Journal of Bioinformatics and Computational Biology

Accurate Refinement of Docked Protein Complexes Using Evolutionary Information and Deep Learning

BAHAR AKBAL-DELIBAS, ROSHANAK FARHOODI, MARC POMPLUN, NURIT HASPEL*

Department of Computer Science, University of Massachusetts Boston
100 Morrissey Boulevard, Boston, MA 02125, USA
abakbal@cs.umb.edu,
rfarhood@cs.umb.edu,
marc.pomplun@umb.edu,
**nurit.haspel@umb.edu (Corresponding author)*

Abstract

One of the major challenges for protein docking methods is to accurately discriminate native-like structures from false positives. Docking methods are often inaccurate and the results have to be refined and re-ranked to obtain native-like complexes and remove outliers. In a previous work we introduced AccuRefiner, a machine learning based tool for refining protein-protein complexes. Given a docked complex, the refinement tool produces a small set of refined versions of the input complex, with lower Root-Mean-Square-Deviation (RMSD) of atomic positions with respect to the native structure. The method employs a unique ranking tool that accurately predicts the RMSD of docked complexes with respect to the native structure. In this work we use a deep learning network with a similar set of features and five layers. We show that a properly trained deep learning network can accurately predict the RMSD of a docked complex with 1.40 Å error margin on average, by approximating the complex relationship between a wide set of scoring function terms and the RMSD of a docked structure. The network was trained on 35 unbound docking complexes generated by RosettaDock. We tested our method on 25 different putative docked complexes produced also by RosettaDock for five proteins that were not included in the training data. The results demonstrate that the high accuracy of the ranking tool enables AccuRefiner to consistently choose the refinement candidates with lower RMSD values compared to the coarsely docked input structures.

1. Introduction

Proteins often bind to other proteins to generate higher-order molecular machines, or complexes. Protein complexes play a central role in nearly every cellular process [1]. Since the structure and function of proteins are closely related, it is important to know the structural properties of complexes in order to understand their function in the cell. Experimental

methods to determine complex structures are often slow and costly, and therefore computational methods can be helpful in supplementing and predicting experimental data. Computational docking methods aim to compute the interaction of two or more proteins and predict the way they bind to each other. Docking methods are typically composed of two main stages: The *search stage* generates a large set of candidate complexes using various structural and geometric techniques, and the *ranking stage* ranks the candidates using a scoring function. Various scoring functions rank decoys according to energetic criteria including electrostatic, Van der Waals, and solvent interactions, or similarity to experimental structures [2–7].

The complexes output by computational docking methods are high-scoring candidate complexes, according to the ranking stage, that are expected to be similar to the native bound complex. However, computational docking methods are not accurate. The energetic difference between the native structure and non-native complexes is often small and the scoring functions employed by docking algorithms are often an estimate of the real forces that govern protein-protein interactions. These scoring functions are often not sensitive enough to distinguish near-native candidates from low-energy false positives. Experimental and computational methods for binding site detection can aid in docking and refinement [8–11], but the correct binding site cannot always be determined correctly and sometimes complexes are docked at a completely different site than the binding site. As a result, structures produced by docking programs often disagree with experimental NMR data [12]. CAPRI (Critical Assessment of PRedicted Interactions) rounds reveal an important insight: even the most accurate docking methods still fall short in predicting the correct binding of complexes [2, 13].

For the reasons mentioned above, docking methods are often aided by an additional refinement stage in order to obtain native-like structures and improve the results. Docking refinement methods accept docking candidates as input and aim to produce a set of refined structures with lower energy, better interface packing and smaller Root-Mean-Square Deviation (RMSD) with respect to the native complex. Refinement methods, just like docking, usually involve a two-phase process to achieve this goal. First, an extensive set of refinement candidates are generated using a wide variety of techniques to search the conformational space, including rigid body transformations with flexible fitting that accounts for changes proteins undergo upon binding. Flexible fitting methods include side chain optimization [14], normal-mode analysis [15], molecular dynamics [16], energy minimization [3], Monte Carlo [17], genetic algorithms [18] and more. Subsequently, the refinement candidates are ranked using scoring functions and a subset of candidates that are expected to be most similar to the native conformation is selected . Refinement scoring functions, like docking scoring functions, are also designed to favor conformations with low binding energy, good geometric fit and clusters of conserved amino acids on the binding interface [19].

