}$$

$$= d^x_{i,j}(0,0) + \delta d^y(0,0) + ... \tag{16}$$

So, if we let

$$x_i(\vec{\delta}) = r \cdot \sum_{j=1}^{i} sin(\phi_j) \cdot cos(\theta_j + \delta_j)$$

$$y_i(\vec{\delta}) = r \cdot \sum_{j=1}^{i} sin(\phi_j) \cdot sin(\theta_j + \delta_j)$$

Then, according to the small angle approximation (valid estimate for sufficiently small $\vec{\delta}$? We have that

$$x_i(\vec{\delta}) \approx x_i(\vec{0}) + \sum_{j=1}^{i} \delta_j \cdot (y_j - y_{j-1}) \tag{17}$$

$$y_i(\vec{\delta}) \approx y_i(\vec{0}) + \sum_{j=1}^{i} \delta_j \cdot (x_j - x_{j-1}) \tag{18}$$

$$z_i(\vec{\delta}) = z_i(\vec{0}) \tag{19}$$

$$x_i(\vec{\delta}) - x_k(\vec{\delta}) \approx x_i - x_k + \sum_{j=i}^{k} \delta_j \cdot (y_j - y_{j-1}) \tag{20}$$

$$y_i(\vec{\delta}) - y_k(\vec{\delta}) \approx y_i - y_k + \sum_{j=i}^{k} \delta_j \cdot (x_j - x_{j-1}) \tag{21}$$

$$z_i(\vec{\delta}) - z_k(\vec{\delta}) = z_i - z_k \tag{22}$$

5 Links to Code

Matt's Copy of Protein Energy Landscape Code:
https://colab.research.google.com/drive/1KByBHADWC6jDyOOeds0bSY8uMFQWOVDb
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5717763/pdf/nihms918588.pdf
https://www.rosettacommons.org/docs/latest/rosetta_basics/scoring/score-types

6 Algorithm for comparing short range potential functions

Definition 6 (Two-Body Potential Function)**.** *A Two-Body Potential Function (just called a Potential Function, when the two-body nature is clear from context) is a map of the form $E(type_1, type_2, distance) \to \mathbb{R}$, which, for each pair of particles, outputs a real number describing their energy contribution, which depends only on the "types" of the two particles and the distance between them.*

Definition 7 (Short-Range Two-Body Potential Function)**.** *A Two-Body Potential Function, E, is Short-Range with truncation radius Ω if $E(t_1, t_2, d) = 0$ for all particle types t_1, t_2 whenever $d \geq \Omega$.*

Definition 8 (Potential Energy of a collection of particles)**.** *Given a collection S of n particles in which the i^{th} has type t_i and position $\vec{x}_i$, the potential energy $E(S)$ of this collection, under potential function E, is $\sum_{1 \leq i < j \leq n} E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|)$.*

Lemma 9. *Given any collection S of n particles in three dimensional space, and any short-range two-body potential function E with truncation radius Ω (and in which every term is negative), we have that the quantity:*

$$E(S) \leq \int E(S \cap B(x, k \cdot \Omega))\mu(x)dx \leq (1 + 1/4k)(1 - 1/2k)^2 E(S)$$

where $E(S \cap B(x, k \cdot \Omega))$ represents the potential energy, under E, of the subset of particles in S which are within distance $k \cdot \Omega$ of point x, and μ is the uniform measure over the smallest sphere $\mathcal{S}$ which contains all points that are within distance $k \cdot \Omega$ of any point in S. We normalize μ so that $\int \mathbb{1}[|x - c_{\mathcal{S}}| \leq k \cdot \Omega]\mu(x)dx = 1$.

