# Improving Decoy Databases for Protein Folding Algorithms

Aaron Lindsey*, Hsin-Yi (Cindy) Yeh*, Chih-Peng Wu*, Shawna Thomas*, Nancy M. Amato*
* Parasol Lab, Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA
Email: {alindsey, hyeh, chinuy, sthomas, amato}@cse.tamu.edu

***Abstract*—Predicting protein structures and simulating protein folding are two of the most important problems in computational biology today. Simulation methods rely on a scoring function to distinguish the native structure (the most energetically stable) from non-native structures. Decoy databases are collections of non-native structures used to test and verify these functions.**

**We present a method to evaluate and improve the quality of decoy databases by adding novel structures and removing redundant structures. We test our approach on 17 different decoy databases of varying size and type and show significant improvement across a variety of metrics. We also test our improved databases on a popular modern scoring function and 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.**

## I. Introduction

Two important problems in computational biology are predicting protein structures and simulating protein folding motions. The protein's most energetically stable structure, 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 and possibly lead to many devastating diseases. 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 [21] is a widely studied area. One notable method is Rosetta [18] which uses a simplified model to predict the low-resolution protein structure. In response to increased research in protein structure prediction, the CASP [16] competition emerged as a platform to test structure prediction methods. Molecular dynamics [14] and Monte Carlo simulations [7] 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.

Decoy databases have been developed to test and verify these scoring functions [19]. 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 a set of incorrect protein structures. 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 [19, 16, 23]. 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.

**Contribution.** This work presents a method for evaluating the quality of decoy databases and improving them by adding novel structures and removing redundant structures. Our specific contributions are as follows:

- We test on 17 different decoy databases 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.
- We test our improved databases on QMEAN [3], a popular modern scoring function, and show that they contain a great number of structures ranked more native-like than the actual native state than the original databases.

## II. Related Work

In this section 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.

### A. Protein Models

A protein is composed of a chain of amino acids that determine its function. Amino acids are distinguished by their side chain. When hydrogen bonds form between atoms on the protein backbone, secondary structures can develop. $\alpha$ helices and $\beta$ sheets make up the majority of secondary structures.

The most accurate protein model is the all-atoms model. However, in many cases the all-atoms model is too computationally expensive, particularly for larger proteins. Therefore, some coarse-grained models [17, 1, 9] have been developed to ease the computational complexity by ignoring some detail information. For example, the Gaussian Network Model (GNM) [17] models amino acids as beads connected by elastic strings. Lattice models [9] constrain the protein as a rigid lattice and each amino acid is represented as a bead on the lattice. Another coarser-grained approach models a protein as a series of $\phi$ and $\psi$ 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 [21]. 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 $C_\beta$ position. This model has been successfully used to simulate the correct order of large folding events for several small proteins [1].

### B. Energy Functions

The protein's atoms interact with each other and 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 are called all-atoms functions, e.g., CHARMM [6] and AMBER [24]. These are the most accurate since they consider all possible interactions. However, they are computationally expensive and infeasible for many large proteins.

Instead of modeling all possible interactions, coarse-grained functions only consider side chain contributions to approximate the potential energy. In the $\phi - \psi$ model, each side chain is modeled as a sphere located at the $C_\beta$ atom. If two side chains are too close (i.e., less than 2.4 Å, the conventional van der Waals contact distance), the conformation is rejected [14]. Otherwise, the energy may be calculated as:

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

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 [14]. $E_{hp}$ represents the hydrophobic effect. This energy function has been shown to produce folding simulation results similar to an all-atoms function in a fraction of the time [20] and is the energy function used in these results.

### C. Decoy Databases

Decoys are computer-generated protein structures. Decoy databases have been used to improve the accuracy of scoring functions [10, 22]. 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 [19], (ii) the Rosetta set [23], and (iii) the Critical Assessment of Protein Structure Prediction (CASP) set [16].

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. It 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 $C_\alpha$ positions in the predicted model and the target structure are calculated and a score is assigned showing how similar the prediction is compared to the target [25].

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 [23]. Other work uses a library of short fragments to generate protein decoys by assembling them together given the protein's geometric constraints [12]. Most assembled proteins are 6Å from the native structure. Fragments of varying lengths are used in [15] 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.

