IMPROVING DECOY DATABASES FOR PROTEIN FOLDING ALGORITHMS

An Undergraduate Research Scholars Thesis

by

AARON LINDSEY

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Research Advisor: Nancy M. Amato

May 2014

Major: Computer Science

TABLE OF CONTENTS

Page

ABSTRACT

Improving Decoy Databases for Protein Folding Algorithms. (May 2014)

Aaron Lindsey
Department of Computer Science and Engineering
Texas A&M University

Research Advisor: Dr. Nancy M. Amato
Department of Computer Science and Engineering

Predicting protein structures and simulating protein folding motions are two of the most important problems in computational biology today. Modern folding simulation methods rely on a scoring function which attempts to distinguish the native structure (the most energetically stable 3D structure) from one or more non-native structures. Decoy databases are collections of non-native structures that are widely used to test and verify these scoring functions.

We present a method to evaluate and improve the quality of decoy databases by adding novel structures and/or removing redundant structures. We test our approach on 13 different decoy databases of varying size and type and show significant improvement across a variety of metrics. The most improvement comes from the addition of novel structures indicating that our improved databases have more informative structures that are more likely to fool scoring functions. We also test our improved databases on a popular modern scoring function. We show that they contain a greater number of native-like structures than the original databases, thereby producing a more rigorous database for testing scoring functions. This work can aid the development and testing of better scoring functions, which in turn, will improve the quality of protein folding simulations.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Nancy M. Amato, for her guidance and for giving me the opportunity to work in her research group.

I would also like to thank my postdoctoral advisor, Dr. Shawna Thomas, and my graduate student mentor, Hsin-Yi (Cindy) Yeh, for the immense amount of time that they have put into mentoring me and helping me with my project.

Thanks to Chih-Peng Wu, for working hard to help me run experiments and obtain good results.

Many thanks to every other member of the Parasol Lab, epecially those in the Computational Biology group: Chinwe Ekenna, Mukulika Ghosh, and Shuvra Nath. Their collaboration and support have made this project a success.

A final thank-you to Jory Denny, for mentoring me when I initially joined the group and continuing to encourage me.

This work is supported in part by NSF awards CRI-0551685, CCF-0833199, CCF-0830753, IIS-096053, IIS-0917266 by THECB NHARP award 000512-0097-2009, by Chevron, IBM, Intel, Oracle/Sun and by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

CHAPTER I

INTRODUCTION

Two important, and related, problems in computational biology are predicting protein structures and simulating protein folding motions. The protein's most energetically stable structure, called the native structure, determines its function and how it interacts with other proteins. Because a protein's structure and function are so intimately related, predicting a protein's structure is of paramount importance. In addition, errors in the protein folding process (i.e., folding from an unstructured chain of amino acids to the native structure) cause a protein to fold incorrectly thereby altering its functional ability. Such alterations are thought to lead to many devastating diseases including Alzheimer's, Mad Cow, and Parkinson's disease [8]. Thus, the folding process itself remains an important area of study.

Many computational tools have been developed to study these problems because they are either too difficult or too expensive to tackle experimentally. Protein structure prediction [26] is a widely studied area. One notable method is Rosetta [22] which uses a simplified model (similar to [28]) to predict the low-resolution protein structure. In response to increased research in protein structure prediction, the CASP [24] competition emerged as a platform to test structure prediction methods. Molecular dynamics [18] and Monte Carlo simulations [9] have been widely used to simulate protein motion. All of these methods rely on a scoring function, typically an energy function. A scoring function attempts to distinguish between native and non-native structures, ranking them in terms of their energetic feasibility. Thus, the accuracy of these methods largely depends on the accuracy of the scoring function used.

In response to this, decoy databases have been developed to test and verify these scoring functions [23]. A decoy is a computer-generated protein structure that is similar to the native structure. Decoys test the ability of a scoring function to identify the protein's native structure from among a set of incorrect protein conformations. If the scoring function can correctly identify the native structure, the function is said to be correct. Such tests where

decoys attempt to "fool" the scoring function are commonly used to test protein folding algorithms. Thus, if we can create higher quality decoy databases, we can improve protein folding algorithms by improving the scoring functions they rely on.

Many large decoy databases for specific proteins have already been compiled for the purpose of testing and improving scoring functions [23, 21, 30]. However, there is not currently a good way to take these existing databases and improve them so that they are more effective at testing modern scoring functions. Here, we strive to generate higher quality decoy sets in order to more rigorously test these functions.

The **contributions** of this research are methods for:

- evaluating the quality of decoy databases,
- improving the quality of decoy databases by adding novel structures, and
- improving the quality of decoy databases by removing redundant structures.

In order to evaluate the quality of decoy databases, we must describe the attributes of a good database. First, an effective database should contain structures with an overall average potential energy that is similar to the energy of the native structure. This makes it difficult for the scoring function to distinguish between the native structure and the other decoys. A high quality database should also contain structures that cover the potential energy space of the protein. This is the set of all possible conformations and their associated energies. We would like to have a wide variety of structures in the set so that we can be sure that various local minima throughout the energy space are represented by the set. Finally, the database should contain as few structures as possible to cover the potential energy space. In other words, it should not contain redundant structures, which are non-informative structures that are very similar to another structure in the set. We later define a variety of metrics to measure the quality of an existing decoy database based on these attributes.

We propose two methods of building high quality decoy databases using existing databases: The addition of novel structures to and the removal of redundant structures from the existing

database. Before the improvement process, we create a set of filters which define the criteria for a decoy structure to be considered acceptable in the final database. Various filters and their objectives are described later. We start by generating a set of sample structures that we use to add to the existing database. The size of the sample set is chosen to achieve maximum improvement while maintaining a reasonable number of samples, and the sample structures are created using information from the native structure and the existing decoys to earn a low potential energy score. We then apply our set of filters to the sample set to obtain a set of novel informative structures and to the existing decoy database to remove any redundant structures. The final database is a collection of structures from the existing database and structures generated using our sampling methods.

We test our approach on 13 different decoy databases of varying size and type and show that we are able to generate higher quality decoy databases across a variety of metrics. We find that most improvement stems from the addition of structures by our sampling methods. Thus, our improved databases contain more informative structures that are more likely to fool modern scoring functions. We also test our improved databases on QMEAN [4], a popular modern scoring function, and show that they contain a greater number of structures ranked more native-like than the actual native state than the original databases for many of the proteins studied. Our hope is that these methods will be applied to many of the existing decoy databases and in turn will lead to more accurate protein folding algorithms.

CHAPTER II

PRELIMINARIES AND RELATED WORK

In this chapter we discuss related work in the areas of protein models, scoring functions, and existing decoy databases. We also discuss existing methods for sampling conformations as we will use these to add conformations to existing decoy sets.

Protein Models

A protein is composed of a chain of amino acids that determine its function. While the amino acids share a common backbone, each has a unique side chain that identifies the type of amino acid. Figure II.1 shows an illustration of an amino acid's structure.

When hydrogen bonds form between atoms on the protein backbone, secondary structures can develop. α helices and β sheets make up the majority of secondary structures. An α protein does not contain any β sheets, and likewise a β protein does not contain any α helices. If a protein contains both α helices and β sheets, we call it an α/β mixed protein.

The most accurate protein model is the all-atoms model. It takes all bond angles and torsional angles into consideration. However, in many cases the all-atoms model is too computationally expensive, particularly for larger proteins.

A coarser-grained approach models a protein as a series of ϕ and ψ torsional angles. All other bond angles and all bond lengths remain fixed. This is a common modeling assumption as bond lengths and angles typically only undergo small fluctuations [26]. In this $\phi - \psi$ model, a protein conformation with n amino acids has $2n$ degrees of freedom. Side chains are modeled as spheres with zero degrees of freedom located at the C_β position. This model has been successfully used to simulate the correct order of large folding events for several small proteins [2].