Proof. Note that:

$$E(S) = \sum_{1\le i<j\le n} E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) = \sum_{1\le i<j\le n} E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \cdot \int \mathbb{1}\big[\|\vec{x} - \vec{x}_i\| \le k\cdot\Omega\big]\mu(x)dx \tag{23}$$

$$= \int \sum_{1\le i<j\le n} E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \cdot \mathbb{1}\big[\|\vec{x} - \vec{x}_i\| \le k\cdot\Omega\big]\mu(x)dx \tag{24}$$

$$\le \int \sum_{1\le i<j\le n} E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \cdot \mathbb{1}\big[\|\vec{x} - \vec{x}_i\|, \|\vec{x} - \vec{x}_j\| \le k\cdot\Omega\big]\mu(x)dx \tag{25}$$

$$= \int E(S \cap B(x, k\cdot\Omega))\mu(x)dx, \tag{26}$$

where the second equality is true because $\mu(x)$ is normalized so that $\int \mathbb{1}\big[\|\vec{x} - \vec{x}_i\| \le k\cdot\Omega\big]\mu(x)dx = 1$ for any value of $\vec{x}_i$, the inequality is true because the $E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|)$ terms are negative by assumption, and the final equality is true by the definition of $E(S \cap B(x, k\cdot\Omega))$.

To prove the second inequality in the Lemma, note that, by Lemma 11 **(Matt C.:** this could use some additional explanation**)** , for each $i \neq j$ with $\|x_i - x_j\| \le \Omega$, we have that:

$$(1 + 1/4k)(1 - 1/2k)^2 \int \mathbb{1}\big[\|\vec{x} - \vec{x}_i\| \le k\cdot\Omega\big]\mu(x)dx \le \int \mathbb{1}\big[\|\vec{x} - \vec{x}_i\|, \|\vec{x} - \vec{x}_j\| \le k\cdot\Omega\big]\mu(x)dx \tag{27}$$

Therefore, for each $i \neq j$, $\|x_i - x_j\| \le \Omega$:

$$\int E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \cdot \mathbb{1}\big[\|\vec{x} - \vec{x}_i\| \le k\cdot\Omega\big]\mu(x)dx = E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \cdot \int \mathbb{1}\big[\|\vec{x} - \vec{x}_i\| \le k\cdot\Omega\big]\mu(x)dx \tag{28}$$

$$\le (1 + 1/4k)(1 - 1/2k)^2 E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \int \sum_{1\le i<j\le n} \cdot\mathbb{1}\big[\|\vec{x} - \vec{x}_i\|, \|\vec{x} - \vec{x}_j\| \le k\cdot\Omega\big]\mu(x)dx \tag{29}$$

$$= (1 + 1/4k)(1 - 1/2k)^2 \int \sum_{1\le i<j\le n} E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \cdot \mathbb{1}\big[\|\vec{x} - \vec{x}_i\|, \|\vec{x} - \vec{x}_j\| \le k\cdot\Omega\big]\mu(x)dx \tag{30}$$

So:

$$E(S) = \sum_{1\le i<j\le n} E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) = \sum_{1\le i<j\le n} \mathbb{1}\big[\|x_i - x_j\| \le \Omega\big] \cdot E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \cdot \int \mathbb{1}\big[\|\vec{x} - \vec{x}_i\| \le k\cdot\Omega\big]\mu(x)dx \tag{31}$$

$$\ge \frac{1}{(1 + 1/4k)(1 - 1/2k)^2} \sum_{1\le i<j\le n} \mathbb{1}\big[\|x_i - x_j\| \le \Omega\big] \cdot E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \cdot \int \mathbb{1}\big[\|\vec{x} - \vec{x}_i\|, \|\vec{x} - \vec{x}_j\| \le k\cdot\Omega\big]\mu(x)dx \tag{32}$$

$$= \frac{1}{(1 + 1/4k)(1 - 1/2k)^2} \int E(S \cap B(x, k\cdot\Omega))\mu(x)dx, \tag{33}$$

where the second equality follows because E is short range with truncation radius Ω, and the inequality follows because $E(t_i, t_j, \|\vec{x}_i - \vec{x}_j\|) \le 0$ for all i, j by assumption.

□

Lemma 10. *[Taken from Wolfram Alpha] The volume of the intersection of two three dimensional spheres of radius R whose centers are at distance d from each other is:*

$$V = \frac{\pi}{12}(4R + d)(2R - d)^2$$

Lemma 11. *Given two three dimensional spheres of radius R, with the property that the distance between their centers is a R/k, the ratio between the volume of their intersection and the volume of one of the spheres is:*

$$(1 + 1/4k)(1 - 1/2k)^2$$

Proof. Using Lemma 10, we have that $\frac{V_{int}}{V_{sphere}} = \frac{R^3(4+1/k)(2-1/k)^2}{8R^3} = (1+1/4k)(1-1/2k)^2$. □

Algorithm 1: Finite-time exhaustive search algorithm for bounding the ratio of potential functions E_1 and E_2 on unbounded molecular configurations.

