Novel Approaches to the Protein Design Problem

Adi Mukund Jennifer Zou

April 20, 2015

1 Introduction

The protein design problem can be described as follows. Given a set of backbone coordinates $\mathbf{c} = (\vec{c_1}, \vec{c_2}, \ldots, \vec{c_n})$ and rotamer library R of length r, identify the optimal rotamer assignment sequence $\vec{r} = (r_1, r_2, \ldots, r_n)$, $r_i \in R$, $1 \leq i \leq n$ according to an energy function $E(\vec{r})$. This assignment sequence is known as the global minimum energy conformation (GMEC). This problem has been proven to be NP-hard [1], and algorithms such as DEE [2] have been created to allow for combinatorial pruning of the residue search space and make the design problem computationally tractable. However, such algorithms cannot guarantee a time complexity less than the worst-case $O(nr^n)$.

Recent graph based algorithms such as BWM* [3] use sparse residue interaction graphs in order to more efficiently compute functions over the residue space and identify optimal assignments more rapidly. Such graph-based algorithms have been able to achieve combinatorial speedups while maintaining provable accuracy and returning ensemble of minimum energy conformations.

Probabilistic models of protein design assign a probability distribution for rotamers in each position of the protein sequence. In algorithms such as belief propagation, the beliefs or the approximate marginal probabilities of each rotamer are computed iteratively. These approximations are computationally less intense, but they are not always provably accurate [4]. Furthermore, such algorithms are only guaranteed to converge on tree graphs, which are not commonly observed in natural proteins [5].

2 Graph Cuts and the GMEC

2.1 The Graph Labeling Problem

The goal of this project was to apply graph-based algorithms to the protein design problem. We began by attempting to characterize the protein design problem as a graph cut problem and identifying relevant algorithms that might allow efficient approximations of the GMEC.

The protein design problem is most accurately represented not as solely a graph cut problem, but as a graph labeling problem, where each rotamer is a label. The Graph Labeling (GL) problem can be stated as follows: classify a set $\mathcal{V}$ of n objects by assigning to each object a label from a given set $\mathcal{L}$ of labels, given a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. For each $p \in \mathcal{V}$ there is a label cost $\mathbf{c}_p(a) \geq 0$ for assigning the label $a = f_p$ to p, and for every edge pq there is a pairwise cost $\mathbf{c}_{pq}(a, b) = w_{pq} d_{pq}(ab)$ where $d_{pq}(ab)$ is the distance between (or cost of) label a on vertex p and b on vertex q. Thus, the cost of a labeling f is as follows:

$$\text{COST}(f) = \sum_{p \in \mathcal{V}} \mathbf{c}_p(f_p) + \sum_{(p,q) \in \mathcal{E}} w_{pq} d_{pq}(f_p, f_q) \tag{1}$$

The most promising algorithm for an efficient solution was [8], which provides an approximation algorithm based on graph cuts for the non-metric labeling problem, which requires a distance function $d(a, b)$ such that $d(a, b) = 0 \iff a = b$ and $d(a, b) \geq 0$. The algorithm provides a labeling with a cost that is an f-approximation to the minimum-cost labeling, where $f = \frac{d_{max}}{d_{min}}$, where d_{max} is the maximum distance between any two rotamers and d_{min} is the minimum distance.

2.2 Application to the Protein Design Problem

The internal energy of a protein can easily be modeled by equation 1, where the function $\mathbf{c}_p(f_p)$ is held to represent the internal energy of a rotamer f_p at position p and the function $w_{pq} d_{pq}(f_p, f_q)$ is held to represent the pairwise interaction of a rotamer f_p at position p and a rotamer f_q at position q. In addition, each residue is modeled by a single node in the vertex set $\mathcal{V}$, and interactions between residues are represented by edges between nodes in the edge set $\mathcal{E}$.

In order to maintain the distance constraint $\mathbf{c}_{pq}(a, b) = w_{pq} d_{pq}(ab)$, the label set L was set as the Cartesian product $R \times \mathcal{V}$ of the set of all rotamers with the set of all positions. Thus, a given label $l \in \mathcal{L}$ represents a specific

rotamer at a particular position. The pairwise interaction between a label and itself can then be set as zero. In order to ensure that a rotamer for position 1 was not assigned to position 2, the cost of labeling a rotamer to the wrong position was set to be prohibitively high.

3 Methods

3.1 Selection of proteins

We selected 3 proteins to investigate. The first, PDB ID FSV1, was the first computationally-designed protein structure, and was thus chosen for its relatively small size and historical significance. The other two proteins, PDB IDs 1CC8 and 3F0Q, were chosen for their use as example proteins in the open-source OSPREY software suite (as such, they had readily available data against which we could compare our graph-based algorithm). For FSV1, positions 10 and 27 were allowed to mutated. For 1CC8, positions 38-44 were allowed to mutate, and for DHFR only position 5 was allowed to mutate. By selecting a range of maximum mutations, we were able to assess the scalability of the algorithm.