### D. Sampling Conformations

Algorithms in the field of motion planning and robotics use sampling-based methods to generate valid robot configurations. Some examples include the Probabilistic Roadmap (PRM) method [11] and Rapidly-Exploring Random Trees (RRT) [13]. Both of these strategies rely on a sampling method to find valid configurations 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 $\phi$ and $\psi$ angle. The conformation $q$ is accepted based on its potential energy $E(q)$ with the following probability:

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

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 $\phi$ and $\psi$ 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. Thus it should be used in conjunction with other sampling methods.

To improve the coverage in the native energy basin, some sampling strategies bias sampling around the known native structure. Gaussian sampling [1] selects values for each $\phi$ and $\psi$ angle from a set of normal distributions centered around the native structure. Iterative Gaussian sampling [2] 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 sampled conformations from prior iterations.

## III. Methods

In this section 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.

### *A. 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 [23]. 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:

$$\textsc{ZScore}(D) = \frac{\mathrm{E}(D.\text{native}) - \mathrm{E}_{avg}(D)}{\mathrm{E}_{std}(D)} \tag{3}$$

where $\mathrm{E}(d)$ is the energy of a structure $d$, $\mathrm{E}_{avg}(D)$ is the average energy of $D$, and $\mathrm{E}_{std}(D)$ 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 improvement indicating that it contains structures with similar energies to the native.

**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. The improvement score between $D$ and $D'$ is:

$$\textsc{Improvement}(D, D') = \frac{\textsc{ZScore}(D')}{|D'|} - \frac{\textsc{ZScore}(D)}{|D|} \tag{4}$$

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 $\delta$.

This metric measures the diversity of structures in the set. As the minimum distance increases, the diversity of structures included in the set also increases. Possible distance functions include Euclidean distance in $\phi - \psi$-space and C$\alpha$RMSD. In this work, we use Euclidean distance.

### *B. 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 Section III-B1. 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 in Section III-B2. Algorithm 1 describes the approach.

**Algorithm 1** $\textsc{ImproveDecoySet}(D, F, n, m)$

```
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 $D'$.
$S \leftarrow \textsc{GenerateSamples}(n, m)$
$D' \leftarrow \textsc{SelectDecoys}(D, D, F) \cup \textsc{SelectDecoys}(D, S, F)$
return $D'$
```

*1) Sample Set Generation:* To improve decoy sets by adding structures, we must first generate a set of samples from which to select. $\textsc{GenerateSamples}(n, m)$ generates $n$ samples by trying the maximum $m$ attempts for each single sample. It uses one of the methods discussed below to generate structures and only the ones that are energetically feasible as given by Equation 2 will be retained.

**Sampling Methods.** We study the following methods:

- *Uniform Sampling.* Returns a structure at a random point on the energy landscape by simply selecting values for each $\phi$ and $\psi$ angle uniformly at random. This will generate many unwanted high-energy structures but provides good coverage of the landscape. Unlike the other methods, it is not biased by any input structure.
- *Sampling with Native Bias.* Returns structures from iterative Gaussian sampling [2]. 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 [2], 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. Here, generated structures have low energies and are not confined to the native energy basin. However, it typically produces samples near the energy basins of 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 [2]. 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. We would like to have an adequate sample set size which can efficiently provide high quality decoy candidates. After some preliminary experiments monitoring how $n$ affects the z-score rate of change, we found that doubling the original set size provides informative structures efficiently.

*2) 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. For the results in this work, we use the energy function in [20].
- *Minimum Distance Filter.* This filter selects structures whose distance to their closest neighbor as determined by some distance metric $\delta$ is greater than some threshold. Here we use Euclidean distance.

SELECTDECOYS$(D, S, F)$ is performed on a decoy set $D$, a sample set $S$, and a set of filters $F$. It first computes the threshold for each filter $f \in F$ by finding average values for $f$ over $D$ and sets the threshold to be one standard deviation below (minimum distance filter) or above (energy filter) the average. 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.

## 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 existing Decoy 'R' Us databases [19] and CASP10 [16] and are listed in Table I. We study both $\alpha$ and $\alpha/\beta$ mixed proteins including larger proteins (e.g., 4fle with 192 residues) and larger decoy sets (e.g., 1eh2 with 2413 conformations). The original decoys are collected from different sets with different features [8]: `lmds` is built and refined by known secondary structure information and an all atom model, `lattice_ssfit` is obtained from lattice models with all-atom energy function, `4state_reduced` sets have correlation between energy and RMSD, `fisa`, `fisa_casp`, and `fisa_casp3` are collected by Baker's group by simulated annealing protocol, `hg_structal` is generated by homology modeling for globins, and `CASP10` is generated from the 2012 CASP submissions. All results are averaged over 10 runs.