Input : Energy potential functions E_1 and E_2, satisfying the conditions and promise in the statement of Lemma 12, as well as the specific constant $\gamma > 0$ in the promises.

Output: Which of the two promise cases in Lemma 12 is satisfied.

1 Fix a three dimensional sphere S, of radius $k \cdot \Omega$, centered at the origin.

2 Let $\mathcal{F}_S$ denote the set of molecular configurations within S which are restrictions, to S, of some configuration from $\mathcal{F}$.

3 Check, by exhaustive search, whether there is any molecular configuration $f \in \mathcal{F}_S$ for which either

1. $E_1(f)/E_2(F) \geq \gamma$
   OR
2. $E_2(f)/E_1(f) \geq \gamma$

if *If neither inequality is ever obtained for any $f \in \mathcal{F}_S$* **then**
  return "Promise 1 is satisfied." **(Matt C.:** indicating that the first promise in Lemma 12 is satisfied**)**
if *There exists $f \in \mathcal{F}_S$ such that $E_1(f)/E_2(F) \geq \gamma$.* **then**
  return $E_1(f) \geq \gamma E_2(f)$
if *There exists $f \in \mathcal{F}_S$ such that $E_2(f)/E_1(f) \geq \gamma$.* **then**
  return $E_2(f) \geq \gamma E_1(f)$

Lemma 12. *Given two potential energy functions E_1 and E_2 which both have truncation radius Ω, and which are valid on a family $\mathcal{F}$ of molecular configurations, Algorithm 1 solves the following promise problem in running time* $\exp(k \cdot \Omega)$*:*

Given $\gamma > 0$, and promised that either:

- *E_1 and E_2 are $(1+1/4k)(1-1/2k)^2/\gamma$ multiplicative approximations of each other on all configurations of particles in three dimensional space.*

  OR

- *There exists some configuration f of particles for which $E_1(f) \geq \gamma \cdot E_2(f)$ (or vice versa).*

determine which is the case.

Proof. As defined in Algorithm 1, let S denote a three dimensional sphere (centered at the origin, say) with radius $k \cdot \Omega$. We denote, by $\mathcal{F}$, the (potentially infinite) set of configurations of amino acids for which we wish to compare energy potential functions E_1 and E_2. Let $\mathcal{F}_S$ denote the set of molecular configurations within S which are restrictions, to S, of some configuration from $\mathcal{F}$.

Since Algorithm 1 performs an exhaustive search over $\mathcal{F}_S$ in Step 3 **(Matt C.:** need to elaborate on why exhaustive search is possible for $\mathcal{F}_S$ even though it is a continuous set**)** , it is immediate that it will return the right answer in the case that the second promise of Lemma 12 is satisfied for some configuration in $\mathcal{F}_S$.

To complete the proof of correctness of Algorithm 1 it remains to show that, if the second promise of Lemma 12 is not satisfied for any configuration in $\mathcal{F}_S$, then the first promise of Lemma 12 is satisfied on all of $\mathcal{F}$. To see this, note that, for any $f \in \mathcal{F}$, we have that:

$$(1+1/4k)(1-1/2k)^2 E_1(f) \geq \int E_1(f \cap B(x, k \cdot \Omega))\mu(x)dx \geq \int \gamma E_2(f \cap B(x, k \cdot \Omega))\mu(x)dx \geq \gamma E_2(f), \tag{34}$$

(Matt C.: actually shouldn't there be a $1/\lambda$ on the RHS of the above equation? This would also invert the position of the lambda in parts of the Lemma which seems to make more sense.**)** where the first inequality follows by Lemma 9, the second inequality follows because $f \cap B(x, k \cdot \Omega) \in \mathcal{F}_S$ for any $f \in \mathcal{F}$ and our current analysis is for the case that

Algorithm 1 has checked that $E_1(h) \geq \gamma E_2(h)$ for all $h \in \mathcal{F}_S$. **(Matt C.:** although the ball may not be centered at the origin, so need to fix that**)** The third inequality again follows by Lemma 9. By the symmetric argument we also have that $(1 + 1/4k)(1 - 1/2k)^2 E_2(f) \geq \gamma E_1(f)$.

