SNAPP, CRACLE, POPP: PREDICTING PROTEIN INTERACTIONS

Stephen J. Bush

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Bioinformatics and Computational Biology.

Chapel Hill
2014

Approved by:

Alexander Tropsha

Charles W. Carter

Aravind Asokan

Brian Kuhlman

Jack Snoeyink

ABSTRACT

Stephen J. Bush: SNAPP, CRACLe, PoPP: Predicting Protein Interactions.
(Under the direction of Alexander Tropsha.)

Protein-Protein Interactions (PPIs) play a central role in all major signaling events that occur in living cells, from DNA replication to complex, post-translational protein-signaling systems. However, many if not most pairs of interacting proteins remain unknown, and the ability to identify and predict protein-protein interaction sites is a key component in systems and structural biology. Computational techniques such as MD simulations and homology- or template-based modeling constitute the main bioinformatics methods applied to study PPIs, and despite many recent developments, fast and reliable predictions of PPI sites remains a challenge.

Using computational geometry, we have developed two novel, geometry-based scoring function called Simplicial Neighborhood Analysis of Protein Packing (SNAPP) for the task of analyzing and predicting protein interactions. SNAPP-Surface calculates the likelihood that an amino acid on the surface of a protein will participate in a protein interaction. SNAPP-Surface is used in our novel algorithm and software for predicting protein-protein and protein-peptide binding sites called Critical Residue Analysis and Complementarity Likelihood (CRACLe). CRACLe was designed for accurate and efficient high-throughput screening of individual proteins for potential binding sites. CRACLe can be effectively applied to identify putative binding sites for novel proteins and potentially for building protein-protein networks. SNAPP-Interface is used in our novel protein-peptide docking algorithm called Prediction of Protein-peptide Packing (PoPP) to evaluate protein-peptide interactions. SNAPP-Interface is also useful for discriminating between native-like and decoy protein-protein interactions. The SNAPP, CRACLe, and PoPP software and all curated protein-protein and protein-peptide datasets are freely available at http://chembench.mml.unc.edu/cracle.

ACKNOLWEDGMENTS

This work would not have been possible without the guidance and support my adviser Alex Tropsha. I stumbled across his path almost as an accidental afterthought, but I couldn't have found a better home for the past five years. His patience and support made this project possible. I would also like to thank Charlie Carter for his guidance, friendship, and the many hours spent discussing science over beer at Top of the Hill. I would like to thank Aravind Asokan – his advice helped shape the project and provide it with the focus it needed. I would also like to thank Brian Kuhlman and Jack Snoeyink for their advice and feedback. I have no doubt the project would have gone in a completely different direction without their wisdom. Thank you to the Bioinformatics training grant and the National Science Foundation for funding the project.

I would also like to thank the many members of the Molecular Modeling Laboratory for their input and encouragement during my time in the lab. I would especially like to thank Denis Fourches, without whom the project would have fallen apart long ago. We didn't always see eye to eye (at least without me standing on a chair first), but despite his immense workload, he was always available, and I have learned so much from him. I must also thank Andy Fant and Eugene Muratov for their friendship and advice. I can also thank my friends and fellow colleagues in the Bioinformatics program for more than a few laughs and for providing notches on the stick for me to strive towards.

Last, but certainly not least, I must thank my family. To my wife Jen: Your support and patience have kept me going; I don't think I could have made it without you. To my daughters Elizabeth and Cecily: You both are the best parts of our time in Chapel Hill; your laughs are infectious, and I love you both so much. Thank you to the rest of my family who have loved and supported me throughout.

To my three girls.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Ab	Antibody
Ag	Antigen
AD	Applicability Domain
CAPRI	Critical Assessment of PRediction of Interactions
CASP	Critical Assessment of protein Structure Prediction
CRACLe	Critical Residue Analysis and Complementarity Likelihood
FAK	Focal Adhesion Kinase
HSI	Hot Spot Identification
IEDB	Immune Epitope Database
Ig	Immunoglobulin
MHC	Major Histocompatibility Complex
PDB	Protein Data Bank
PICES	A PDB sequence culling server with a poorly documented acronym
POPP	Prediction of Protein-peptide Packing
PPI	Protein-Protein Interaction
QSAR	Quantitative Structure-Activity Relationship
RMSD	Root Mean Squared Deviation
RT500	Richard Top 500
SNAPP	Simplicial Neighborhood Analysis of Protein Packing
SPPR	Single-point-per-residue

CHAPTER 1

Introduction

Protein-protein interactions (PPIs) play an important role in systems biology, particularly for understanding inter- and intra-cellular biological networks [1, 2, 3]. Identifying PPIs using experimental techniques, such as X-ray crystallography, mutagenesis, or mass spectrometry, is cost and time intensive [2, 4]; as a result, computational algorithms have been developed to predict PPI, but these algorithms are still time consuming and often require some *a priori* knowledge of the interaction in question [1, 5, 6]. Therefore, a computational method devoid of the above limitations is highly desirable to quickly and accurately identify potential binding sites on a protein surface and evaluate the likelihood of an interaction between two proteins [2, 6].

More than a dozen protein-protein docking software packages [7] have been described in the literature (Table 1.1), implementing various types of protein-protein docking algorithms; of these, HADDOCK [8, 9] and RosettaDock [10, 11] are among the most well-known within the structural bioinformatics community. Docking algorithms are regularly and rigorously benchmarked for their prediction accuracy under the framework of the CAPRI challenge [12, 13, 14], where modelers are invited to submit their predicted structures for various protein complexes with unknown (at the time of prediction) structures. Although every current approach has unique features, all follow the same overall, three-stage workflow: (i) representation of the system (e.g., coarse-grained or all atom models), (ii) sampling or generation of docking poses within a conformational search space (e.g., stochastic Monte Carlo or Molecular Dynamics), and (iii) ranking of potential conformations using a scoring function (e.g., knowledge or physical force field based). Most docking algorithms rely on the idea that proteins form specific interactions requiring geometric, electrostatic, and/or hydrophobic complementarity [15, 7]. The electrostatic and hydrophobicity terms are generally accepted as the most important [7, 16] and are usually expressed in the scoring function to evaluate the correctness of every generated protein-protein conformation; however, the more complex the conformational search algorithm and the scoring function are, the more time-consuming the calculations become. Therefore, numerous studies [17, 3, 18, 19, 20] have focused on creating new scoring functions and on the identification of PPI hot spots, i.e., solvent-exposed residues critical for specific interactions, to limit the conformational search space. Further details about PPIs and computational techniques to predict them can be found in several recent publications [21, 15, 7, 1].

Despite significant advances in protein-protein docking, there are still unsolved challenges and numerous weaknesses: CAPRI results [22, 10, 12, 13, 14, 23, 24] show that many approaches are able to accurately predict PPI only for relatively small contact interfaces with small conformational perturbations required for complex formation. Indeed, the vast majority of protein docking methods treat proteins as rigid bodies and assume that the overall conformations of the bound chains will be the same as unbound conformations. Unfortunately, this assumption is not always true, and approaches such as HADDOCK have begun taking into account both side-chain and backbone flexibility (at the refinement stage only) to obtain better prediction performance with flexible proteins. Another important and still unresolved weakness is that most docking algorithms return the ”best” pose for a given pair of protein chains, even if the two chains do not actually interact in a biological system.

One possible method for improving protein-protein docking is to limit the sampling step using interaction hot spots. Hot Spots are amino acids found on a protein surface that account for a significant portion of the binding free energy in a given PPI [25]. Identification of hot spots is crucial for studying or modifying PPIs [3, 17, 26], and many algorithms have been

	Access		Ligand			Search									Scoring and Refinement						
Software	Public	Server	Protein	Peptide	Small Molecule	Local	Global	Rigid-body	Flexible	Fast Fourier Transform	Grid-based	Monte Carlo	Genetic Algorithm	Geometric Hashing	Rigid-Body	Flexible	Rotational Sampling	Energy-based	Geometry-based	Restraint-based	Shape Complementarity
AutoDock	✓		✓		✓		✓	✓			✓		✓			✓	✓	✓	✓		
ClusPro/PIPER	✓	✓	✓			✓		✓		✓	✓				✓		✓	✓			✓
DOT 2	✓		✓				✓	✓		✓	✓				✓			✓			
DynaDock	?			✓			✓	✓				✓				✓		✓			
GRAMM-X	✓	✓	✓				✓	✓		✓	✓				✓			✓	✓		
HADDOCK	✓	✓	✓				✓	✓				✓					✓	✓		✓	
Hex	✓	✓	✓				✓	✓		✓							✓	✓			✓
PatchDock	✓	✓	✓				✓	✓						✓	✓				✓		✓
RosettaDock	✓	✓	✓			✓		✓				✓				✓		✓			
ZDock	✓	✓	✓				✓	✓		✓	✓				✓			✓			✓

Table 1.1: [Note to committee: I am currently updating this table.] Characteristics of popular protein-protein docking software.

	Access		Ligand				Algorithm			Features*				Returns	
Name	Public*	Online	Protein	Peptide	Small Molecule	Nucleic Acid	Homology	Machine Learning	Molecular Dynamics	Max Submissions	PDB Structure	Accepts Ligand Information	Accepts Docked Structures	Hot Spot Residues	Binding Sites
CPORT	✓	✓	✓	-	-	-	†	†	†	1	✓	-	-	✓	-
PredUs	✓	✓	✓	✓	-	-	✓	-	-	5	✓	✓	-	✓	-
IBIS	✓	✓	✓	✓	✓	✓	✓	-	-	1	✓	-	-	✓	✓
HotPoint	✓	✓	✓	✓	-	-	-	-	✓	1	✓	R	R	✓	-
PEPSITE 2	✓	✓	-	✓	-	-	✓	-	-	1	✓	R	-	✓	✓
PCRPi-W	-	✓	✓	✓	-	-	-	✓	-	1	✓	R	-	✓	-
PocketQuery	✓	✓	✓	-	✓	-	-	-	✓	1	✓	R	R	✓	✓
HSPred	✓	✓	✓	-	-	-	-	✓	✓	1	✓	R	R	✓	-
Robetta Server	✓	✓	✓	-	✓	-	-	-	✓	1	✓	R	R	✓	✓
iPred	-	-	✓	✓	-	-	-	✓	-	-	✓	-	-	✓	✓
Metz, et al.	-	-	-	-	✓	-	-	-	✓	-	-	-	-	✓	✓
FTMap	?	?	?	?	✓	?	?	?	?	?	?	?	?	?	?
Nisius et al.	-	-	✓	?	✓	?	?	?	?	?	?	?	?	?	?
SiteHound	✓	✓	?	?	✓	?	?	?	?	?	?	?	?	?	?

Table 1.2: Characteristics of existing algorithms for hot spot and binding site identification.

* 'Public' refers to unrestricted access of the software; R stands for required.

†CPORT returns a consensus from 6 different algorithms: WHISCY, PIER, ProMate, cons-PPISP, SPPIDER, and PINUP. These six, freely available packages are not detailed here as CPORT reported significantly improved results even above the best of them.

proposed for hot spot identification (HSI) . The majority of HSI algorithms analyze protein surfaces for specific patterns of chemical and geometrical properties, such as charge, polarity, hydrophobicity, shape, or sequence [18, 27, 28, 29], while several recent algorithms utilize homology models to transitively identify binding sites [30, 31]. Table 1.2 provides a list of several popular algorithms.

Existing methods for predicting protein-protein binding sites are limited by a lack of accessibility, functionality, and overall accuracy. Unfortunately, not all are publicly available as either a web server or standalone program (see Table 1.2), and for those that are, users must typically submit each protein or complex separately and wait several minutes to hours for the results. At the time of this writing, we are unaware of a publicly-accessible method that is capable of handling a large volume of queries. Most algorithms require that users supply the protein or peptide ligand, and in some cases, the supplied ligand must already be in a native or native-like docking pose [26], further hindering the ability of these algorithms for discovering unknown PPIs. The reported prediction accuracy has not been high; few algorithms have reported a prediction accuracy above 60% [32]. Recent reviews commented on the difficulty of comparison between different approaches [6, 32] because each algorithm was tested and benchmarked on a different data set and validated with the different metrics. In general, homology-based algorithms have been shown to yield more accurate predictions, but by definition these are applicable only to proteins with known structural homologs[30, 32].

With these limitations in mind, we set out to develop a series of algorithms to predict protein binding sites and interactions based on two hypotheses: (1) The geometry and composition of residues involved in protein-protein interactions are conserved versus residues found on the rest of the protein surface; and (2) this conservation can be used to predict other protein-protein interactions. Over the next the chapters we will discuss the development and benchmarking of our novel scoring function called Simplicial Neighborhood Analysis of Protein Packing (SNAPP); our novel binding site prediction algorithm called Critical Residue Analysis and Complementarity Likelihood (CRACLe); and our novel protein-peptide docking algorithm called Prediction of Protein-peptide Packing (POPP). Each part of SNAPP, CRACLe, PoPP provides a rapid and efficient geometry-based algorithm based on a combination of techniques from cheminformatics and computational geometry. Each part was implemented with high-throughput analysis in mind and requires only protein crystal structures as input.

CHAPTER 2

Development of a SNAPP Scoring Function for Analysis of Protein Interactions

2.1 Creation of the SNAPP scoring function

The SNAPP scoring function was originally developed by the Tropsha laboratory in the late 1990s as a method to evaluate protein structure [33] and pioneered the use of a computational geometry technique called Delaunay tessellation [34] for protein structure analysis[1]. Since its creation, SNAPP has been used to recognize protein folds [37], predict protein stability [38], simulate protein folding [39], identify structural motifs in protein folds [40], identify fold nuclei [41], distinguish between native and native-like versus decoy protein folds [41], and automate protein-function annotation [42]. The initial development of SNAPP has been summarized in a review [43].

2.1.1 Protein Representation

Two representations are applied both in the creation and application of the SNAPP score: (1) coarse-grained representation of protein structure using a single-point-per-residue (SPPR) model; and (2) partitioning of the protein into an aggregate of four-body interactions via Delaunay tessellation [34]. Both of these representations deviate from the standard models for protein structure analysis, which typically use an all-atom representation with one or more

[1] Previous studies had used related techniques such as Voronoi diagrams [35] and α shapes [36]; however, SNAPP is the first direct application of Delaunay tessellation to protein structure found in the literature at the time

energy functions [44]. The goodness of a representation depends upon its application, and our coarse-grained representation emphasizes speed and stability over structural precision.

The SPPR representation of a protein employed by Singh et al. [33] originally used the C_α of each amino acid; however, C_αs were quickly replaced with the side-chain centroids, including the C_α, for each residue. The use of side-chain centroids was found to be more predictive [41] and results in a tessellation that is more stable against perturbation [45] when compared to the use of C_αs. Furthermore, each centroid is more robust against errors in structural data versus an all atom model, where the loss of an atom may change the results of an energy calculation or predicted hydrogen bonding. A centroid minus an atom still retains the properties of its amino acid type but suffers a slight coordinate change. Although such a change could result in a modified Delaunay tessellation, a previous study found that Delaunay tessellation is sufficiently robust to handle centroid perturbations [45]. Centroid coordinates are less sensitive to side chain rotamers: Not only can the rotational movement be accounted for with a single translation, but the movement will be less drastic due to the constancy of the atoms that do not move. Even more importantly, side chain centroids lower the complexity of residue interactions from a multi-body to a single-body problem.

We are currently using a modified version of the Bowyer-Watson algorithm [46, 47] for Delaunay tessellation. Delaunay tessellation provides a method for partitioning a three-dimensional protein structure into simpler and more manageable polyhedrons. Given a series of coordinates in three-dimensional space, Delaunay tessellation generates an aggregate of space-filling, non-overlapping, irregular tetrahedra, known as simplices, and each Delaunay tetrahedron objectively and uniquely defines the vertices as four nearest-neighbor points. When applied to a SPPR model, the aggregate of tetrahedra form a network of contacts between residues, reducing a complex, three-dimensional structure to a collection of explicit quadruplet structural motifs and a unique connected graph for every protein. To better define the protein structure, Delaunay edges longer than 11.5 Å are removed from the tessellation. These four-body simplices are the smallest possible constructs both necessary and sufficient to preserve structural information and allow for comparison between protein geometries [40, 43]. Figure 2.1A and B show an example of a tessellated protein

Tetrahedra are further classified by their residue composition and sequence adjacency (Figure 2.1C), which simply means the types of residues involved and the peptide bonding between each of the residues. Although the exact distribution of tetrahedra is unique for each protein structure, the Tropsha group found that particular types of tetrahedra occur more often than statistically expected. This finding culminated in the SNAPP database and a novel, four-body statistical scoring function.

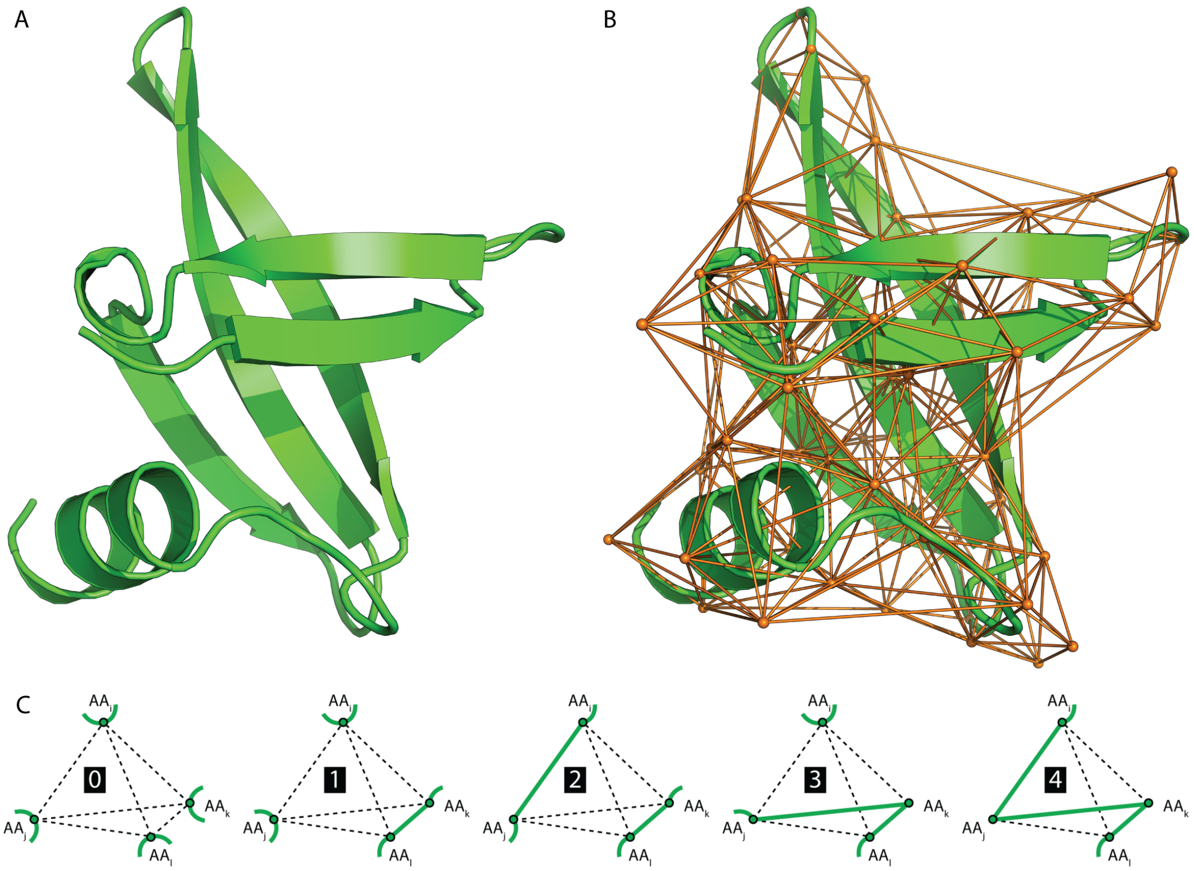

Figure 2.1: (A & B) The tessellated structure of a DNA binding protein (PDB code 1C8C) with edges greater than 11.5 Å trimmed. (C) Five tetrahedral types based on peptide bonds between adjacent amino acids, ranging from zero peptide bonds between residues (type 0) to three peptide bonds (type 4).

2.1.2 Calculating the SNAPP scoring function

The original SNAPP scoring function was defined by Singh, et al. [33] as follows: Given a training set of protein structures, all proteins are tessellated, and SNAPP scores are calculated for every possible simplex combination of amino acid composition and sequence adjacency according to the following equation:

$$q_{ijkl} = \log\left(\frac{f_{ijkl}}{p_{ijkl}}\right) \tag{2.1}$$

where q is the SNAPP score for a simplex with amino acids $ijkl$; and f and p are the observed and expected frequencies of the simplex, respectively:

$$f_{ijkl,t} = \frac{|s_{ijkl}|}{|S|} \tag{2.2}$$

$$p_{ijkl,t} = Ca_i a_j a_k a_l \tag{2.3}$$

where $|S|$ is the cardinality of Delaunay simplices, s is a subset of all simplices S in the dataset, and a_i is the observed frequency of amino acid i in the dataset. In the expected probability p, C defines a combinatorial factor that accounts for redundancy of amino acid composition, e.g., $ijkl = jkli$:

$$C = \frac{4!}{\prod_n^i |a_i|} \tag{2.4}$$

where n is the number of unique amino acids in the simplex, and $|a_i|$ is the cardinality of amino acids of type i.

The SNAPP score q is a statistical likelihood than estimates how likely it is that a particular simplex would be found in a protein based on our knowledge of existing protein structures. Combining scores will yield the likelihood that simplices will be found together, and because each score q is a logarithm of the likelihood function, we can add the scores rather than dealing with products. Thus, once the SNAPP scores have been computed, calculating the SNAPP score Q for a new protein is simple: tessellate the protein; score each simplex according to its composition and sequence adjacency; and sum the individual SNAPP scores:

$$Q = \sum q_i \tag{2.5}$$

2.2 Variations on a Theme

There have been three variations of the SNAPP methodology since its inception. Each follows the same basic structure given above, but includes slight modifications to refine and specialize the potential. These modifications include a filter on the scoring function output, slight changes to the scoring function itself, and alterations to the representation of the protein, which in turn alter the scoring function.

2.2.1 SNAPP and Protein Tertiary Structure

The first variation of SNAPP by Carter et al. [38] began as an attempt to approximate the free-energy difference, $\Delta(\Delta G)$, of protein folding by evaluating native protein tertiary structure using SNAPP. Carter et al. found that simplices composed of four hydrophobic residues occurred more frequently than expected by random chance, and proposed that these simplices encode information relevant to thermodynamically significant tertiary interactions. To test their hypothesis, Carter et al. selected five proteins with a total of seventy-six mutations with experimentally tested ΔG values. They identified core residues for each of the five proteins from either the literature or based on cumulative SNAPP scores greater than 1.5 Å, and generated a series of variant proteins with single point mutations for each core residue. They found that the difference in SNAPP scores, Δ SNAPP, for hydrophobic simplices in the core of a protein correlated with experimental $\Delta(\Delta G)$ values.

In the course of their study, Carter et al. recompiled the SNAPP scoring function using an improved dataset containing 1,200 single chain proteins versus the original 103. Additionally, the study introduced two changes to the SNAPP algorithm as set forth by Singh et al. First, Carter et al. focused on protein tertiary structure and therefore removed many simplices involving residues adjacent in the primary sequence as such simplices were said to be uninvolved in tertiary interactions. The second change was the removal of any simplices with a vertex-to-vertex edge distance greater than 10 Å on the basis that direct interactions will occur only at shorter distances. However, both of these changes were introduced as filters for the study rather than canonical changes to the SNAPP score.

One year later, Cammer et al. [40] introduced the first SNAPP variation in a study using SNAPP to identify tertiary packing motifs. This study expanded on the previous alterations, maintaining the 10 Å edge cutoff and explicitly stating the removal of all simplices except type 0 (Figure 2.1C), i.e., simplices without any sequence-adjacent residues. Cammer et al. used SNAPP to identify common sequence-structure motifs among simplices with similar residue composition. They found that simplices containing a balance of hydrophobic and polar residues occurred far more frequently than simplices with singular compositions. Furthermore, they found specific residue-sequence motifs for three separate protein families, suggesting that some of the motifs could be used as markers for protein functional families.

The SNAPP score variation introduced by Cammer et al. (referred to as SNAPP-Cammer) has been used in other studies [43, 48], but is typically reserved for evaluating tertiary interactions. In addition to the log-likelihood functions, the SNAPP-Cammer database contains a plethora of additional data for each type 0 simplex composition; however, many of the implementation details have been lost, and exactly how this additional information was applied to the scoring function, if at all, is unknown. As a result, the SNAPP-Cammer scoring function used for comparison performs scores only type 0 simplices as described in the paper.