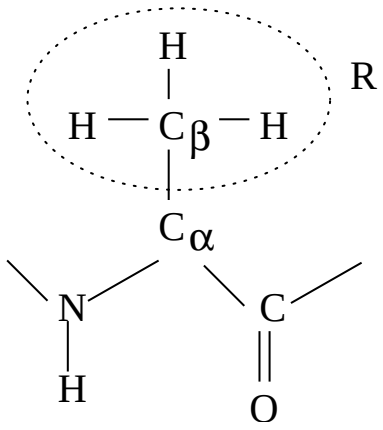


Fig. II.1. An example of our amino acid model for Alanine. We model the side chain as a sphere around C_β and do not explicitly represent the hydrogen atoms attached to C_β.

Regardless of the model, a protein's potential energy landscape is defined as the set of all conformations and their associated potential energies. Figure II.2 shows a theoretical energy landscape where all possible conformations are represented by the $x - y$ plane. The native structure of a protein is thought to be located at the lowest point on a funnel-shaped landscape. The landscape may contain several local minima which are distinct from the native energy basin.

Energy Functions

The atoms in the protein interact not only with each other but also with the surrounding solvent through bonds and non-bond interactions such as electrostatic interaction and van der Waals forces. A potential energy function determines conformation validity by taking into account these different atom interactions.

Generally, potential functions that compute all pairwise interactions for a molecule are called all-atoms functions, e.g., CHARMM [6] and AMBER [31]. All-atoms functions are the most accurate since they consider all possible interactions. However, they are computationally expensive and not feasible for many large proteins.

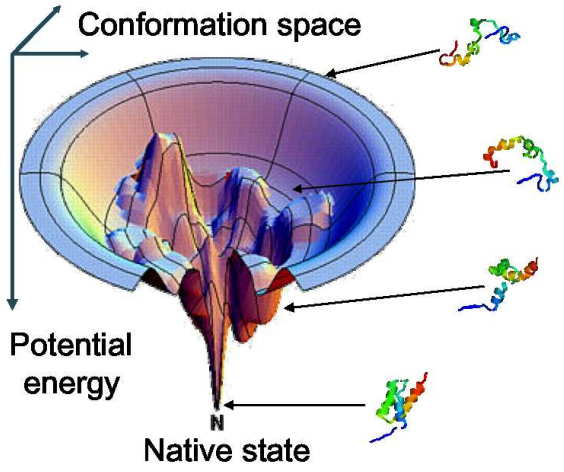


Fig. II.2. The potential energy landscape of a protein is the set of all conformations and their associated potential energies [7]. The conformation space of a protein can span hundreds of degrees of freedom; it is not limited to only the $x - y$ plane.

Instead of modeling all possible interactions, coarse-grained functions only consider side chain contributions to approximate the potential energy. Recall that in the $\phi - \psi$ model, each side chain is modeled as a sphere located at the C_β atom. If two side chains are too close (less than 2.4 Å), the conformation is rejected. Otherwise, we use the energy equation from Levitt et. al.:

$$U_{tot} = \sum_{restraints} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_c\} + E_{hp}, \tag{II.1}$$

where K_d is 100 kJ/mol, d_i is the distance of a hydrogen or disulphide bond in native structure, and $d_0 = d_c = 2$Å [18]. This energy function has been shown to produce folding simulation results similar to an all-atoms function in a fraction of the time [25].

Decoy Sets

Decoys are computer-generated protein structures. Decoy databases have been used to improve the accuracy of scoring functions [13, 19, 27]. A scoring function is the component

of a protein folding algorithm that distinguishes between native and non-native structures. Thus, the performance of the algorithm is dependent on the accuracy of the scoring function. Decoy databases attempt to "fool" a scoring function into choosing a non-native structure as the native. Some existing decoy databases include (i) the Decoys 'R' Us set [23], (ii) the Rosetta set [30], and (iii) the Critical Assessment of Protein Structure Prediction (CASP) set [21].

The Decoys 'R' Us set contains three subsets: the single decoy set, the multiple decoy set, and the loop decoy set. The single decoy set only contains the native structure and one decoy structure. The purpose of this set is to test whether a scoring function can distinguish between these two structures. The multiple decoy set and the loop decoy set each contain many decoy structures, and they are both used to verify that a scoring function can select a conformation with low RMSD to the native structure.

The Rosetta set is generated by the Rosetta protein structure prediction method developed in the Baker Laboratory. The Rosetta prediction method can generate low-resolution structures by adding side chains and making structure adjustments [5].

CASP is a protein structure prediction competition held every other year. Competition submissions are collected as a decoy database. Participants use their own approaches to predict the three-dimensional structure of the given amino acid sequence. In order to evaluate the results, the distances between the C_α positions in the predicted model and the target structure are calculated [10] and a score is assigned showing how similar the prediction is compared to the target [32].

Some work has been proposed to improve protein decoy sets. For example, the Rosetta set has been improved by adding back the side chains and running the structures through an energy minimizer [30]. Other work uses a library of short fragments to generate protein decoys by assembling them together given the protein's geometric constraints [16]. Most assembled proteins are 6Å from the native structure. Fragments of varying lengths are used in [20] to refine near-native protein decoy structures. While this multi-level approach

produces decoy structures closer to the native structure, this method is dependent on the quality of the input fragments.

Sampling Conformations

Algorithms in the field of motion planning and robotics use sampling-based methods to generate valid robot conformations. Some examples include the Probabilistic Roadmap (PRM) method [15] and Rapidly-Exploring Random Trees (RRT) [17]. Both of these strategies rely on a sampling method to find valid conformations for a robot in its environment.

In the context of protein folding, sampling methods generate protein conformations by setting a value for each degree of freedom in the protein model. Thus, for the $\phi - \psi$ protein model, a conformation q is generated by assigning a value to each ϕ and ψ angle. The conformation is accepted based on its potential energy $E(q)$ with the following probability:

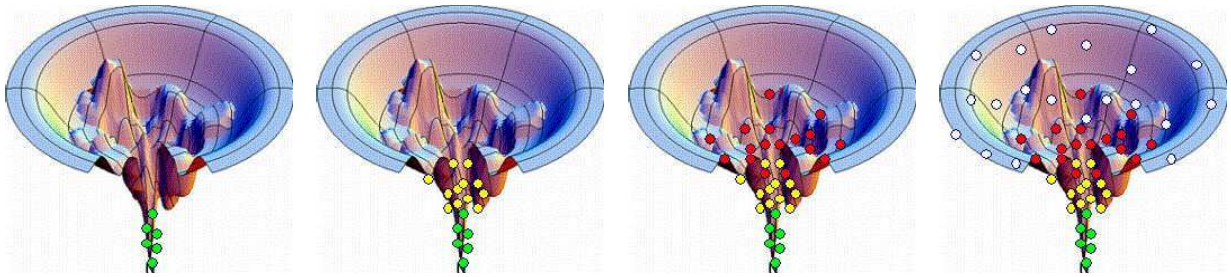
$$P(\text{accept}q) = \begin{cases} 1 & \text{if } E(q) < E_{min} \\ \frac{E_{max} - E(q)}{E_{max} - E_{min}} & \text{if } E_{min} \leq E(q) \leq E_{max} \\ 0 & \text{if } E(q) > E_{max} \end{cases}. \tag{II.2}$$

Values for each degree of freedom may be generated either directly from the amino acid sequence or using an existing conformation to bias generation. This existing conformation, for example, may be the native structure or a decoy structure.

The simplest strategy is to sample each ϕ and ψ angle uniformly at random. The resulting sample may be passed through an energy minimization function to improve its energetic feasibility. While this scheme has the ability to generate samples all across the energy landscape, it does not provide dense coverage around more interesting regions, e.g., the native energy basin. Due to the high dimensionality of the energy landscape, it would require an infeasible number of samples to cover these areas well.