6.1 Stability approximation for low-stretch protein configurations

We can also use ideas from the proof of Lemma 12 to compute an approximation to the energy landscape of large low-stretch protein fold configurations in time that scales only polynomially in the size of the large protein, but exponentially in the stretch, truncation radius, and quality parameter for the approximation.

Lemma 13. *placeholder*

Claim 14 (Informal)**.** *For a helix h with n amino acids, which has stretch σ, and an energy potential function E with truncation radius Ω, there exists the following local approximation to the potential energy $E[h]$ of the helix.*

Choose a window-length parameter k, and define sliding window W_i to be the window containing exactly the atoms in the helix starting from the $(i-k)^{th}$ amino acid, and proceeding to the i^{th} amino acid, including all atoms along the helix in between. Let $E_{W_i}[h]$ be the real valued quantity obtained by summing exactly the energy terms from $E[h]$ for which both atoms are contained in the window W_i. Then:

$$E[h] \geq \frac{1}{2k} \sum_{i=1}^{n+k} E_{W_i}[h]$$

(Matt C.: *this mathematical statement still need to be completed***)**

Proof. The idea is that, in the sum $\sum_{i=1}^{n+k} E_{W_i}[h]$ every term from the sum $E[h]$ appears at most $2k$ times, and at least $2k - \lceil \frac{\Omega\sigma}{\ell} \rceil$ **(Matt C.:** perhaps elaborate on why this is true.**)** (where ℓ denotes the length in space of a covalent bond in the protein). Therefore, the normalized expression $\frac{1}{2k} \sum_{i=1}^{n+k} E_{W_i}[h]$ contains exactly the same terms in the sum $E[h]$ where each term is re-weighted by a positive quantity in the interval $[\frac{2k - \frac{\Omega\sigma}{\ell}}{2k}, 1]$ which is the same as interval $[1 - \frac{\Omega\sigma}{2k\ell}, 1]$. □

(Matt C.: Need to change the notation in Lemma 12 and in Algorithm 1 so that f refers to a collection of particles, and B refers to a sphere of radius omega!!!!**)** □

- To Do:
- Determine whether anyone has made available code for a true exhaustive search over all valid conformations of very small proteins (a few amino acids). Maybe Matt should email someone and ask (perhaps the UW people).
- Also, ask, in the email, if Rosetta has a specification for how regular chemical bonds between amino acids constraint the relative positions of the amino acids (is it just distance and angle between connected atoms?).
- Think more about the proof complexity for statements of this form: Protein conformation S lies at the bottom of a potential well (w.r.t. Ref2015 potential function) of height H and "uneveneness" ϵ.
- Open Question: Can we *design* a protein sequence that has a short proof of the stability property above?
- Link to colab for alphafold: https://colab.research.google.com/github/deepmind/alphafold/blob/main/notebooks/
- : Very useful colab for dowloading data from the protein data bank and displaying it: `https://colab.research.google.com/github/pb3lab/ibm3202/blob/master/tutorials/lab02_molviz.ipynb#scrollTo=S2GfVk67gisv`

Pseudocode for Exhaustive Search Algorithm:

1. Produce an approximate ϵ net over 3 dimensional orthogonal matrices. One quick way to code this would be to take $O(1/\epsilon^3)$ (?) samples from a uniformly random orthogonal matrix generator, which should accomplish the task with high probability.

2. Reverse engineer the bond constraints between pairs of amino acids by using statistics of the appearances of those pairs of amino acids in the PDB, which can be accessed using wget as in https://colab.research.google.com/github/pb3lab/ibm3202/blob/master/tutorials/lab02_molviz.ipynb#scrollTo=S2GfVk67gisv.
3. For a length n protein chain, we try all possible combinations of $n-1$ independent 3 dimensional orthogonal matrices from our ϵ net. These unitaries specify the relative position between neighboring amino acids in the chain.

