

# Progress Report: Graph-Based Representations of the Protein Design Problem

Adi Mukund

Jennifer Zou

This project consists of two main components: (1) an attempt to apply graph cuts to the residue interaction graph in order to see if such algorithms hold any potential for developing new, efficient approaches to solving the protein design problem, and (2) a graph decomposition-based modification of cluster expansion in order to retain the computational benefits that cluster expansion provides while minimizing the gap between the values returned by cluster expansion and more traditional energy functions.

## 1 Graph Cuts and the GMEC

The main idea behind this component of the project is to try to apply graph cut-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:

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

The most promising algorithm for an efficient solution was [1], 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.

The steps over the next few weeks are as follows:

1. Implement the algorithm as an extension to OSPREY
2. Identify how the conformations from by the algorithm differ from those provided by OSPREY.
3. Determine if the  $f$ -approximation is sufficiently tight so as to produce useful results.

## 2 Modified cluster expansion

Cluster expansion (CE) can be used to approximate the energy of proteins from sequence features or cluster functions (CFs). The algorithm fits a set of coefficients called effective cluster interaction (ECI) values that reflect the energetic contribution of a CF. After defining a set of candidate CFs, an iterative algorithm called CLEVER can be used to determine which CFs contribute significantly to total energy.

Even though low-order interactions between residues are assumed to contribute more to total energy, high-order interactions can affect prediction accuracy of CE. [2] However, inclusion of many high-order interactions in CE is computationally intense and may lead to overfitting.

The triplet option in CLEVER considers all triple CFs that are spanned by three pair CFs. [3] This filter chooses triplets CFs that are collinear with the pair CFs and may neglect important interactions. Rather than choosing triplet CFs from the pair CFs, we hope to use strongly connected components in sparse residue interaction graphs to create high-order interaction terms that match the substructure of graphs. Alternatively, terms can be generated by finding articulation points within branch decompositions of the sparse residue graphs.

The size of strongly connected components in sparse residue graphs is generally too large to include in cluster expansions. The distribution could be shifted by sparsifying the graphs, but this may introduce additional bias.

Large strongly connected components could potentially be broken up into several CFs, but it is difficult to determine how this will impact the accuracy the CE.

Tasks for the remaining of the semester include

1. Implement branch decomposition
2. Run cluster expansion with new interaction terms
3. Compare results with CLEVER triplet option

## References

- [1] 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
- [2] Grigoryan G, et al. Ultra-fast evaluation of protein energies directly from sequence. PLoS Computational Biology 63,0551-0563 (2006).
- [3] Negron C, Keating A. Multistate Protein Design Using CLEVER and CLASSY. Methods in Enzymology, Vol.523.