To improve the coverage in the native energy basin, some sampling strategies bias sampling around the known native structure. Gaussian sampling [2] selects values for each ϕ and

ψ angle from a set of normal distributions centered around the native structure. Iterative Gaussian sampling [1] applies such perturbations iteratively. Instead of always sampling from a set of normal distributions centered around the native structure, the normal distributions are centered around seeds, or sampled conformations from prior iterations. Figure II.3 illustrates this process.



(a) (b) (c) (d)

Fig. II.3. Iterative Gaussian sampling gives a denser distribution near the native structure and a sparse distribution as they grow outward.

CHAPTER III

METHODS

In this chapter we describe our approach to evaluating and improving the quality of decoy databases. We first discuss how to evaluate a decoy set using various metrics. We then present two types of improvement operations: adding novel structures to the set and removing redundant structures from the set.

Decoy Set Evaluation

Because our methods improve existing decoy sets, we first develop strategies for analyzing the quality of decoy sets. These are used later to show what advantages the improved set provides over the original. We present several quantitative metrics to compare decoy sets and describe how their values are calculated in the experiments.

Z-Score

The z-score (or standard score) indicates the number of standard deviations between the native structure energy and the average energy of a decoy set [30]. Researchers frequently use z-score to determine the likelihood that a scoring function would pick the native structure from the other structures in the set. A positive z-score means the native structure has a higher energy than the average energy of the set, and a negative z-score means the native structure energy is lower than the average energy. A z-score of zero indicates the native structure energy is exactly the same as the average energy of the decoys. The z-score of a decoy set D is:

$$\text{ZSCORE}(D) = \frac{\text{E}(D.\text{native}) - \text{E}_{avg}(D)}{\text{E}_{std}(D)} \tag{III.1}$$

where $\text{E}(d)$ is the energy of a structure d, $\text{E}_{avg}(D)$ is the average energy of D, and $\text{E}_{std}(D)$ is the standard deviation of the energies in D. A desirable decoy set has structures with low

energies close to the native structure. Thus, we would like to see the z-score approach zero after the improvement process.

Improvement Score

Given an original decoy set D and an improved decoy set D', the improvement score returns the change in z-score per sample between the two sets. We define this metric to help determine the optimal sample set size for our decoy selection algorithm when improving decoy sets. The improvement score between D and D' is:

$$\textsc{Improvement}(D, D') = \frac{\textsc{ZScore}(D')}{|D'|} - \frac{\textsc{ZScore}(D)}{|D|}. \tag{III.2}$$

Higher values indicate greater changes in z-score.

Minimum Distance

The minimum distance metric measures the average minimum distance from each decoy structure to any other decoy structure in the set. In other words, it is the average distance of each structure to its closest neighbor measured by some distance metric δ, (see Algorithm 1).

Algorithm 1 $\textsc{MinDist}(D, \delta)$

Input. A decoy set D and a distance metric δ.
Output. The average minimum distance from a structure in D to its closest neighbor as measured by δ.

```
1: d_total ← 0
2: for every i ∈ D do
3:    d_min ← ∞
4:    for every j ∈ D \ {i} do
5:       d_min = min(d_min, δ(i, j))
6:    end for
7:    d_total ← d_total + d_min
8: end for
9: return d_total/|D|
```

We define this metric to measure the diversity of structures in the set. As the minimum distance metric increases, the diversity of structures included in the set also increases. Possible distance functions include Euclidean distance in $\phi-\psi$-space, CαRMSD, and a distance based on the differences between the rigid and flexible regions in each structure [29].

Decoy Set Improvement

There are two main phases in the improvement of decoy sets. First, samples are generated on the protein's energy landscape. This set may be generated in a variety of ways and is discussed in further detail below. In the decoy selection phase, some structures are chosen from the original set D to be removed and some are chosen from the sample set S to be added. Decoy selection is discussed below. Algorithm 2 describes the approach.

Algorithm 2 ImproveDecoySet(D, F, n, m)

Input. A decoy set D, a set of filters F, a number of samples to generate n, and a number of attempts to generate a single sample m.
Output. An improved decoy set D'.

1: $S \leftarrow$ GenerateSamples(n, m)
2: $D' \leftarrow$ SelectDecoys$(D, D, F) \cup$ SelectDecoys(D, S, F)
3: **return** D'

Sample Set Generation

To improve decoy sets by adding structures, we must first generate a set of samples from which to select. Algorithm 3 describes this process where GenerateSample() returns a structure created using one of the methods discussed below and Valid(s) returns true if the structure s is energetically feasible as given by Equation II.2.

Sampling Methods. We study several different sampling methods described below.

- *Uniform Sampling.* GenerateSample() returns a structure at a random point on the energy landscape by simply selecting values for each ϕ and ψ angle uniformly at random. This will generate many unwanted high-energy structures but provides good

Algorithm 3 GENERATESAMPLES(n, m)

Input. A number of samples to generate n and a maximum number of attempts to generate a single sample m.

Output. A set of samples S.

```
1: S ← ∅
2: for i = 0 ... n do
3:    for j = 0 ... m do
4:       s ← GENERATESAMPLE()
5:       if VALID(s) then
6:          S ← S ∪ {s}
7:          break
8:       end if
9:    end for
10: end for
11: return S
```

coverage of the landscape. Unlike the other methods, it is not biased by any input structure.

- *Sampling with Native Bias.* GENERATESAMPLE() returns structures from iterative Gaussian sampling [1]. This sampling approach has been successfully applied to simulate the folding process on larger proteins. It generates many low energy samples, but they are usually confined to the native energy basin.

- *Biased Sampling from Low-Energy Decoys.* Instead of starting from the native structure as iterative Gaussian sampling [1], this approach begins the iteration from the decoy structures with the lowest energy. To our knowledge, this is a novel approach to generating low-energy structures. As with native bias sampling, perturbations are selected from a set of normal distributions. The advantage is that generated structures have low energies and are not confined to the native energy basin. However, it typically produces samples near the energy basins of the selected decoys.

These methods may be combined to form a hybrid sampler that exploits the strengths of each method. Such a sampler first adds the native structure to the set of seeds as in iterative Gaussian sampling [1]. For the remaining seeds in the set, it selects half from the lowest

energy decoy structures and half from uniform sampling. This ensures that there are plenty of low-energy structures in the final set that are located throughout the energy landscape in many different local minima. Such structures are important to include because they are most likely to confuse a scoring function.

Calculating Sample Set Size. For each sample set, we must specify n, the number of sample structures to generate. This value affects the quality of the final decoy set because the decoy selection algorithm uses the sample set size to determine how strict to make its acceptance criteria. To calculate the ideal sample set size for a specific decoy set, we monitor the rate of change in z-score with increased n and select the ideal size as the value at which this rate of change peaks. This is a common learning algorithm technique (called the *elbow criterion*) used to help determine parameter values [14].

Figure III.1 shows the z-score and its rate of change for various sample set sizes using a decoy set for 4pti [23]. The original decoy set contained 334 structures. We search set sizes above and below this value and average our results over 10 runs. The set size that maximizes the rate of change in the z-score for this run is approximately 1350 (denoted by the triangle).

Decoy Selection

Given an existing decoy set D and a set of sample structures S, we would like to add viable structures from S to D and remove redundant structures from D. To select such structures, we apply a filter to each one. We investigate the following filters:

- *Energy Filter.* This filter chooses all structures whose energy is less than some threshold.
- *Minimum Distance Filter.* This filter selects structures whose distance to their closest neighbor as determined by some distance metric δ is greater than some threshold.

Filter thresholds are made stricter if the given sample set is large or more lenient if the sample set is small.