Despite the additional refinement, computational docking methods still produce a large number of false positives. The inaccuracy of docking and refinement methods can be attributed in part to the difficulty of ranking and scoring functions to model the complex interactions between protein interfaces. The scoring functions aim to detect native-like decoys

in a large set of candidates. However, these scoring functions are often inaccurate. Although state-of-the-art docking algorithms can rank a few near-native conformations among their top solution candidates, highest rank candidates are often false positives [2, 20]. In other words, the ranking suggested by scoring functions is usually not highly correlated with the RMSD between the docking candidates and the native complex. A survey of various scoring functions showed that although some of these functions have meaningful individual components, none of them could predict binding affinity reliably [21]. The docking community agrees that there is a relationship between favorable intermolecular interactions (e.g., Van der Waals, electrostatic, solvation and others) and the similarity of a docked complex to its native form. However, the exact nature of this relationship is unknown. Docking and refinement algorithms formulate this relationship as a weighted sum of selected terms and calibrate their weights using specific training data. Yet, the widespread inaccuracy of ranking functions indicates that the relationship between molecular interaction terms and the RMSD of a conformation may be much more complex than a simple linear function.

Different fields of artificial intelligence (AI) such as signal processing, pattern recognition, and computer vision often approximate complex, nonlinear functions. Neural networks are one of the efficient methods for estimating the functional relationship between a set of input and output variables and are frequently used in the artificial intelligence and machine learning community. By tuning the weights assigned to different processing units (neurons), the neural network gains its ability to recognize complex patterns hidden in the data. Recently more sophisticated types of neural networks, called deep networks, have gained popularity, which are composed of multiple layers of non-linear feature representation. Deep learning or representation learning has become a promising area in machine learning, which has shown great success in the domain of image processing, speech recognition and bioinformatics [22]. Compared to shallow architectures, where only a single layer of nonlinear transformation is adopted, in deep architectures multiple layers of nonlinear mappings are stacked on top of one another to capture the nonlinearities in the representation of training data. These multiple hidden layers build a hierarchy that transforms the raw input to a feature space that is often initially unobservable. As one goes higher in this hierarchy more and more abstract representations of the data become available thanks to the nonlinear processing units used in the previous layers. Nowadays, availability of vast volumes of data at low cost as well as advances in hardware used for data processing enable the efficient training of deep networks. There are several supervised and unsupervised deep architectures for tackling different domain-specific problems such as deep belief networks, convolutional neural nets, stacked auto-encoders and multilayer neural networks. By using sophisticated learning algorithms and taking advantage of large and diverse training datasets, deep networks can accurately approximate complex functions. The backpropagation learning algorithm [23] was the first successful approach to the training of multilayer networks with continuous input and output, and it is still the most widely used neural network learning algorithm.

In our previous work we proposed AccuRefiner [24], a novel method to refine docked protein-protein complexes. AccuRefiner utilizes AccuRMSD [25,26], a ranking tool we developed to predict the RMSD* (RMSD* is used to distinguish predicted values from actual

RMSD values) of a docked refinement candidate with respect to the native structure. The previous version of AccuRMSD used a trained two-layer neural network to approximate the complex relationship between a diverse set of scoring function terms and the RMSD of a docked structure. In this work, we enhance AccuRMSD to employ a deep learning network. Instead of a two-layer neural network, a supervised multilayer neural network is devised to predict the RMSD value of a given protein structure using several features that are shown to be important by multiple docking and refinement algorithms. Also, training and testing of the network are conducted with new proteins that are selected from a benchmark that is widely used in the field as explained in Section 2.2.3. The high accuracy of AccuRMSD enables AccuRefiner to consistently select refinement candidates with lower RMSD values compared to the input of coarsely docked structures.

2. Methods

2.1. *Decoy Generation*

We used protein complexes produced by a docking method as the input for our method. We considered the larger chain of the heterodimer as the receptor and the smaller one was treated as the ligand whose relative position we aim to improve via rigid-body rotations. The search was performed on the vicinity of the input conformation with the objective of lowering the computational cost and to prevent major alterations to the structure. We perturbed the structures by rotating them by a randomly chosen angle within a predefined range around an arbitrary axis passing through the centroid of the structure. Each rotation generated a new decoy. Then we minimized the energy of the decoys for 200 steps, using NAMD [27], in order to resolve local clashes and present small-scale local flexibility (larger scale flexibility will be pursued in a future work). A limited number of minimization steps were applied to relax the complex locally without introducing significant changes to the original conformation. Once the minimization was done, we predicted the RMSD* of each decoy by employing a novel method described in Section 2.2. We ranked the decoys based on RMSD* values and with the aid of these rankings a selection probability was given to each decoy. Then a subset of these decoys were selected as the output.

For each input conformation, 500 decoys were generated by rotations around an arbitrary axis passing through the centroid of the ligand. Since our method assumes a roughly correctly docked conformation as the input and aims at refining it through small rotations without significantly perturbing the overall shape, we considered -5 to 5 degrees a reasonable range for the rotation angle. We chose a rotation axis from the set of all three-dimensional unit vectors in a unit sphere whose center is at the centroid of the chain. A three-dimensional vector V can be represented by two angles: the angle between V and X-axis (α) and the angle between V and Z-axis (β). Then the x, y, z components of V can be expressed as follows: $V_x = \cos\alpha\,; V_y = \sin\alpha\,; V_z = \cos\beta$. The arbitrary rotation axis is then selected among 360×360 three-dimensional unit vectors by randomly choosing α and β values from integers between 1 and 360. For each refinement candidate we randomly selected a number from the range of [1-15] that we used as the number of consecutive rotations to be applied to the input structure.