2.2.2 Predicting Native-like versus Decoy Structures

Although the original paper by Singh et al. mentioned the five different types of simplices (Figure 2.1C), the five types were noticeably absent in the scoring function. Gan et al. [39] attempted to incorporate the sequence adjacency into the formula, along with redefining SNAPP as a multi-body contact energy:

$$Q^{\alpha}_{ijkl} = -k_B T \ln \frac{f^{\alpha}_{ijkl}}{p_{ijkl}} \tag{2.6}$$

where α represents simplex type, and the observed frequency f was redefined to

$$f^{\alpha}_{ijkl} = \frac{|s^{\alpha}_{ijkl}|}{|s^{\alpha}|} \tag{2.7}$$

where $|s|$ is the cardinality of simplices in the dataset with a given type α and composition $ijkl$. Unfortunately, the inclusion of the type did not extend into the expected frequency p. Gan et al. made a number of additional changes, including using a varied edge cutoff of either 8 or 11 Å to allow for comparison of SNAPP scores created from datasets with fewer structures. The refined scoring was unsuccessfully used to select native-like conformations from a series of decoys.

Decoy discrimination still presented an inviting target for SNAPP, and Krishnamoorthy et al. [41] re-purposed SNAPP for the task. Like Gan et al. they saw the importance of including the simplex type, but Krishnamoorthy et al. also recalculated the expected simplex frequency:

$$p^{\alpha}_{ijkl} = C a_i a_j a_k a_l p_{\alpha} \tag{2.8}$$

where p_{α} is the frequency of type α tetrahedra in the dataset. Additionally, the $-k_B T$ term was removed, as it would be constant, the natural log, $\ln$, was replaced with log base 10, and the edge cutoff was reset to 10 Å. The training set was also curated to remove any protein chains with missing atoms or residues to ensure the SNAPP potentials were developed without any irregularities. The resulting SNAPP potentials (henceforth called SNAPP-Bala) were able to accurately distinguish native-like protein folding from decoy conformations generated for a single protein, and were later applied with marginal success to evaluate the effects of mutations on protein stability and reactivity [49].

2.2.3 Accounting for Structural Variation

Protein structures obtained from X-ray crystallography or NMR can be imprecise: Experimental error and protein flexibility could lead to variation in the atomic coordinates, possibly resulting in a different Delaunay tessellation. In 2004, Bandyopadhay and Snoeyink [45] developed almost-Delaunay simplices to identify potential quadruplets that would occur if vertices within a point set were allowed small perturbations. Given a Delaunay tessellation, they identified possible Delaunay edges for all vertices within a minimum distance threshold of 10 Å and identified all possible simplices given these additional edges. Each of these simplices was said to be almost-Delaunay *iff* the simplex would be a Delaunay simplex after neighboring vertices were perturbed by a minimum distance $\varepsilon \geq 0$.

Bandyopadhyay and Snoeyink were able to visualize and quantify α-helices, β-sheets, and β-turns using almost-Delaunay simplices; however, they also found that fewer almost-Delaunay simplices were created as proteins became increasingly structured and when side-chain centroids were used instead of C_αs. Additionally, they weighted the SNAPP potentials based on the almost-Delaunay simplices and found that both versions were able to discriminate native-like from decoy protein folding. Although almost-Delaunay is a unique and potentially useful technique for analysis of protein structure, we chose not to use it in this project due to the additional computational complexity and overhead required.

2.3 Modern Modifications (M^2)

Given the relative success of SNAPP for evaluating a variety of different protein folding problems, we wanted to see if the scoring function could predict protein-protein interactions; since the most recent iterations of SNAPP were almost a decade old, at the very least, we needed to recalculate SNAPP using a training set with updated structures. First, we decided to use the variations set forth by Krishnamoorthy et al. as a control to allow for comparison between the old and newer scoring functions. Second, we suspected that the expected simplex frequency used in the previous SNAPP iterations might be too simple to accurately portray the complexity of protein interactions; we set out to remodel the expected simplex frequency based on a multi-body chemical reaction. Third, we designed a set of novel cheminformatics-like descriptors to account for simplex features ignored by the SNAPP potentials.

2.3.1 The Current SNAPP Scoring Function

As a part of updating the SNAPP score, we needed to recompile the training sets. Unfortunately, recompiling the training sets on PPI data meant that comparison against the old SNAPP-Bala potentials would not be accurate; we needed to recalculate a new set of SNAPP potentials on a set of single-chain proteins using the algorithm set for by Krishnamoorthy et al. We compiled a set of single-chain protein structures from the Richardson Top 500 [50], PICES [51], and a subset of the PDB [52], which we define in greater detail in Chapter 2.4.1. To help differentiate between other SNAPP scores, we refer to the updated potentials as SNAPP-Fold.

The SNAPP-Fold potentials were created using the following equations, which include the simplex type. The SNAPP score for a single simplex $q_{ijkl,t}$ with amino acids $ijkl$ in configuration type t is defined by:

$$q_{ijkl,t} = \log\left(\frac{f_{ijkl,t}}{p_{ijkl,t}}\right) \tag{2.9}$$

which, remains a log ratio of the observed f over the expected p frequencies. The frequencies f and p have likewise changed to account for the simplex type:

$$f_{ijkl,t} = \frac{|s_{ijkl,t}|}{|s_t|} \tag{2.10}$$

$$p_{ijkl,t} = Ca_i a_j a_k a_l f_t \tag{2.11}$$

where s is a subset of all simplices S in the dataset, a_i is the observed frequency of amino acid

i in the dataset, and f_t is the frequency of type t simplices in the dataset:

$$f_t = \frac{|s_t|}{|S|} \tag{2.12}$$

2.3.2 Redefining the Expected Frequency

When we first set out to redefine the SNAPP scoring function, our first concern was how the expected frequency $p_{ijkl,t}$ was calculated. In all of the SNAPP variations, the expected frequency estimates the likelihood that four particular residues will associate with each other due to random chance. We hypothesized that tetrahedral formation was not entirely due to random chance, but was constrained by the existing peptide bonds between sequential amino acids. To test our hypothesis, we designed three new expected frequencies based on (1) the distribution of Delaunay edges found in proteins, (2) the frequency of interaction between amino acids, as defined by Delaunay tessellation, and (3) the occupation frequency for cooperative binding.

Edge Frequency

Similar to the amino acid frequency a_i used in the original SNAPP equation, the edge frequency $f_{e_{ij}}$ gives a ratio of the occurrence of an edge between residues of type a_i and a_j across the dataset:

$$a_i = \frac{n_{a_i}}{N_{residues}} \tag{2.13}$$

$$f_{e_{ij}} = \frac{n_{e_{ij}}}{N_{edges}} \tag{2.14}$$

The edge frequency directly replaces the amino acid frequency, but allows the scoring function to take into account the frequency of peptide versus non-peptide edges, which inherently includes the simplex type:

$$p_{ijkl} = \prod_{x}^{s_e} f_{e_x} \tag{2.15}$$

$$= \prod_{x}^{\pi_{\texttt{peptide}}(s_e)} f_{e_x} \prod_{y}^{\pi_{non-peptide}(s_e)} f_{e_y} \tag{2.16}$$

where s_e is the set of edges for a given simplex and $\Pi_{peptide}$ is the projection of edges s_e that are peptide bonds. Like the original equation, we must also account for the redundancy of permutations due to amino acid and edge types:

$$p_{ijkl} = C \sum^{\sigma(s_{E:t})} \prod_{x}^{\Pi_{peptide}(s_e)} f_{e_x} \prod_{y}^{\Pi_{non-peptide}(s_e)} f_{e_y} \tag{2.17}$$

where $\sigma(s_{E:t})$ is the selection of all possible edge permutations given a simplex of type t. The final edge frequency serves as a basis for the other two scoring functions, substituting f_{e_x} for the respective frequencies.

Interaction Frequency

Instead of using amino acid distributions alone to calculate likelihood potentials, we could consider the formation of a simplex similar to that of a chemical reaction in equilibrium:

$$A + B \rightleftharpoons AB \tag{2.18}$$

with a first order reaction rate r and equilibrium coefficient of K,

$$r = k_1[A][B] - k_2[AB] \tag{2.19}$$

$$K = \frac{k_1}{k_2} \tag{2.20}$$

$$= \frac{[AB]}{[A][B]} \tag{2.21}$$

where $[A]$ is the concentration of A, and k_1 and k_2 are the rate coefficients. By treating the amino acid and edge distributions as the concentrations of each, we may approximate the frequency of interaction, f_I, between two residues:

$$f_I = \frac{f_{e_{ij}}}{f_{a_i} f_{a_j}} \tag{2.22}$$

Occupation Frequency

The final potential builds upon the interaction frequency and approximates a cooperative binding frequency f_V that is loosely based on the Adair-Klotz equation [53], which gives the fractional occupation v:

$$v = \frac{\sum_i^n i \prod_j^i K_i [A]^j}{1 + \sum_i^n \prod_j^i K_i [A]^j} \tag{2.23}$$

$$f_V = \frac{f_{e_{i \to j}} f_{e_{j \to i}} f_{a_i} f_{a_j}}{1 + f_{e_{i \to j}} f_{a_i} + f_{e_{j \to i}} f_{a_j}} \tag{2.24}$$

where the edge frequency has been given a direction $i \to j$ to indicate i binding to the structure before j. Here, the directed edge frequency is substituted for the rate constant, and the amino acid frequency is substituted for the concentration. The occupation frequency ignores the higher order reactions: The values calculated for $i > 1$ were several orders of magnitude smaller and had little effect on the frequency.

Cheminformatics-like Descriptors for Simplices

Until now, the SNAPP score utilized only two traits to characterize the tessellation of a protein structure: the amino-acid composition, and the sequence adjacency; however, Delaunay tessellation does not depend on either vertex composition or order of occurrence. Instead, Delaunay tessellation depends on and provides additional information about the spatial arrangement of the points, or amino acids, within the set. To make use of this additional information, we applied cheminformatics-like descriptors. In cheminformatics, chemical compounds are described by numerical parameters called descriptors that encode its physical and chemical characteristics. These descriptors range from constitutional traits, such as the number of atoms, to more complex topological indices, often based on the number of bonds per atom, also known as the vertex degree. Descriptors are a core component of cheminformatics algorithms and are utilized to help predict experimental outcomes. In order to better evaluate the structural diversity of protein packing, we developed a series of (i) chemistry-based descriptors that describe inherent structural characteristics, (ii) geometry-based descriptors to characterize the three-dimensional conservation of residue quadruplets and (iii) topology-based using well-defined constitutional and topological indices (e.g., Kier & Hall, Randic) [54, 55]. A complete list of the calculated protein descriptors can be found in Appendix 5.3[2].

Geometric descriptors characterize simplices by quantitatively scoring the conservation of their three-dimensional structure, such as volume, surface area, inter-residue distance and angles, tetrahedrality (i.e., a measure of deviance from an ideal tetrahedron) [33], and chirality. In particular, tetrahedral chirality uniquely characterizes protein structure by identifying not only nearest-neighbor residues but also their spatial orientation with one another. Because the underlying structure is always a tetrahedron, these data are quickly calculated and provide a simple comparison between tetrahedra with the same residue composition and sequence adjacency.

Topological descriptors aid in describing, discriminating, and qualitatively comparing PPI structure through graph theory, which is widely used in cheminformatics and has also been useful for studying protein structure [56, 57, 58], protein flexibility [59, 60], PPI structure [61], and protein-protein docking [62, 63]; however to our knowledge, we are the first group to apply graph theory to PPI described using Delaunay tessellation. Topological descriptors are quickly calculated and describe both connectivity and branching complexity, expressed as graph indices. Examples of graph indices include: the Wiener index, i.e., the length of the shortest path across a graph, which correlates to van der Waals surface area [64]; various

[2] The appendices do not seem to be correctly labeled

vertex centralities, e.g., vertex degree (Equation 2.25) and Eigenvector centrality (Equation 2.26), which measure the importance of a vertex within a graph:

$$\bar{v}_i = \frac{1}{M \sum_{v=1}^{M} v_i} \tag{2.25}$$

$$x_i = \frac{1}{\lambda \sum_{j}^{N} A_{ij} x_j}, \tag{2.26}$$

the Randic connectivity index (Equation 2.27), which expresses the level of graph branching [65]:

$$_R = \sum_{\textit{all edges}} (v_i \cdot v_j)^{-\frac{1}{2}}, \tag{2.27}$$

and the Estrada index (Equation 2.28), which characterizes protein folding [66]:

$$EE(G) = \sum_{i}^{n} e^{\lambda i}, \tag{2.28}$$

Descriptor calculation follows a simple workflow (Figure 2.2). First, each protein complex is subjected to Delaunay tessellation. Second, the calculation of these SNAP protein descriptors generates a series of numerical values for (a) each residue vertex, (b) each simplex, (c) each protein, and (d) special subsets of vertices, such as surface or interfacial residues. These numerical values specifically describe the constitutional, geometrical, and topological characteristics of each part of a protein, resulting in a protein fingerprint that can be used to analyze, sort, cluster, and model tetrahedra.

2.4 Validating the New Scoring Functions

In order to validate the new scoring functions, we needed to compare the new potentials against the old, which meant first testing on protein folding. To this end, we recompiled the SNAPP potentials using an updated set of single protein chains and tested the ability each

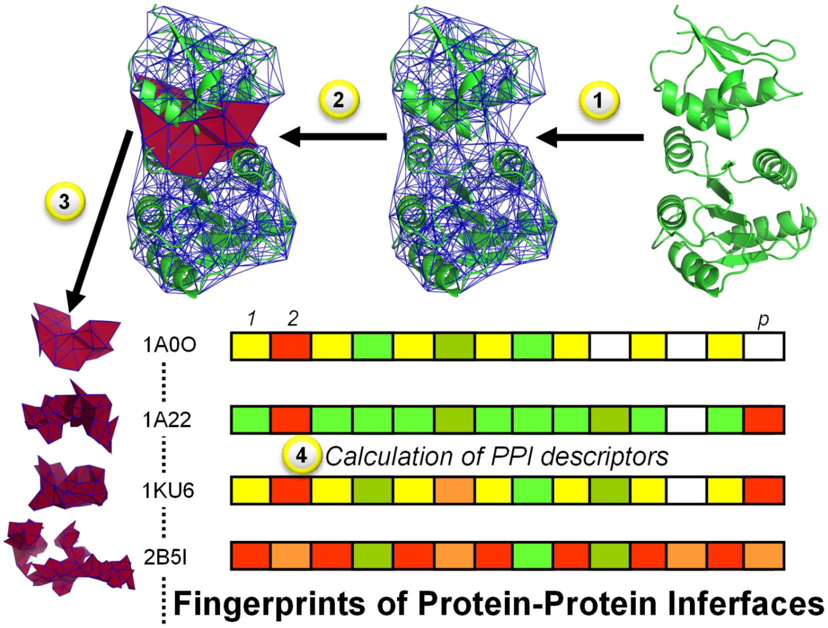
1
2
3
1A0O
1A22
1KU6
2B5I
1
2
p
4
Calculation of PPI descriptors
Fingerprints of Protein-Protein Inferfaces

Figure 2.2: Workflow to derive PPI fingerprints: (1) Tessellate a protein complex (PDB code 1A0O in this example); (2) Identify interfacial quadruplets; (3) Extract interfacial quadruplets; and (4) Calculate PPI descriptors (e.g., volume, exposed surface area, Kier & Hall indices).

of the six scoring functions to discriminate between native-like and decoy protein folds. We used SNAPP-Cammer, SNAPP-Bala, and with the newer SNAPP-Fold as controls to test the modified SNAPP potentials based on edge frequency, interaction frequency, and occupation frequency.

2.4.1 Compiling the Training Sets

The selection and curation of the data used to create the scoring function directly relates to the algorithms efficacy and applicability. The databases used for training are summarized in Figure 2.3A, and the creation of the algorithm is described below.

We trained the new SNAPP scoring function using three datasets: the Richardson Top 500 [50]; a specialized, single chain subset of structures from the PDB [52]; and a collection of structures selected by R-Factor using PICES [51]. The Richardson dataset contains 500 high-quality, manually curated crystal structures; however, as the last update was in 2000 [50][3]

[3] Since the compilation of SNAPP-Fold, the Richardson laboratory has replaced the Top 500 dataset with a newer Top 8000, which has not yet been used in SNAPP training.

and due to the low number of structures, we chose to add protein structures from additional sources. We selected a subset of single chain structures from the PDB with high resolution ($<$ 2 Å) and low sequence similarity ($< 35\%$) that contained only the protein itself, i.e., no ligands, co-factors, or nucleic acids. Further structures were added from PICES, a web server that ”culls” the PDB for structures according to R-factor.

All three training sets were curated, and any structures with one or more of the following problems were removed: missing atoms; missing entire residues; or containing an insertion code, or iCode (Figure 2.3B). The latter filter was chosen due to the inconsistent implementation and poor quality of structures containing iCode data. Duplicate structures were removed with preference given to the Richardson dataset, followed by the PDB subset. Although the majority of each dataset was removed, the remaining datasets had surprisingly little overlap. The resulting database of 1,473 unique single-chain protein structures was tessellated and used to recompile the SNAPP scoring function (Figure 2.3C).

2.4.2 Benchmarking and comparison of new and old SNAPP scores

Two separate tests were used to validate the SNAPP scoring functions: (1) the Baker decoy set containing 60 protein backbones, i.e., only the C_αs for each protein; and (2) the Rosetta all-atom data set with 59 proteins [67].

The Baker decoy set [68] consists of sixty protein backbones, each with one native-like and three decoy structures. Each set of protein C_αs were tessellated and scored using SNAPP-Bala, SNAPP-Cammer, SNAPP-Fold, and the three novel Edge, Interaction, and Occupancy frequencies. For each of the scoring functions, the highest scoring protein was predicted to also have the lowest RMSD in relation to the native protein. Unfortunately, none of the scoring functions were able to consistently distinguish between native-like and decoy protein structures (Figure 2.4): SNAPP-Cammer proved the worst, correctly predicting 14 of the 60 structures; SNAPP-Bala and the Edge, Interaction, and Occupancy frequencies performed slightly better with 18-19 correct predictions; and SNAPP-Fold lead the pack with a measly

(A) SNAPP Training Sets

				Dockground	
	RT500	PDB	PICES	Homo	Hetero
	Single-Chain Proteins			Protein Complexes	
Starting structure count	500	1,106	5,149	1,325	525
N Structures Missing atoms	110	309	1,869	-	-
N Structures Missing residues	108	312	2,487	-	-
N Structures with an iCode	13	12	34	20	70
N Remaining after curation	*310*	*562*	*877*	*1,305*	*455*
Total number of unique structures	**1,473**			**1,305**	**455**
Total number of simplexes	**1,437,278**			**253,031**	**104,165**

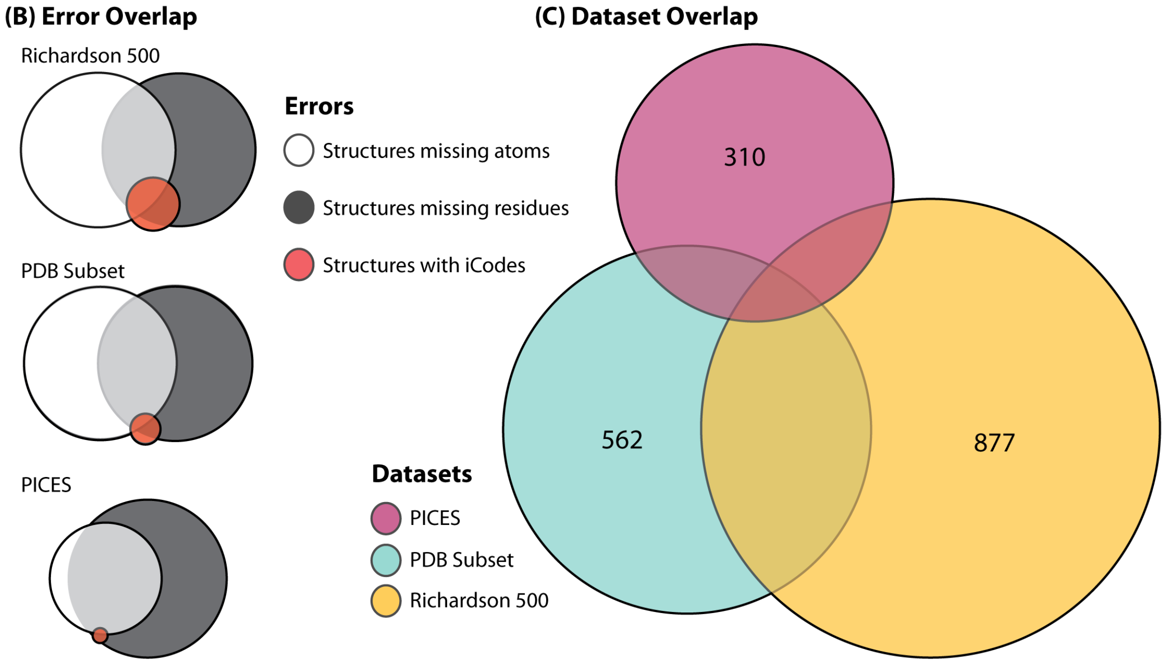

Figure 2.3: (A) Training set databases for SNAPP-Fold, SNAPP-Surface, and SNAPP-Interface. (B) For each dataset, a Venn diagram shows the overlap of structures that one or more errors. (C) A Venn diagram showing the contribution of each dataset in the SNAPP-Fold training set, resulting in a total of 1,473 unique protein structures.

24 accurate predictions. When including predictions where the lowest RMSD had the second highest SNAPP score, SNAPP-Bala eked ahead with 38 correct predictions over SNAPP-Fold's 36. Regardless, none of the scoring functions perform well, achieving little more than 60% native-like prediction accuracy for the dataset when the prediction standards are lowered. We hypothesize that some part of the poor predictions may likely be attributed to the use of C_αs for the protein structures rather than the side-chain centroids that SNAPP was trained on. To validate this hypothesis, we decided to retest the SNAPP scoring functions using the Rosetta all-atom decoy set.

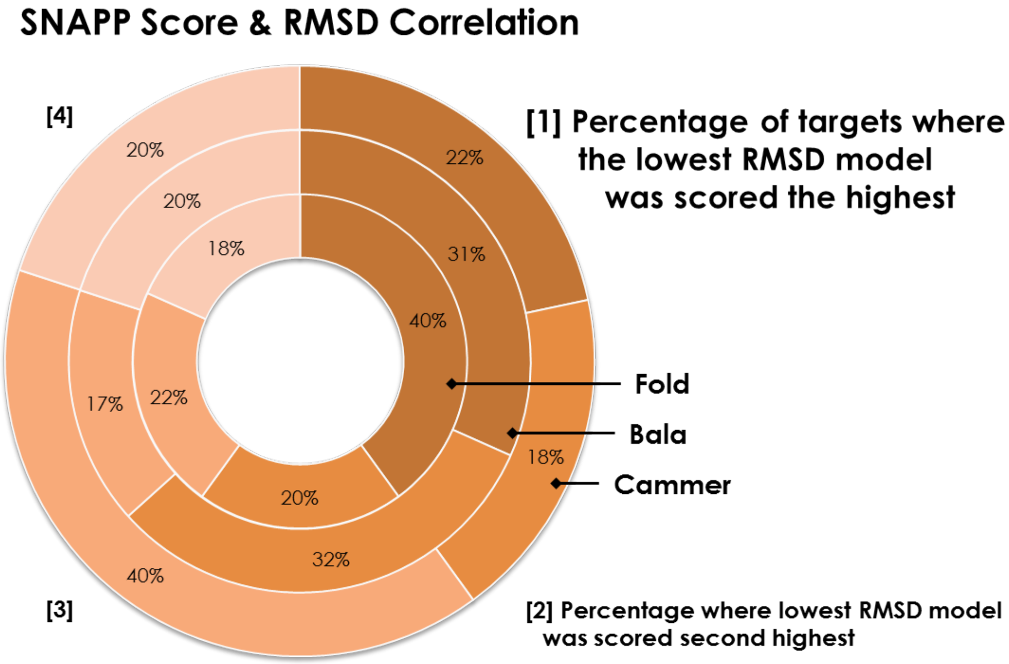

Figure 2.4: [Note to committee: This figure is confusing. I am fixing it and will send the updated figure.] The correlation between RMSD and SNAPP for the Baker decoy dataset. Proteins were put into one of four classes based on the SNAPP score rank of the lowest RMSD, e.g., if the highest scoring structure also has the lowest RMSD, it is put into class one. Shown are the percentages of proteins in each class as defined by the results from each of the three SNAPP scoring functions.

The Rosetta all-atom decoy set [67] contains 59 different sets of proteins, each set containing 1 native protein structure, a series of 20 Rosetta-refined native-like structures, and 100 low scoring decoys from 10,000 produced by the Rosetta structure prediction algorithm, resulting in a total of 121 structures per protein. For each protein group, we returned the rank of the native structure (Table 2.1), and calculated the sensitivity and specificity of the SNAPP scores based on the number of native and native-like structures as defined by an RMSD threshold of 1 Å (Table 2.2), 2 Å (Table 2.3), and 4 Å (Table 2.4). All of the SNAPP scores performed much better with the all-atom structures; however, SNAPP-Fold still outperformed the other scoring functions, followed closely by SNAPP-Bala, then by the Edge, Interaction, and Occupancy frequencies, with SNAPP-Cammer trailing behind. In fact, for 43 of the 59 proteins, SNAPP-Fold scored the native protein within the top 10 highest scores of the other 121 in each set; of those 43 sets, the native was scored within the top 5 for 36 and as the highest scoring for 22 of the protein sets. Although none of the SNAPP scoring functions successfully found the native pose for more than 73% of the proteins, SNAPP-Fold consistently outperformed the other SNAPP variations, which unfortunately included the novel frequency variations.