TABLE I
DECOY SETS STUDIED FROM DECOYS 'R' US [19] AND CASP10 [16].

| Type | Protein | Residue | Set Name | Original Size | Improved Size | |
|---|---|---|---|---|---|---|
| | | | | | Avg. | Std. |
| $\alpha/\beta$ | 1fca | 55 | `lattice_ssfit` | 2001 | 2024.90 | 21.19 |
| | 4pti | 58 | `lmds` | 334 | 361.80 | 23.12 |
| | 1igd | 61 | `lmds` | 501 | 512.30 | 9.53 |
| | 1sn3 | 65 | `4state_reduced` | 660 | 630.50 | 4.03 |
| | 1ctf | 68 | `4state_reduced` | 630 | 604.50 | 6.25 |
| | 4icb | 76 | `fisa` | 500 | 579.70 | 8.01 |
| | 1eh2 | 79 | `fisa_casp3` | 2413 | 2546.40 | 13.88 |
| | 4fr9 | 141 | `CASP10` | 406 | 496.90 | 4.30 |
| | 4gb5 | 148 | `CASP10` | 217 | 228.90 | 4.89 |
| | 4f54 | 184 | `CASP10` | 322 | 310.90 | 3.67 |
| | 4fle | 192 | `CASP10` | 182 | 183.40 | 0.66 |
| $\alpha$ | 1r69 | 63 | `4state_reduced` | 676 | 744.70 | 9.59 |
| | 2cro | 65 | `lmds` | 501 | 619.20 | 9.11 |
| | 1nkl | 78 | `lattice_ssfit` | 1995 | 2293.80 | 21.41 |
| | 1jwe | 114 | `fisa_casp` | 1407 | 1452.40 | 29.27 |
| | 1ash | 147 | `hg_structal` | 30 | 36.00 | 1.41 |
| | 1gdm | 153 | `hg_structal` | 30 | 33.20 | 1.72 |

### A. Decoy Selection

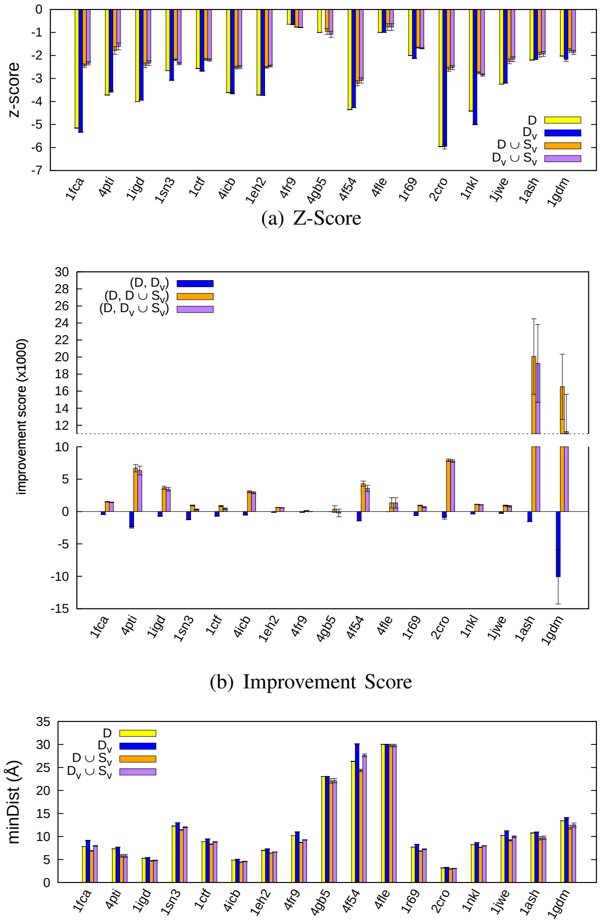
The original decoy set $D$ and the sample set $S$ can be broken down into four subsets:

- 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 $S$.

Table I provides the resulting set sizes after improvement. For all proteins, the resulting size is comparable to the original.