2.2. *Ranking Decoys*

We employed AccuRMSD [25, 26] to accurately discriminate refinement candidates produced from putative docked protein complexes. AccuRMSD predicts the RMSD* of each refinement candidate with respect to the native conformation. It uses the values of 11 features representing the conformation and then ranks the candidates based on RMSD* values. In the following subsections, we provide an overview of AccuRMSD and its validation on an extensive test set of 12,500 refinement candidates. More details about AccuRMSD can be found in [26]. It should be mentioned that finding ways to improve our decoys, possibly by introducing more flexibility and energy consideration, is part of on-going research.

2.2.1. *Features*

We devised a multilayer neural network to approximate the relationship between 11 different features listed below and the RMSD value of a protein structure. The majority of these features are used as scoring function terms by a wide variety of docking and refinement methods. The previous version of this work [24] included two more features (i.e. the ratios of atoms belonging to charged and aliphatic interface residues to the total interface size) which are excluded here as the same information is implicitly available through other features (i.e. ratios of positively charged, negatively charged and hydrophobic residues on the interface).

- Van der Waals (VdW): The VdW force for interface atoms (defined as the atoms within at most 6Å to the adjacent chain atoms) is computed using a soft Lennard-Jones potential [19].
- Electrostatic: Computed for interface atoms, based on Coulomb's law as explained in [19].
- Conservation: Evolutionary Traces (ET) are based on the finding that residues on functional interfaces are important for correct binding, and more likely to be conserved throughout evolution. Therefore, we define the interface conservation score as: $\sum_{atoms} k \cdot c_i$ where c_i is the ET coverage value of the residue to which the interface atom i belongs. c_i ranges between 0 and 1, where lower values imply higher evolutionary importance [28]. k is -1 if c_i is less than the threshold defined as below; otherwise it is 1.

  $$threshold = \mu - \sigma/2$$

  where μ and σ are the mean and standard deviation of ET coverage values in the chain, respectively. c_i is multiplied with this constant to avoid bias towards conformations with smaller interfaces.
- Interface Conserved Atom Ratio (ICAR): The ratio of the evolutionarily conserved interface atoms to the total interface size. We define atoms belonging to a residue with ET coverage value less than the $threshold$ as conserved.
- Conservation per Interface Atom (CPIA): Average conservation score for interface atoms. While ICAR favors conformations with more conserved interface atoms,

CPIA favors conformations with interface atoms that have higher conservation value (e.g. ET coverage 0.1 vs. 0.4).

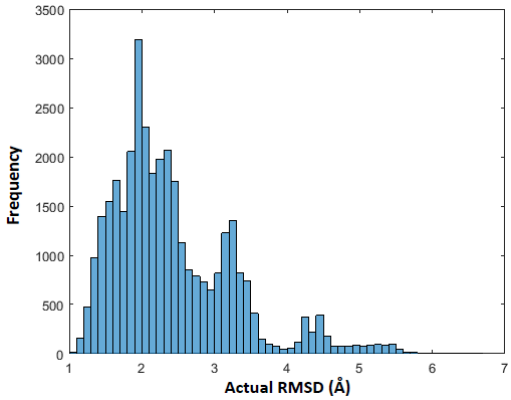
- Protein Category: The numeric representation of the protein category (1:enzyme/inhibitor, 2:antibody). In this work other protein categories including signaling, transferase and cell adhesion have not been investigated.
- The ratios of interface atoms belonging to certain types of residues to the total interface size: Hydrophobic (A, C, G, I, L, M, P, V); Positively Charged (H, K, R); Negatively Charged (D, E); Polar (N, Q, S, T); Aromatic (F, H, W, Y).