We also compared our results for the Rosetta all-atom decoy set against those of Arnautova et al. [69]. Arnautova et al. developed three force fields for evaluating protein stability and tested their energy functions against 45 of the 59 proteins in the Rosetta all-atom decoy set. For each protein, their algorithm generated an additional 6,000 decoys to provide a smoother energy landscape for identifying the lowest energy conformation. They evaluated their scoring function based on the RMSD of the decoy with the lowest energy. The highest scoring protein found with SNAPP-Fold had a lower RMSD for 30 of the 45 protein sets, versus 26 and 17 for SNAPP-Bala and SNAPP-Cammer, respectively.

Overall, all of the SNAPP functions performed adequately for decoy fold prediction, including the previously published SNAPP variations. The lack of reproducibility between the published results and our tests could be caused by a number of reasons. First, none of the published results covered a very large test set, at most including a handful of different proteins. Our results with the Rosetta and CASP9 test sets showed that SNAPP did indeed discriminate between native-like and decoy conformations very well for some proteins, and we hypothesize that this improved prediction is likely a result of over-fitting to the training set. Second, none

PDB ID	*N* Native-like	*N* Decoy	Fold	Bala	Cammer	Edge	Interaction	Occupancy	mean
1a19	21	100	17	16	19	25	31	28	22.67
1a32	88	33	116	88	60	119	105	116	100.67
1a68	21	100	35	37	25	74	71	73	52.50
1acf	21	100	1	1	14	17	18	19	11.67
1ail	25	96	1	1	5	37	90	61	32.50
1aiu	102	19	92	89	35	107	91	102	86.00
1b3a	19	102	7	7	10	13	11	11	9.83
1bgf	21	100	7	15	24	16	24	18	17.33
1bk2	21	100	4	16	108	3	6	5	23.67
1bkr	21	100	16	21	11	23	36	26	22.17
1bm8	21	100	1	1	4	1	3	2	2.00
1bq9	21	100	14	20	34	26	8	20	20.33
1c8c	31	90	105	108	79	101	109	105	101.17
1c9o	21	100	92	96	36	95	102	99	86.67
1cc8	21	100	26	27	31	35	42	39	33.33
1cei	21	100	5	8	52	11	23	15	19.00
1cg5	21	100	6	6	5	9	11	10	7.83
1ctf	21	100	5	5	5	69	86	85	42.50
1dhn	21	100	2	5	17	19	25	21	14.83
1e6i	21	100	14	9	5	70	58	69	37.50
1elw	121	0	22	16	27	2	1	1	11.50
1enh	46	75	92	64	32	54	73	64	63.17
1ew4	21	100	1	1	3	1	3	1	1.67
1eyv	21	100	11	10	11	32	28	34	21.00
1fkb	21	100	4	5	26	7	14	8	10.67
1fna	21	100	5	6	47	1	3	1	10.50
1gvp	13	108	34	40	42	72	85	75	58.00
1hz6	21	100	18	11	46	42	38	40	32.50
1ig5	22	99	56	45	22	86	60	73	57.00
1iib	28	93	10	10	6	31	48	41	24.33
1kpe	21	100	4	5	5	28	27	27	16.00
1lis	21	100	1	8	13	5	22	8	9.50
1lou	21	100	16	14	25	14	23	18	18.33
1nps	21	100	10	10	18	22	20	20	16.67
1opd	22	99	29	25	52	27	33	28	32.33
1pgx	121	0	5	18	27	14	20	16	16.67
1ptq	21	100	86	89	54	46	32	54	60.17
1r69	79	42	64	69	42	72	110	91	74.67
1rnb	21	100	3	3	12	4	7	5	5.67
1scj	21	100	46	28	24	13	17	17	24.17
1shf	21	100	72	77	75	93	78	87	80.33
1ten	21	100	2	2	5	2	10	6	4.50
1tig	21	100	26	35	39	23	18	21	27.00
1tul	21	100	15	18	18	21	22	22	19.33
1ubi	21	100	64	54	22	106	108	109	77.17
1ugh	21	100	1	1	3	5	2	4	2.67
1urn	21	100	1	1	19	1	1	1	4.00
1utg	14	107	119	119	120	121	121	121	120.17
1vcc	21	100	17	33	33	33	45	46	34.50
1vie	21	100	9	50	4	17	40	28	24.67
1vls	17	104	21	28	9	32	33	34	26.17
1who	21	100	21	21	22	22	22	22	21.67
256b	53	68	1	1	44	2	20	4	12.00
2acy	21	100	3	3	5	13	22	16	10.33
2chf	21	100	10	8	14	13	8	14	11.17
2ci2	21	100	18	41	26	6	25	10	21.00
4ubp	20	101	43	36	28	69	64	74	52.33
5cro	21	100	16	17	9	91	108	102	57.17
mean			26.59	27.55	27.72	36.43	40.71	39.09	

Table 2.1: The rank of the native protein from the Rosetta all-atom decoy set according to the SNAPP score.

			SNAPP variations											
			Fold		Bala		Cammer		f_{edge}		$f_{interaction}$		$f_{occupancy}$	
PDB ID	N Native-like	N Decoy	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp
1a19	21	100	0.99	0.95	0.99	0.95	0.97	0.86	0.97	0.86	0.95	0.76	0.97	0.86
1a32	10	111	0.92	0.10	0.92	0.10	0.91	0.00	0.92	0.10	0.91	0.00	0.92	0.10
1a68	21	100	0.93	0.67	0.95	0.76	0.89	0.48	0.86	0.33	0.86	0.33	0.87	0.38
1acf	4	117	0.97	0.25	0.98	0.50	0.97	0.25	0.98	0.50	0.98	0.50	0.98	0.50
1ail	1	120	1.00	1.00	1.00	1.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00
1aiu	21	100	0.86	0.33	0.86	0.33	0.83	0.19	0.90	0.52	0.91	0.57	0.91	0.57
1b3a	12	109	0.92	0.25	0.92	0.25	0.95	0.58	0.90	0.08	0.90	0.08	0.90	0.08
1bgf	12	109	0.94	0.50	0.94	0.42	0.94	0.42	0.93	0.33	0.95	0.58	0.95	0.58
1bk2	21	100	0.86	0.33	0.82	0.14	0.79	0.00	0.95	0.76	0.98	0.90	0.96	0.81
1bkr	21	100	1.00	1.00	1.00	1.00	0.97	0.86	0.99	0.95	0.99	0.95	0.99	0.95
1bm8	20	101	0.99	0.95	0.99	0.95	0.99	0.95	0.97	0.85	0.98	0.90	0.98	0.90
1bq9	21	100	0.83	0.19	0.83	0.19	0.79	0.00	0.84	0.24	0.87	0.38	0.85	0.29
1c8c	19	102	0.81	0.00	0.81	0.00	0.81	0.00	0.81	0.00	0.81	0.00	0.81	0.00
1c9o	18	103	0.83	0.00	0.83	0.00	0.83	0.00	0.83	0.06	0.83	0.06	0.83	0.06
1cc8	21	100	0.95	0.76	0.94	0.71	0.93	0.67	0.92	0.62	0.88	0.43	0.91	0.57
1cei	18	103	0.93	0.61	0.93	0.61	0.83	0.00	0.92	0.56	0.91	0.50	0.93	0.61
1cg5	18	103	0.97	0.83	0.96	0.78	0.95	0.72	0.96	0.78	0.96	0.78	0.96	0.78
1ctf	2	119	0.98	0.00	0.98	0.00	0.98	0.00	0.98	0.00	0.98	0.00	0.98	0.00
1dhn	14	107	0.91	0.29	0.89	0.14	0.88	0.07	0.87	0.00	0.87	0.00	0.87	0.00
1e6i	14	107	0.90	0.21	0.91	0.29	0.97	0.79	0.87	0.00	0.89	0.14	0.87	0.00
1elw	84	37	0.22	0.65	0.22	0.65	0.30	0.69	0.30	0.69	0.30	0.69	0.30	0.69
1enh	19	102	0.87	0.32	0.89	0.42	0.92	0.58	0.84	0.16	0.84	0.16	0.84	0.16
1ew4	16	105	0.94	0.62	0.93	0.56	0.95	0.69	0.90	0.38	0.90	0.38	0.91	0.44
1eyv	1	120	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00
1fkb	21	100	0.98	0.90	0.94	0.71	0.86	0.33	0.88	0.43	0.87	0.38	0.87	0.38
1fna	18	103	0.96	0.78	0.95	0.72	0.83	0.06	0.94	0.67	0.93	0.61	0.94	0.67
1gvp	1	120	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00
1hz6	18	103	0.90	0.44	0.92	0.56	0.88	0.33	0.85	0.17	0.86	0.22	0.86	0.22
1ig5	21	100	0.82	0.14	0.84	0.24	0.90	0.52	0.79	0.00	0.79	0.00	0.79	0.00
1iib	21	100	0.80	0.05	0.80	0.05	0.94	0.71	0.81	0.10	0.80	0.05	0.81	0.10
1kpe	17	104	0.97	0.82	0.97	0.82	0.93	0.59	0.93	0.59	0.94	0.65	0.92	0.53
1lis	1	120	1.00	1.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00
1lou	21	100	0.97	0.86	0.97	0.86	0.99	0.95	0.98	0.90	0.98	0.90	0.98	0.90
1nps	21	100	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.95	1.00	1.00	1.00	1.00
1opd	21	100	0.88	0.43	0.91	0.57	0.80	0.05	0.88	0.43	0.86	0.33	0.86	0.33
1pgx	71	50	0.26	0.48	0.28	0.49	0.40	0.58	0.40	0.58	0.52	0.66	0.46	0.62
1ptq	10	111	0.91	0.00	0.91	0.00	0.91	0.00	0.91	0.00	0.91	0.00	0.91	0.00
1r69	22	99	0.81	0.14	0.84	0.27	0.79	0.05	0.89	0.50	0.87	0.41	0.89	0.50
1rnb	19	102	0.98	0.89	0.97	0.84	0.97	0.84	0.89	0.42	0.89	0.42	0.90	0.47
1scj	21	100	0.79	0.00	0.79	0.00	0.95	0.76	0.80	0.05	0.80	0.05	0.80	0.05
1shf	21	100	0.87	0.38	0.87	0.38	0.79	0.00	0.88	0.43	0.92	0.62	0.90	0.52
1ten	21	100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1tig	21	100	0.96	0.81	0.96	0.81	0.79	0.00	0.95	0.76	0.95	0.76	0.97	0.86
1tul	21	100	1.00	1.00	1.00	1.00	0.99	0.95	0.99	0.95	0.99	0.95	0.99	0.95
1ubi	21	100	0.80	0.05	0.79	0.00	0.79	0.00	0.89	0.48	0.90	0.52	0.89	0.48
1ugh	6	115	0.96	0.17	0.96	0.17	0.96	0.17	0.96	0.17	0.96	0.17	0.96	0.17
1urn	20	101	0.92	0.60	0.91	0.55	0.88	0.40	0.94	0.70	0.92	0.60	0.94	0.70
1utg	2	119	0.98	0.00	0.98	0.00	0.98	0.00	0.98	0.00	0.98	0.00	0.98	0.00
1vcc	9	112	0.96	0.44	0.94	0.22	0.93	0.11	0.93	0.11	0.92	0.00	0.92	0.00
1vie	21	100	0.87	0.38	0.80	0.05	0.97	0.86	0.81	0.10	0.80	0.05	0.79	0.00
1vls	1	120	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00	0.99	0.00
1who	21	100	1.00	1.00	1.00	1.00	0.99	0.95	0.99	0.95	0.99	0.95	0.99	0.95
256b	2	119	0.99	0.50	0.99	0.50	0.98	0.00	0.99	0.50	0.98	0.00	0.98	0.00
2acy	21	100	1.00	1.00	1.00	1.00	0.96	0.81	1.00	1.00	0.99	0.95	1.00	1.00
2chf	21	100	0.98	0.90	0.97	0.86	0.99	0.95	0.96	0.81	0.94	0.71	0.96	0.81
2ci2	20	101	0.84	0.20	0.83	0.15	0.89	0.45	0.88	0.40	0.87	0.35	0.88	0.40
4ubp	3	118	0.97	0.00	0.97	0.00	0.97	0.00	0.97	0.00	0.97	0.00	0.97	0.00
5cro	21	100	1.00	1.00	1.00	1.00	0.96	0.81	0.82	0.14	0.83	0.19	0.81	0.10

Table 2.2: The specificity (Sp) and sensitivity (Sn) of SNAPP for decoy discrimination based on a native-like threshold of 1 Å for proteins from the Rosetta all-atom decoy set.

			SNAPP variations											
			Fold		Bala		Cammer		f_{edge}		$f_{interaction}$		$f_{occupancy}$	
PDB ID	N Native-like	N Decoy	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp
1a19	21	100	0.99	0.95	0.99	0.95	0.97	0.86	0.97	0.86	0.95	0.76	0.97	0.86
1a32	88	33	0.21	0.70	0.33	0.75	0.03	0.64	0.39	0.77	0.33	0.75	0.36	0.76
1a68	21	100	0.93	0.67	0.95	0.76	0.89	0.48	0.86	0.33	0.86	0.33	0.87	0.38
1acf	21	100	0.96	0.81	0.97	0.86	0.95	0.76	0.98	0.90	0.98	0.90	0.98	0.90
1ail	25	96	0.86	0.48	0.82	0.32	0.75	0.04	0.74	0.00	0.74	0.00	0.74	0.00
1aiu	102	19	0.47	0.90	0.42	0.89	0.47	0.90	0.42	0.89	0.47	0.90	0.42	0.89
1b3a	19	102	0.86	0.26	0.86	0.26	0.95	0.74	0.84	0.16	0.84	0.16	0.83	0.11
1bgf	21	100	0.98	0.90	0.98	0.90	0.97	0.86	0.94	0.71	0.95	0.76	0.94	0.71
1bk2	21	100	0.86	0.33	0.82	0.14	0.79	0.00	0.95	0.76	0.98	0.90	0.96	0.81
1bkr	21	100	1.00	1.00	1.00	1.00	0.97	0.86	0.99	0.95	0.99	0.95	0.99	0.95
1bm8	21	100	0.99	0.95	0.99	0.95	1.00	1.00	0.96	0.81	0.98	0.90	0.97	0.86
1bq9	21	100	0.83	0.19	0.83	0.19	0.79	0.00	0.84	0.24	0.87	0.38	0.85	0.29
1c8c	31	90	0.66	0.00	0.66	0.00	0.66	0.00	0.66	0.00	0.66	0.00	0.66	0.00
1c9o	21	100	0.79	0.00	0.79	0.00	0.79	0.00	0.80	0.05	0.82	0.14	0.80	0.05
1cc8	21	100	0.95	0.76	0.94	0.71	0.93	0.67	0.92	0.62	0.88	0.43	0.91	0.57
1cei	21	100	0.94	0.71	0.93	0.67	0.79	0.00	0.91	0.57	0.90	0.52	0.91	0.57
1cg5	21	100	0.99	0.95	0.99	0.95	0.98	0.90	0.99	0.95	0.98	0.90	0.98	0.90
1ctf	21	100	0.87	0.38	0.87	0.38	0.85	0.29	0.79	0.00	0.79	0.00	0.79	0.00
1dhn	21	100	0.86	0.33	0.83	0.19	0.84	0.24	0.80	0.05	0.81	0.10	0.80	0.05
1e6i	21	100	0.85	0.29	0.88	0.43	0.94	0.71	0.79	0.00	0.87	0.38	0.82	0.14
1elw	121	0	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00
1enh	46	75	0.73	0.57	0.73	0.57	0.81	0.70	0.73	0.57	0.68	0.48	0.71	0.52
1ew4	21	100	0.95	0.76	0.94	0.71	0.99	0.95	0.91	0.57	0.91	0.57	0.91	0.57
1eyv	21	100	0.99	0.95	0.99	0.95	0.99	0.95	0.92	0.62	0.94	0.71	0.92	0.62
1fkb	21	100	0.98	0.90	0.94	0.71	0.86	0.33	0.88	0.43	0.87	0.38	0.87	0.38
1fna	21	100	0.98	0.90	0.98	0.90	0.81	0.10	0.96	0.81	0.94	0.71	0.95	0.76
1gvp	13	108	0.89	0.08	0.88	0.00	0.93	0.38	0.88	0.00	0.88	0.00	0.88	0.00
1hz6	21	100	0.90	0.52	0.93	0.67	0.91	0.57	0.83	0.19	0.86	0.33	0.84	0.24
1ig5	22	99	0.82	0.18	0.84	0.27	0.91	0.59	0.79	0.05	0.79	0.05	0.79	0.05
1iib	28	93	0.76	0.21	0.75	0.18	0.97	0.89	0.76	0.21	0.76	0.21	0.76	0.21
1kpe	21	100	0.99	0.95	0.99	0.95	0.95	0.76	0.93	0.67	0.95	0.76	0.94	0.71
1lis	21	100	0.89	0.48	0.84	0.24	0.85	0.29	0.91	0.57	0.86	0.33	0.89	0.48
1lou	21	100	0.97	0.86	0.97	0.86	0.99	0.95	0.98	0.90	0.98	0.90	0.98	0.90
1nps	21	100	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.95	1.00	1.00	1.00	1.00
1opd	22	99	0.88	0.45	0.90	0.55	0.79	0.05	0.88	0.45	0.86	0.36	0.87	0.41
1pgx	121	0	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00
1ptq	21	100	0.79	0.00	0.79	0.00	0.79	0.00	0.82	0.14	0.80	0.05	0.81	0.10
1r69	79	42	0.55	0.76	0.55	0.76	0.71	0.85	0.48	0.72	0.52	0.75	0.48	0.72
1rnb	21	100	0.99	0.95	0.98	0.90	0.96	0.81	0.89	0.48	0.89	0.48	0.90	0.52
1scj	21	100	0.79	0.00	0.79	0.00	0.95	0.76	0.80	0.05	0.80	0.05	0.80	0.05
1shf	21	100	0.87	0.38	0.87	0.38	0.79	0.00	0.88	0.43	0.92	0.62	0.90	0.52
1ten	21	100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1tig	21	100	0.96	0.81	0.96	0.81	0.79	0.00	0.95	0.76	0.95	0.76	0.97	0.86
1tul	21	100	1.00	1.00	1.00	1.00	0.99	0.95	0.99	0.95	0.99	0.95	0.99	0.95
1ubi	21	100	0.80	0.05	0.79	0.00	0.79	0.00	0.89	0.48	0.90	0.52	0.89	0.48
1ugh	21	100	0.98	0.90	0.98	0.90	0.94	0.71	0.90	0.52	0.91	0.57	0.91	0.57
1urn	21	100	0.92	0.62	0.91	0.57	0.88	0.43	0.93	0.67	0.91	0.57	0.93	0.67
1utg	14	107	0.87	0.00	0.87	0.00	0.87	0.00	0.87	0.00	0.87	0.00	0.87	0.00
1vcc	21	100	0.93	0.67	0.91	0.57	0.85	0.29	0.86	0.33	0.79	0.00	0.79	0.00
1vie	21	100	0.87	0.38	0.80	0.05	0.97	0.86	0.81	0.10	0.80	0.05	0.79	0.00
1vls	17	104	0.88	0.29	0.88	0.29	0.91	0.47	0.92	0.53	0.92	0.53	0.92	0.53
1who	21	100	1.00	1.00	1.00	1.00	0.99	0.95	0.99	0.95	0.99	0.95	0.99	0.95
256b	53	68	0.56	0.43	0.59	0.47	0.50	0.36	0.63	0.53	0.66	0.57	0.65	0.55
2acy	21	100	1.00	1.00	1.00	1.00	0.96	0.81	1.00	1.00	0.99	0.95	1.00	1.00
2chf	21	100	0.98	0.90	0.97	0.86	0.99	0.95	0.96	0.81	0.94	0.71	0.96	0.81
2ci2	21	100	0.84	0.24	0.83	0.19	0.88	0.43	0.88	0.43	0.88	0.43	0.89	0.48
4ubp	20	101	0.88	0.40	0.91	0.55	0.95	0.75	0.81	0.05	0.80	0.00	0.80	0.00
5cro	21	100	1.00	1.00	1.00	1.00	0.96	0.81	0.82	0.14	0.83	0.19	0.81	0.10

Table 2.3: The specificity (Sp) and sensitivity (Sn) of SNAPP for decoy discrimination based on a native-like threshold of 2 Å for proteins from the Rosetta all-atom decoy set.

			SNAPP variations											
			Fold		Bala		Cammer		f_{edge}		$f_{interaction}$		$f_{occupancy}$	
PDB ID	N Native-like	N Decoy	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp	Sn	Sp
1a19	21	100	0.97	0.90	0.97	0.90	0.93	0.79	0.96	0.86	0.96	0.86	0.96	0.86
1a32	88	33	-	0.94	-	0.94	-	0.94	-	0.94	-	0.94	-	0.94
1a68	21	100	0.93	0.67	0.95	0.76	0.89	0.48	0.86	0.33	0.86	0.33	0.87	0.38
1acf	21	100	0.98	0.92	0.98	0.92	0.94	0.76	0.97	0.88	0.97	0.88	0.97	0.88
1ail	25	96	0.87	0.62	0.83	0.53	0.70	0.16	0.65	0.03	0.64	0.00	0.64	0.00
1aiu	102	19	-	0.98	-	0.98	-	0.98	-	0.98	-	0.98	-	0.98
1b3a	19	102	0.85	0.62	0.83	0.56	0.90	0.74	0.78	0.50	0.80	0.50	0.79	0.47
1bgf	21	100	0.98	0.90	0.98	0.90	0.97	0.86	0.94	0.71	0.95	0.76	0.94	0.71
1bk2	21	100	0.86	0.33	0.82	0.14	0.79	0.00	0.96	0.76	0.98	0.90	0.96	0.81
1bkr	21	100	1.00	1.00	1.00	1.00	0.97	0.86	0.99	0.95	0.99	0.95	0.99	0.95
1bm8	21	100	0.99	0.96	0.99	0.96	0.99	0.96	0.95	0.83	0.97	0.87	0.96	0.83
1bq9	21	100	0.74	0.70	0.68	0.62	0.57	0.50	0.63	0.55	0.68	0.62	0.65	0.59
1c8c	31	90	0.33	0.98	0.33	0.98	-	0.97	-	0.97	-	0.97	-	0.97
1c9o	21	100	0.62	0.95	0.69	0.96	0.54	0.94	0.69	0.96	0.62	0.95	0.69	0.96
1cc8	21	100	0.69	0.75	0.69	0.75	0.65	0.72	0.70	0.70	0.65	0.72	0.67	0.73
1cei	21	100	0.94	0.71	0.93	0.67	0.79	0.00	0.92	0.57	0.90	0.52	0.91	0.57
1cg5	21	100	0.99	0.95	0.99	0.95	0.98	0.90	0.97	0.95	0.98	0.90	0.98	0.90
1ctf	21	100	0.85	0.61	0.86	0.64	0.89	0.70	0.67	0.18	0.68	0.15	0.69	0.18
1dhn	21	100	0.86	0.33	0.83	0.19	0.84	0.24	0.79	0.05	0.81	0.10	0.80	0.05
1e6i	21	100	0.85	0.29	0.88	0.43	0.94	0.71	0.80	0.00	0.87	0.38	0.82	0.14
1elw	121	0	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00
1enh	46	75	0.20	0.97	0.20	0.97	0.40	0.97	-	0.96	-	0.96	-	0.96
1ew4	21	100	0.95	0.76	0.94	0.71	0.99	0.95	0.91	0.57	0.91	0.57	0.91	0.57
1eyv	21	100	0.99	0.95	0.99	0.95	0.99	0.95	0.90	0.62	0.94	0.71	0.92	0.62
1fkb	21	100	0.98	0.90	0.94	0.71	0.86	0.33	0.88	0.43	0.87	0.38	0.87	0.38
1fna	21	100	0.97	0.90	0.97	0.90	0.82	0.41	0.91	0.76	0.93	0.79	0.93	0.79
1gvp	13	108	0.84	0.24	0.83	0.19	0.93	0.67	0.79	0.00	0.79	0.00	0.79	0.00
1hz6	21	100	0.27	0.90	0.20	0.89	0.20	0.89	0.20	0.89	0.13	0.88	0.20	0.89
1ig5	22	99	0.56	0.78	0.56	0.78	0.63	0.81	0.37	0.68	0.59	0.79	0.44	0.71
1iib	28	93	0.71	0.85	0.71	0.85	0.62	0.80	0.67	0.82	0.67	0.82	0.67	0.82
1kpe	21	100	0.99	0.95	0.99	0.95	0.95	0.76	0.94	0.67	0.95	0.76	0.94	0.71
1lis	21	100	0.89	0.48	0.84	0.24	0.85	0.29	0.87	0.57	0.86	0.33	0.89	0.48
1lou	21	100	0.97	0.86	0.97	0.86	0.99	0.95	0.97	0.90	0.98	0.90	0.98	0.90
1nps	21	100	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.95	1.00	1.00	1.00	1.00
1opd	22	99	0.87	0.70	0.87	0.70	0.75	0.43	0.86	0.68	0.80	0.54	0.83	0.62
1pgx	121	0	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00
1ptq	21	100	0.79	0.00	0.79	0.00	0.79	0.00	0.80	0.14	0.80	0.05	0.81	0.10
1r69	79	42	0.50	0.99	0.50	0.99	-	0.98	-	0.98	-	0.98	-	0.98
1rnb	21	100	0.99	0.95	0.98	0.90	0.96	0.81	0.91	0.48	0.89	0.48	0.90	0.52
1scj	21	100	0.56	0.10	0.56	0.10	0.89	0.78	0.57	0.15	0.63	0.25	0.58	0.15
1shf	21	100	0.88	0.45	0.87	0.41	0.78	0.00	0.89	0.50	0.92	0.64	0.90	0.55
1ten	21	100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1tig	21	100	0.96	0.82	0.97	0.86	0.79	0.05	0.97	0.82	0.95	0.77	0.97	0.86
1tul	21	100	1.00	1.00	1.00	1.00	0.99	0.95	0.99	0.95	0.99	0.95	0.99	0.95
1ubi	21	100	0.50	0.96	0.50	0.96	0.50	0.96	0.50	0.96	0.50	0.96	0.50	0.96
1ugh	21	100	0.98	0.90	0.98	0.90	0.94	0.71	0.91	0.52	0.91	0.57	0.91	0.57
1urn	21	100	0.92	0.62	0.91	0.57	0.88	0.43	0.95	0.67	0.91	0.57	0.93	0.67
1utg	14	107	0.77	0.00	0.77	0.00	0.78	0.04	0.77	0.00	0.77	0.00	0.77	0.00
1vcc	21	100	0.93	0.67	0.91	0.57	0.85	0.29	0.79	0.33	0.79	0.00	0.79	0.00
1vie	21	100	0.87	0.38	0.80	0.05	0.97	0.86	0.80	0.10	0.80	0.05	0.79	0.00
1vls	17	104	0.88	0.45	0.88	0.45	0.91	0.59	0.89	0.59	0.90	0.55	0.91	0.59
1who	21	100	1.00	1.00	1.00	1.00	0.99	0.95	0.99	0.95	0.99	0.95	0.99	0.95
256b	53	68	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00	-	1.00
2acy	21	100	1.00	1.00	1.00	1.00	0.96	0.81	1.00	1.00	0.99	0.95	1.00	1.00
2chf	21	100	0.78	0.84	0.76	0.83	0.82	0.87	0.71	0.79	0.78	0.84	0.73	0.80
2ci2	21	100	0.84	0.24	0.83	0.19	0.88	0.43	0.88	0.43	0.88	0.43	0.89	0.48
4ubp	20	101	0.87	0.38	0.90	0.52	0.96	0.81	0.79	0.05	0.79	0.00	0.80	0.05
5cro	21	100	1.00	1.00	1.00	1.00	0.96	0.81	0.81	0.14	0.83	0.19	0.81	0.10

Table 2.4: The specificity (Sp) and sensitivity (Sn) of SNAPP for decoy discrimination based on a native-like threshold of 4 Å for proteins from the Rosetta all-atom decoy set.

