A machine learning based approach to improve sidechain optimization

Sabareesh Subramaniam
Department of Computer Sciences
University of Wisconsin-Madison
sabarees@cs.wisc.edu

Sriraam Natarajan
Translational Science Institute
Wake Forest University
snataraj@wfubmc.edu

Alessandro Senes
Department of Biochemistry
University of Wisconsin-Madison
senes@wisc.edu

ABSTRACT

Side chain optimization is the process of packing the sidechains of a protein onto a fixed backbone structure, such that the energy of the resultant structure is minimized. The continuous space of sidechain conformations is typically handled by discretizing (sampling) into a finite set of representative conformations called a "conformer library". The key contribution of this work is to use machine learning methods to distribute (conformational) sampling among different positions in a protein. The idea is that different positions in a protein backbone have different sampling requirements, for example, solvent exposed positions require less sampling than positions in the core of a protein. We propose a 3-ary categorization of every position in a target protein based on its sampling requirements and evaluate it by comparing against an unbiased distribution of conformers. Our results demonstrate that this strategy helps to distribute the sampling more efficiently for sidechain optimization.

Categories and Subject Descriptors

[**Protein and RNA Structure**]: BioInformatics

1. INTRODUCTION

Most sidechain optimization procedures revolve around four key elements: 1) the protein backbone; 2) a library of discrete sidechain conformations[1] that provides sidechain mobility; 3) a set of energy functions for scoring structures and 4) a search strategy to identify the best solution. The backbone and the library together define the optimization problem and the search strategy attempts to find the global minimum energy configuration (GMEC). A number of algorithms exist to search for the GMEC from among the conformers in the library, however, irrespective of the algorithm used, the final result can approach the GMEC only if the correct sidechain conformations were provided by the library. This work focuses on improving the library used for sidechain optimization.

The level of sampling required to fit a sidechain in a position depends on the constraints imposed by the environment. For example, some positions may be present in the protein core surrounded by immobile backbone atoms, such positions require specific conformations to interact favourably with the surrounding atoms. On the other hand, positions on the surface of a protein are relatively more flexible. However, an exact relationship between a position and its sampling requirement is difficult to identify. The question that we address in this paper is the following: *How do we identify the set of sidechain conformations for each position in a protein that will maximize performance in sidechain optimization?*. Here, we present a machine learning (ML) method that analyzes the structural characteristics of each position in a protein backbone and identifies its sampling requirements. We compile a dataset of high resolution structures from the Protein Data Bank[2], label each position in this dataset and train a classification algorithm to predict each label based on environmental features.

This paper makes several key contributions: First, it considers the difficult problem of distributing sampling among different positions in a protein. To our knowledge, there is no method that attempts to differentiate between positions of the same amino acid type. Second, the proposed method is directly based on the objective function of the optimization (*energy*) and uses structural features that allow for generalization. Third, evaluation on 44 proteins shows that the ML-based approach can yield a better search space compared to the baseline method with similar memory and runtime constraints.

2. CONFORMER LIBRARY

All experiments in this paper are based on the conformer library we created in our previous work[9] : a library that distributes sampling based on the energetic impact of sidechain conformation instead of pure geometry, compiled using the following strategy for each amino acid:

1: An exceedingly fine-grained library of N conformers is created from high resolution crystal structures.
2: A large number M of natural environments(protein positions) that contain the same amino acid type are selected from the PDB.

[1] Note that sidechain optimization can be performed without the use of a rotamer library but as far as we are aware, rotamer/conformer libraries seem to be the popular choice.

ACM-BCB '11, August 1-3, Chicago, IL USA

3: The native sidechains in the environments are remodeled into the shape of each one of the N conformers and the energy is measured (with the CHARMM-22[3] all-atoms force field).The operation produces an $N \times M$ table of energies.

4: Using a threshold, each energy is converted to a boolean value to indicate if the conformer fits (T) or does not fit (F) the environment.

5: The conformer that fits the largest number of environments is selected as the top-ranking conformer and is added to the sorted library.

6: All environments that have been already satisfied by the previous conformers in the list are no longer considered.

7: The conformer that fits the largest number of remaining environments is selected as the best complement to the previous and added to the list.

8: The process is continued until all environments are covered.

The method produced a ranked list of conformers for each amino acid type where the first n conformers is probably the best set of 'n' conformers.

3. MACHINE LEARNING TO DISTRIBUTE CONFORMER SAMPLING

We propose to classify each position in a protein into one of three categories (easy, medium and hard) based on its sampling requirement.