2.2.2. *AccuRMSD Neural Network*

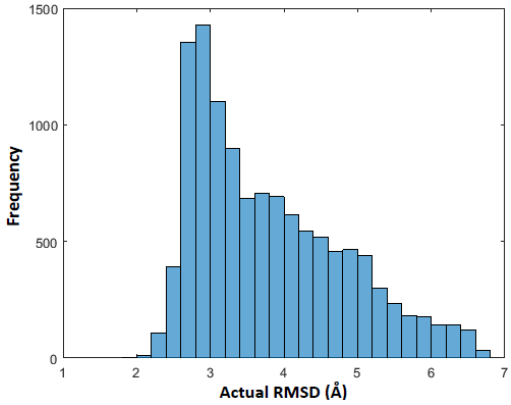
The network has 11 input neurons which receive the input data including 11 features that characterize a protein structure. These features consist of continuous values and were initially normalized to the range between 0 and 1 before being used for training. To determine the optimal architecture of the network we used a fixed set of training data and altered the network depth between 2 to 5, while observing the prediction error. Also, in separate experiments we kept the number of hidden layers fixed and varied the number of nodes in each hidden layer while observing the network performance on our test set. This stage of parameter tuning resulted in a network of 3 layers of hidden units, with 50 neurons in the first hidden layer and 70 and 100 neurons in the next two hidden layers, that gave us the smallest error and fastest convergence. The output layer consisted of a single neuron that produces the predicted RMSD value. Similar to the input values, this output value is in the range between 0 and 1 and needed to be scaled to represent the final predicted RMSD* value. Also, each hidden and output-layer neuron received an additional input that was constantly set to 1 (often referred to as bias), enabling offset adjustment through the corresponding weight during the training phase. The network was trained using the back-propagation learning algorithm to predict the RMSD value of a given structure based on the input values of the 11 selected features. We trained the network using 10-fold cross validation for 300 epochs. In this phase, the dataset was randomly divided into 10 sub-datasets where 9 of these sub-datasets were used for training and the remaining sub-dataset was used for testing. This process was repeated until all 10 sub-datasets were used as the test set once in an iterative fashion. The best performing model was recorded and used for further validation of the new protein structures. The neural network was implemented using the Matlab deep learning toolbox.

2.2.3. *Training Dataset*

We trained AccuRMSD with an extensive dataset, composed of 35,000 samples of 35 unbound dimer proteins listed in the Protein-Protein Docking Benchmark 4.0 [29]: 1B6C, 1EFN, 1EWY, 1FFW, 1CL1, 1GLA, 1GPW, 1GXD, 1H9D, 1US7, 1J2J, 1JTG, 1OC0, 1OYV, 1PVH, 1S1Q, 1T6B, 1XD3, 1YVB, 1Z0K, 1Z5Y, 1ZHH, 1ZHI, 2AST, 2AJF, 2B42, 2FJU, 2HLE, 2HQS, 2J0T, 2O8V, 2OOB, 2VDB, 3DSS and 4CPA. We focused on the dimers from the rigid-body category in this benchmark, and among those we selected the



(a) Training dataset



(b) Refinement candidates

Fig. 1: RMSD distribution of (a) the training dataset and (b) the refinement candidates test set. The RMSD is with respect to the native PDB structure.

proteins for which the corresponding evolutionary trace files existed on the ET Server [30]. For each protein, we created 1,000 docked structures by feeding unbound ligands and receptors to RosettaDock [31]. We then analyzed interfaces of each structure and calculated the values of the network features.

2.2.4. *RMSD Prediction Validation*

The prediction accuracy of AccuRMSD was validated on refinement candidates produced from coarsely docked protein complexes generated by RosettaDock. The refinement candidates were generated from 25 different docked complexes of 5 proteins: 1GL1, 2A9K, 2AYO, 2G77, 2SNI, which are again from Protein-Protein Docking Benchmark 4.0 [29]. For each input docked complex, 500 refinement candidates are created by applying small-scale rigid body rotations. It is worth noting that the test data included a completely dif-

Table 1: Correlation coefficient and prediction error for the RMSD of 25 refinement candidate test cases with respect to the native structure (500 samples for each).

PDB ID : complex#	Correlation	Error (Å)
1GL1 : complex1	0.46	1.62
1GL1 : complex2	0.45	1.53
1GL1 : complex3	0.24	1.30
1GL1 : complex4	0.41	1.31
1GL1 : complex5	0.56	1.31
2A9K : complex1	-0.27	1.89
2A9K : complex2	0.20	1.19
2A9K : complex3	0.09	2.96
2A9K : complex4	0.28	1.56
2A9K : complex5	0.26	1.99
2AYO : complex1	0.42	1.52
2AYO : complex2	0.20	1.59
2AYO : complex3	0.21	1.69
2AYO : complex4	0.41	1.50
2AYO : complex5	0.19	1.60
2G77 : complex1	0.35	1.77
2G77 : complex2	0.29	1.87
2G77 : complex3	0.48	1.96
2G77 : complex4	0.26	1.68
2G77 : complex5	0.48	1.67
2SNI : complex1	0.65	0.41
2SNI : complex2	0.65	0.28
2SNI : complex3	0.75	0.33
2SNI : complex4	0.70	0.30
2SNI : complex5	0.75	0.29
Overall	**0.38**	**1.40**

ferent set of proteins than the training data to avoid any bias in the prediction. The overall error was 1.40 Å while the correlation coefficient between predicted and actual RMSDs was 0.38. Figure 1(b) displays the RMSD distribution of the refinement candidates. Table1 lists the correlation and error values for each test case. One observation is worth noting. As shown in Table1 the smallest RMSD prediction error was achieved for protein 2SNI, which has the largest correlation coefficients among its samples in the testset. As the correlation among samples decreases, the RMSD prediction error tends to increase.