	PDB ID	Arnautova	Fold	Bala	Cammer	Edge	Interaction	Occupancy
α-helical proteins	1a32	1.38	8.39	8.39	2.58	1.29	1.29	1.29
	1ail	3.77	0.00	0.00	6.90	7.48	7.92	8.26
	1bgf	10.74	0.98	1.07	10.54	10.54	10.54	10.54
	1bkr	0.97	0.62	0.62	14.12	0.62	0.62	0.62
	1cei	1.08	0.68	0.68	14.00	12.31	12.37	12.37
	1cg5	1.34	0.97	0.97	0.97	0.87	0.87	0.87
	1e6i	1.34	8.01	8.01	6.55	6.55	6.55	6.55
	1enh	2.74	2.85	2.85	1.89	0.94	1.03	0.94
	1eyv	1.68	1.52	1.52	1.59	1.27	7.15	7.15
	1lis	1.24	0.00	10.55	7.83	11.51	1.44	1.44
	1r69	1.00	1.35	1.35	1.48	0.61	2.54	0.61
	1utg	4.63	4.46	4.46	4.73	4.66	4.66	4.66
	1vls	10.67	1.47	1.47	7.01	1.47	7.01	1.47
α/β proteins	1a19	1.08	0.73	0.73	2.74	0.73	0.75	0.72
	1a68	0.77	10.29	10.29	12.61	0.67	11.43	11.43
	1acf	3.66	0.00	0.00	3.14	1.03	0.93	0.93
	1aiu	1.61	1.39	0.78	1.51	1.55	0.76	1.55
	1bm8	0.85	0.00	0.00	0.93	0.00	0.61	0.73
	1ctf	1.45	3.34	3.34	3.98	8.89	8.89	8.89
	1dhn	1.87	9.78	9.78	11.18	14.59	16.68	14.59
	1ew4	1.45	0.00	0.00	0.85	0.00	7.41	0.00
	1hz6	3.66	1.04	1.04	4.22	3.83	3.89	3.83
	1iib	1.02	1.86	1.86	1.93	2.76	2.70	2.69
	1kpe	1.51	1.26	1.26	12.25	6.81	1.26	1.26
	1lou	0.95	6.16	6.16	16.38	6.16	6.16	6.16
	1opd	0.93	4.62	4.62	2.72	4.62	0.58	0.58
	1pgx	1.07	0.86	1.24	1.12	1.30	1.34	1.30
	1rnb	2.06	0.95	0.95	13.41	13.88	15.87	15.87
	1scj	7.73	6.74	2.60	5.63	6.68	7.16	6.68
	1tig	1.06	0.81	0.87	11.56	0.81	0.81	0.81
	1ubi	0.83	2.24	2.24	2.78	2.73	0.65	2.63
	1ugh	1.55	0.00	0.00	7.38	8.57	8.83	8.83
	1vcc	1.65	0.87	6.17	5.05	7.23	7.45	7.23
	2chf	0.67	0.70	0.70	0.68	0.54	4.32	0.67
	2ci2	9.60	9.59	11.63	9.38	0.72	9.51	0.72
	4ubp	9.27	8.79	8.79	11.61	7.43	8.07	7.43
	5cro	8.34	0.68	0.70	0.70	6.61	6.61	6.61
β proteins	1bk2	7.36	7.22	7.22	7.14	0.62	0.83	0.62
	1fna	0.90	1.21	3.28	4.06	0.00	3.28	0.00
	1gvp	15.04	14.54	14.78	9.35	14.54	14.54	14.54
	1shf	0.90	4.44	4.44	7.38	3.45	3.45	3.45
	1ten	0.69	0.62	0.62	0.62	0.61	0.62	0.62
	1tul	1.16	0.81	0.65	0.90	0.65	0.65	0.65
	1vie	6.41	8.06	8.06	0.52	7.79	7.79	7.79
	1who	0.95	0.73	0.73	0.88	0.88	0.88	0.88
Unpredicted by Arnautova et al.	1b3a	-	7.95	7.95	0.73	7.95	7.95	7.95
	1bq9	-	4.41	4.41	6.67	2.67	2.67	2.67
	1c8c	-	2.52	2.52	2.33	2.52	2.52	2.52
	1c9o	-	2.79	3.12	2.90	3.52	2.74	3.52
	1cc8	-	2.96	2.96	7.64	8.06	8.06	8.06
	1elw	-	0.78	1.68	1.68	1.68	0.00	0.00
	1fkb	-	0.66	13.95	13.95	14.05	14.05	14.05
	1ig5	-	1.72	1.72	4.04	4.04	4.08	4.08
	1nps	-	0.62	0.62	0.65	0.62	0.62	0.62
	1ptq	-	11.62	10.48	11.74	10.48	10.48	10.48
	1urn	-	0.00	0.00	9.22	0.00	0.00	0.00
	256b	-	0.00	0.00	2.10	1.68	2.03	2.03
	2acy	-	0.61	0.61	0.65	0.66	0.66	0.66

Table 2.5: The rank of the native protein (NR), and specificity (Sp) and sensitivity (Sn) of SNAPP for decoy discrimination of proteins from the Rosetta all-atom decoy set.

of the published SNAPP variations provided exact details of the implementation method used to score tetrahedra and their proteins, and our implementation may have differed from that used in the literature. We suspect this is especially the case with SNAPP-Cammer.

Unfortunately, none of our frequency variations fared as well as SNAPP-Fold or SNAPP-Bala. In certain cases, the frequency variations performed exceptionally well when all other SNAPP potentials failed, but a quick glance did not reveal a prediction pattern for the frequency variations. We propose that a complete redesign of the SNAPP potentials using the frequency variations could improve decoy fold prediction; however, we leave that experiment for future studies.

Although we had hoped to see a cleaner discrimination between native-like and decoy folded protein structures, the purpose of testing against protein folding was not to improve decoy fold discrimination, but to compare our new SNAPP potentials against the existing variations. To this extent, the above experiments proved useful: We found that SNAPP-Fold consistently outperformed all other SNAPP potentials, and we will use SNAPP-Fold as a control when designing the SNAPP potentials for PPI.

2.4.3 Evaluation of SNAPP descriptors

To our knowledge, cheminformatics-like descriptors have not been previously applied to evaluate protein packing in relation to either protein folding or protein interactions. Without previous results or an established benchmark to compare against, we opted to forgo descriptor analysis on protein folding and instead focus on descriptors for protein interactions. We calculated the novel SNAPP descriptors for docking decoys in the Dockground decoy dataset [70], which contains 61 different protein complexes, each with one native complex, one to twelve native-like complexes, and one hundred decoy poses. We define the *target interface* as the interface between the chains given by the dataset, and we tested to see if the SNAPP descriptors could discriminate between native-like and decoy complexes.

For each protein in the Dockground decoy dataset, we tessellated each complex and selected all interfacial simplices, which we define as simplices that contain vertices from both chains at the target interface. Descriptors were calculated for each simplex individually and applied to describe the interface as a whole, depending on the trait in question. For instance, the volume of an interface was calculated by adding the volumes of all participating simplices, whereas the surface area of an interface included only those triangular faces external to the interface, and tetrahedrality descriptors were averaged across all interfacial simplices. Any descriptors caught deserting were immediately put to the sword. Unfortunately, we found very little correlation across the entire dataset between any single or paired descriptor and the RMSD of the complex, although the descriptor-RMSD correlation varied from complex to complex. When we plotted the RMSD-descriptor points, we found that the native and native-like interfaces clustered somewhere along the y-axis while the decoys showed a u-shaped RMSD-descriptor correlation Figure 2.5, which is to say no correlation at all. For more than 60% of the protein complexes in the dataset, most of the native-like complexes were easily identifiable using one or more of the descriptors; however, the native complex often ended up buried beneath the native-like complexes and two or three high-RMSD decoys.

The small range of descriptor values displayed by native-like complexes suggested a potential problem with discrimination of high-resolution structures, and the small number of decoys for each complex limited our ability to evaluate the descriptors. As an additional test of decoy discrimination, we decided to evaluate our descriptors against a random dataset. Using our POPP docking algorithm, described in Chapter 4.1.1, we compiled a series of 6,000 randomly generated docking poses based on the native structure for phospholipase A2 in complex with a synthetic pentapeptide (PDB code 1TKJ). We calculated descriptors for each pose and checked for a correlation with RMSD (Figure 2.6). Unfortunately, we found even less correlation between the descriptors and RMSD when the RMSD range was lowered. Although many of the docking poses are native-like, none of the descriptors were sensitive enough to identify the native pose, and only a few of the descriptors were able identify native-like poses.

Although the SNAPP descriptors were unable to efficiently differentiate between decoys

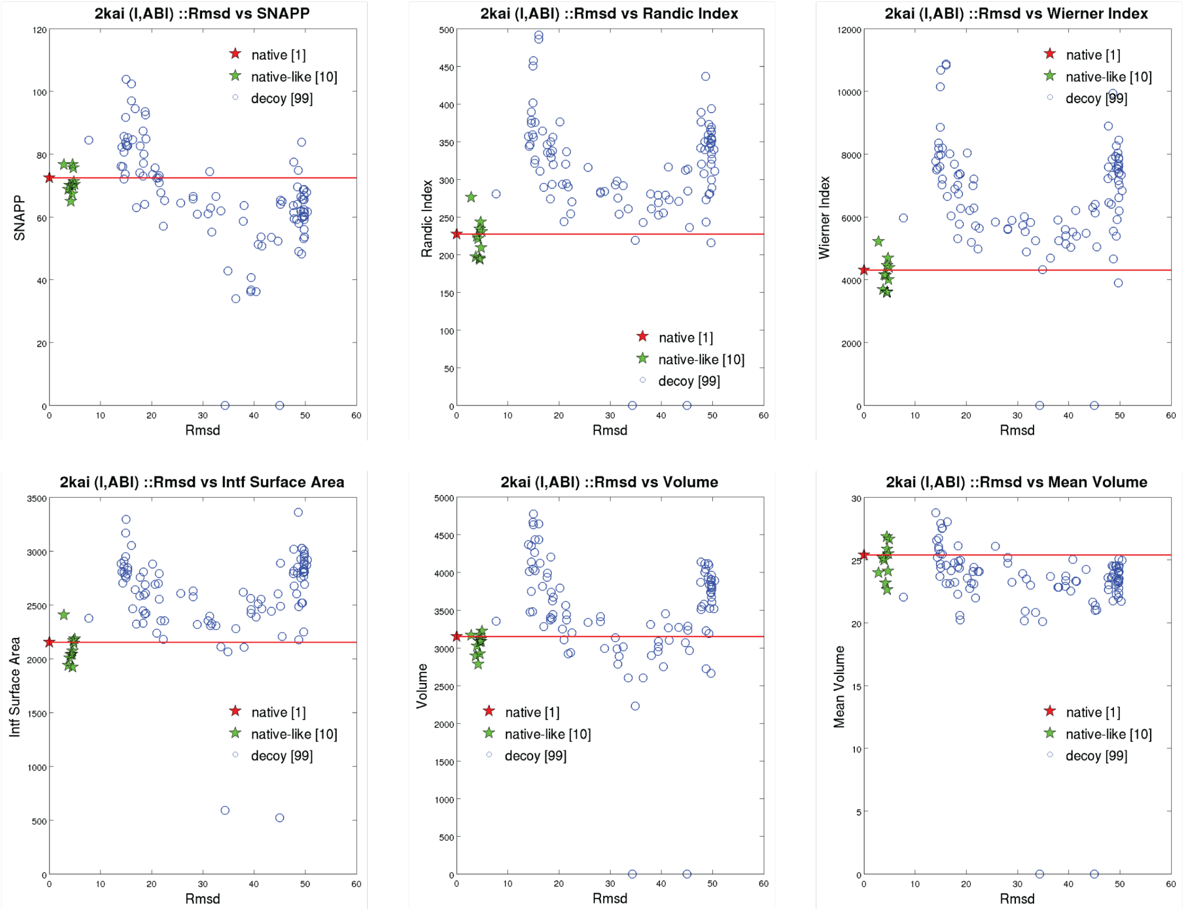

Figure 2.5: RMSD versus SNAPP Descriptors for porcine kallikrein A bound to bovine trypsin inhibitor (PDB code 2KAI). The red horizontal line in each graph shows the value of a descriptor for the native complex relative to the others. Despite the lack of a correlation with RMSD, three of the descriptors (the Randic and Weiner Indices and the interfacial surface area) were able to discriminate between most of the native-like and the decoy poses.

for similar structures with similar RMSD, we hypothesized that the SNAPP descriptors could be used differentiate between complexes with different structural interfaces. It is known that many interfaces have conserved structure and sequence to ensure functional domains remain intact [71, 1, 72, 73]. To test whether SNAPP descriptors could be used to identify functionally distinct groups of proteins, we generated descriptor fingerprints for each of the native complexes in the Dockground decoy dataset. The fingerprints were clustered using the `dendogram` function in MatLab (Figure 2.7), and we were able to identify several subgroups of functionally related proteins within the dendogram clusters.

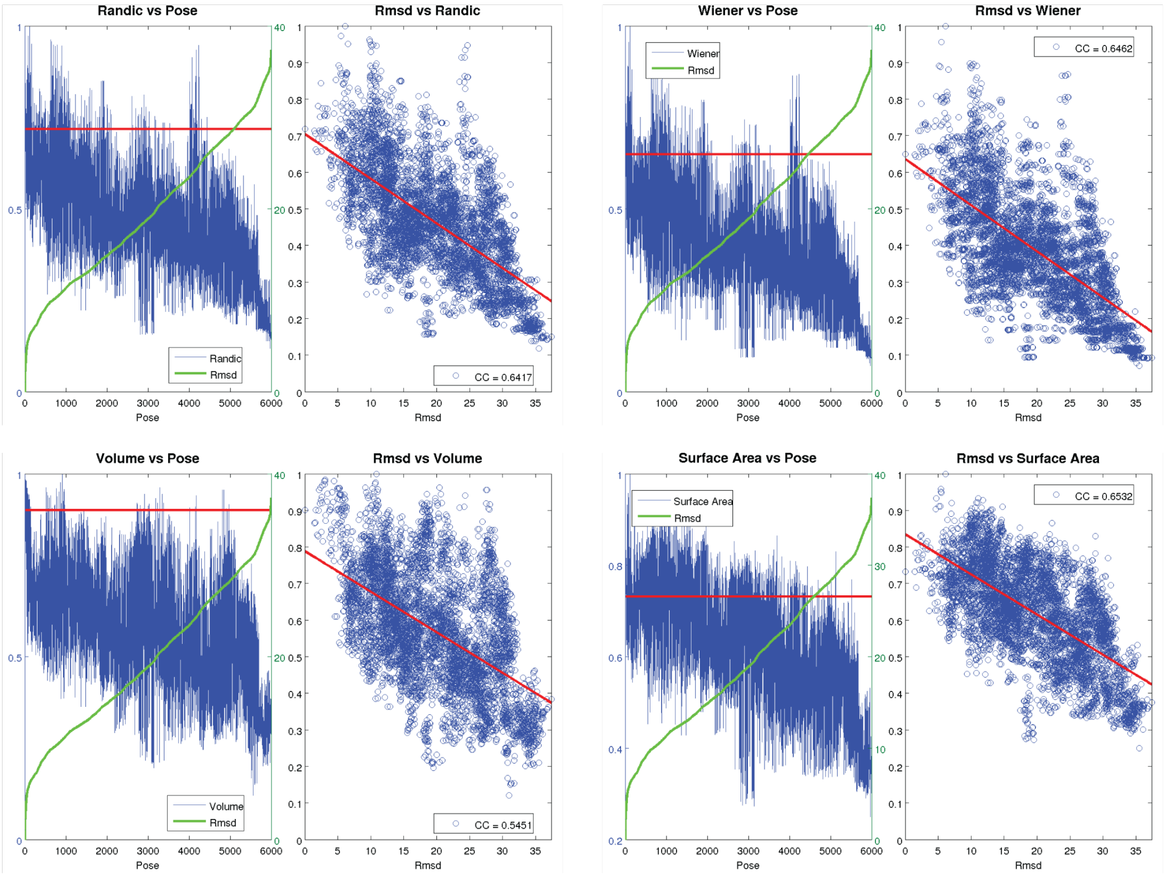

Figure 2.6: The distribution of descriptor values for 6,000 decoy poses for phospholipase A2 in complex with a synthetic pentapeptide (PDB code 1TKJ), ranked by RMSD. The left side of side of each graph also shows the RMSD for each pose as a green line and the descriptor value for the native complex is given as a red line. On the right, the linear fit is given as a red line.

Overall, the current protein descriptors were only able to weakly discriminate between docking decoys, but our results suggest that they may be able to differentiate between functionally related proteins. However, as the implementation suggests, the use of the interfacial descriptors requires a three-dimensional structure of both the protein and the ligand in question; although potentially useful for studies where the interaction is already known, we decided to instead focus on generating a set of SNAPP potentials that could be used to evaluate proteins and protein interactions without *a priori* knowledge of the interface. We propose that the protein descriptors could be further refined for inclusion as a refinement step during protein-protein or protein-peptide docking in future work.

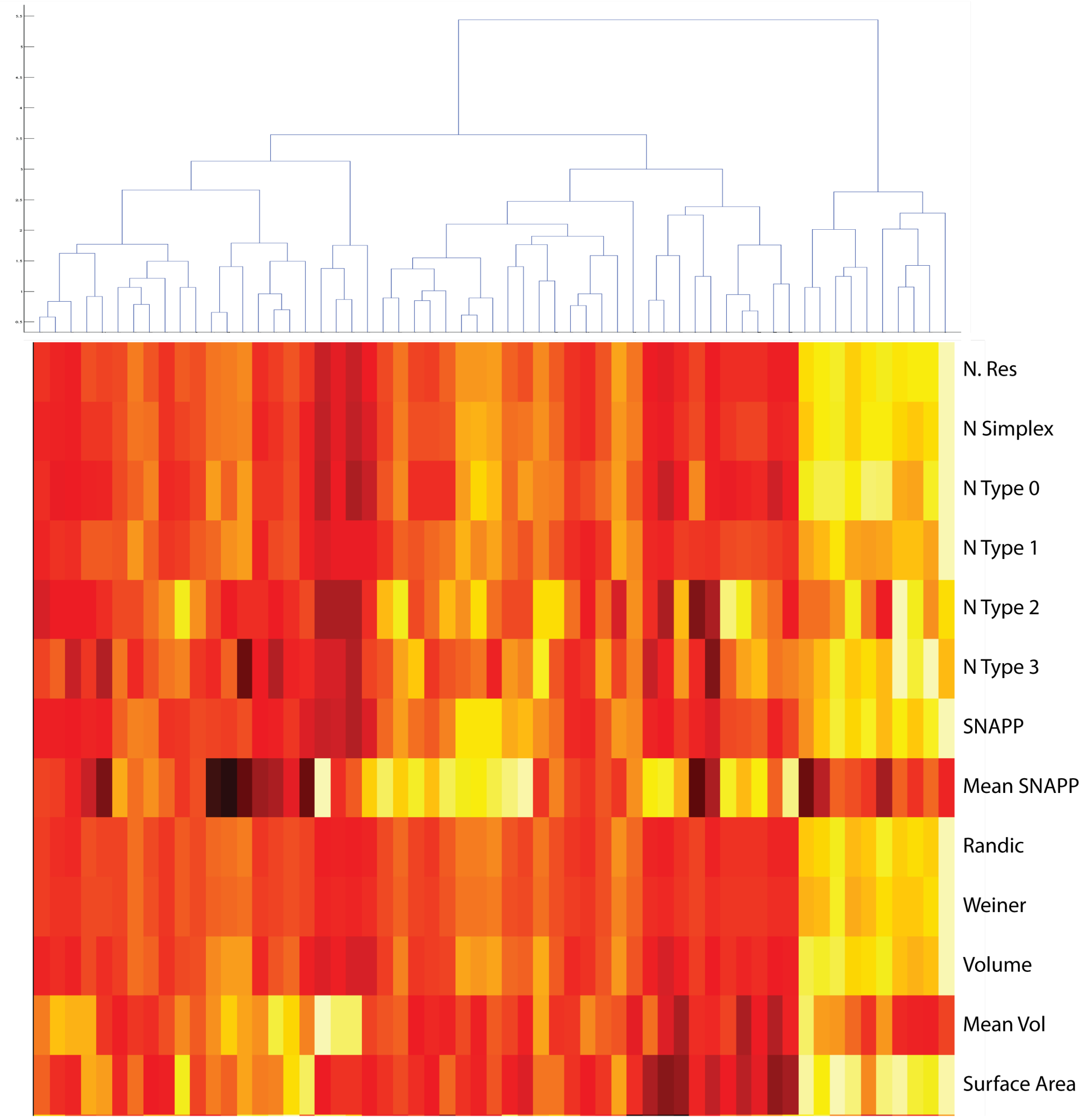

Figure 2.7: The dendogram and heat map of the SNAPP descriptors for the Dockground decoy dataset.