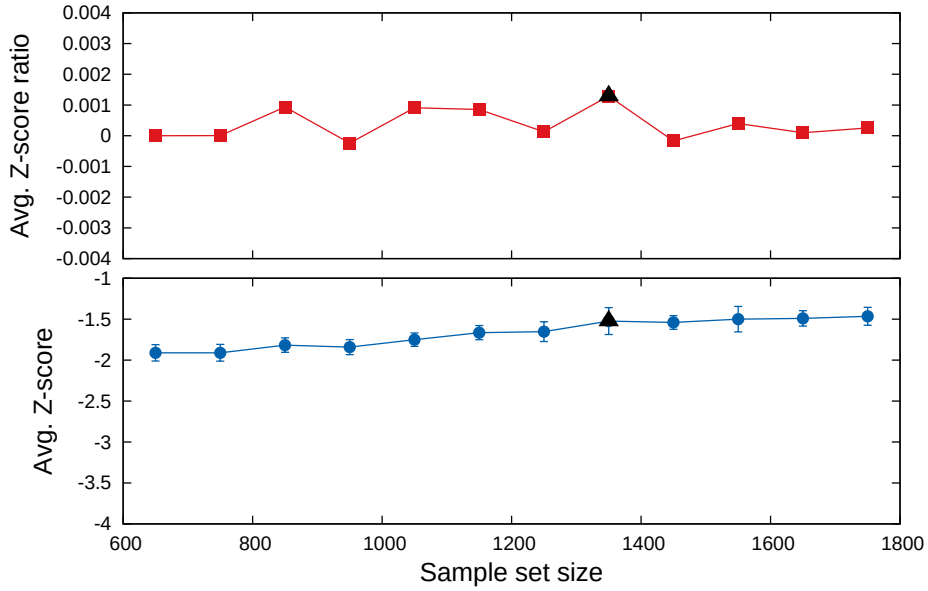


Fig. III.1. Z-Scores (below in blue) and their rate of change (above in red) using the 4pti decoy set. The rate of change peaks at approximately 1350 samples (denoted by the triangle).

Algorithm 4 illustrates how a viable set is formed from a decoy set D, a sample set S, and a set of filters F. The function SetThreshold(D) computes the threshold for the filter f by finding average values for f over D. Once a threshold is computed, structures are removed if they fail to meet the threshold.

In the case where S is a generated sample set, new structures will be chosen. In the case where $S = D$, only the viable structures from D will be returned.

Algorithm 4 SELECTDECOYS(D, S, F)

Input. A decoy set D, a sample set S, and a set of filters F.
Output. A set of viable decoy structures V from S.

```
1: V ← S
2: for every f ∈ F do
3:    f.SETTHRESHOLD(D)
4:    for every s ∈ S do
5:       if ¬f.PASS(s) then
6:          V ← V \ {s}
7:       end if
8:    end for
9: end for
10: return V
```

CHAPTER IV

RESULTS AND DISCUSSION

We apply our methods to existing decoy sets and show that they are able to generate sets with lower energies and more diverse structures that are more likely to fool protein folding scoring functions. All decoy sets were obtained from the Decoys 'R' Us database [23] and are listed in Table IV.1. We study both α proteins and α/β mixed proteins including larger proteins (e.g., 1gdm with 153 residues) and larger decoy sets (e.g., 1eh2 with 2398 conformations). The original decoys are collected from different sets with different features [12]: `lmds` is built and refined by known secondary structure information, `lattice_ssfit` is obtained from lattice models, `4state_reduced` sets have correlation between energy and RMSD, `fisa`, `fisa_casp`, and `fisa_casp3` are collected by Baker's group, and `hg_structal` is generated by homology modeling for globins. All results are averaged over 10 runs.

Table IV.1
Decoy sets studied from the Decoys 'R' Us database [23].

Type	Protein	Residue	Set Name	Original Size	Improved Size	
					Avg.	Std.
α/β	1fca	55	`lattice_ssfit`	1994	2024.20	15.22
	4pti	58	`lmds`	334	358.50	29.86
	1igd	61	`lmds`	501	512.40	6.92
	1sn3	65	`4state_reduced`	660	630.20	7.92
	1ctf	68	`4state_reduced`	630	592.90	12.41
	4icb	76	`fisa`	500	591.70	7.46
	1eh2	79	`fisa_casp3`	2398	2568.10	13.08
α	1r69	63	`4state_reduced`	676	753.60	5.12
	2cro	65	`lmds`	501	588.20	7.28
	1nkl	78	`lattice_ssfit`	1376	1575.90	17.50
	1jwe	114	`fisa_casp`	1407	1768.10	4.11
	1ash	147	`hg_structal`	30	37.00	0.45
	lgdm	153	`hg_structal`	30	34.00	1.41

Sample Set Generation

Recall from Chapter III that we select the desired sample set size as the one that maximizes the z-score rate of change over various set sizes. For each protein, the entire process of sample set generation and decoy selection is carried out for different sample set sizes. We use the hybrid sampling scheme to generate samples. Figures IV.1 and IV.2 display the z-scores and their rates of change for different sample set sizes for α/β proteins and α proteins, respectively. The set with the highest improvement score rate of change (denoted by the triangle) is used for all subsequent experiments.

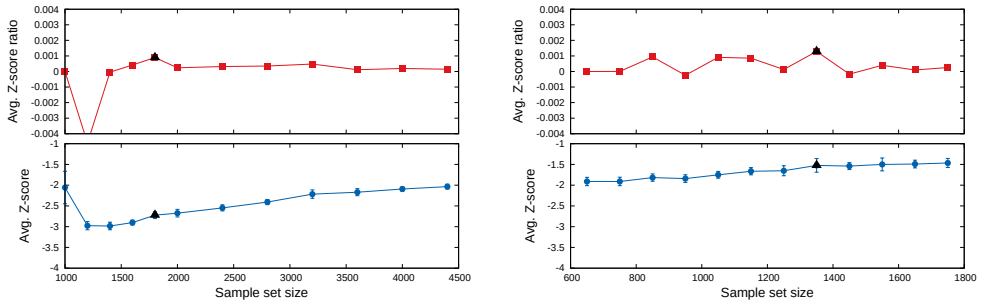
In most cases, we find a peak in z-score rate of change. This peak will be selected as the sample set size for generating candidate decoy structures. For protein 1ash whose original decoy set size is small (30 structures), the sample set size is selected as the second highest z-score ratio within double of the original size because the sample set size with the maximum ratio is too small of a set size.

Decoy Selection

After we compute the desired sample set size, we perform our decoy selection methods (see Chapter III). The original decoy set D and the sample set S can be broken down into four subsets:

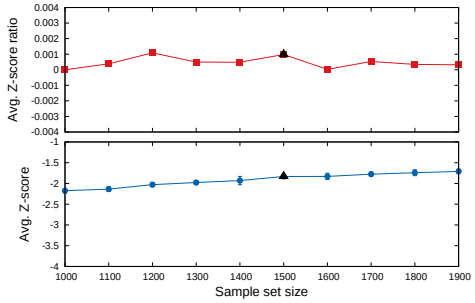
- redundant decoy structures D_D from D,
- viable decoy structures D_V from D,
- redundant sampled structures S_D from S, and
- viable sampled structures S_V from S.

Figure IV.3 shows the relationship of these four subsets. The final improved decoy set is $D_V \cup S_V$ and is a combination of two operations: removing redundant structures from D (yielding D_V) and adding new structures from S (yielding S_V).