Idea for New Potential Energy Function in the presence of Catalysts:

$$H_{catalyst}(P) \propto E_{C\sim\mu_{catalyst(P)}}[|H_{Rosetta}(C)|^{\lambda}H_{Rosetta}(C)] \tag{35}$$

$$H_{catalyst}(P) \propto E_{C\sim\mu_{catalyst(P)}}[\exp(T\cdot|H_{Rosetta}(C)|)H_{Rosetta}(C)] \tag{36}$$

Definition 15 ($\mu_{catalyst(P)}$)**.** *$\mu_{catalyst(P)}$ is defined to the be distribution over configurations of protein fold P surrounded by catalyst molecules in which the protein ....*

To Do List: Matt should ask Biologists about bond lengths between Amino acids. Matt should ask about using UMD/NIST clusters for our project. How to gain access and what programming language is most amenable. Should we use C or C++ or Python, or something else?

Links for requesting py Rosetta and the Rosetta license:
https://els2.comotion.uw.edu/product/pyrosetta
Rosetta Commons License username and password:
user: levinthal
pass: paradox
To Do List (6-30-2023)

- Validity checker subroutine
- Sample from the L1 ball in angle space.
- Reverse engineer whether temperature typically decreases linearly or geometrically in pyRosetta montecarlo.
- Write proof outline for computer aided proof of bounded infinite-time escape probability.
- Figure out how fine of an epsilon net is needed for exhaustively checking the steepness of energy wells (this may require computing the Liphschitz constant for the energy of proteins as a function of their angles....or RMS distance from F_0). It seems that this Liphschitz constant will be much smaller in RMS distance than in angle L1 distance (perhaps by a factor of the number of amino acids, or more)?
- The following question may be important for constructing an epsilon net: Is there an algorithm that can take a fold and check (in the worst case) whether it is close to a valid fold (either in angle distance or RMS distance)?
- A divide-and-conquer analysis might allow us to get away with a smaller epsilon net than might otherwise be required.

It seem worthwhile to study the special case of Protein Side Chain Packing. Dead End Elimination Theorem for Protein Side Chain Packing: https://www.nature.com/articles/356539a0
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2142755/pdf/7756990.pdf
https://www.sciencedirect.com/science/article/pii/S0006349594809233
References on Biocomputing:

- Dan Boneh, Richard Lipton, et al, Theory paper on DNA computing. https://pdf.sciencedirectassets.com/271602/1-s2.0-S0166218X00X00327/1-s2.0-S0166218X96000583/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjED0aCXVzLWVhc3QtMSJHMEUCIQDDaAdmFgz%2FDG%2BG4E20LI9ndE1yYSuVk8OxoJ8a5u%

2FeKQIgDbd4GOzgb6z4aQ9%2FizOPQyAbK%2BrIc00ovSGmrECxrPcqvAUIxv%2F%2F%2F%2F%2F%2F%2F%
2F%2FARAFGgwwNTkwMDM1NDY4NjUiDAbEDH%2FZqjs%2FgZ%2FxAiqQBbNMqgduPBvB6KZLspfnIYxisD0bqp%
2Fvo0ekMhbgzgyb0b%2FoZcNpMLp%2B7p6fF5QkgBRdRwtPTKm1kQ1x6tY29RM1KcsROSFEirgCc3K%2BCO%2FRNgduzYLo1QFF
2FxaERKnCZbq652OAFGBVgHvILYpHzE%2FGtGHte1ee0W%2Bl0%2BOHV4pHS3bf4lTjep1dIbbJeEbmMqa6s8%
2FYVBZDmHBNAJCnMunnp3ceqGiyfk24nVKC9zBuMAJE2igCN1%2FsNMidto3GEZgMGfCp64Km7WpIKhiKcAy4Sst3q568e0yxLo
2Fkhut%2Bpyb3%2B0Of2KcXKeTZa671U%2FFwnynEz6hpczsAfBdR0%2BOZIUW3TPiwmkKrwGBXo5IpdD0bcoHFb3efYRHK1R%
2BwCj7zB9Huwzx42tqkFwl0MsoHixfpqp8XDceQERSkpeWHotgZuSpYbEKIgS9kzhH3f9KQ2p64No5eC8tsCHHRTgcBkQix7Sqi
2FMxNRWdcsdwellawB7mr5BwhUrsr8Ur7o2NZPAU8XUur%2BR%2BINYqGl11ZTvKmwa8juvAts04TdNHthQWfpFIEynl73pjKV6
2FPV4XGM3BN92VMxF545roZA%2BJnRybU8qWUd4IkBsnw7jmrkQnnBdW217jA1Ekah6hpQ33DiyZWyl27Hm5C1OwsvscWUrEjE%
2Bve7ywXPRb5%2FOcAwX%2BGhwa8ntYSVrhPDNVfadVG8vroROgYCFdQe75WZMLnS66UGOrEB5pEz%2B2BDLva9%
2BB4cMcl9ywvpdDQgJ8qJJP%2FT2FcIuZAbZvuXpFg3Dfgu8D6T3AnbwXSU3zYkTx4O6AyvO41wdVc4tXMgSZUoz5sniYuXOQs9
2FX4KdpCkuNEGAfHgJfHzTm%2B3ps0X%2FHUPLg9p8fP9e2ifkfBwrmp6JsHUI24zUEuovn6wBSznYB&X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20230721T211509Z&X-Amz-SignedHeaders=host&X-Amz-
Expires=300&X-Amz-Credential=ASIAQ3PHCVTY2LCT3S73%2F20230721%2Fus-east-1%2Fs3%2Faws4_
request&X-Amz-Signature=a7b6fb522c6728212476a0b2efa8ff036e88cfb4729814fc87aad5d7f13b9edd&
hash=1825d143f93f5f4dc6e5ce749e69e8be70e43302e1924476b543f622d9a45c90&host=68042c943591013ac2b2430a
pii=S0166218X96000583&tid=spdf-62b6ecea-d5aa-4462-99c8-b04b61902bc0&sid=75a693563f3223458508482-
59250c6dcbbdgxrqa&type=client&tsoh=d3d3LnNjaWVuY2VkaXJlY3QuY29t&ua=0f1559070353565254515f&
rr=7ea67527094607d7&cc=us