2.5 Specializing for Protein Interactions

In cheminformatics, an Applicability Domain (AD) of a Quantitative Structure Activity Relationship (QSAR) model is the region of chemical space that is similar to compounds found in the training set for which a model is expected to yield accurate predictions [74, 75]. In the same manner, we must consider the AD for SNAPP; all previous versions of SNAPP were trained, tested and applied to single chain, folded protein structures and are potentially outside AD for PPI prediction. Ofran and Rost [76] quantified the difference between six types of protein-protein interactions, including two internal (intra-domain and domain-domain) and four external (homo-obligomers, homo-complexes, hetero-obligomers, hetero-complexes) interactions, and found different amino acid distributions and pairwise contacts for each of the six types. They found that the residue and contact differences between each of the six types of interfaces was in fact sufficient to quantitatively discriminate between the other interface types – even between the two internal interactions. With this in mind, we set out to define a new set of SNAPP potentials specifically designed to predict protein interactions.

2.5.1 Redesigning SNAPP for Protein Interactions

For this project, our goal is two-fold: (1) to predict where protein interactions occur, i.e., binding sites, on protein surfaces; and (2) predict and evaluate conformations of protein interactions. Although very similar, both problems require subtly different approaches. In addition, we further split each set of potentials based on the type of interface, i.e., from either a homo- or hetero-complex, used to train.

SNAPP for Binding Site Prediction

In SNAPP-Fold, the observed and expected frequencies f and p reflect the distribution of simplices and amino acid residues of single chain, folded protein structures; to evaluate the likelihood that a particular simplex will form between two proteins, we need to evaluate the likelihood that any given residue will participate in the interface. Thus, we let the observed frequency f be the frequency of interfacial simplices found in a given dataset, and we let the expected frequency reflect the amino acids available to form interfacial simplices, i.e., surface residues. We redefined the amino acid frequency a_i from Equation 2.11 to reflect the frequency that a given amino acid i will occur on the surface:

$$a_i = \frac{|A_{surface}(i)|}{|A_{surface}|} \tag{2.29}$$

where $|A_{surface}(i)|$ is the number of times amino acid type i is found on the surface of a protein in the dataset versus all amino acids on the surface of all proteins of the dataset $|A_{surface}|$. The new SNAPP-Surface potential reflects the likelihood that a simplex will form between two protein surfaces. However, a simplex formed from two separate protein chains cannot have four sequentially adjacent residues, and the type 4 tetrahedra (Figure 2.1C) will never occur. Fortunately, the potentials will naturally reflect this change, and no special cases need to be written into the algorithm.

SNAPP for Interface Prediction

To evaluate the likelihood of a given interface, we developed the SNAPP-Interface potentials. In the same manner as the SNAPP-Surface potentials, the SNAPP-Interface potentials redefine the data used to compute the observed frequency f and expected frequency p. The observed frequency f also uses the interfacial simplices found in the dataset; however, the expected frequency p instead uses the frequency of amino acids found at an interface:

$$a_i = \frac{|A_{\mathit{interface}}(i)|}{|A_{\mathit{interface}}|} \tag{2.30}$$

Homo- versus Hetero-complexes

We decided to create two sets of potentials for both SNAPP-Surface and SNAPP-Interface based on separate training sets containing either homo- or hetero-complexes. Ofran and Rost further defined protein interactions as either obligatory, i.e. an obligomer, or transient, i.e., a complex. Obligatory interactions were defined as any interaction that typically lasted the life of the protein, such as the interaction between different chains of a hemoglobin molecule, whereas the transient interactions were temporary, such as that of an enzyme and substrate. Due to the small number of crystal structures, we differentiate only between homo- and hetero-complexes, resulting in a total of four SNAPP scoring functions specifically designed for analysis of protein interfaces: SNAPP-Surface:Homo, SNAPP-Surface:Hetero, SNAPP-Interface:Homo, and SNAPP-Interface:Hetero. A brief comparison between each of the five scoring functions is given in Table 2.6.

	Trained On				Tested On				Used to Predict		
SNAPP Version	Single Chain Proteins	Homo-complexes	Hetero-complexes	Protein-Peptide Complexes	Single Chain Proteins	Homo-complexes	Hetero-complexes	Protein-Peptide Complexes	Protein Folding	Binding Sites	Protein-Peptide Docking
Fold	✓				✓	✓	✓	✓	✓	✓	
Surface:Homo		✓				✓	✓	✓		✓	✓
Surface:Hetero			✓			✓	✓	✓		✓	✓
Interface:Homo		✓				✓	✓	✓			✓
Interface:Hetero			✓			✓	✓	✓			✓

Table 2.6: A breakdown of the types of data each of the SNAPP scores was trained on and how they are applied.

2.5.2 Training Set – Dockground Database

The Dockground dataset [77] was used to train two separate SNAPP scoring functions for identifying protein hot spots, SNAPP-Homo and SNAPP-Hetero (see Figure 2.3A for details about the dataset). We downloaded the list of automatically selected representative complexes from the Bound-Bound dataset, removed any self-interacting proteins, i.e., proteins whose interaction is listed as the same chain, and split the dataset into homo- and hetero-complexes. Each complex consisted of two or more protein chains; however, the dataset identifies a single target interface by specifying exactly two interacting chains. We decided to limit curation of the Dockground dataset to removal of PDBs containing an iCode due to two limiting factors: First, the use of only interfacial simplices yielded far less data with which to train the scoring functions, a full order of magnitude less than the equivalent amount of data for protein folding. Second, residues on the surfaces of proteins typically assume multiple rotameric states, and as such, the side chain atoms may be missing atoms or absent entirely in the crystal structure. The SNAPP-Surface:Homo and SNAPP-Surface:Hetero scoring functions are used in the CRACLe algorithm, and their evaluation will be discussed in the next chapter.

2.6 Conclusion

We have described the development of three new SNAPP potentials, including SNAPP-Fold, SNAPP-Surface, and SNAPP-Interface, suitable for the purpose of evaluating protein packing of protein folding and protein interfaces. SNAPP-Fold was created to take advantage of the explosive growth of protein structures available in the PDB and provide an updated version of the existing SNAPP scores. We also developed three new variations of the SNAPP scoring function that used a modified expected simplex frequency. Although each of these variations outperformed the other versions of SNAPP in several cases, their overall performance was decreased. Instead, we found that SNAPP-Fold outperformed both the old and new variations. The equations used to create SNAPP-Fold were also used to compile the two new SNAPP scores, SNAPP-Surface and SNAPP-Interface. These new SNAPP scores were designed to evaluate two separate aspects of protein interactions, and will be discussed in more depth in the next two chapters.

CHAPTER 3

Predicting Sites of Protein Interactions

In this chapter, we cover the development of the Critical Residue Analysis and Complementarity Likelihood (CRACLe) algorithm and software for identifying hot spot residues and binding sites. CRACLe has been developed as a rapid method for computational high-throughput screening of individual proteins to identify potential binding sites on the protein surface rather than a singular, time-consuming, high-resolution docking solution. We show that CRACLe is capable of correctly predicting binding sites in more than 85% of proteins from the PepX test set [78], 88% from the ZDock Bound test set [79], and 83% from the ZDock Unbound test set [79]. CRACLe is computationally efficient, capable of predicting binding sites for over 1,000 proteins in under seven minutes on a standard desktop computer versus PredUs [30], which required the same amount of time to run a single protein on its web server. This high-throughput prediction of putative protein-protein binding sites could enable building of protein interaction networks, provide an assessment of potential drug targets for peptide inhibitors, or provide a scoring filter in PPI decoy discrimination.

3.1 Materials and Methods

As mentioned in the previous chapter, our research group pioneered the use of a computational geometry technique called Delaunay tessellation in protein structure analysis, which resulted in the SNAPP scoring function. The last iteration of SNAPP was compiled in 2003, and due to the explosive growth in crystal structures in the PDB, we recently compiled a representative and highly curated dataset of 1,473 non-redundant single chain proteins from three independent databases. These SNAPP-Fold potentials outperformed previous SNAPP iterations in decoy discrimination on three independent datasets. Due to the difference in nature between internal protein folding and PPIs, we defined and compiled two novel SNAPP scoring functions called SNAPP-Surface and SNAPP-Interface that were trained on tessellated protein-protein interfaces for homo-complexes and hetero-complexes.

In this chapter, we focus on the SNAPP-Surface potentials. We used 1,448 and 540 protein complexes from the Dockground dataset to train two the new SNAPP-Surface scoring functions for evaluating protein surfaces, respectively called SNAPP-Surface:Homo and SNAPP-Surface:Hetero. Both of these scoring functions use the same equations given for SNAPP-Fold (Equation 2.9-2.12) with two important distinctions: First, the observed frequency f included only simplices whose residues were found at protein-protein interfaces. Interfacial residues are commonly identified by their proximity to residues of their binding partners [80, 81]. Similarly, we defined interfacial residues based on the presence of a Delaunay edge, i.e., an edge defined by Delaunay tessellation, between the residues of interacting proteins. Second, the expected frequency p utilized only amino acid frequencies a_i of surface residues, i.e., solvent-exposed residues on protein surfaces. The use of surface residue frequencies was intended to enable SNAPP to discriminate between interfacial and non-interfacial surface residues rather than evaluating the likelihood of the interface itself.

3.1.1 Using Surface Residue Triplets to Identify Putative Binding Sites

As previously described, the Delaunay tessellation of a set of protein-points yields a convex hull composed of Delaunay tetrahedra. The convex hull defines a set of simplices with one or more triangular faces that are not shared with an adjacent simplex; however, as the convex hull does not accurately describe the shape of the protein, we removed all simplices with edge lengths greater than a certain threshold. After some experimentation, we selected a threshold of 11.5 Å as the minimum distance to allow all residues to retain Delaunay edges.

The resulting hull is no longer convex, but effectively defines the solvent accessible surface of the protein by the unshared triangular faces that we call surface residue triplets. These surface residue triplets characterize the surface topology of a protein (Figure 3.1A) and provide a unique and critical basis for scoring protein surface residues using SNAPP. Surface residue triplets define a surface topology that is dependent on the distance threshold for edge removal; although other methods such as α-shapes [36] have been used to remove Delaunay edges, we have found removal of edges based on length is not only consistent but computationally simpler.

By definition, triplets cannot be scored using the four-body SNAPP scoring function. However, a triplet at a protein-protein interface would form a new simplex when tessellated with the binding partner, resulting in a SNAPP-scorable, four-body simplex. Such an interfacial tetrahedron would have a constrained simplex type (limited to type 0, 1, or 3–see Figure 2.1C for type definition) based on the sequence adjacency of the surface residue triplet. Based on this concept, we define an *ad hoc* simplex built on the triplet, but we allow the composition of the fourth residue to vary, yielding a modular but SNAPP-capable scoring function. We hypothesize that particular fourth residue compositions may provide additional stability and a lower binding free energy for the PPI and that these particular compositions will also yield higher SNAPP scores, allowing us to identify (1) triplets that are likely to form more favorable interfacial tetrahedra and (2) the composition of potential surface residues that will maximize the stability of a particular interfacial tetrahedron formed with a given triplet.

Therefore, for each surface residue triplet t_i with a given residue composition and sequence adjacency, we define an *ad hoc* simplex (Figure 3.1B) whose tetrahedral type is defined by the sequence adjacency of the triplet residues and the non-adjacent residue X, thus limiting each *ad hoc* simplex to type 0, 1, or 3. Composition of each ad hoc simplex is defined by the triplet residues and an *ad hoc* residue X, where X represents the set of all 20 naturally

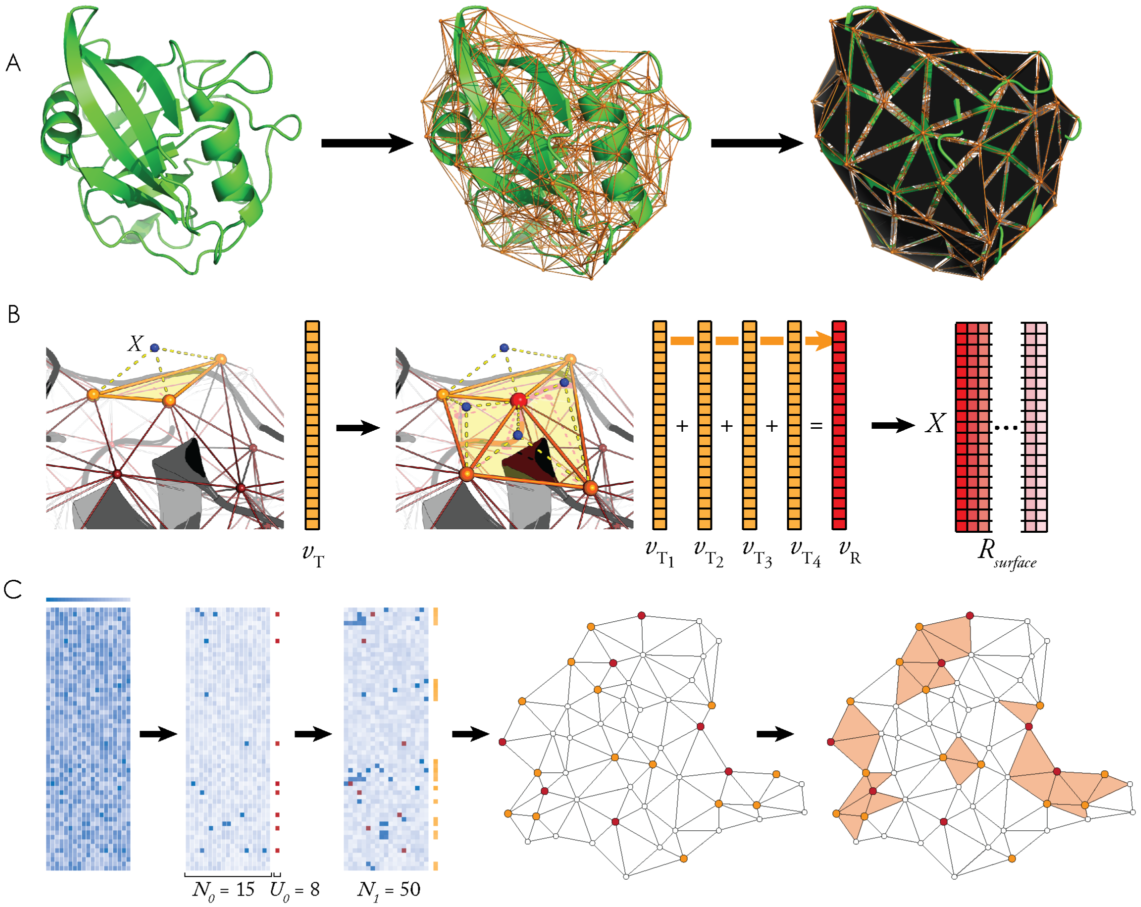

Figure 3.1: The CRACLe workflow. (A) Delaunay tessellation of a protein structure using a single-point-per-residue model to identify the solvent-exposed simplex faces, i.e., surface residue triplets. (B) For each surface residue triplet, we evaluate the likelihood of a potential interaction between the triplet and each of the 20 standard amino acids, represented by the imaginary residue X, resulting in a triplet feature vector v_T of 20 SNAPP scores. A summation of all triplet feature vectors that contain a single residue in common yields a residue feature vector v_R for each surface residue. We then concatenate each surface residue feature vector to form the SNAPP pairing matrix, where each cell contains the pairing potential between a surface residue and a particular amino acid. (C) Each pairing potential in the SNAPP pairing matrix is ranked according to the highest potential. The top N_0 pairing potentials are identified, and up to U_0 unique surface residues are identified as primary critical residues. The top N_1 pairing potentials are then identified as secondary critical residues and mapped onto the protein surface. Binding sites are predicted based on the clustering of primary and secondary critical residues. Both the function-based and the maximum-potential algorithms follow this generic workflow.

occurring amino acids, resulting in a 1×20 triplet feature vector of SNAPP scores, v_T:

$$v_T(i,j) = \big[q(t_i, X_j)\big]_{20} \tag{3.1}$$

Each of the twenty SNAPP scores in v_T is a likelihood function of simplex occurrence, but because it is a logarithm, we are able to use vector summation to calculate the likelihood of any two triplets occurring together. Such a summation essentially calculates a local SNAPP score, similar to Equation 2.5, that is dependent on the value of X.

All triplet feature vectors that contain a mutual vertex are added together using a vector summation to generate a residue feature vector v_R for each surface residue r_i (Figure 3.1B). Thus, each of the twenty scores in v_R is the summation of SNAPP scores for a simplex composed of a particular residue X the neighboring triplets. Each v_R score estimates the likelihood that the surface residue r_i will interact with a particular residue X, and we call this statistical likelihood a pairing potential. A second residue feature vector, $v_R\prime$, is also created by dividing each residue feature vector v_R by the number of contributing triplets.

$$v_R(r_i) = \sum v_T : r_i \in v_T \tag{3.2}$$

$$v_R\prime = \frac{v_R}{|v_T|} \tag{3.3}$$

Both the v_R and $v_R\prime$ feature vectors are independently normalized using a z-score and concatenated to form two independent, protein-specific SNAPP pairing matrices (Fig. 2b) with dimensions $N_{AA} \times N_{SR}$, where N_{AA} is the number of amino acids in the alphabet and N_{SR} is the number of surface residues on the protein. Columns contain the scores for each surface residue ($1 \times N_{AA}$), and rows contain the scores for each *ad hoc* residue X ($1 \times N_{SR}$). By definition, each cell contains the pairing potential s_{ij} for an interaction between a given surface residue r_i and a particular *ad hoc* amino acid X_j. Both scoring matrices are used to predict residues in the protocol described below.

3.1.2 Critical Residue Analysis

CRACLe provides two algorithms for binding site prediction. Both use the same underlying methodology, but the first returns four sets of predictions, one for each of the three SNAPP scoring functions (SNAPP-Fold, SNAPP-Surface:Homo, and SNAPP-Surface:Hetero) and a consensus of the three, while the second returns only a single set of predictions based on the maximum pairing potentials from each of the three SNAPP scoring functions. In both algorithms, CRACLe utilizes the two SNAPP pairing matrices independently to identify likely interface residues in three stages (Figure 3.1C). The first two stages identify the most likely critical residues by selecting the highest scoring pairing potentials; the third stage uses surface topology to group selected critical residues into potential binding sites. The maximum-potential algorithm is set by default, but the function-based algorithm may be invoked with the '`-cracle_function_based`' flag.

Function-based Algorithm

The first and second stages utilize the pairing matrices to identify primary and secondary critical surface residues most likely to participate in PPIs. In the first stage, all of the pairing potentials in a given matrix are sorted, and the top N_0 highest-scoring pairing potentials (typically set at 10-20) are selected to find up to U_0 (typically 5-10) unique residues. These primary critical residues provide a starting point from which other critical residues are selected, and as a result, both N_0 and U_0 parameters play an important role in determining the sensitivity of the algorithm for correctly identifying critical residues. In the next stage, CRACLe extends the previous search from the top N_0 residues to the top N_1 residues (typically 30-50); however, these secondary critical residues must share a Delaunay edge with a primary residue, i.e., have a surface tessellation graph distance of one.

The third stage uses sub-graph mining of the Delaunay tessellation to cluster primary and secondary critical residues into potential binding sites of two or more surface residues. To participate in a sub-graph, a residue vertex must share a Delaunay edge with at least two other predicted residues or a single Delaunay edge with a critical residue and common non-critical neighboring surface residue. CRACLe returns all sub-graphs with three or more vertices as potential binding sites, and any sub-graphs with only two vertices as potential binding site extensions. Isolated primary and secondary residues are ignored.

Maximum-Potential Algorithm

Primary and secondary critical residues are selected in the same manner as described above with two important distinctions. First, the U and N parameters are defined dynamically using a $\log$ function dependent on the number of surface residues for a given a protein. Second, secondary residues are not required to have any connection with a primary residue. Residues are selected from both of SNAPP pairing matrices from each of the SNAPP scoring functions.

In the third stage, critical residues are clustered in three steps. In the first step, each primary critical residue along with surface-adjacent secondary residues are clustered to form binding sites. Next, clusters of three adjacent secondary residues form additional binding sites. Lastly, binding sites from both of the previous steps are merged; any binding sites that share two or more residues are merged into a single binding site. At each step, the edges between any two adjacent surface residues are verified by comparison against existing surface triplets to ensure the edge exists on the surface rather than simply between two surface residues. Isolated primary and secondary residues are ignored, and binding sites containing only two critical residues are ignored. Predicted critical residues that participate in a binding site are predicted to be hot spots.

3.1.3 Training and Test Sets

The selection and curation of the data used to create the scoring function directly relates to the algorithms efficacy and applicability. The databases used for training are summarized in Figure 2.3 and are described in greater detail in Chapter 2.5.2. The datasets used for testing the SNAPP-Fold, SNAPP-Surface:Homo, and SNAPP-Surface:Hetero scoring functions are summarized in Table 3.1. Both the Dockground and PepX datasets were analyzed using both CRACLe algorithms; the ZDock dataset was tested only on the maximum-potential algorithm.

	Dockground			ZDock	
	Homo	Hetero	PepX	Bound	Unbound
			Protein Complexes		
N structures	1,325	525	1,431	176	176
Missing atoms	-	-	-	-	-
Missing residues	-	-	-	-	-
Contained iCode	20	70	354	41	47
After curation	*1,305*	*455*	*1,077*	*135*	*129*
N unique	**1,305**	**455**	**1,077**	**135**	**129**
N simplexes	**253,031**	**104,165**	-	-	-

Table 3.1: SNAPP-Surface Test Sets

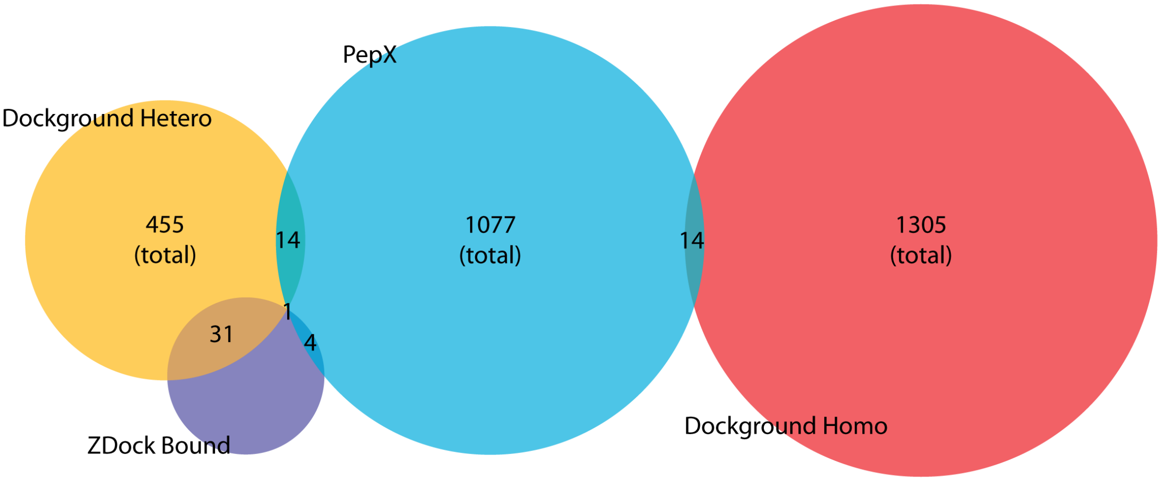

Figure 3.2: The overlap between the Dockground, PepX and ZDock Bound test sets.

The PepX dataset [78] was selected as an independent dataset for testing each of the three scoring functions. Each of the 1,431 complexes contains the crystal structure of one or more peptides bound to one or more proteins. For each complex, the interacting chains were identified as follows: if the structure contained only two chains, the smaller was identified as the peptide, and the larger was analyzed using CRACLe; if the structure contained more than two chains, we visually selected the chains in complex with the peptide(s) based on the crystal structure of the complex.

The ZDock Protein-Protein Benchmark 4.0 [79] contains 176 test cases of protein-protein interactions, including 123 rigid-body, 29 medium, and 24 difficult interactions. Furthermore, each test case not only includes the crystal structure of the complex, but also of the unbound forms of each binding partner. The unbound structures for each protein were curated by the ZDock team and aligned to the corresponding bound structure. As with the Dockground dataset, two chains defining the target interface are provided for each complex; thus each complex contains a single hetero-dimer. For easier computational analysis we renamed the chains of each of the unbound proteins to match the chain ids given by the complex. Of the 176 test cases, 41 bound and 47 unbound complexes were removed due to an iCode or other problem with the structure. The ZDock dataset has 31 complexes in common with the Dockground Hetero dataset and 4 with the PepX dataset (Figure 3.2).

3.1.4 Validation of Predicted Binding Sites

We used CRACLe to analyze each protein in the Dockground, PepX, and ZDock datasets independently of its binding partner. As mentioned above, we defined the target interface for a given complex based on information provided by the dataset (see Figure 3.3A for an example target interface); accuracy is based solely on whether a binding site was predicted for a single specific interface for each protein, i.e., the set of target interfaces is not exhaustive. For this project, we evaluated our predictions based on whether or not a CRACLe predicted a binding site at the target interface using the following metrics: (i) a protein has a correctly predicted interface if at least one predicted binding site was found at the target interface; and (ii) a predicted binding site was found at the interface if at least 60% of its residues participate in the target interface. For the maximum-potential algorithm, we also calculated the specificity and sensitivity for the prediction of interfacial residues, i.e., an interfacial residue found in a predicted binding site is a true positive and a non-interfacial residue not in a predicted binding site is a true negative. While we must keep in mind that most proteins are promiscuous and have alternative interfaces [82, 83] and that many and more PPIs have not been experimentally validated [1], only the target interfaces found in the datasets are relevant, and only the statistics regarding the target interface are relevant to evaluating the predictions at this point.