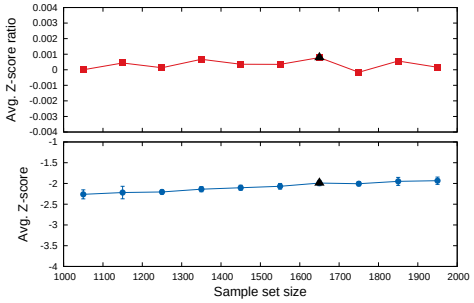


(a) 1fca

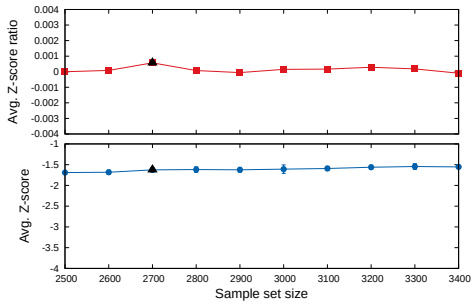
(b) 4pti



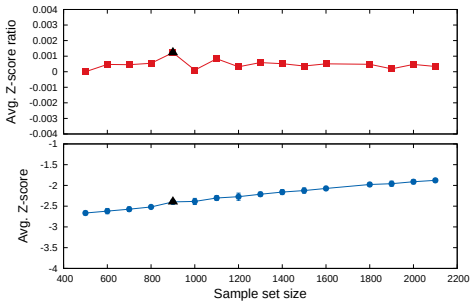
(c) 1igd



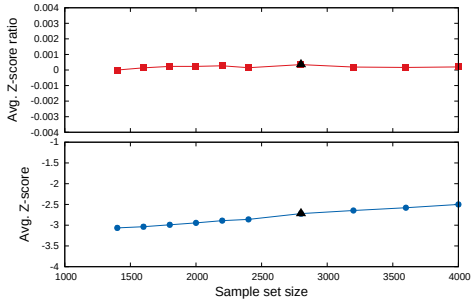
(d) 1sn3



(e) 1ctf

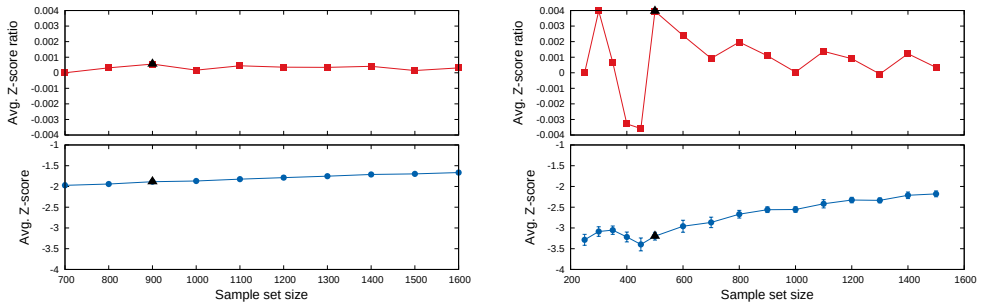


(f) 4icb



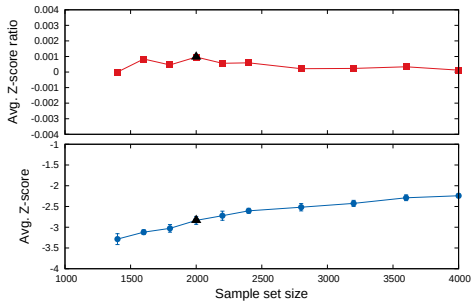
(g) 1eh2

Fig. IV.1. Improvement scores (below in blue) and their rate of change (above in red) for each α/β protein. The final selected sample set size has the peak in the rate of change (denoted by the triangle).

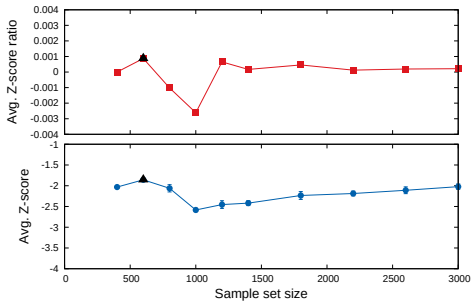


(a) 1r69

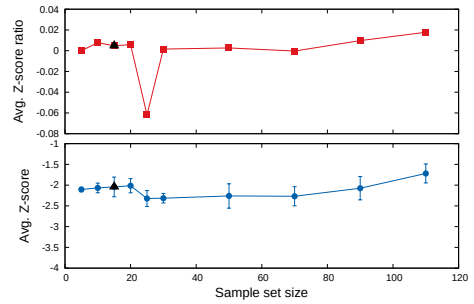
(b) 2cro



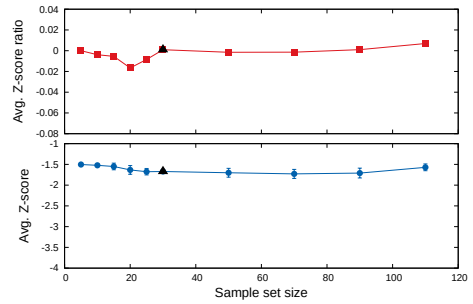
(c) 1nkl



(d) 1jwe



(e) 1ash



(f) 1gdm

Fig. IV.2. Improvement scores (below in blue) and their rate of change (above in red) for each α protein. The final selected sample set size has the peak in the rate of change (denoted by the triangle).

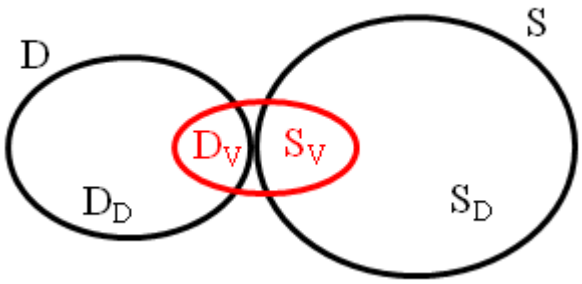
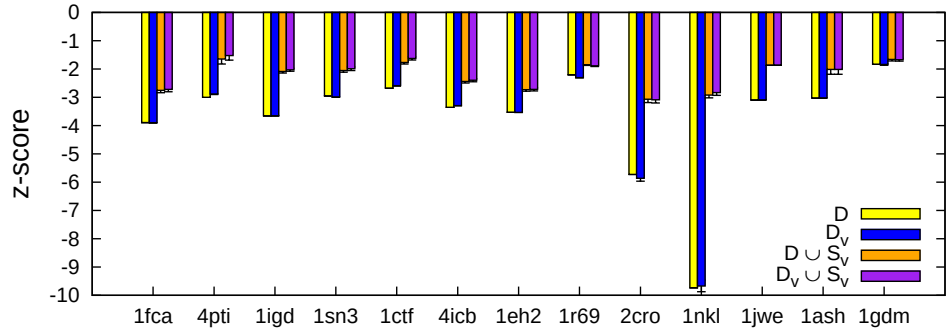


Fig. IV.3. The relationship between each of the subsets D_D, D_V, S_D and S_V from the original decoy set D and the sample set S. The final set, in red, is $D_V \cup S_V$.

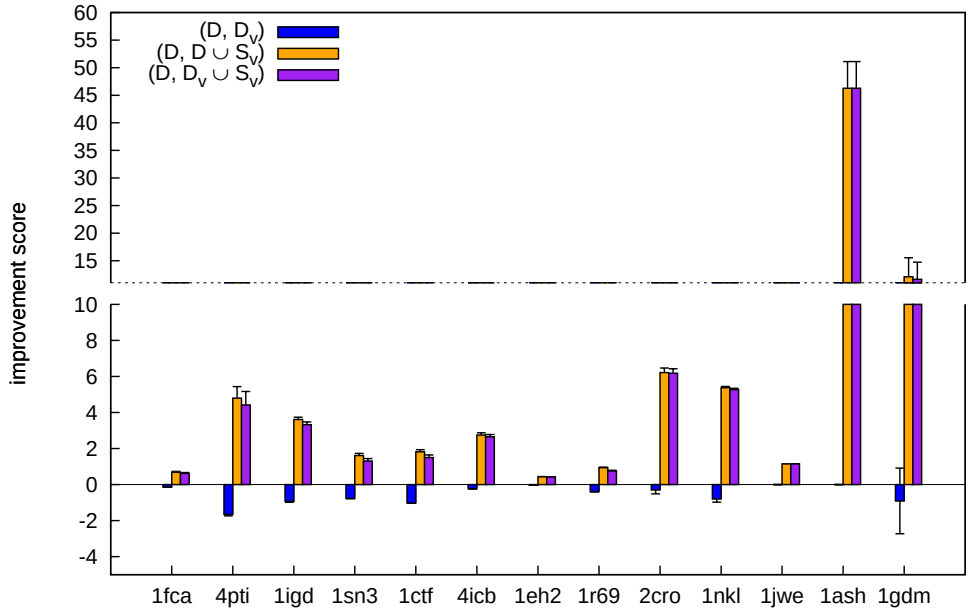
Figure IV.4 summarizes the resulting z-score, improvement score, and minimum distance value for each protein. For each metric, we show the contribution from each operation (removing redundant decoys (D_V) and adding new samples ($D \cup S_V$)) and from their combination ($D_V \cup S_V$).