3.2 Generation of pairwise interaction matrices

OSPREY ([6], [7]) was used to generate pairwise interaction matrices and provide a baseline against which to compare the results generated by the graph -based algorithm.

3.3 Optimizing the metric labeling problem

The FastPD Markov Random Field optimization library ([8] and [9]) was used in order to test the graph algorithm. The FastPD algorithm transforms the initial GL problem into the minimization of a linear program according to [10].

$$
\begin{aligned}
min & \sum_{p \in V} \sum_{a \in L} c_p(a) x_p(a) + \sum_{(p,q) \in E} w_{pq} \sum_{a,b \in L} d(a,b) x_{pq}(a,b) \\
s.t. & \sum_{a} x_p(a) = 1 \qquad \forall p \in V \\
& \sum_{a} x_{pq}(a,b) = x_q(b) \qquad \forall b \in L, (p,q) \in E \\
& \sum_{b} x_{pq}(a,b) = x_p(a) \qquad \forall a \in L, (p,q) \in E
\end{aligned}
\tag{2}
$$

Equation 2 shows a formulation of an integer linear program, where $x_p(a)$ is a binary variable that indicates whether vertex p is assigned to label a, and $x_{pq}(a,b)$ indicates that vertices p and q are assigned the labels a and b. The first constraint requires each vertex to be assigned to one label, and the other constraints maintains consistency between the labeled vertices and edges.

Optimizing the cost of the metric labeling problem is NP-hard, and many approximation algorithms (α-expansion, α-β-swap) either assume metric distances that satisfy the triangle inequality or provide no guarantees about optimality. Since pairwise interaction energies between protein residues are inherently nonmetric distances, these algorithms are not adequate for the protein design problem.

FastPD only requires a nonmetric distance function that satisfies $d(a,b)$ such that $d(a,b) = 0 \iff a = b$ and $d(a,b) \geq 0$. Since the pairwise interaction energy between a residue and itself is zero, and all energies are greater than zero, this algorithm can be applied to the protein design problem.

Relaxation of the integer linear program constraints transforms the NP-hard optimization problem into one that can be solvable in polynomial time. The FastPD algorithm modifies the constraints to $x_p(\cdot) \geq 0$ and $x_{pq}(\cdot,\cdot) \geq 0$.

3.4 The Primal-Dual Schema

The FastPD algorithm utilizes the primal-dual schema in order to rapidly obtain an f-approximation to the optimal solution. Given a primal program

$$
\begin{aligned}
& \min \mathbf{c}^T \mathbf{x} \\
& \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0
\end{aligned}
$$

the dual program can be formulated as follows:

$$\max \mathbf{b}^T\mathbf{y}$$
$$\text{s.t. } \mathbf{A}^T\mathbf{y} \leq \mathbf{c}$$

The primal-dual principle states that if $\mathbf{x}$ and $\mathbf{y}$ are solutions satisfying $\mathbf{c}^T\mathbf{x} \leq f \cdot \mathbf{b}^T\mathbf{y}$, then $\mathbf{x}$ is an f-approximation to the optimal solution $\mathbf{x}*$. FastPD starts with initial guesses for $\mathbf{x}$ and $\mathbf{y}$ and repeatedly improves them by making every variable in the program a node in the graph and using a max-flow/min-cut algorithm to alter the assigned values to each variable. The algorithm converges to an approximation factor $f_{app} = 2\frac{d_{max}}{d_{min}}$ where d_{max} is the largest distance between two labels and d_{min} is the smallest distance between two labels.

Thus, the FastPD algorithm returns a conformation whose energy is an f- approximation to the GMEC. This project aimed to test whether or not this would be sufficiently accurate so as to produce useful results.

4 Results

After 100 iterations, FastPD assigned residues to selected positions. In Tables 1, 2, and 3, the results for these assignments are summarized. The first column in each table contains the position to be optimized, and the second contains the assigned label. The n-th position corresponds to the n-th residue in the amino acid chain. The assigned label format indicates the amino acid and position assigned to the site ([amino acid]-[position]). Many of the selected labels did not correspond to the correct position or interest, indicating limitations of the algorithm.

Table 1: DHFR

Position	**Assigned Label**
5	Leu-0

4.1 Residue Labeling

There were numerous problems with applying the FastPD algorithm to the protein design problem. Most notably, for the 1FSV and 1CC8 proteins both resulted in labels applied to the wrong residue (Tables 2, 3). For 1FSV, allowing mutations at residue 10 and 27 resulted in the algorithm

Table 2: 1CC8

Position	Assigned Label
38	Ser-2
39	Val-6
40	Val-6
41	Ala-6
42	Ley-6
43	Ala-6
44	Leu-6

Table 3: 1FSV

Position	Assigned Label
10	Glu-27
27	His-27

attempting to place the glutamine reserved for residue 27 at residue 10. While we did attempt to prevent this by increasing the cost of applying a label to the wrong residue to the maximum value possible, this strategy was evidently unsuccessful.