For each dataset, we classified each predicted binding site into one of three categories according to their participation in the target interface: (1) putative, i.e., 0% participation; (2) overlapping, i.e., less than 60% participation; and (3) interfacial, i.e., at least 60% participation (Figure 3.3), where participation is defined as the percentage of predicted binding site residues that are also interfacial, i.e., in the target interface. We selected the 60% threshold (i) to ensure that large binding sites would not be declared interfacial unless the majority was truly at the target interface and (ii) to account for some degree of promiscuity for the binding site: many proteins have multiple binding partners, but these different partners often use the same hot spots [84].

To fully understand CRACLe's binding site predictions, we must make note of an important caveat: CRACLe is not meant to predict the entire interface of a PPI. The critical residue analysis algorithm takes *only* single proteins as input, not protein complexes, and trying to predict an interface with an unknown protein is not only difficult but presumptuous. We have used well-known protein-protein datasets to train and test our algorithm, but experimental hot spot identification is costly [85, 86], and we do not have experimental hot spot data for every protein in the data set. While some hot spot datasets have been compiled [87, 88], it is not uncommon for predictions to be evaluated based on whether or not the predicted residues are in the interface [30, 83]. For now, we limit the validation of our algorithm based on interfacial data, not hot spots.

3.2 Results and Discussion

Using the function-based algorithm, CRACLe correctly predicted binding sites for 88%, 85%, and 78% of individual proteins from the target interface in the Dockground homo-complex, Dockground hetero-complex, and PepX datasets, respectively. The results for binding site prediction per protein are summarized in Table 3.2. Because analysis of the complex requires predictions from both binding partners, no accuracy results are given for PepX complexes as the peptide was not independently analyzed in this study.

				Ratio of Binding Sites Found					
				By Complex			**By Protein**		
Dataset	**N Complex**	**N Protein**	**Function**	Any Site	At Interface	With Extensions	Any Site	At Interface	With Extensions
Dockground Hetero	457	914	Fold	1.00	0.89	0.92	1.00	0.69	0.77
			Hetero	1.00	0.93	0.95	1.00	0.76	0.84
			Homo	1.00	0.88	0.93	1.00	0.66	0.76
			All	1.00	0.95	0.97	1.00	0.84	0.88
Dockground Homo	1,317	2,634	Fold	1.00	0.76	0.83	1.00	0.66	0.75
			Hetero	1.00	0.79	0.86	1.00	0.72	0.80
			Homo	1.00	0.76	0.83	1.00	0.67	0.78
			All	1.00	0.86	0.91	1.00	0.80	0.85
PepX	1,076	,1077	Fold	–	–	–	1.00	0.61	0.71
			Hetero	–	–	–	1.00	0.63	0.70
			Homo	–	–	–	1.00	0.55	0.65
			All	–	–	–	1.00	0.70	0.78

Table 3.2: CRACLe results using the function-based algorithm for the Dockground and PepX data sets. The rows corresponding to training sets are highlighted. Shown are the ratios of complexes and proteins for which CRACLe was able to identify 1) at least one binding site, 2) at least one binding site at the interface with at least three defined residues, and 3) at least one binding site at the interface with extensions included. A complex is said to have a binding site at the interface if at least one of its participants has a binding site at the interface; note that for the PepX database, only the receptor is analyzed.

3.2.1 Participation of Predicted Binding Sites in the Target Interface

Using the metrics described above, we looked at how many of the predicted binding sites participated in the target interface across each dataset Figure 3.4. We found that most of the predicted binding sites were not at the target interface, but of those classified as interfacial binding sites, the majority had 90% participation or greater. Although we hypothesize that many of the putative predictions are in fact binding sites, we simply do not have the data to test our hypothesis at this time.

For the Dockground hetero- and homo-complex data sets, about 13% of the predicted binding sites were overlapping, and 42% were interfacial. Within the interfacial classification, the large majority of binding sites had a 90% participation or better—accounting for 33% of all predicted binding sites in each dataset. From the PepX dataset, only 28% of all predicted binding sites were classified as interfacial, but as with Dockground, most of the interfacial predictions had at least 90% participation. Due to the smaller size of protein-peptide interfaces, we expected to find fewer predicted sites at the interface, but the overall lower percentage of PepX proteins with interfacial predictions (78% of target PepX protein-peptide binding sites identified versus 85% and 88% for Dockground homo- and hetero-complex protein-protein binding sites, respectively) suggests that the SNAPP scoring functions may be less suited to prediction of protein-peptide binding sites than protein-protein binding sites.

3.2.2 Dockground Complexes

The Dockground dataset was used to train both of the SNAPP-Surface scoring functions; however, both the homo- and hetero-complexes were also used to test the opposite scoring function along with SNAPP-Fold. Examples of binding site predictions from the Dockground dataset are shown in Figure 3.5A–E.

Dockground Homo-complexes

Training. Using SNAPP-Surface:Homo, CRACLe was able to identify a binding site at the target interface for 77.5% of the 2,622 proteins in the data set and 83.4% of the 1,311 complexes formed by these proteins. Interestingly, SNAPP-Surface:Homo benefited the most from the inclusion of binding site extensions, correctly predicting the target binding sites for another

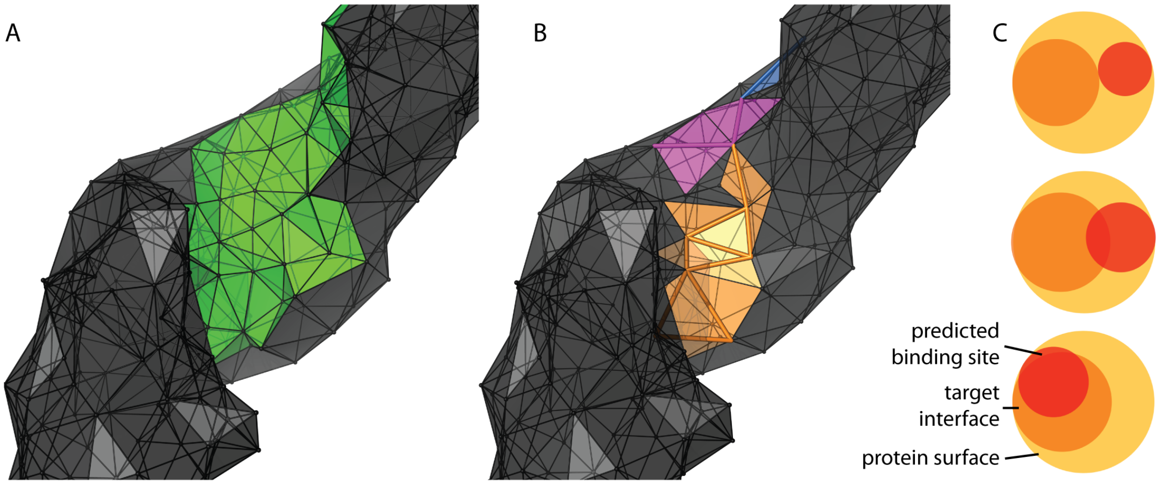

Figure 3.3: A comparison between the target interface and the predicted binding sites of β-catenin (PDB 1jdh). (A) The target interface, shown in green, as defined by Delaunay tessellation, i.e., all of the green triplets form simplices in the native interface. (B) Two predicted binding sites (orange and pink) and one binding extension (blue) found at the target interface. The surface tessellations are slightly different due to tessellation of the complex versus the protein in A and B, respectively. (C) A graphical representation of the three classes of predicted binding sites: (top) putative—no participation in the target interface; (middle) overlapping—less than 60% participation; and (bottom) interfacial—at least 60% participation.

10.7% (up from 66.8% to 77.5%) of homo-complex proteins using binding site extensions.

Testing. Using SNAPP-Fold, CRACLe identified the target binding site for 75.2% of the Dockground homo proteins and 82.5% of the complexes. SNAPP-Surface:Hetero performed even better, identifying 79.6% and 85.7% of the target binding sites for proteins and complexes, respectively. Curiously, SNAPP-Surface:Hetero outperformed SNAPP-Surface:Homo on its own training set. Previous work has found that the interfaces in homo and hetero complexes differ in their residue compositions [76] and would suggest the opposite results. We have found that the tessellation on the surface of a protein will often incur slight-to-moderate conformational changes between the bound and unbound protein structures, typically around shallow pockets. These predictions suggest that homo-complex interactions may be more likely to alter the tessellation on the surface. We hypothesize that the tessellation of an unbound protein may create triplets similar to those found at hetero-complex interactions and

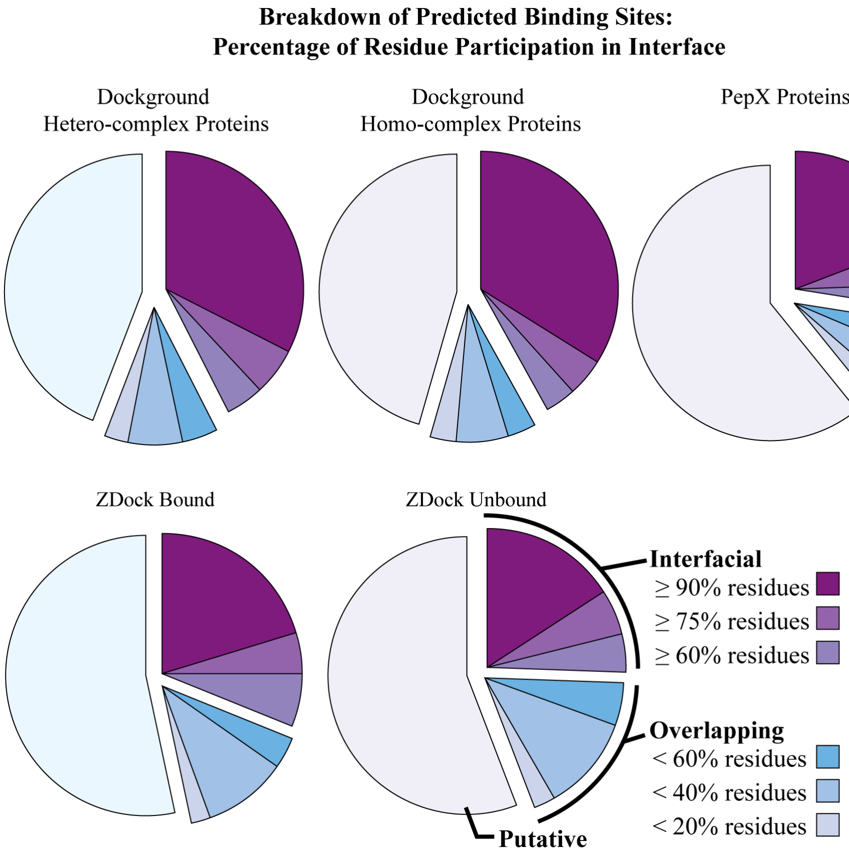

Figure 3.4: Percentage of residue participation in the target interface for all predicted binding sites, excluding binding site extensions. Binding sites are said to be in the interface if 60% of their residues are in the interface, e.g., at least 2 residues for a 3 residue binding site, and 3 residues for a 4 residue binding site.

that residues exposed due to conformational changes may create the homo-complex specific triplets.

Dockground Hetero-complexes

Training. CRACLe binding sites predicted using the SNAPP-Surface:Hetero scoring function identified the target binding site in 95.4% of complexes and 83.6% of proteins in the Dockground hetero-complexes dataset.

Testing. Neither SNAPP-Fold nor SNAPP-Surface:Homo matched the prediction rate of SNAPP-Surface:Hetero, predicting the target binding site for 91.9% and 93.2% of complexes

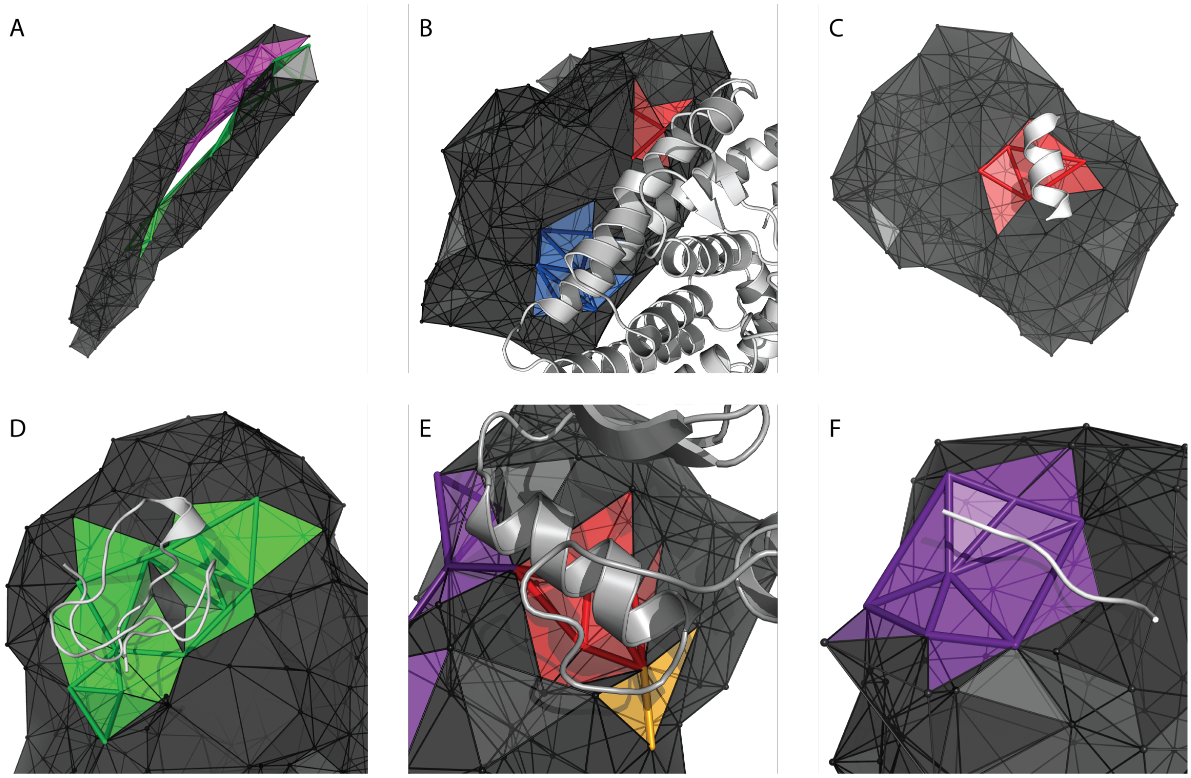

Figure 3.5: Examples of predicted binding sites. The binding sites are colored for easier visual identification. (A) Leucine zipper formed from transcription factors ATF-4 and C/EBP β in the absence of DNA (PDB 1ci6); (B) Interfacial binding site for peroxisome proliferator activated receptor γ bound to retinoic acid receptor RXR-α (PDB 1fm6); (C) A putative binding site showing a steroid receptor co-activator bound to retinoic acid receptor RXR-α (PDB 1fm6); (D) α-amylase bound to inhibitor (PDB 1clv); (E) T4 lysozyme dimer (PDB 137L); (F) HIV Gag HAGPIA hexa-peptide bound to cyclophilin A (PDB 1awq).

and 77.0% and 75.6% of proteins, respectively. However, both SNAPP-Fold and SNAPP-Surface:Homo were able to identify unique target binding sites: Using a consensus of all three scoring functions, CRACLe was able to identify the target binding sites for a total of 96.7% of hetero-complexes and 87.8% of hetero-complex proteins. The improvement from using the consensus scoring function could result from the identification of stabilizing hot spots from SNAPP-Fold and SNAPP-Surface:Homo. We hypothesize that SNAPP-Fold and SNAPP-Surface:Homo are more likely to predict hot spot residues that improve interaction stability through either hydrophobic side chain or backbone hydrogen bonding, whereas SNAPP-Surface:Hetero may predict hot spots that contribute to interaction specificity.

3.2.3 PepX Protein-Peptide Complexes

For PepX data set, CRACLe was able to identify the target protein-peptide binding site for 77.5% of the proteins. The decreased overall prediction accuracy versus the Dockground datasets is likely due to known differences between protein-protein and protein-peptide interactions. For instance, although protein-protein interfaces are often fairly planar, especially when compared to protein-small molecule interactions [89], protein-peptide interfaces are typically even more planar and tend to pack more tightly together [90], which could suggest that protein-peptide interactions more closely resemble interactions found in protein folding. In fact, rather than trailing behind SNAPP-Surface:Hetero as seen from the Dockground predictions, SNAPP-Fold outperformed both SNAPP-Surface:Hetero and SNAPP-Surface:Homo by a small margin. Protein-peptide interactions are also less likely to induce a conformational change in the receptor [90]. An example binding site prediction from PepX is shown in Figure 3.5F.

Putative Predicted Binding Sites

Proteins are promiscuous with respect to interactions with other proteins [82, 91, 61]: They typically have more than one binding site, and those binding sites might overlap; however, because many binding sites are not known or the structural data is not available, identifying and quantifying these sites can be problematic. In this study, we found that many predicted binding sites were in fact true positives for non-target interactions, i.e., interactions not specified in the Dockground data or any protein-protein interactions in the PepX dataset but are known otherwise.

One notable example is an MHC-I α chain present in the PepX dataset (Figure 3.6A) that has a target interaction with an HIV Gag nona-peptide [92] (Figure 3.6B). In addition to correctly identifying a large portion of the MHC-I antigen peptide binding groove, CRACLe predicted three other binding sites (Figure 3.6C,D). Each of the three putative predictions are found at different portions of the interface between the α chain and the $\beta 2$ subunit of the

MHC-I protein. The first putative prediction (Figure 3.6C) is found at the protein-protein interface between the $\alpha 1$ domain and the $\beta 2$ subunit with 100% participation from twelve residues. The second and third putative predictions (Figure 3.6D) are found at the interface between the $\alpha 3$ domain and the $\beta 2$ subunit with 75% (of four residues) and 50% (of eighteen residues) participation, respectively.

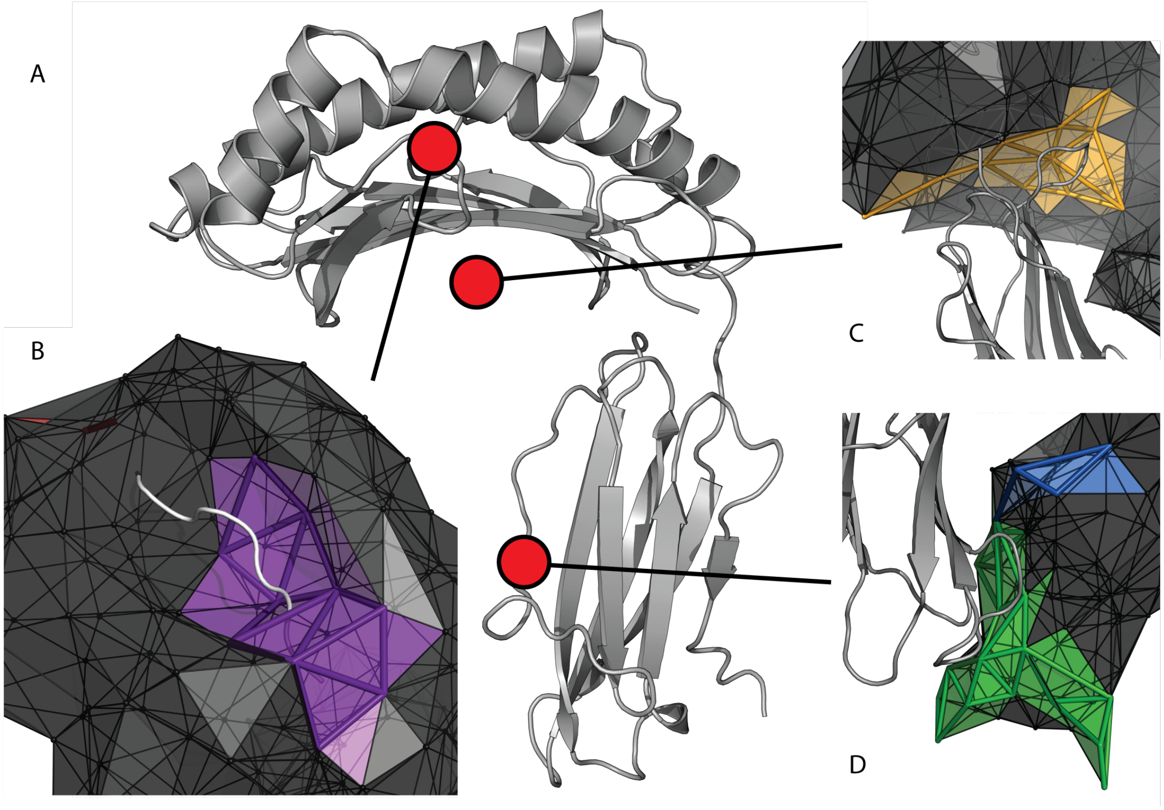

Figure 3.6: Predicted binding sites for MHC class I (PDB 1a1m). (A) The α chain; (B) the expected and predicted binding site and extension for the peptide; and (C,D) putative predicted binding sites at the interface with the $\beta 2$ subunit. These binding sites contain experimentally and computationally validated hot spots.

Although all four of the putative predicted binding sites did participate in the $\beta 2$ subunit interface, the largest site (Figure 3.6E in green) only had 50% participation—nine residues were outside of the $\alpha 3$-$\beta 2$ subunit interface. Those nine residues are split into two groups: (a) H191 and P193-D196 and (b) W274-H278, except for P276. A closer look at the CRACLe prediction revealed that each of the three scoring functions did predict binding sites at the $\alpha 3$-$\beta 2$ interface (Figure 3.7); however, SNAPP-Surface:Homo predicted a single large binding site below the $\alpha3$-$\beta2$ interface (Figure 3.7D), containing both sets of non-interfacial residues, and SNAPP-Fold predicted a smaller binding site containing only the C-terminus residues W274 to H278. Although we were unable to identify a known interaction for residues P193 to D195, residues W274 to H278 of the C-terminus contain a portion of a poly-histidine tail that was added to enable purification of the protein. Thus, a portion of the larger predicted $\alpha3$-$\beta2$ binding site is still found at an experimentally validated, if artificial, binding site.

The larger putative predicted binding site reflect a potential lack of specificity in the function-based algorithm: Distinct but overlapping binding sites may be reported as a single site or in close proximity between a true positive and a false positive, resulting in a predicted binding site that is only half correct. This merging of seemingly unrelated binding sites led to the development of the max-potential algorithm, which is discussed in more detail below. Using the max-potential algorithm, CRACLe was able separate the larger putative binding site into several smaller binding sites. Interestingly, the max-potential algorithm also predicted a binding site that partially overlaps the experimentally suggested CD8 binding site—a site that neither the function-based algorithm nor PredUs (details below) was able to identify.

Comparison with the PredUs Algorithm

Due to the availability of experimental data, we used the aforementioned MHC-I complex to compare CRACLe against existing algorithms. Most of the existing algorithms for binding site prediction focus entirely on hot spot prediction and none of them, save one, allow for more than a single analysis at a time. In fact, only PredUs [30] was similar enough in objective and capabilities to allow us to compare and verify our results. Other programs gave limited results, or were not accessible.