When the z-score approaches zero, the native structure energy is harder to distinguish among the energies of the other structures in the set. For every protein in Figure IV.4(a), the z-scores of D and D_V are very similar. Thus, simply removing structures does not greatly impact the z-score. However, once we add new structures from our sampling approach ($D \cup S_V$), the z-score drops drastically with comparable z-scores to the final set ($D_V \cup S_V$). Therefore, the main contributors to z-score improvement are the structures generated by our sampling approach.

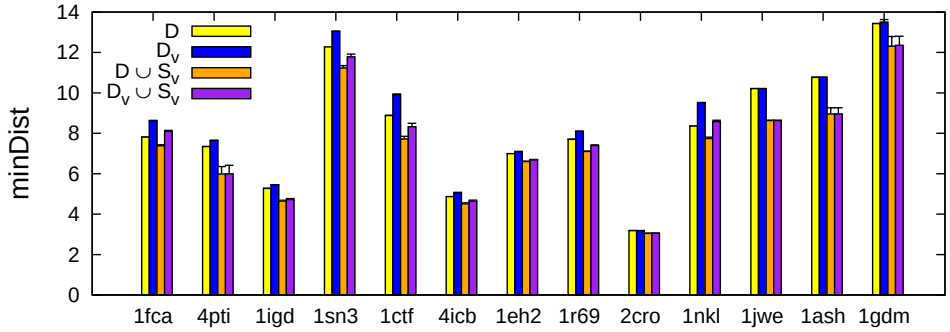
Recall that the improvement score shows the change in z-score per sample between two sets. A higher value indicates that the change (either structure addition, removal, or both) has a greater impact on the z-score. Figure IV.4(b) displays the improvement scores across all tested proteins. We again see that adding structures provides a decoy set with better quality than simply removing redundant structures. Proteins 1ash and 1gdm with the smallest original sets show the largest improvement scores.



(a) Z-Score



(b) Improvement Score



(c) Minimum Distance

Fig. IV.4. Resulting metrics of improved decoy sets and their subsets.

The last metric we examine is the minimum distance between neighboring structures which indicates the diversity of the structures. A larger distance signifies greater structural diversity and implies a greater ability to fool different scoring functions. Figure IV.4(c) shows how this metric changes for each operation. As expected, when decoys are removed (D_V), the minimum distance increases, and when adding decoys ($D \cup S_V$), the minimum distance decreases. For all proteins studied, the minimum distance isn't affected significantly by adding decoys ($D \cup S_V$) implying that they are informative structures.

Set Distribution

Table IV.2 shows the contributions from each type of sample (e.g., from the original decoy set, from uniform sampling, etc.) to the final improved decoy set. In the final improved decoy set, most samples come from the original set (D_V) and the remaining largely come from native bias sampling. Uniform sampling and decoy bias sampling play a very small role because they typically fail to pass the filters employed in the decoy selection phase, either because their energies are too high or they are too similar to structures already in the final set.

Figures IV.5 and IV.6 show the potential vs. CαRMSD from the native structure distribution for α/β and α proteins, respectively. In each plot, three subsets are displayed: D_V, D_D and S_V. We see that in general, the added structures have lower energies and cover a wider range of CαRMSD than the original structures. While most added structures are located near the native structure, there are also some structures with high CαRMSD. These structures are especially valuable because they could be located in a local minima of the energy landscape and may be more likely to fool scoring functions.

Improved Decoy Sets in Practice

Here we assess the ability of our improved decoy sets to "fool" a modern scoring function. In protein structure prediction, scoring functions are often used to guide the search for the native structure. Thus, they must be able to accurately detect the native structure from a

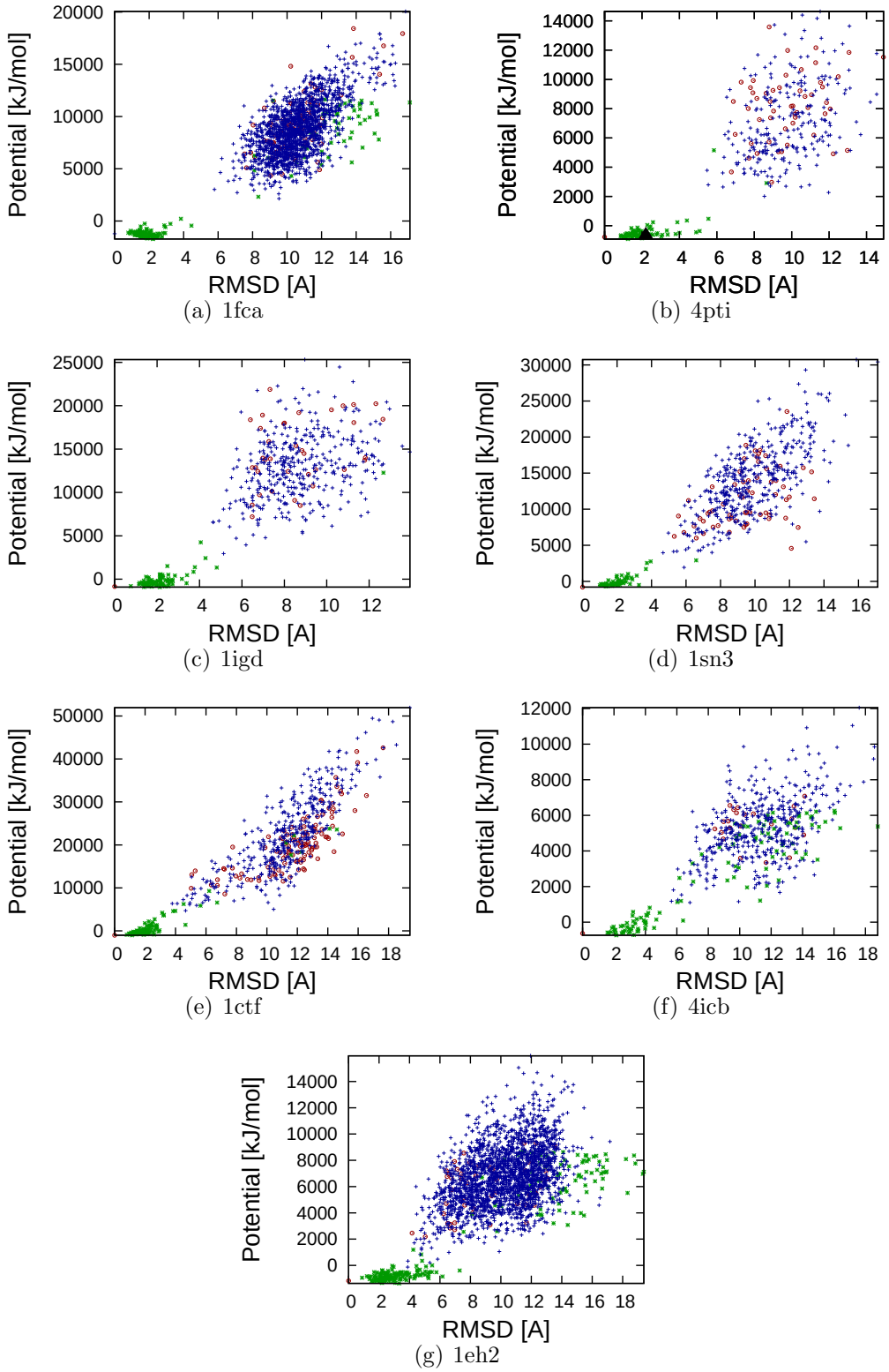


Fig. IV.5. Plots of potential vs. CαRMSD for D_D (red circles), D_V (blue '+'s), and S_V (green '*'s).