The most likely cause of this is the f-approximation produced by the FastPD algorithm, where $f = 2\frac{d_{max}}{d_{min}}$ and the distance functions represent pairwise interaction energies. Because the maximum pairwise interaction energy would be immensely high (representing steric clash), and because the minimum pairwise interaction energy would be extremely low (representing the stabilizing influence of, for example, a hydrogen bond), the value of f will be far too high to return a useful approximation to the GMEC.

4.2 Further Implications

The results from this indicate that, on the whole, applying graph-based algorithms designed for image segmentation problems may not be a particularly fruitful endeavor. There are three reasons for this.

4.2.1 f-Approximation

Typical graph-based approximation algorithms return an f-approximation to the optimal solution [10]. The viability of this approach relies on an assumption that "good is good enough" with regards to the returned so-

lution. However, due to the multitude of local minima observed in the conformational landscape of any peptide [11], merely being *close* to the optimal configuration is not sufficient. Thus, while comparatively efficient f-approximations may be useful in image segmentation problems when slightly suboptimal solutions may not be of extreme consequence, even slight gaps between the returned solution and optimal configuration are of significant concern.

4.2.2 Multiple Labels

In addition, while the algorithm used in this project featured an approximation that scaled with the ratio of pairwise interaction distances, more commonly used algorithms feature approximations that scale with the number of possible labels on the graph [12]. In order to maintain distance constraints that permit the application of such algorithms, the number of labels must be multiplied by the number of residues, as was done here. However, this in turn reduces the accuracy of the approximation, making graph-based algorithms less useful for the protein design problem.

5 Conclusion

This project attempted to apply graph-based algorithms to the protein design problem to see if such approaches held any promise for future improvements to current algorithms. We applied the FastPD Markov Random Field optimization library to three proteins, with varying numbers of mutable residues, and found that the approximations returned were poor enough so as to be functionally meaningless. These results seem to indicate that similar graph-based approaches that provide factor-approximations to the optimal solution are poor choices for further investigation with regards to the protein design problem. It remains to be seen whether efficient ϵ-approximation algorithms exist, and, if so, whether they might be feasible options instead.

References

[1] Pierce, Niles A. and Winfree, Erik. Protein Design is NP-hard Protein Eng. (2002) 15 (10): 779-782 doi:10.1093/protein/15.10.779

[2] Dahiyat, B. I. De Novo Protein Design: Fully Automated Sequence Selection. Science 278, 82-87 (1997)

[3] Jou, J.D. Jain, S. Georgiev, I. Donald, B.R. BWM*: A Novel, Provable, Ensemble-based Dynamic Programming Algorithm for Sparse Approximations of Computational Protein Design. RECOMB (2015) Warsaw, Poland. April 12, 2015 (In Press)

[4] Kamisetty H1, Xing EP, Langmead CJ. Free energy estimates of all-atom protein structures using generalized belief propagation. J Comput Biol. 2008 Sep;15(7):755-66. doi: 10.1089/cmb.2007.0131.

[5] M. Fromer, C. Yanover, and M. Linial. Design of multispecific protein sequences using probabilistic graphical modeling. Proteins 75;3(2009 May 15):682-705.

[6] C. Chen, I. Georgiev, A. C. Anderson, and B. R. Donald. Computational structure-based redesign of enzyme activity. PNAS USA, 106(10): 37643769, 2009.

[7] P. Gainza, K. E. Roberts, I. Georgiev, R. H. Lilien, D. A. Keedy, C. Chen, F. Reza, A. C. Anderson, D. C. Richardson, J. S. Richardson, and B. R. Donald. OSPREY: Protein design with ensembles, flexibility, and provable algorithms. Methods in Enzymology, 523:87-107, 2013.

[8] Komodakis, N.; Tziritas, G., "Approximate Labeling via Graph Cuts Based on Linear Programming," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.29, no.8, pp.1436,1453, Aug. 2007 doi: 10.1109/TPAMI.2007.1061

[9] N. Komodakis, G. Tziritas and N. Paragios, "Performance vs Computational Efficiency for Optimizing Single and Dynamic MRFs: Setting the State of the Art with Primal Dual Strategies". Computer Vision and Image Understanding, 2008 (Special Issue on Discrete Optimization in Computer Vision).

[10] C. Chekuri, S. Khanna, J. Naor, and L. Zosin, Approximation Algorithms for the Metric Labeling Problem via a New Linear Programming Formulation, Proc. 12th Ann. ACM-SIAM Symp. Discrete Algorithms, pp. 109-118, 2001.

[11] Dobson CM (2000-12-15). "The nature and significance of protein folding". In RH Pain. Mechanisms of Protein Folding (2nd ed.). Oxford, UK: Oxford University Press. ISBN 0-19-963788-1.

[12] Y. Boykov, O. Veksler, R. Zabih. Fast approximate energy minimization via graph cuts. IEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-1239. Nov 2001.