3.1 Labeling

We *score* each position in the training set as follows: we traverse the sorted list and for each position, find the number of conformers that result in a "good" interaction i.e) sufficiently low energy. An important thing to note is that all the other positions retain their crystal structure. The *score* is then the number of conformers that are a good fit for the position. Every position with a high score can be fit by a large number of conformers, therefore, it is easy to find a conformer in the library that fits this position and so it will be labeled as "easy" . A position with a low score, on the other hand, will be labeled as "hard". We label the top and bottom 25 percentile of positions (based on the score) as "easy" and "hard" respectively, and all positions in between are labeled "medium".

3.2 Classification

In the labeling step, it is worth noting that the sidechain atoms of other postions need to be present(in their crystal states) and so the above strategy cannot be applied directly on an unknown protein. Hence, we use a set of structural features that can be obtained easily from the backbone to train the classifier.The features include: (a) the *backbone dihedral angles* (ϕ and ψ) for 4 positions before and after the current, 16 in total (b) *local sequence* information (i.e., the residues for 4 positions before and after r_i^j), 9 features (c) the *number of backbone atoms* not in the local sequence and are within $\langle 4, 8, 12, 16 \rangle$ Å and (d) *solvent accessible surface area.* This data set is then used for our second step where we learn a model to predict the category of each residue (position) for a given backbone.We employ different learning algorithms ranging from Naive Bayes to decision trees to ensemble methods such as bagging and boosting in our experiments. The result of the classification step is a label indicating the "hardness" of a position. Given this label, we use a preset number of conformers to fit at each position.

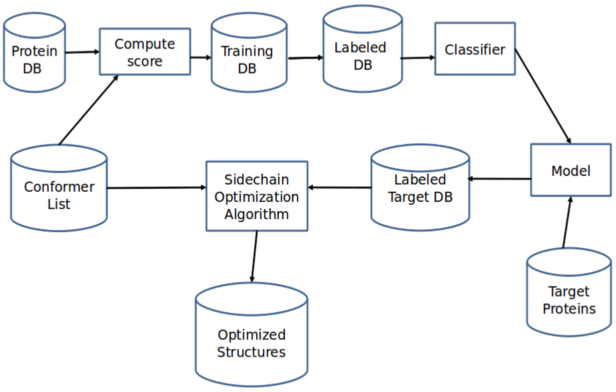

Figure 1: Schematic representation of the biased sampling strategy.

3.3 Algorithm for distributing sampling

The different steps involved in our algorithm are presented visually in Figure 1. The first-step is to compute the scores and assign a label for each position. Then, we learn a model that learns a mapping from the features to labels. This model is then used to label each residue of a seperate test set of proteins. Then a sidechain optimization algorithm is used to identify the GMEC for the current protein.

4. EMPIRICAL EVALUATION

Our training dataset consists of approximately 676 proteins and the test set contains around 44 proteins. The proteins were modeled using MSL[1], which implements all[2] the energy terms in the CHARMM[3] potential along with the hydrogen bonding term implemented in the SCWRL[7] program. Classification is then performed on this dataset for each amino acid separately using the WEKA[5] package. The MSL program used to perform sidechain optimization implements dead-end elimination[4], followed by (if resultant search space cannot be enumerated) self-consistent mean field[8] and metropolis monte carlo search[6]. Our goal in this section is to answer the following questions:

Q1: Does the labeling scheme really help achieve the goals of reduced search spaces and/or improved energies?

Q2: How do the classification methods compare against a baseline method that does not differentiate between the different positions (but uses a search space of similar size)?

4.1 Effectiveness of the Labels

We analyze the precision of the labeling process by performing sidechain optimization using our biased sampling scheme and comparing energies obtained with an unbiased sampling scheme. We assign a *low* level of sampling to the easy positions and *high* level of sampling to the hard positions. The "medium" positions and the unbiased scheme are assigned the same number of conformers such that the difference in the search spaces for the biased sampling and the

[2] We included only the bond, angle, dihedral, improper and Van der Waals energy terms in our experiments.