To compare CRACLe against PredUs, we used the MHC-I α chain discussed above. PredUs completed the analysis in a time frame similar to what it took CRACLe to analyze all proteins in the Dockground homo-complex data set. In general, the predictions generated from both algorithms correlated well, especially for the peptide groove (Figure 3.8). PredUs

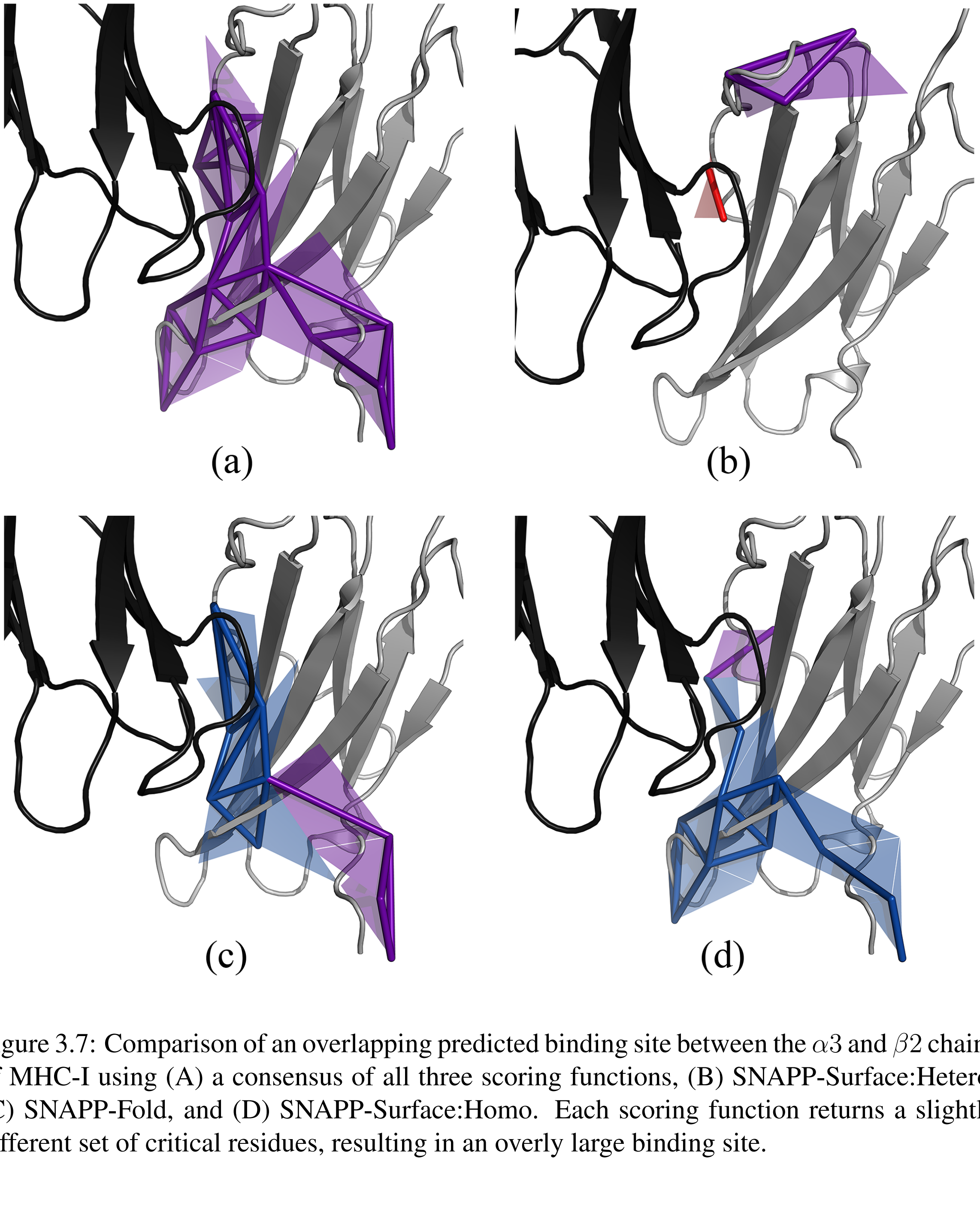

Figure 3.7: Comparison of an overlapping predicted binding site between the $\alpha 3$ and $\beta 2$ chains of MHC-I using (A) a consensus of all three scoring functions, (B) SNAPP-Surface:Hetero, (C) SNAPP-Fold, and (D) SNAPP-Surface:Homo. Each scoring function returns a slightly different set of critical residues, resulting in an overly large binding site.

predicted 77 hot spot residues, and each one was found either in or immediately adjacent to a predicted binding site. The only exception was the aforementioned, larger putative prediction (Figure 3.6D). As expected, PredUs did not predict any hot spots near residues P193 to D195, further suggesting a false positive; however, PredUs did not identify the C-terminal residues W274 to H278 that in fact constitute a binding site.

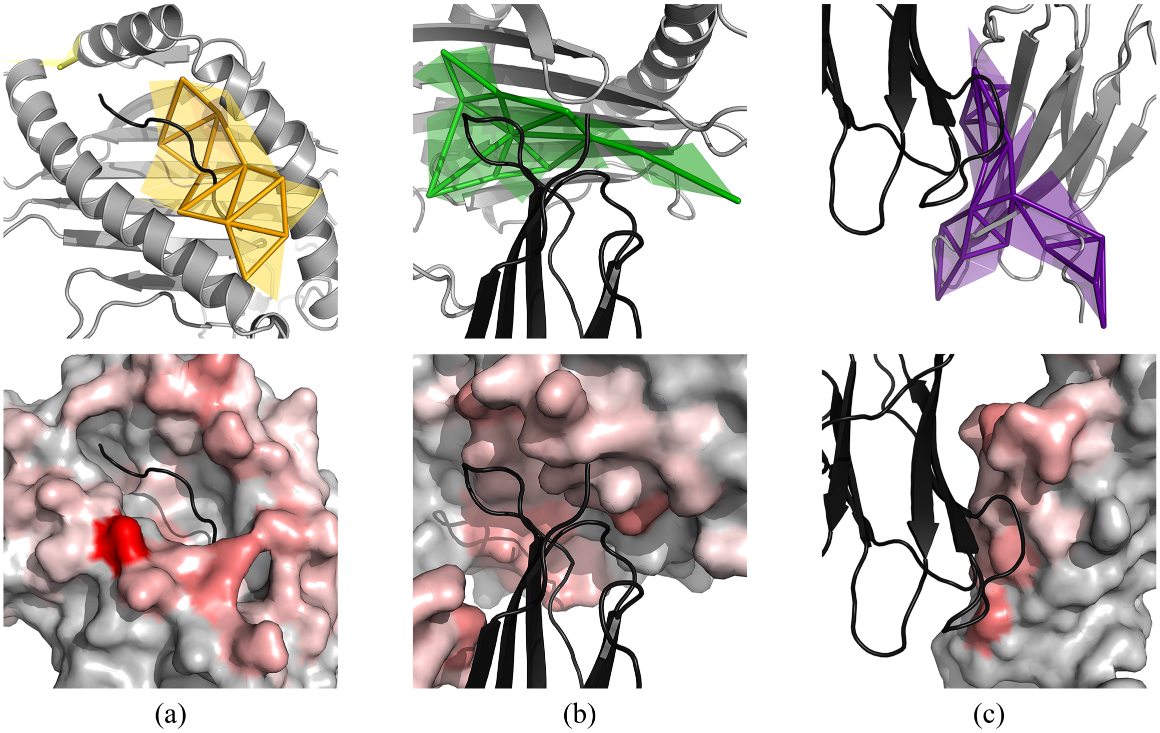

Figure 3.8: Comparison of CRACLe (top) against PredUs (bottom) for (A) the peptide groove and (B,C) the interactions with the $\beta2$ subunit. CRACLe and PredUs returned very similar results, identifying many of the same residues.

3.2.4 The Maximum-Potential Algorithm and the ZDock Benchmark

The over-prediction of the MHC-I $\alpha3$-$\beta2$ binding site led us to analyze several of the larger binding site predictions in greater detail: We found that many of the larger binding sites were actually smaller groups of selected critical residues bound together by a single edge between a residue from each. In many cases, this connection caused for the larger binding site to be classified as overlapping when two binding sites should have been classified as interfacial and putative instead. To overcome this problem, we simplified the existing algorithm and increased the requirements for merging binding sites, resulting in the maximum-potential algorithm. The new algorithm initially selects many more potential critical residues and refines the list based on proximity other potential critical residues. The new algorithm also predicts twice as many binding sites due to the stricter requirements for merging two or more binding sites. Another important change is the removal of binding site extensions; due to inclusion of secondary critical residue clusters, the addition of critical residue pairs provided little information above the existing data.

In addition to the previously used metrics, we also calculated the sensitivity and specificity for each dataset (Table 3.4), where residues are classified based on whether or not they (i) are interfacial or non-interfacial and (ii) are predicted to be hot spots, i.e., whether or not they participate in a predicted binding site. Unfortunately, specificity and sensitivity defined in this manner only serve to evaluate whether or not CRACLe can identify the target interface. This classification of residues is problematic for a number of reasons. First, defining which residues do in fact participate in an interface presents several additional problems. There is evidence that hot spots do tend to cluster near the center of an interface and that the surrounding residues provide some stabilization of the interaction, much like an o-ring in pipe fitting [88]; however, determination of where an interface begins and ends is tenuous at best. Second, all of the measures used to validate the predictions account only for the target interface, ignoring any other interactions a protein may have. Third, the specificity reflects the fact that the data set is unbalanced: There are typically four to five times more non-interfacial residues than there are interfacial residues. A high specificity provides little information other than to verify that CRACLe does not over-predict the entire surface of a protein. Fourth and especially, CRACLe was intended to predict only hot spot residues—not the interface—and not every residue in an interface is a hot spot. In other words, we never intended to identify the entire interface. Thus, the sensitivity reflects our ability to predict *interfacial* residues, not hot spots, and should be low. Unfortunately, we must make use of the interfacial data; experimental validation of hot spots is costly and therefore not available for many proteins. As a result all of the standard statistical methods fall outside of the applicability domain. We defined the sensitivity, specificity, precision, and accuracy as follows using the confusion matrix shown in Table 3.3:

$$\textit{sensitivity} = \frac{TP}{TP + FN} \tag{3.4}$$

$$\textit{specificity} = \frac{TN}{TN + FP} \tag{3.5}$$

$$\textit{precision} = \frac{TP}{TP + FP} \tag{3.6}$$

$$\textit{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.7}$$

We retested the Dockground and PepX datasets using the maximum-potential algorithm and saw a large jump in the prediction of interfacial binding sites. Due to the minimal overlap with Dockground and the inclusion of the unbound forms of proteins, the ZDock Protein-Protein Benchmark 4.0 presents an optimal test test. Not only do the test cases provide an external test set, but they also provide a more practical evaluation of CRACLe in an experimental setting. As expected, the sensitivity and precision are low and the specificity and accuracy are high for predictions from all three sets, suggesting that the predictions are largely found at the interface, but they cover very little of the interface defined using Delaunay tessellation. Previous studies have suggested that a sensitivity and/or precision of greater than 40% is a good measure of stability [83, 93], while reporting that in most cases, the sensitivity increased when more residues were predicted as interfacial. De Vries and Bonvin [83] report that CPORT predictions had a sensitivity of 48% and precision of 28% for the unbound proteins from the Protein-Protein Benchmark 3.0 [94], which was a significant improvement on previous results. Using the unbound proteins from the newer Benchmark 4.0 [79], CRACLe achieved a sensitivity of 34% and precision of 31% while predicting 20 fewer residues on average for each complex. As previously mentioned, we expected a lack of sensitivity: CRACLe predicts roughly the same number of residues as are found on average across all interfaces in the dataset; furthermore, CRACLe does not attempt to predict all of the residues in an interface but those residues which are critical to the interaction.

				Interfacial Residues			
				Experimental			
				Yes		No	
Binding Site Residues	*Predicted*	Yes	**True Positives**	13,097	**False Positives**	21,515	Dockground Hetero
				37,792		62,288	Dockground Homo
				10,492		36,140	PepX
				3,328		7,572	ZDock Bound
				3,007		7,608	ZDock Unbound
		No	**False Negatives**	27,933	**True Negatives**	111,731	
				86,734		342,428	
				29,294		145,739	
				6,814		33,527	
				6,503		32,409	

Table 3.3: The confusion matrix for experimental interfacial residues versus predicted binding site residues from CRACLe predictions.

We also looked at how well CRACLe was able to predict binding sites within each of the three classes in the Benchmark 4.0. For the bound test cases, CRACLe correctly predicted interfacial binding sites for 88% (165 of 188) of the rigid-body proteins, 91% (40 of 44) of the medium proteins, and 89% (34 of 38) of the difficult protein test cases. To evaluate the unbound test cases, we identified interfacial residues from the bound complexes and checked whether or not those same residues were present in the binding sites predicted for the unbound proteins. CRACLe correctly predicted binding sites for 83% (132 of 160) of the rigid-body proteins, 83% (30 of 36) of the medium proteins, and 84% (32 of 38) of the difficult unbound test cases.

Interestingly, several studies have reported on the difficulty of predicting interaction sites for antibody-antigen (Ab-Ag) complexes [95, 93, 83]. The difficulty arises in part because antibodies nearly always bind at their Complementarity Determining Region, which is largely determined by the hypervariable V(D)J region of DNA and contains strong desolvation signals, and in part because the antigenic epitope is not easily distinguishable from the rest of the protein [83, 96, 97]. In fact, Kufareva et al. [96] suggested that binding antibodies was not a biological function of antigens, thus epitopes should not be considered as biological interfaces. However, biological systems are not run on enzymes alone, and a problem should not be ignored because it is non-standard or difficult. CRACLe predicted binding sites at the target Ab-Ag interface for 10 out of 14 of the antigens and 10 out of 14 of the antibodies analyzed in the ZDock Unbound dataset. Of those Ab-Ag complexes, only one—PDB code 2HMI, listed as a difficult complex—was without a predicted binding site for both the antibody and the antigen. These results suggest that there antigenic epitopes likely do have some distinguishable feature that is recognizable in some part from the neighborhood of the epitope. We have begun a separate study, discussed below, to analyze antigenic epitopes using CRACLe

Dataset	**N Complex**	**N Protein**	Avg. $N_{surface}$ Residues	Avg. $N_{interface}$ Residues	Avg. $N_{critical}$ Residues	Prediction Ratio	Avg. Sensitivity	Avg. Specificity	Avg. Precision	Avg. Accuracy
Dockground Hetero	457	914	145	44	30	0.94	0.28	0.85	0.44	0.70
Dockground Homo	1,317	2,634	155	47	30	0.93	0.24	0.86	0.43	0.71
PepX	1,076	1,077	164	35	34	0.85	0.21	0.82	0.27	0.70
ZDock Bound	135	270	151	36	39	0.89	0.31	0.80	0.34	0.70
ZDock Unbound	129	258	151	36	40	0.83	0.27	0.77	0.31	0.70

Table 3.4: CRACLe results for the test and training sets using the maximum-potential algorithm.

3.3 Practical Applications

In addition to validating CRACLe using the ZDock Benchmark 4.0, CRACLe has been applied to two additional projects. The first is a collaboration with the Cance laboratory from Roswell Park Cancer Institute in Buffalo, New York to develop a small molecule inhibitor for the interaction between the Focal Adhesion Kinase (FAK) and Human Epidermal growth factor Receptor 2 (HER2). The second is a collaboration with the Asokan laboratory at the University of North Carolina at Chapel Hill to predict antigenic epitopes for the viral envelope protein gp120.

3.3.1 Predicting the FAK-HER2 Interaction

FAK is a tyrosine kinase that plays an important role in a number of cellular functions, including integrin-mediated signaling, cellular motility, and protection against apoptosis, and HER2 has been used as a marker to evaluate the aggressiveness of a particular cancer [98]. Several studies have shown that FAK plays an important role in upregulation of the HER2 signaling pathway [98], and the Cance laboratory has produced experimental evidence that FAK not only plays a role in the signaling pathway but actually binds HER2. Based on this knowledge, their laboratory is attempting to design a small molecule inhibitor to disrupt this interaction.

We used CRACLe to suggest possible binding sites on the surface of the FAK FERM domain. CRACLe predicted two binding sites that have been previously validated in the literature (Figure 3.9). The first site consists of residues Y180 and V196 on the FERM F2 domain. These residues form part of a hydrophobic pocket that binds and inhibits the FAK kinase domain [99]. The second predicted binding site consists of residues K218 and K222, both of which were found to be critical for an interaction with the proto-oncogene c-Met [100].

CRALCe also predicted two additional binding sites on the FERM domain that are undergoing experimental validation for binding activity with HER2. Our collaborators have designed small molecules inhibitors to bind at each of these predicted binding sites and are

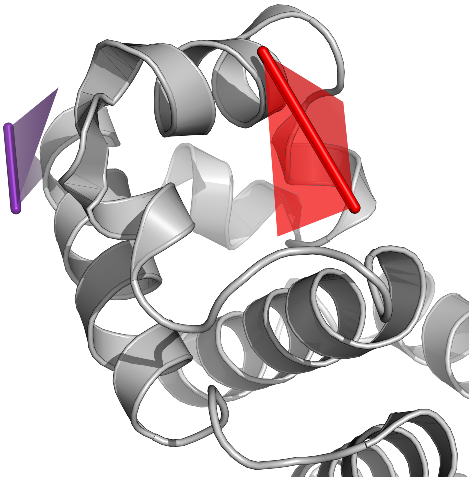

Figure 3.9: CRACLe predicted two binding sites on the FAK FERM domain (PDB code 2AL6) that correspond to experimentally validated hot spots. (Red) Predicted binding site with residues Y180 and V196; this site corresponds to a hydrophobic pocket formed by residues Y180, M183, V196, and L197 that binds the FAK kinase domain. M183 is a part of the surface triplet, and L197 is not on the surface in the tessellation. (Purple) Predicted binding site with residues L281 and L222; both of these residues are experimentally validated hot spots for the interaction with c-Met.

currently testing to see the affect each small molecule has on the FAK-HER2 interaction. Inhibitors for the first predicted site have been found to reduce cell viability in cancer cell lines; unfortunately, the results suggest that this binding site does not interact with HER2. However, inhibitors for the second binding site do appear to be disrupting the FAK-HER2 interaction. We are unable to provide additional data at this time as the results are not yet published.

3.3.2 Predicting Antigenic Epitopes

In order to fight infection, B cells produce antibodies that are able to identify and neutralize foreign proteins called antigens. Antibodies bind these antigens at a unique, and usually conserved, part of the protein, referred to as antigenic epitopes ([101]). Prediction of antigenic epitopes could lead to the development of specialized antibody drugs and a better understanding of host-pathogen interactions [101, 102]. Ofttimes, antigenic epitopes are continuous, i.e., formed by a strand of sequentially adjacent peptides, but many known epitopes are discontinuous. The former are much easier to predict, and many various methodologies have been developed for that purpose; the latter much less so [103, 101]. Based in part on the ZDock Unbound results for the Ab-Ag complexes, we hypothesize that CRACLe could be used to predict antigenic epitopes on pathogenic proteins. To this end, we decided to predict the binding sites for the well-known HIV-I envelope protein gp120.

We have compiled a dataset of 18 gp120 and gp160 precursor proteins from the Immune Epitope Database (IEDB) based on sequence similarity for known antigenic epitopes. Our initial results have not correlated well with the known antigenic epitopes; however, visual inspection of the crystal structure shows that the binding sites predicted by CRACLe may be correct. For example, we analyzed the structure of gp120 co-crystallized with CD4 and an antibody (PDB code 2QAD) and found that the antigenic epitopes given in the IEDB do not occur where the antibody is bound (Figure 3.10A), but instead in an internal part of the protein, at the interface with a gp120 dimer, and on an α-helix on the opposite side of the protein from where the antibody is bound. These differences may be explained by problems interpreting the crystal structure or existence of multiple antibodies that bind in different locations. Interestingly, CRACLe predicted three binding sites, one at the interface with the antibody in the crystal structure, another at the gp120 dimer interface, and a third structurally adjacent to the antigenic epitope found on the α-helix. Further analysis is required before our predictions may be validated; however, these initial results suggest that CRACLe may be useful for prediction of new and validation of existing antigenic epitopes.

3.4 Conclusions and Future Work

In this study, we have shown that the CRACLe algorithm is capable of identifying binding sites on protein surfaces, correctly predicting the target native protein-protein binding site for more than 85% of all individual proteins in the Dockground data sets and 77% of target protein-peptide binding site for the PepX database. However, roughly two thirds of the predicted binding sites were putative; we have provided examples showing that some putative

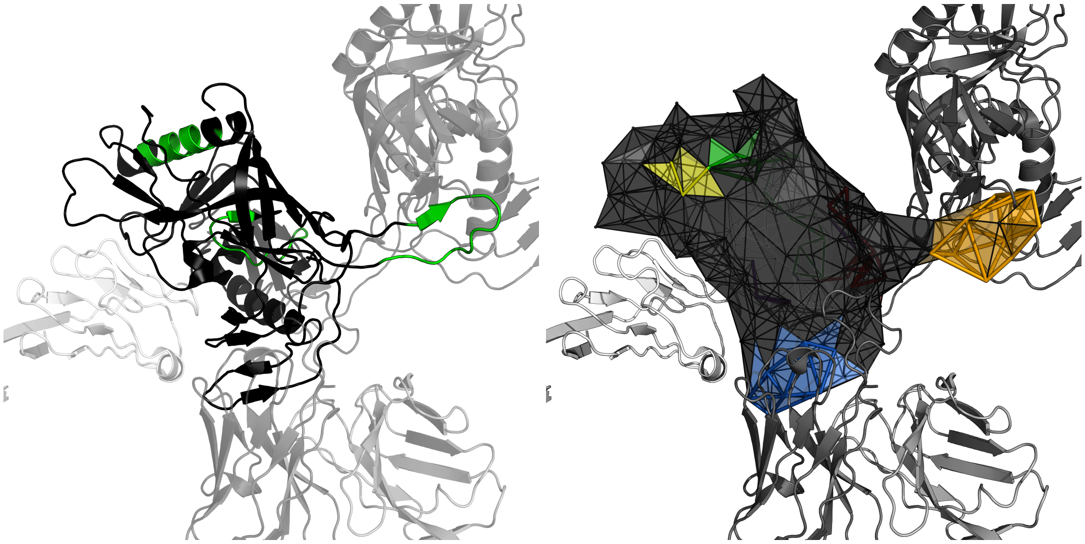

Figure 3.10: (A) The crystal structure of gp120 bound by CD4 and an antibody. The structure of gp120 is shown in black and the antigenic epitopes found in the IEDB are highlighted in green. (B) CRACLe predicted four binding sites: one at the interface with the antibody; another at the gp120 dimer interfaces (this interaction may be a result of crystallization); and another structurally adjacent to the upper antigenic epitope on the α-helix.

predictions do correspond to actual binding sites. We hypothesize that many more of putative predictions do correspond to actual binding sites, and as more experimental data becomes available, we will continue to verify these putative binding sites.

Our SNAPP-Fold scoring function performed surprisingly well considering it was built using a non-protein interaction training set; this observation provides further support for the similarity between protein folding and protein interactions [90] but also highlights the need for high-quality data curation [104]: Additional PPI structures and stricter data curation may significantly improve predictions of the SNAPP-Surface:Hetero and SNAPP-Surface:Homo scoring functions. In contrast, we were also surprised to see SNAPP-Surface:Hetero perform better than SNAPP-Surface:Homo given the fewer number of complexes in the training set. However, we hypothesize that homo-complexes instead present different surface residue triplets when a protein is tessellated by itself versus with its binding partner, e.g., when an interaction occurs at a deep or narrow binding pocket.

CRACLe's accuracy could also be affected by the use of bound structures to train the

SNAPP-Surface scoring functions. As the results show, the prediction ratio for the ZDock Unbound dataset was 6& lower than the Bound dataset. Unfortunately a number of factors limit our use of unbound structures. First, compilation of a dataset of unbound structures is not a trivial task and would require additional time and resources. Second, the data may not exist. Third, unbound structures may not easily map onto bound structures: Chain and residue numbering will likely differ between bound and unbound structures, and some structural features may differ between the two structures. The ZDock team has now released four versions of their Protein-Protein Benchmark, and their dataset is only a fraction of the size of the Dockground dataset. Until a much larger dataset of bound and unbound proteins is compiled and made publicly available, SNAPP-Surface will most likely continue to be trained on bound structures.

Overall, CRACLe accurately identified the target binding site for thousands of proteins in less than seven minutes on a standard desktop computer (see Table C.1). Additionally, CRACLe needs less *a priori* knowledge of an interaction to make a prediction, requiring only the three-dimensional structure of a single protein. As a result, CRACLe opens new possibilities for computational research on protein-protein interactions, especially for use in discovering and analyzing protein interaction networks, aiding in the design of new protein interactions, or simply as an additional filter for scoring and decoy discrimination in protein-protein docking algorithms. We also envision CRACLe as a new tool to guide experimental studies, for example identifying surface residues for mutagenesis studies or suggesting potential secondary interactions for side effect screenings.

3.5 Availability

The current version of CRACLe is available upon request. In the near future, we plan to make CRACLe publicly available under the ChemBench online cheminformatics portal (http://chembench.mml.unc.edu) and modular computing cloud developed in our laboratory [105].

CHAPTER 4

Predicting Protein-peptide Docking

Prediction of Protein-peptide Packing, or PoPP, provides a SNAPP-based approach to predicting Protein-PEptide interactions (PPE). Given the three-dimensional structure of a receptor protein and a peptide sequence, PoPP predicts the most likely docking pose based on a global search of the receptor's surface. In this chapter, we discuss the development of the novel GridDock algorithm behind PoPP, the current status of the project, and an additional application of GridDock towards prediction of protein pockets, called PickPocket.

Throughout this chapter, we use two terms to describe how a peptide ligand is docked to a receptor: conformation and pose. A binding conformation refers directly to the structure of the peptide, i.e., the relative coordinates of its residue vertices, and has no connection to the receptor. A binding pose refers to the conformation of a peptide in relation to the receptor, i.e., the absolute coordinates of peptide vertices in a particular position relative to the receptor.

Like most docking algorithms, PoPP may be broken down into three basic steps: (1) generation of docking poses; (2) scoring of poses; and (3) pose refinement. SNAPP handles the scoring, but pose generation and refinement is handled largely by a novel three-dimensional lattice algorithm called GridDock. A basic workflow is shown in Figure 4.1.