Figure 1 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 other structures in the set. For every protein in Figure 1(a), the z-scores of $D$ and $D_V$ are very similar. Hence, simply removing structures does not greatly impact z-score. Once we add new structures from our sampling approach ($D \cup S_V$), the z-score drops drastically with scores comparable to the final set ($D_V \cup S_V$). Thus, the main contributors to z-score improvement are the structures generated by our sampling approach.



(a) Z-Score

(b) Improvement Score

(c) Minimum Distance

Fig. 1. Resulting metrics of improved decoy sets and their subsets, where $D$ is the original set, $D_V$ is after redundant structures are removed, and $S_V$ is the set of sampled structures to be added.

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 1(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 ones. Proteins 1ash and 1gdm with the smallest original sets show the largest improvement scores. Since the original set sizes are small, removing structures causes significant decrease in the improvement scores.

The last metric we examine is the minimum distance between neighboring structures which indicates set diversity. A larger distance signifies greater structural diversity and implies a greater ability to "fool" different scoring functions. Figure 1(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 is not affected significantly by adding decoys ($D \cup S_V$) implying that they are informative structures.

## B. *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 set of possible candidates. Qualitative Model Energy ANalysis (QMEAN) [3] 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 II compares the number of structures QMEAN ranked higher than the native state between the original decoy dataset and our improved decoy dataset. The QMEAN webserver was used to generate rankings [4]. In 7 out of 17 proteins studied, our improved decoys sets were able to produce more structures that "fooled" the scoring function than the original set, sometimes finding a large number of new structures as in 1eh2. 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 $D_D$.

TABLE II
COMPARISON OF THE NUMBER OF STRUCTURES RANKED HIGHER THAN THE NATIVE STATE BY THE QMEAN SCORING FUNCTION [3].

| Type | Protein | # Structures Ranked Higher than Native | | |
|---|---|---|---|---|
| | | Original | Improved | Impr. - Orig. |
| $\alpha/\beta$ | 1fca | 0 | 8 | 8 |
| | 4pti | 0 | 0 | 0 |
| | 1igd | 0 | 0 | 0 |
| | 1sn3 | 0 | 10 | 10 |
| | 1ctf | 0 | 2 | 2 |
| | 4icb | 0 | 0 | 0 |
| | 1eh2 | 0 | 45 | 45 |
| | 4fr9 | 0 | 0 | 0 |
| | 4gb5 | 0 | 0 | 0 |
| | 4f54 | 0 | 1 | 1 |
| | 4fle | 0 | 0 | 0 |
| $\alpha$ | 1r69 | 0 | 0 | 0 |
| | 2cro | 0 | 0 | 0 |
| | 1nkl | 0 | 3 | 3 |
| | 1jwe | 7 | 13 | 6 |
| | 1ash | 0 | 0 | 0 |
| | 1gdm | 0 | 0 | 0 |

## 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 17 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.

## Acknowledgments

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).

## References

[1] 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.

[2] N. M. Amato, Ken 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.

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

[4] Pascal Benkert, Michael Künzli, and Torsten Schwede. QMEAN server for protein model quality estimation. *Nucleic Acids Res.*, 37:W510–W514, 2009.

[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.E. Bruccoleri, B.D. Olafson, D.J. 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] David G. Covell. Folding protein $\alpha$-carbon chains into compact forms by Monte Carlo methods. *Proteins: Struct. Funct. Bioinf.*, 14(3):409–420, 1992.

[8] 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.

[9] N. Go. Theoretical studies of protein folding. *Annu. Rev. Biophys. Bioeng.*, 12:183–210, 1983.

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

[11] 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.

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

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

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

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

[16] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure predictionround x. *Proteins Struct. Funct. Bioinf.*, 82(S2):1–6, 2014.

[17] A.J. Rader and Ivet Bahar. Folding core predictions from network models of proteins. *Polymer*, 45:659–668, 2004.

[18] Carol A. Rohl, Charlie E. M. Strauss, Kira M. S. Misura, and David Baker. Protein structure prediction using rosetta. *Methods Enzymol.*, 383:66–93, 2004.

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

[20] G. Song, S.L. Thomas, K.A. Dill, J.M. Scholtz, and N.M. 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.

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

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

[23] Jerry Tsai, Richard Bonneau, Alexandre V. Morozov, Brian Kuhlman, Carol A. Rohl, and David Baker. An improved protein decoy set for testing energy functions for protein structure prediction. *Proteins Struct. Funct. Bioinf.*, 53(1):76–87, 2003.

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

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