- This paper studies proteins that have multiple stable 3D folds instead of just one unique fold: https://journals.plos.org
- AlphaFold is Low depth according to this article: https://www.frontiersin.org/articles/10.3389/frai.2022.875587/full
- Look up *Robust* Low-Rank Matrix Completion results, and see if they might already address the weak membership problem! In particular, look at Theorem 1 in this paper https://arxiv.org/pdf/1412.8132.pdf, which claims (in the introduction) to be the first paper to consider robust low-rank matrix completion "with noise". (Also, look at this Candes paper https://arxiv.org/pdf/0903.3131.pdf which claims low rank noisy matrix completion, but "with an error that is proportional to the noise level".)

Definition 16 (ϵ-pairwise-invalid Fold)**.** *We call a protein fold configuration ϵ-pairwise-invalid if every pair of covalently bonded atoms in the protein are exactly the correct distance from each other, and every other pair of atoms in the protein are at least distance $\alpha - \epsilon$ apart, where α is the canonical minimum distance for non-bonded atoms.*

Claim 17 (Weak Membership Conjecture 1)**.** *There exists a function $f(n, \epsilon)$, with $\lim_{\epsilon \to 0} f(n, \epsilon) = 0$ such that every ϵ-pairwise-invalid Fold of any protein with n amino acids is within $f(n, \epsilon)$ of a valid fold in RMS distance.*

Definition 18 (δ-cover)**.** **(Matt C.:** *A δ-cover is like a δ-net except that a δ-net requires that its elements must be members of the set that it is approximating, whereas a δ-cover can have elements from a larger space.***)**

Definition 19 (Prot(b) and Ang(F))**.** *Given a vector b of torsion angles corresponding to a particular n-amino acid sequence, let Prot(b) be the fold on the n amino acid chain which is determined by the torsion angles given in b. Inversely, given a fold F of a particular n-amino acid sequence, Ang(F) be the vector of torsion angles corresponding the fold F.*

Algorithm 2: Constructing a δ-cover over ϵ-pairwise-invalid folds on a specific n amino acid chain.

Input : Constants $\epsilon, \kappa, \beta > 0$ and an ordered list of n amino acids.
Output: A δ-cover of ϵ-pairwise-invalid folds of the given n amino acid chain.