2.3. *Probabilistic Selection of Candidates*

The refinement candidates were ranked based on RMSD* values and a subset of them were selected as the refinement output. Despite the high predictive power of AccuRMSD, RMSD* values rarely correlate strongly with the actual RMSD values as shown in Table 1. Hence, we adopted a probabilistic selection strategy for selecting the refinement solutions among the large set of evaluated candidates. We first rank the decoys based on RMSD* values. Then, we assign a probability of selection to each decoy based on its ranking, as shown in Table 2. Finally, we select 20 decoys and return them as the refinement solutions according to the probability distribution in Table 2. The cumulative probability of selection

Table 2: Selection probability for decoys according to their ranking in RMSD* values. The relative probability is with respect to the selection probability of decoys in the top 20.

Decoy Rank	Relative Probability	Selection Probability
1 – 20	1	0.0266
21 – 40	0.5	0.0133
41 – 80	0.1	0.0027
81 – 120	0.02	0.0005
121 – 400	0.01	0.0003

for the top 40 decoys is about 80%.

3. Results and Discussion

We tested AccuRefiner on 25 different putative docked complexes, generated from RosettaDock. For each coarsely docked input structure we did the following: (i) generated 500 refinement candidates through small-scale rigid body rotations; (ii) predicted the RMSD* of each candidate with respect to the native structure using AccuRMSD; (iii) ranked the candidates based on RMSD* values; (iv) assigned a probability of selection to each candidate based on its relative ranking; and (v) selected 20 decoys randomly as the refinement solutions, using the probability distribution function in Table 2.

Refinement results of our program for 25 different test cases are summarized in Table 3 with actual RMSDs. For each test case, the following information is provided: (a) the RMSD of the input coarsely docked structure with respect to the native conformation; (b) the number of refinement solutions that are better than the input docked structure in terms of the RMSD value with respect to the native; (c) maximum improvement achieved through refinement in comparison to RMSD of the input structure with respect to the native form; (d) maximum improvement that was possible among 500 refinement candidates (if AccuRefiner could select the lowest-RMSD candidate); (e) RMSD values of the best five refinement solutions; and (f) the RMSD value of the best refinement candidate among 500 decoys evaluated. In addition, several examples of the docked input structure, the best refinement solution, and the corresponding native conformation are depicted in Figure 2 for visual comparison of the interface packing. Several important observations are worth noting about the refinement results:

- Despite relatively low-RMSD starting points, AccuRefiner consistently produces better solutions than the input coarsely docked structure. On average, 30% (6 out of 20) of the refinement solutions have lower RMSD than the input docked structures with respect to the native conformation. The number of lower-RMSD refinement solutions can go up to 60% (12 out of 20).
- Even though AccuRefiner yields better solutions than the input structure, the magnitude of improvement (in RMSD) varies from case to case. In all but one case, at least one of the top five solutions has lower RMSD than the input docked complex.
- Lowest-RMSD Candidate column in Table 3 lists the smallest-RMSD decoy

Table 3: Refinement results of AccuRefiner on 25 different docked protein complexes generated by RosettaDock. The Lower-RMSD Solutions column specifies how many of the 20 solutions produced by AccuRefiner are better than the input docked structure in terms of the RMSD values. The Lowest-RMSD Candidate column lists the RMSD values of the best structures among the 500 decoys generated for each input docked complex. All RMSD values are actual RMSDs with respect to the native structures.