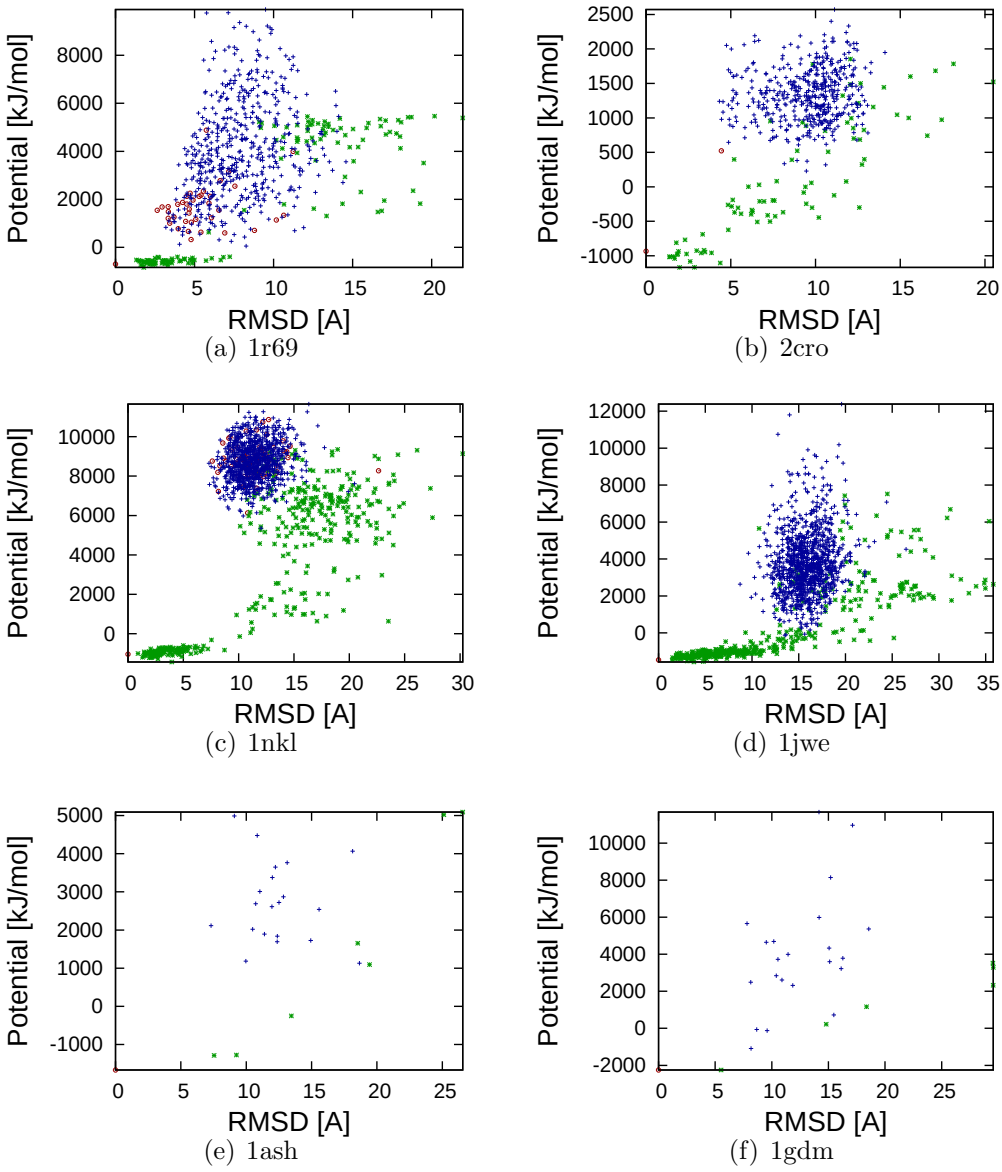


Fig. IV.6. Plots of potential vs. CαRMSD for D_D (red circles), D_V (blue '+'s), and S_V (green '*'s).

Table IV.2
Set distributions for the initial generated samples S and the final improved decoy sets $D_V \cup S_V$ from Table IV.1.

Type	Protein	D_D/D (%)	% Samples in S			% Samples in $D_V \cup S_V$			
			Uniform	Native Bias	Decoy Bias	Uniform	Native Bias	Decoy Bias	D_V
α/β	1fca	6.37	10.17	89.45	0.38	0.00	8.49	0.00	91.51
	4pti	16.83	1.70	98.13	0.16	0.00	22.48	0.00	77.49
	1igd	11.28	11.90	88.00	0.10	0.00	13.25	0.00	86.75
	1sn3	13.62	5.76	94.04	0.20	0.00	9.54	0.00	90.46
	1ctf	21.95	7.01	92.73	0.26	0.00	17.07	0.00	82.93
	4icb	4.94	7.55	92.40	0.05	0.00	19.67	0.00	80.33
	1eh2	2.16	14.34	85.62	0.04	0.00	8.64	0.00	91.36
α	1r69	6.52	9.59	90.27	0.14	0.00	16.12	0.00	83.85
	2cro	0.18	5.15	94.55	0.30	0.00	14.84	0.03	85.02
	1nkl	10.76	8.94	90.43	0.63	0.00	22.07	0.00	77.92
	1jwe	0.00	0.07	99.92	0.02	0.00	20.42	0.01	79.58
	1ash	0.00	0.00	100.00	0.00	0.00	18.92	0.00	81.08
	1gdm	0.67	0.00	100.00	0.00	0.00	12.35	0.00	87.65

set of possible candidates. Qualitative Model Energy ANalysis (QMEAN) [4] is a composite scoring function incorporating several different structural descriptors including local geometry features for discriminating native-like torsional angles from others, secondary structure features for long-range interactions, burial status, and solvent accessibility. QMEAN showed a statistically significant improvement over 5 other well-established scoring functions on decoy sets compiled from molecular dynamics simulations and CASP competition predictions.

Table IV.3 compares the number of structures QMEAN ranked higher than the native state between the original Decoys 'R' Us dataset [23] and our improved decoy dataset. The QMEAN webserver was used to generate rankings [3]. In 3 out of 13 proteins studied, our improved decoys sets were able to produce more structures that "fooled" the scoring function than the original set. Thus, even on a sophisticated, modern scoring function, our improved decoy sets are able to indicate areas of weakness in the scoring function. Note that our improved sets are never worse than the original sets. This means that their quality does not decrease after we remove the structures in D_D.

We also examine the structural differences between the native structure and the decoy with the highest QMEAN score for the 4pti set (denoted by the triangle in Figure IV.5(b)). Figure IV.7 shows the superimposition between the native structure (displayed in green) and the highest QMEAN ranked decoy (displayed in blue) for 4pti. The decoy has an extra α helix (top right) and shorter β sheets which cause it to score higher in QMEAN.

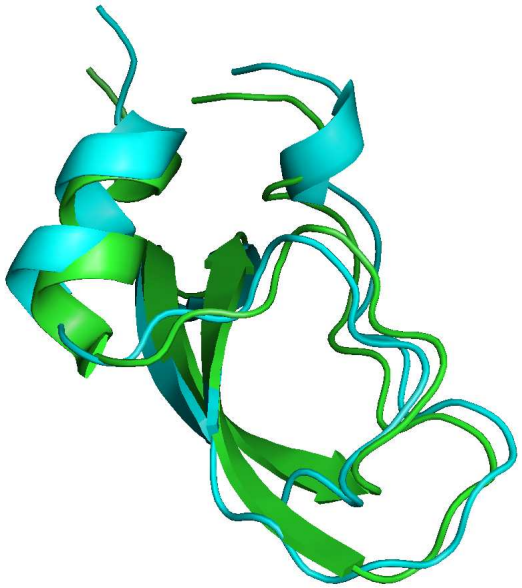


Fig. IV.7. Superimposing the native structure (shown in green) and the decoy scored the highest in QMEAN (shown in blue) for 4pti by PyMOL [11]. The decoy has an extra piece of secondary structure causing it to score higher on QMEAN.