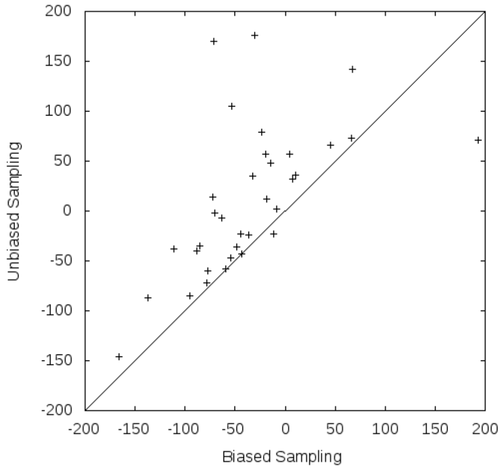

Figure 2: Energies obtained by our biased sampling scheme (using the actual labels) are lower than those obtained by an unbiased scheme.

unbiased sampling scheme is minimal. Results in Figure 2 show that sampling less/more on the positions labeled as easy/hard indeed enables the sidechain optimization algorithm to achieve better energies than an unbiased sampling scheme, thus answering $Q1$ affirmatively.

4.2 Experiments with classification algorithms

We use the following classification algorithms from the WEKA package: a) *Bagging* b) *simpleCART* - decision tree learner c) *LogitBoost* - Boosting of decision stumps and d) *Naive Bayes*. Figure 3 presents the results of the comparison of the models against the baseline method which uses an unbiased sampling for all positions. For each of the classification methods, we compare the energies produced on the classified and baseline models. We compute the fraction of proteins in which the energy for one method is significantly ($> 2\ kcal.mol^{-1}$) lower than the other. The figure presents the results for all the classifier methods listed above. As can be seen, the use of the classification methods yields a superior performance for a larger fraction of proteins compared to the baseline method. The performance seems to be comparable for Bagging, CART and NaiveBayes with LogitBoost performing slightly less efficiently. Hence, we can answer $Q2$ positively i.e., the classifiers perform better than a baseline that is unbiased.

Thus we see that the labels produced by the classifiers improve performance in a vast majority of the test proteins.

5. CONCLUSION

In this work, we addressed the challenging problem of distributing sampling among different positions in a protein. To achieve this, we developed a two-stage approach that used an underlying conformer library. In the first-step, we labeled the positions based on the number of conformers in the library that can fit in the position. In the second step, we used a different set of features (i.e., structural features) to learn a classifier that is able to predict the hardness of the different positions. We evaluated the two-stage approach on 44 proteins from the protein data bank. Our evaluation allowed us to answer affirmatively a variety of questions ranging from the validity of the labeling strategy to the effectiveness of the various classification algorithms. Our results strongly demonstrate that the use of machine learning indeed results in superior performance compared to a method that does not explicitly differentiate the positions.

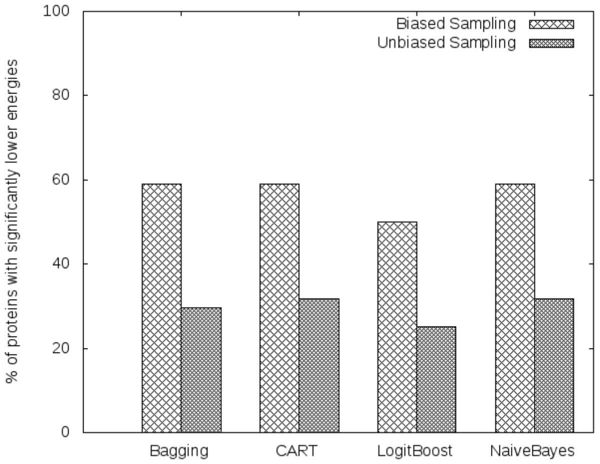

Figure 3: Classification methods provide labels that help attain lower energies in more test proteins than the unbiased scheme.

6. REFERENCES

[1] Molecular Simulation Library. `http://msl-libraries.org`.

[2] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[3] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4(2):187–217, Feb 1983.

[4] R. F. Goldstein. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophys J*, 66(5):1335–1340, May 1994.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P.Reutemann, and I. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009.

[6] L. Holm and C. Sander. Fast and simple monte carlo algorithm for side chain optimization in proteins: Application to model building by homology. *Proteins: Structure, Function, and Bioinformatics*, 14(2):213–223, 1992.

[7] G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins*, 77(4):778–795, Dec. 2009.

[8] J. Mendes, C. M. Soares, and M. A. Carrondo. Improvement of side-chain modeling in proteins with the self-consistent mean field theory method based on an analysis of the factors influencing prediction. *Biopolymers*, 50(2):111–131, 1999.

[9] S. Subramaniam and A. Senes. An energy optimized conformer library for sidechain optimization. In preparation, 2011.