PDB ID : complex#	Input RMSD	Improved Solutions	Max. Improvement Achieved	Max. Improvement Among Candidates	Refinement Solution 1	Refinement Solution 2	Refinement Solution 3	Refinement Solution 4	Refinement Solution 5	Lowest-RMSD Candidate
1GL1 : complex1	3.03	9	9%	21%	2.75	2.82	2.85	2.87	2.89	2.39
1GL1 : complex2	3.00	6	11%	16%	2.66	2.72	2.84	2.84	2.92	2.51
1GL1 : complex3	2.59	1	5%	8%	2.46	2.62	2.66	2.71	2.76	2.38
1GL1 : complex4	2.89	4	3%	9%	2.81	2.83	2.84	2.84	2.95	2.62
1GL1 : complex5	2.73	5	13%	13%	2.38	2.41	2.56	2.61	2.65	2.38
2A9K : complex1	4.50	4	5%	15%	4.26	4.29	4.34	4.38	4.54	3.81
2A9K : complex2	3.17	11	20%	20%	2.54	2.61	2.70	2.71	2.78	2.54
2A9K : complex3	5.00	6	5%	9%	4.74	4.79	4.81	4.83	4.91	4.57
2A9K : complex4	3.80	4	13%	14%	3.29	3.40	3.62	3.63	3.93	3.26
2A9K : complex5	4.01	12	7%	10%	3.74	3.77	3.79	3.82	3.84	3.62
2AYO : complex1	3.20	9	9%	11%	2.91	2.96	3.02	3.05	3.08	2.84
2AYO : complex2	3.03	6	14%	15%	2.60	2.62	2.71	2.84	2.84	2.59
2AYO : complex3	3.27	8	12%	16%	2.89	2.90	2.99	3.11	3.16	2.74
2AYO : complex4	3.11	3	2%	5%	3.04	3.05	3.06	3.11	3.13	2.94
2AYO : complex5	2.96	6	12%	12%	2.60	2.62	2.68	2.75	2.82	2.60
2G77 : complex1	2.56	11	18%	23%	2.09	2.25	2.26	2.33	2.37	1.96
2G77 : complex2	2.60	5	5%	12%	2.48	2.50	2.54	2.55	2.57	2.3
2G77 : complex3	2.89	1	3%	4%	2.80	2.80	2.90	2.91	2.97	2.78
2G77 : complex4	2.41	8	8%	10%	2.21	2.28	2.31	2.31	2.32	2.18
2G77 : complex5	2.43	0	0%	1%	2.51	2.63	2.63	2.67	2.78	2.41
2SNI : complex1	2.59	3	1%	4%	2.56	2.57	2.57	2.59	2.65	2.49
2SNI : complex2	2.75	4	1%	2%	2.71	2.72	2.73	2.74	2.75	2.69
2SNI : complex3	2.63	9	2%	3%	2.58	2.58	2.58	2.59	2.59	2.56
2SNI : complex4	2.73	3	1%	1%	2.70	2.70	2.71	2.74	2.75	2.69
2SNI : complex5	2.70	10	3%	3%	2.63	2.63	2.64	2.64	2.64	2.61

among 500 refinement candidate structures generated for each input docked structure. Comparison of the best refinement solutions and the lowest-RMSD candidates evaluated by AccuRefiner reveals the high accuracy of our tool. For 25 different test cases, the average RMSD value of the best refinement solutions chosen by AccuRefiner are only 0.1Å higher than the lowest-RMSD candidates evaluated. For some cases, the RMSD difference between the best refinement solution and the lowest-RMSD candidate is less than 0.1Å.

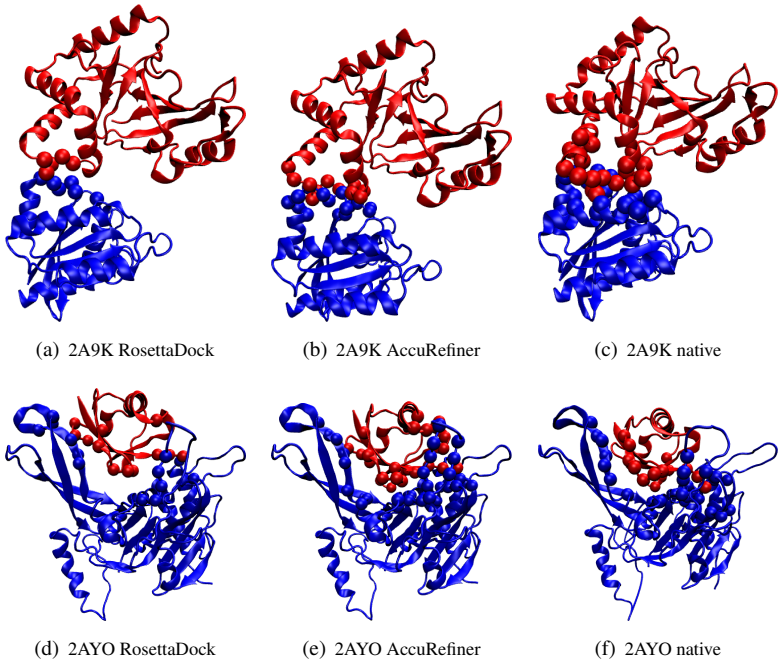
- Comparing the maximum improvement achieved and the maximum improvement possible among 500 refinement candidates for each test case also reveals the high accuracy of the ranking tool of AccuRefiner. For 25 test cases, the maximum RMSD improvement possible among candidates was only 10% on average. De-

spite such limited opportunity for improvement, AccuRefiner accomplished 7% improvement on average relative to RMSD values of the input coarsely docked structures.