Table IV.3
Comparison of the number of structures ranked higher than the native state by the QMEAN scoring function [4].

Type	Protein	# Structures Ranked Higher than Native		
		Original	Improved	Impr. - Orig.
α/β	1fca	0	0	0
	4pti	0	46	46
	1igd	0	0	0
	1sn3	0	0	0
	1ctf	0	9	9
	4icb	0	0	0
	1eh2	0	6	6
α	1r69	0	0	0
	2cro	0	0	0
	1nkl	0	0	0
	1jwe	7	7	0
	1ash	0	0	0
	1gdm	0	0	0

CHAPTER V

CONCLUSION

We describe a new method for evaluating and improving the quality of decoy databases. Our method removes redundant structures and generates new low energy structures in varied locations on the energy landscape resulting in higher quality decoy sets that are more likely to fool the scoring functions of modern protein folding algorithms.

We tested our approach on 13 different decoy databases of varying size and type and showed significant improvement over the original set. Interestingly, most of the improvement came from adding structures not originally covered by the set indicating a capacity to fool more scoring functions. We also show that our improved databases produced a greater number of structures ranked more native-like by a popular modern scoring function than the original databases for many of the proteins studied.

In the future, we plan to implement a web service to improve user-submitted decoy databases. Our hope is that others can use these improved databases to develop better protein folding algorithms and more accurate folding simulations.

REFERENCES

[1] N. M. Amato, K. A. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *J. Comput. Biol.*, 10(3-4):239–255, 2003. Special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2002.

[2] N. M. Amato and G. Song. Using motion planning to study protein folding pathways. *J. Comput. Biol.*, 9(2):149–168, 2002. Special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2001.

[3] P. Benkert, M. Künzli, and T. Schwede. QMEAN server for protein model quality estimation. *Nucleic Acids Res.*, 37:W510–W514, 2009.

[4] P. Benkert, S. C. E. Tosatto, and D. Schomburg. QMEAN: A comprehensive scoring function for model quality assessment. *Proteins*, 71:261–277, 2008.

[5] R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. M. Strauss, and D. Baker. Rosetta in CASP4: progress in ab initio protein structure prediction. *Proteins Struct. Funct. Bioinf.*, Suppl 5:119–126, 2001.

[6] B. Brooks, R. Bruccoleri, B. Olafson, D. States, S. Swaminathan, and M. Karplus. Charmm: a program for macromolecular energy, minimization and dynamics calculations. *Journal of Computational Chemistry*, 4:187–217, 1983. http://yuri.harvard.edu/.

[7] H. S. Chan and K. A. Dill. Protein folding in the landscape perspective: Chevron plots and non-arrhenius kinetics. *Proteins: Structure, Function, and Bioinformatics*, 30(1):2–33, 1998.

[8] F. Chiti and C. M. Dobson. Protein misfolding, functional amyloid, and human disease. *Annu. Rev. Biochem.*, 75:333–366, 2006.

[9] D. G. Covell. Folding protein α-carbon chains into compact forms by Monte Carlo methods. *Proteins: Struct. Funct. Bioinf.*, 14(3):409–420, 1992.

[10] D. Cozzetto, A. Kryshtafovych, K. Fidelis, J. Moult, B. Rost, and A. Tramontano. Evaluation of template-based models in CASP8 with standard measures. *Proteins Struct. Funct. Bioinf.*, 77 Suppl 9(S9):18–28, 2009.

[11] W. DeLano. The pymol molecular graphics system (2002). *DeLano Scientific, Palo Alto, CA, USA.*, 2002.

[12] F. Fogolari, L. Pieri, A. Dovier, L. Bortolussi, G. Giugliarelli, A. Corazza, G. Esposito, and P. Viglino. Scoring predictive models using a reduced representation of proteins: model and energy definition. *BMC Structural Biology*, 7(15), 2007.

[13] J. Handl, J. Knowles, and S. C. Lovell. Artefacts and biases affecting the evaluation of scoring functions on decoy sets for protein structure prediction. *Bioinformatics*, 25(10):1271–1279, 2009.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[15] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, August 1996.

[16] R. Kolodny and M. Levitt. Protein decoy assembly using short fragments under geometric constraints. *Biopolymers*, 68(3):278–285, 2003.

[17] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 473–479, 1999.

[18] M. Levitt. Protein folding by restrained energy minimization and molecular dynamics. *J. Mol. Biol.*, 170:723–764, 1983.

[19] M. Mirzaie, C. Eslahchi, H. Pezeshk, and M. Sadeghi. A distance-dependent atomic knowledge-based potential and force for discrimination of native structures from decoys. *Proteins Struct. Funct. Genet.*, 77(2):454–463, 2009.

[20] K. Molloy and A. Shehu. Biased decoy sampling to aid the selection of near-native protein conformations. In *BCB*, pages 131–138. ACM, 2012.

[21] J. Moult, K. Fidelis, A. Kryshtafovych, B. Rost, T. Hubbard, and A. Tramontano. Critical assessment of methods of protein structure predictionround vii. *Proteins Struct. Funct. Bioinf.*, 69(S8):3–9, 2007.

[22] C. A. Rohl, C. E. M. Strauss, K. M. S. Misura, and D. Baker. Protein structure prediction using rosetta. *Methods Enzymol.*, 383:66–93, 2004.

[23] R. Samudrala and M. Levitt. Decoys 'R' Us: A database of incorrect conformations to improve protein structure prediction. *Protein Sci.*, 9(7):1399–1401, 2008.

[24] V. Sobolev, T. Moallem, R. Wade, G. Vriend, and M. Edelman. Casp2 molecular docking predictions with the ligin software. *Proteins*, 1:210–214, 1997.

[25] G. Song, S. Thomas, K. Dill, J. Scholtz, and N. Amato. A path planning-based study of protein folding with a case study of hairpin formation in protein G and L. In *Proc. Pacific Symposium of Biocomputing (PSB)*, pages 240–251, 2003.

[26] M. J. Sternberg. *Protein Structure Prediction*. OIRL Press at Oxford University Press, 1996.

[27] A. Subramani, P. A. DiMaggio, and C. A. Floudas. Selecting high quality protein structures from diverse conformational ensembles. *Biophys. J.*, 97(6):1728–1736, 2009.

[28] S. Sun. Reduced representation model of protein structure prediction: statistical potential and genetic algorithms. *Protein Sci.*, 2(5):762–785, 1993.

[29] S. Thomas, X. Tang, L. Tapia, and N. M. Amato. Simulating protein motions with rigidity analysis. In *Proceedings of the 10th annual international conference on Research in Computational Molecular Biology*, RECOMB'06, pages 394–409, Berlin, Heidelberg, 2006. Springer-Verlag.

[30] J. Tsai, R. Bonneau, A. V. Morozov, B. Kuhlman, C. A. Rohl, and D. Baker. An improved protein decoy set for testing energy functions for protein structure prediction. *Proteins Struct. Funct. Bioinf.*, 53(1):76–87, 2003.

[31] P. Weiner and P. Kollman. Amber: Assisted model building with energy renement, a general program for modeling molecules and their interactions. *J. Comp. Chem.*, 2:287–303, 1981.

[32] A. Zemla. LGA: a method for finding 3D similarities in protein structures. *Nucleic Acids Res.*, 31(13):3370–3374, 2003.