1 Suppose that the given n amino acid chain has m torsion angles in total.
2 Create a β-net, $\mathcal{B}$, over the space $[0, 2\pi]^m$.
`/* In order to understand the return line below, recall, for each element` $b \in \mathcal{B}$`, the definition of Prot(b) in Definition 19. */`
3 **return** The covering set $\mathcal{D} \equiv \{Prot(b) : b \in \mathcal{B}$, and $Prot(b)$ is κ-pairwise-invalid$\}$

Claim 20 (ϵn^2-Cover for ϵ-pairwise-invalid folds.)**.** *There exists a constant c such that, if we run Algorithm 2 with $\beta = \frac{c\epsilon}{n^2}$ and $\kappa = 2\epsilon$, it outputs a set $\mathcal{D}$ which is a $O(\epsilon \cdot n^2)$-cover, in RMS distance, of the set of all ϵ-pairwise-invalid folds. Furthermore, note that every element of the set $\mathcal{D}$ is a 2ϵ-pairwise-invalid fold by construction.*

Proof. Let F be any ϵ-pairwise-invalid fold, recall, from Definition 19, that Ang(F) is the set of torsion angles defining F. By construction, the β-net, $\mathcal{B}$, constructed in Algorithm 2 contains an element b such that $|b - Ang(F)|_1 \leq \beta$. We will now show that the fold Prot(b) is within $O(\epsilon)$ RMS distance of F. We will also show that Prot(b) is a 2ϵ-pairwise-invalid fold. These two statements, together, prove the desired result.

Showing that Prot(b) is within $O(\epsilon)$ RMS distance of F: Fix any two such atoms in F, say, the i^{th} and j^{th} atoms, and let $d_{i,j}(F)$ denote the distance between these atoms in F. Now imagine that we change the angles in F, which initially have values Ang(F) by definition, one-by-one to be the angles in b instead. By the time we are done with these changes the fold F will have changed to the fold Prot(b), and the distance between atoms i and j will have changed to some new value $d_{i,j}(Prot(b))$. Since $\sin(x) \leq x$ and the entire n-amino-acid protein chain has length at most $(2n + 40)\ell$ (where ℓ is a stand in variable for the maximum length of a covalent bond in proteins), each time we change the k^{th} angle by an amount θ_k the distance $d_{i,j}$ between atoms i and j can change by at most $2 \cdot (2n + 40)\ell \cdot |\theta_k|$. So, accounting for every one of the angle changes iteratively, we have that

$$|d_{i,j}(Prot(b)) - d_{i,j}(F)| \leq \sum_k 2 \cdot (2n + 40)\ell \cdot |\theta_k| \leq 2 \cdot (2n + 40)\ell \cdot \beta = 2 \cdot (2n + 40)\ell \cdot \frac{c\epsilon}{n^2} = O(\frac{\ell\epsilon}{n}) \quad (37)$$

So, if we define the matrix $M(F)$ whose i, j entry is $d_{i,j}(F)$ (resp. $M(Prot(b))$ and $d_{i,j}(Prot(b))$, then the RMS distance between F and Prot(b) is:

$$RMS(M(F), M(Prot(b))) = \sqrt{\sum_{i,j}(d_{i,j}(F) - d_{i,j}(Prot(b))^2} \leq \sqrt{O(n^2) \cdot O(\frac{\ell^2\epsilon^2}{n^2})} = O(\ell\epsilon) \quad (38)$$

Showing that Prot(b) is a 2ϵ-pairwise-invalid fold: F is an ϵ-pairwise-invalid fold by assumption, so $d_{i,j}(F) \geq \alpha - \epsilon$ for all atom pairs i, j which are not connected by a covalent bond. So, again using Equation 37, we see that $d_{i,j}(Prot(b)) \geq \alpha - \epsilon - O(\frac{\ell\epsilon}{n}) \geq \alpha - 2\epsilon$. This proves the desired result.

□

7 Locally Bounding the Liphschitz Constant of the potential energy function for a specific protein

7.1 Elementary Bounds

Some bounds on the Lipschitz constant can be derived from direct computation, for example, by assuming the maximum contribution from each pairwise energy term within a small interval of the pairwise distance for those atoms in the grid fold. **(Matt C.:** elaborate more later**)**

7.2 An SDP bound on the local Lipschitz constant in the neighborhood of a given protein fold.

Suppose that we are given a specific fold configuration for a specific protein which we will call the "reference fold", and we want to locally bound the Lipschitz constant of the potential energy function (Ref15) applied to that protein for configurations which lie within a small neighborhood of the reference fold. Here we give an SDP for computing such bounds.

Suppose that