- Out of these 25 test cases, three of them seem like outliers compared to others. For 1GL1:complex3 and 2G77:complex3, AccuRefiner was able to produce only one solution that was better than the input structure in terms of RMSD while for 2G77:complex5, no solutions had lower RMSD than the input. Further analysis of these three cases reveals the following insight: the room for possible improvement was extremely small for 2G77:complex5 as AccuRefiner could produce only five candidates that had lower RMSD values compared to the input (the best candidate's RMSD value was only 0.02 Å less than the input's). Although the room for improvement vas relatively larger for 1GL1:complex3 and 2G77:complex3, the number of candidates with better RMSD values was as low as 30 (i.e., 6% of all 500 candidates).
- Analysis of the refinement results on these 25 test cases also indicate that the decoy generation step of AccuRefiner was not able to generate candidates that were significantly better compared to the input docked structure. On average, the lowest RMSD refinement candidate was only 10% better in terms of RMSD. One possible explanation for this could be that the input docked structures had relatively low RMSD with respect to their native forms. However, considering the demonstrated high accuracy of its ranking tool, AccuRefiner could yield much better solutions if it could produce relatively better refinement candidates in the first place.
- Finally, as shown in Figure 2, the refinement solutions produced by AccuRefiner are better than the input docked structures not only in terms of the RMSD values with respect to the native conformation but also visibly better in terms of interface packing.

4. Conclusions

We presented AccuRefiner, a novel docking refinement method that accepts docked conformations and returns similar conformations with better RMSD with respect to the native structure. The novelty of our method lies in the unconventional approach we took for evaluating the solution candidates. We formulated a backpropagation neural network to approximate the complex, non-linear relationship between a large set of features used by different scoring functions and a structure's similarity to its native conformation. We are able to predict the RMSD* of a refinement candidate with less than 1.40Å error margin and rank the candidates accurately based on RMSD* values. The high accuracy of the ranking tool enables AccuRefiner to consistently choose the refinement candidates with lower RMSD values compared to the input coarsely docked structures. We tested our method on a large number of docked complexes produced by three different docking programs. We were able to produce complexes with lower RMSDs in nearly all of the test cases. In some cases the improvement in RMSD was nearly 20%. Current and future work are two-fold. We are working on further enhancing the accuracy of the ranking tool by training the network with



(a) 2A9K RosettaDock (b) 2A9K AccuRefiner (c) 2A9K native

(d) 2AYO RosettaDock (e) 2AYO AccuRefiner (f) 2AYO native

Fig. 2: Interface packing comparison for docked and refined structures with respect to the native conformation. Docked structures generated by RosettaDock are shown in the first column; the second column shows the best refinement solution generated from the initial structures; the third column shows the corresponding native structures on each row. Interface atoms are drawn as spheres. Chains are represented in different colors.

a richer dataset and optimizing the selected features. Finally, we plan to incorporate flexible fitting of loops and side chains to improve interface packing in the decoy generation stage. Executables will be available upon request. In the future, we plan to make AccuRefiner available as a web service.

Acknowledgements

The research was funded in part by NSF grant CCF-1421871 (NH).

References

1. D.S. Goodsell and A.J. Olson. Structural symmetry and protein function. *Annu. Rev. Biophys. Biomol. Struct.*, 29(1):105–153, 2000.
2. S. Vajda and D. Kozakov. Convergence and combination of methods in protein-protein docking. *Curr. Opinion Struct. Biol.*, 19(2):164–170, 2009.
3. S. J. de Vries, A. D. J. van Dijk, M. Krzeminski, M. van Dijk, A. Thureau, V. Hsu, T. Wassenaar, and A.M. J. J. Bonvin. Haddock versus haddock: New features and performance of haddock2.0 on the capri targets. *Proteins: Struct., Funct., Bioinf.*, 69(4):726–733, 2007.
4. D. Schneidman-Duchovny, Y. Inbar, R. Nussinov, and H. J. Wolfson. Geometry based flexible and symmetric protein docking. *Proteins: Struct. Funct. Bioinf.*, 60(2):224–231, 2005.

5. C. Wang, P. Bradley, and D. Baker. Protein-protein docking with backbone flexibility. *J. Mol. Biol.*, 373(2):503–519, 2007.
6. P. Ferrara, H. Gohlke, D.J. Price, G. Klebe, and C.L. Brooks III. Assessing scoring functions for protein-ligand interactions. *Journal of medicinal chemistry*, 47(12):3032–3047, 2004.
7. Sheng-You Huang. Search strategies and evaluation in protein–protein docking: principles, advances and challenges. *Drug Discovery Today*, 19(8):1081–1096, 2014.
8. Irina S. Moreira, Pedro A. Fernandes, and Maria J. Ramos. Hot spots–a review of the protein-protein interface determinant amino-acid residues. *Proteins: Structure, Function, and Bioinformatics*, 68(4):803–812, 2007.
9. Salam A. Assi, Tomoyuki Tanaka, Terence H. Rabbitts, and Narcis Fernandez-Fuentes. Pcrpi: Presaging critical residues in protein interfaces, a new computational tool to chart hot spots in protein interfaces. *Nucleic Acids Research*, 38(6):e86, 2010.
10. Yanay Ofran and Burkhard Rost. Predicted protein-protein interaction sites from local sequence information. *FEBS Letters*, 544(1-3):236 – 239, 2003.
11. V. Srinivasa Rao, K. Srinivas, G. N. Sujini, and G. N. Sunand Kumar. Protein-protein interaction detection: Methods and analysis. *International Journal of Proteomics*, 2014:12, 2014.
12. S. Potluri, A. K. Yan, J. J. Chou, B. R. Donald, and C. Bailey-Kellogg. Structure determination of symmetric homo-oligomers by a complete search of symmetry configuration space, using NMR restraints and van der waals packing. *Proteins: Struct. Funct. Bioinf.*, 65(1):203–219, 2006.
13. Dima Kozakov, Dmitri Beglov, Tanggis Bohnuud, Scott E. Mottarella, Bing Xia, David R. Hall, and Sandor Vajda. How good is automated protein docking? *Proteins: Structure, Function, and Bioinformatics*, 81(12):2159–2166, 2013.
14. E. Mashiach, R. Nussinov, and H. J. Wolfson. Fiberdock: Flexible induced-fit backbone refinement in molecular docking. *Proteins: Struct., Funct., Bioinf.*, 78(6):1503–1519, 2010.
15. Andreas May and Martin Zacharias. Energy minimization in low-frequency normal modes to efficiently allow for global flexibility during systematic protein-protein docking. *Proteins: Struct., Funct., Bioinf.*, 70(3):794–809, 2008.
16. M. Krol, A.L. Tournier, and P.A. Bates. Flexible relaxation of rigid-body docking solutions. *Proteins: Struct., Funct., Bioinf.*, 68(1):159–169, 2007.
17. S. Chaudhury, A. Sircar, A. Sivasubramanian, M. Berrondo, and J. J. Gray. Incorporating biochemical information and backbone flexibility in rosettadock for capri rounds 6-12. *Proteins: Struct., Funct., Bioinf.*, 69(4):793–800, 2007.
18. J. Esquivel-Rodriguez, Y.D. Yang, and D. Kihara. Multi-lzerd: Multiple protein docking for asymmetric complexes. *Proteins: Struct., Funct., Bioinf.*, 80(7):1818–1833, 2012.
19. B. Akbal-Delibas, I. Hashmi, A. Shehu, and N. Haspel. An evolutionary conservation based method for refining and reranking protein complex structures. *J Bioinform Comput Biol*, 10(3):1242002, 2012.
20. I. Halperin, B. Ma, H. Wolfson, and R. Nussinov. Principles of docking: an overview of search algorithms and a guide to scoring functions. *Proteins: Struct., Funct., Bioinf.*, 47(4):409–443, 2002.
21. Panagiotis L. Kastritis and Alexandre M. J. J. Bonvin. Are scoring functions in protein-protein docking ready to predict interactomes? clues from a novel binding affinity benchmark. *Journal of Proteome Research*, 9(5):2216–2225, 2010.
22. Jrgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117, 2015.
23. Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. *Elements of artificial neural networks*. MIT press, 1997.
24. B. Akbal-Delibas, M. Pomplun, and N. Haspel. Accurefiner: A machine learning guided refinement method for protein-protein docking. In *proc. of BICoB (7th international conference on Bioinformatics and Computational Biology)*, March 2015.

25. B. Akbal-Delibas, M. Pomplun, and N. Haspel. Accurmsd: A machine learning approach to predicting structure similarity of docked protein complexes. In *ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, 2014.
26. B. Akbal-Delibas, M. Pomplun, and N. Haspel. Accurate prediction of docked protein structure similarity. *J. Comp. Biol.*, To Appear, 2015.
27. J.C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R.D. Skeel, L. Kale, and K. Schulten. Scalable molecular dynamics with namd. *J. Comput. Chem.*, 26(16):1781–1802, 2005.
28. A. Wilkins, S. Erdin, R. Lua, and O. Lichtarge. Evolutionary trace for prediction and redesign of protein functional sites. *Methods Mol Biol.*, 819:29–42, 2012.
29. Howook Hwang, Thom Vreven, Joël Janin, and Zhiping Weng. Protein–protein docking benchmark version 4.0. *Proteins: Structure, Function, and Bioinformatics*, 78(15):3111–3114, 2010.
30. I. Mihalek, I. Res, and O. Lichtarge. Evolutionary trace report maker: a new type of service for comparative analysis of proteins. *Bioinformatics*, 22(13):1656–7, 2006.
31. Sergey Lyskov and Jeffrey J Gray. The rosettadock server for local protein–protein docking. *Nucleic acids research*, 36(suppl 2):W233–W238, 2008.