4.1 Modeling the Problem

The representation of the receptor and ligand structures is crucial first step that defines the accuracy and computational efficiency of a docking algorithm. In order to dock the ligand

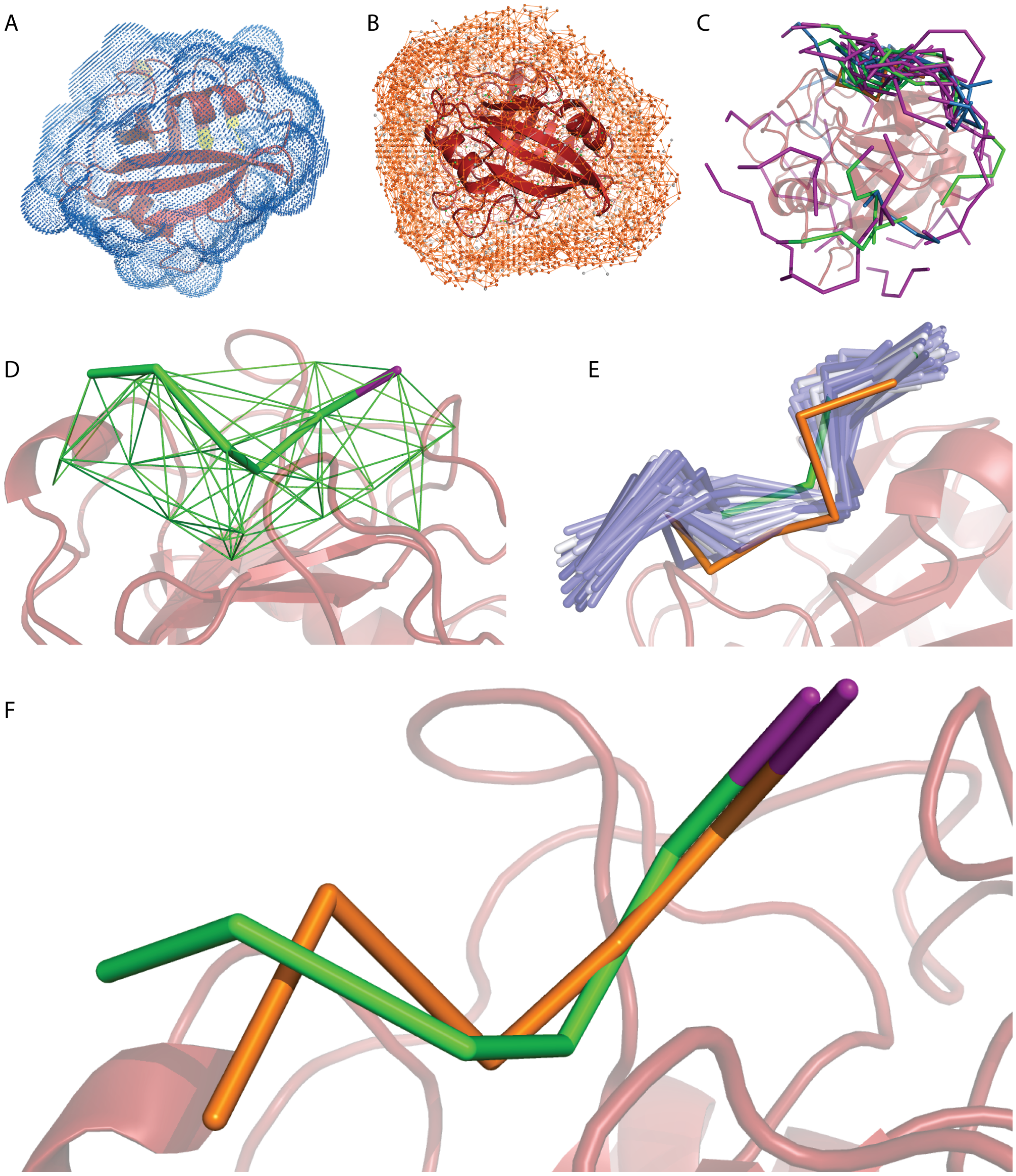

Figure 4.1: The PoPP workflow. (A) First the interaction grid is created around the receptor protein, and (B) peptide poses are randomly initialized within the interaction grid. (C) High ranking poses are selected using a Metropolis Monte Carlo algorithm based on the SNAPP-Interface score calculated from (D) a local Delaunay tessellation. (E) For each selected ligand-receptor pose, the ligand undergoes multiple perturbations, which undergo the same process of selection and perturbation. (F) The highest scoring poses are selected and returned. The predicted pose (green) is the highest scoring initial pose with an RMSD of 3.46 Å to the native peptide (orange). (PDB code 1AWQ)

with the receptor, we must first define the area around the receptor where the ligand is allowed to reside, which we term the *interaction space*. This interaction space serves to limit the movement of the ligand during pose generation and refinement to ensure that the peptide ligand does not (i) stray too far from the receptor or (ii) invade the space occupied by the receptor. The simplest method used to restrict the interaction space is to place the receptor in a grid and constrain vertices of the ligand to these grid points. We have developed a special grid defined by a two-dimensional array of binary strings, where each bit represents a single three-dimensional coordinate. This novel implementation allows for rapid comparison between grid objects and easy access to multiple grid points at once.

To create the grid, we first convert the atomic structure of the receptor to the single-point-per-residue model that is used with both SNAPP and CRACLe. The entire structure is then geometrically translated so that all residue vertices exist in positive coordinates, and the spacing between grid vertices defaults to 0.5 Å, defined by the `resolution` parameter. Each residue vertex is assumed to have a constant sphere of exclusion to account for steric hindrance—no grid points are allowed within the sphere. The radius of the exclusion sphere is defined by the `fit` parameter (Figure 4.2B), so called because it determines how the grid vertices fit around the shape of the receptor. Each residue vertex is also given a constant inclusion sphere that defines the area around a vertex where grid vertices may exist. The radius of the interaction sphere is given by the `fit` parameter in conjunction with a `thickness` parameter that defines, as its name suggests, the width of the grid surrounding a residue vertex (Figure 4.2C). The exclusion and inclusion spheres are combined using a binary `OR` operation to generate an interaction annulus that represents the space around each residue vertex where an interaction with another residue may occur—grid vertices may exist only within the interaction annulus. To define the interaction space for a protein, exclusion and inclusion spheres are applied to each residue vertex to create two independent grid objects (Figure 4.2BC). A simple binary `OR` operation between the two grid objects generates the interaction annulus for each residue vertex with any grid vertices removed if they would exist within the exclusion sphere of another residue vertex (Figure 4.2D). Each of the parameters and shapes are defined in Table 4.1

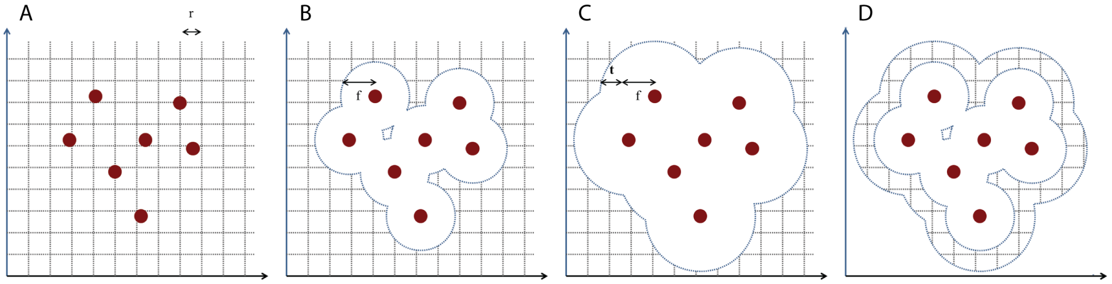

Figure 4.2: Cartoon representation of the PoPP parameters used to define the grid. (A) The distance between grid points is defined by the `resolution` r. (B) The exclusion sphere defines the minimum distance between receptor or ligand vertices, defined by the `fit` parameter f. (C) The inclusion sphere is defined by the `thickness` parameter t in conjunction with the `fit` parameter. (D) The interaction annulus is obtained by subtracting the exclusion sphere from the interaction sphere.

Parameters	**Default Value**	**Definition**
`resolution`	0.5 Å	defines the distance between grid vertices
`fit`	4.0 Å	defines the minimum distance between grid and residue vertices
`thickness`	6.0 Å	defines the maximum depth of the grid extending from any given residue vertex
Grid Shapes	**Defined By**	**Represents**
exclusion sphere	$radius = \texttt{fit}$	space physically occupied by a given residue; van der Waals radius
inclusion sphere	$radius = \texttt{fit} + \texttt{thickness}$	space in which a given residue may interact with another residue, ignoring steric hindrance
interaction annulus	exclusion — inclusion	space in which a given residue may interact with another residue, accounting for steric hindrance

Table 4.1: GridDock parameters and shapes.

As mentioned above, the grid is defined by an array of bit strings, where each active bit represents a single grid vertex. This implementation provides a number of advantages over other grid implementations, such as minimizing the memory footprint, reducing of the number of steps required to access multiple points, and easy access to entire sections of the grid. Basic grid storage on a 64-bit machine would usually require $3 \times 64 = 196$ bits to represent coordinates for a single vertex, or O($M_X M_Y M_Z$) storage, where M is the number of grid points along axis X. More efficient grid implementations can achieve O($M_X M_Y + M_3$) or even O(M_3) for specialized data structures [106]. Our bit string implementation achieves O($nM_X M_Y$) storage, where the n is the number of 64-bit segments required to represent the z-axis[1]. For example, a one-dimensional $1 \times 1 \times 64$ bit string grid would require 64 bits, where the x and y coordinates represent the index in a two-dimensional array, and the z axis data is stored in the bit string. Granted, storing the third dimension in a bit-string limits access to individual vertices, but the access to the third dimension is quickly enabled using efficient bit manipulation and a precomputed table of Hamming weights. Inactive bits may be easily flipped to active bits, allowing the grid to quickly change if necessary (hint: protein flexibility). The bit string implementation also means that multiple vertices may accessed simultaneously, whether for comparison of two grid structures, determining overlap of interaction spheres, or simply counting the number of grid vertices. Bit manipulations equate to improved access speed for multiple vertices, with greater speed gained for each additional point accessed simultaneously. The unfortunate trade-off is that a single vertex is not as easily accessed and requires greater computational complexity. However, easy accessibility to multiple vertices at once also enables large sections of the grid to be identified quickly without additional calculations. Grid vertices within a certain distance of a set of ligand vertices are quickly identified using a bounding box and a number of bit strings equal to the product of the x and y range. The bit string lattice does increase the computational overhead for accessing a single point; however, the storage mechanism provides a marked improvement during grid

[1] The efficiency of the bit string data structure is dependent on the processor used. A 32-bit machine will require twice the memory and number of calculations for the same operations on a 64-bit machine. The GridDock was created, tested, and optimized on a Linux desktop with a 64-bit first generation Intel core I7 processor. The software has not been fully tested on a 32-bit machine.

creation or when checking to see if two residue vertices are within each others' interaction spheres. The size of the grid depends directly on the rotation of the receptor protein in the grid. Rotating the protein to maximize the length along the z axis and minimizing the distance along the x and y axes will minimize the spatial complexity and maximize access efficiency for multiple vertices along the z axis. Conversely, maximizing the range of the x and y axes and minimizing along the z axis will yield a slight improvement for single vertex access for greater distances along the z axis.

The grid covers the entire surface of the receptor, and ligand conformations are placed anywhere on the grid; however, the available interaction space may be narrowed using the CRACLe algorithm. Once CRACLe predicts likely binding sites for a given protein, PoPP can create a second grid object with the same dimensions as the first where the critical residues found by CRACLe are used to generate the interaction space. This second grid can be used as a filter to limit or give preference to grid vertices for placement of ligand residue vertices.

4.1.1 Docking the Ligand

Once the interaction grid is created using the GridDock algorithm described above, PoPP randomly initializes ligand poses on the grid (up to 10,000x). In the current implementation, PoPP allows ligand conformations to remain flexible, with constraints in place to ensure a physically possible conformation, while keeping the receptor rigid. PoPP accepts only the peptide sequence and generates conformations by placing the peptide on the grid one residue at a time. The initial peptide vertex is randomly selected and then randomly placed on any of the possible interaction space grid vertices; if the critical residue grid is used, the initial placement is constrained to a more limited set of grid points. Subsequent placement of ligand residues will begin with either of the sequentially adjacent residues and alternate between extending in both the N- and C-terminal directions until all ligand residues have been placed on the grid. Subsequent ligand vertices are initially constrained to a distance between $1.0-1.4\times$ the value of the `fit` parameter and to a $120°$ angle from the previous two peptide vertices; a smaller grid object is used to define exclusion spheres for each peptide vertex to prevent the peptide from folding back on itself. Subsequent ligand vertices may also be forced to use a critical grid vertex or weighted to prefer a critical residue grid vertex. The resulting conformation mimics a simplified protein backbone and allows for a surprising number of unique conformations; however, the poses a peptide is allowed to sample is directly influenced by the `thickness` of the grid: A thicker grid allows more flexible conformations while a thinner grid forces the initial peptide conformation to `fit` more closely to the receptor surface.

Once PoPP generates the initial peptide poses, the pose may be translated, rotated, or flipped to generate a series of new poses. Both translation and rotation are initially applied to a single peptide vertex, selected at random, and transformed by a random amount. Translation is limited to up to three times the `resolution` of the grid. Rotation randomly selects the ϕ and ψ angles and one of adjacent peptide vertices to use as the origin of rotation. For both types of perturbations, adjacent residues may undergo similar perturbations to help ensure the structure remains physically feasible. With a rotational perturbations for example, the same rotation is often applied to residues further along the peptide sequence using the same origin; if subsequent residues would end up outside of the interaction grid, further perturbations are performed to ensure the entire peptide remains within the acceptable region. Flipping affects the entire pose without directly affecting the conformation: The residue composition is reversed so the ends are swapped although the coordinates of the peptide remain the same. The peptide may assume any conformation provided its vertices (a) remain within the interaction grid and (b) do not enter the exclusion sphere of other peptide vertices.

All poses are scored and ranked using a local Delaunay tessellation and the SNAPP-Interface scores outlined in Chapter 2.5.2. To score a pose, PoPP identifies local receptor residues within a distance of six Å from any peptide vertex, calculates the local Delaunay tessellation of the ligand and receptor vertices, and scores each of the interfacial simplices. For the initial poses, PoPP calculates the SNAPP-Interface score distribution and selects poses based on a Metropolis Monte Carlo algorithm [107], where all poses at the eighty-fifth percentile or higher are automatically chosen, and poses below the threshold are chosen based on a probability distribution described by the Metropolis table. Ligands from selected poses undergo perturbation as described above and subsequently scored. Newly generated poses are selected or discarded as before using the ninetieth percentile from the previous set of scores as the new threshold. The scoring-perturbation loop is continued for 1,000 iterations or until the standard deviation of RMSD for poses in the ninetieth percentile has fallen below 0.2 Å. PoPP then selects up to 1,000 poses with the highest SNAPP-Interface score across all iterations to continue with pose refinement.

Once PoPP selects the top 1,000 poses, each of the ligands from the selected poses undergoes a refinement step where the ligand residues are allowed to deviate from the discrete coordinates given by the interaction grid[2]. Each ligand and nearby receptor vertex is weighted using the data generated from CRACLe. Ligand vertices are translated to maximize the expected edge distance between each it and every other residue vertex, both ligand and residue, it shares an edge with. At each stage of refinement, PoPP performs a local Delaunay tessellation and rescores the pose. Poses are clustered based on pairwise RMSD between other poses, and for each cluster, PoPP calculates a mean consensus pose that is returned to the user.

Unfortunately, the coarse-grained SPPR model used throughout SNAPP, CRACLe, and PoPP does not lend itself to a pretty, high resolution, finalized structure; we must first convert the ligand residues from an SPPR to a full atomic model. This conversion is not a trivial task, and our implementation is not currently available in the current code distribution. Using structural data collected from Dockground, we apply a vector to each peptide vertex with a direction and magnitude to represent the location of the C_α relative to the side chain centroid. Each vector is based on the residue composition of both the peptide vertex and the residue vertices that share a Delaunay edge with it. Once the residue vector has been calculated for each peptide residue, the atomic coordinates of the side chain and peptide backbone are

[2] The refinement algorithm is partially implemented. Only generation of the consensus pose is currently in use.

filled in. The fine-grained peptide pose is returned to the user and may be submitted to other programs such Molprobity [108, 109] for structure refinement.

4.2 Initial Results

Although the algorithm is not fully implemented, we have tested our design using six, experimentally derived PPE, with ligands ranging from 5 to 9 residues in length. For each PPE, we generated initial peptide poses and scored using SNAPP-Bala. In all six examples, we saw the high-scoring initial poses cluster together; in three of the PPE, the main cluster covered the known binding site (Figure 6), though high-scoring initial poses were still found at the target interface in close proximity to the native peptide (Figure 7). After running the PPE through the algorithm several times, GridDock repeatedly identified one or more, high-scoring initial poses closely resembling the native peptide pose for four of the six PPE; these initial poses were all ranked within the top five. Interestingly, some of these unrefined initial poses were found to have similar or improved RMSD to the native pose than those published in an independent study [110]. Refinement of these poses could significantly improve upon our initial results.

These six examples provide proof-of-concept for our algorithm. Even with the initial poses, GridDock has shown some discrimination between native-like and decoy protein-peptide poses. These smaller types of interfaces are particularly common in viral interactions and typical of most peptide drugs. In fact, two of the PPI tested (PDB codes 1AWQ [111] and 1M4P [112]) involve interfaces with HIV proteins, and two are synthetic interfaces designed as an anti-inflammatory peptide and a model to analyze snake venom -bungarotoxin, PDB codes 1ABT [113] and 1TJK, respectively.

4.3 PickPocket

We have also begun development of a novel algorithm that uses GridDock to identify exposed and buried pockets on protein surfaces. Such pockets are desirable targets for drug

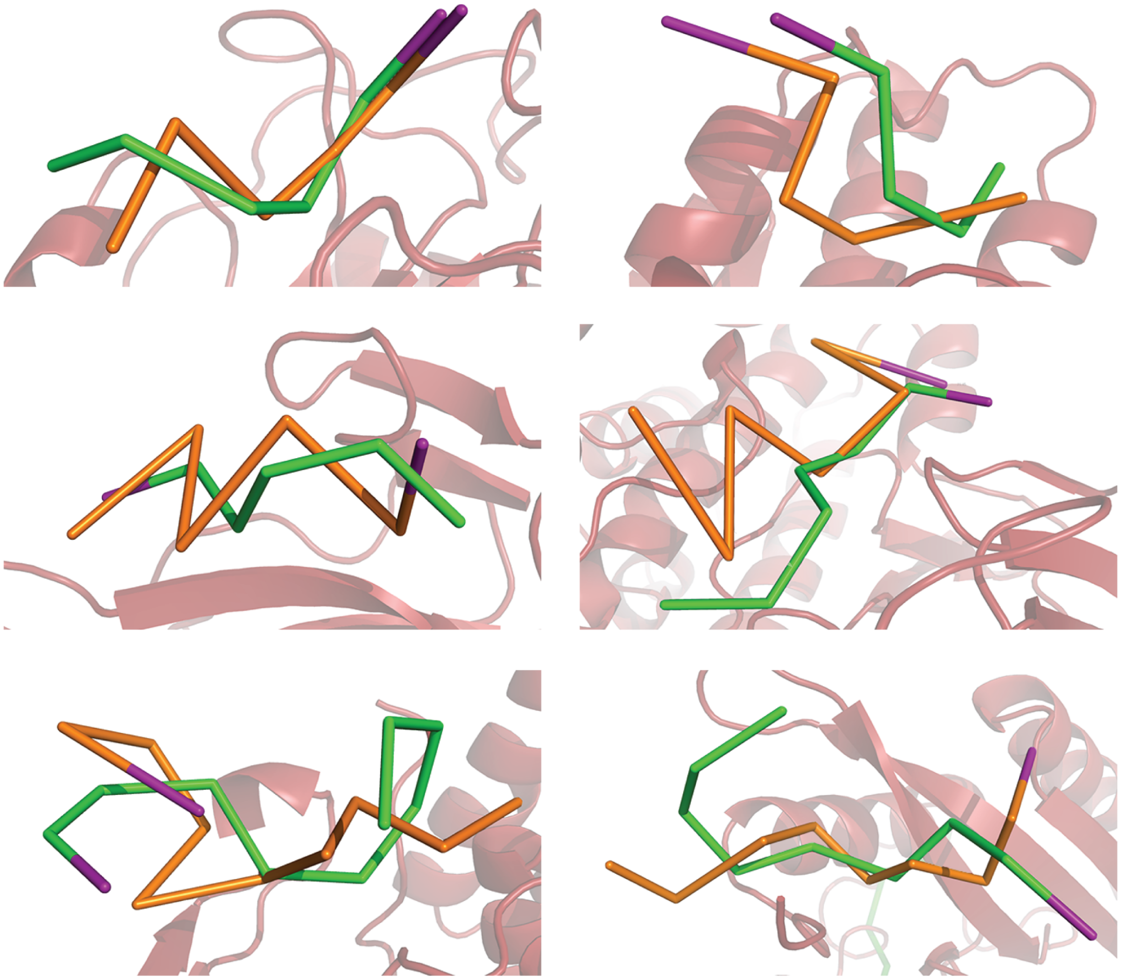

Figure 4.3: PoPP initial results. (A) The HAGPIA sequence from HIV Gag protein (orange) in complex with cyclophilin A [PDB code 1AWQ, RMSD = 3.46]. This pose improves upon results from Tsai et al. (B) A synthetic peptide sequence FLSTK (orange) designed to bind group II phospholipase A in an anti-inflammatory role [PDB code 1TJK, RMSD = 5.15]. (C) A synthetic peptide, sequence KHWVYY (orange), mimics the binding site of nicotinic acetylcholine receptor bound to snake venom-derived alpha-bungarotoxin (BGTX) [PDB code 1ABT, RMSD = 12.61, 4.04 (flipped)]. (D) A synthetic peptide sequence RQMSFRL (orange) in complex with phosphorylase kinase. [PDB code 2PHK, RMSD=11.2]. (E) Residues 22-31, sequence SYTTNAFPGE (orange), of Rac.GTP in complex with p67phox [PDB code 1e96, RMSD = 6.47]. (F) The HIV PTAP domain, sequence PEPTAPEE, (orange), in complex with the UEV domain of human Tumor Susceptibility Protein 101 (Tsg101) [PDB code 1M4P, RMSD = 7.52]

design studies, and a number of studies have developed computational algorithms to identify protein surface pockets [114, 115, 116, 117, 118]. Exposed pockets are more easily identified, but very difficult to define [119]; each algorithm tends to have a different definition of how the border of the pocket is defined [120], and some pockets may overlap one another, making their identification more difficult [119]. Buried pockets are far easier to define due to the lack of an opening on the surface [119] but can be more difficult to detect [117], especially as most algorithms tend to identify pockets by rolling spheres over protein surfaces. In contrast, PickPocket uses the GridDock algorithm to identify pockets based on overlap of the interaction annulus. Pockets, clefts, and cavities on a protein surface are, by their very nature, surrounded by amino acids that will potentially interact with any small molecules that may bind; Thus, we hypothesize that we can detect pockets based on the overlap of their interaction annuli. Although such an overlap could be used to identify shallow pockets on the protein surface, our goal for PickPocket is easier detection of buried pockets.

PickPocket was designed to take a three-dimensional protein structure of a protein and return a list of surface features, e.g., clefts and pockets, on or near the surface of a protein. Although PickPocket does need only protein structure, it currently returns an extensive list of all grid points found in any cleft or pocket. Ideally, each cleft and pocket would be filtered and ranked according to various descriptor characteristics, but currently, PickPocket filters out only very small, independent pockets. We suspect that the hot spots predicted by CRACLe could be used to narrow the list surface features; however, we have not yet benchmarked the algorithm.

PickPocket begins by defining the interaction space for a given protein as described above; however, PickPocket uses an atomic level resolution rather than the previously used SPPR to provide greater resolution. The resulting grid is used to define the space occupied by the protein, providing a reference index that allows small partitions of the grid to maintain a spatial relationship with the protein and each other. PickPocket creates a separate partition for each residue of the protein, defining the interaction space for each residue based on the inclusion and exclusion spheres for each atom in the residue. Each residue partition is said to have an overlap value of 1, i.e., the interaction space defined for the partition is the overlap of the interaction space for exactly 1 protein—itself. Pairwise processing of each of these single residue partitions through a binary AND gate results in a collection of dual residue partitions with an overlap value of 2. Subsequent AND operations generate additional degrees of overlap, which directly affects the number and size of pockets predicted (Figure 4.4). We tested degrees of overlap ranging from 3 to 15 and found that the change in predicted pockets was most drastic between an overlap degree from 6 to 10. A qualitative, visual analysis of the results showed that an overlap of 10 removed the majority of shallow clefts on the protein surface but also resulted in fewer and smaller buried pockets; in contrast, less overlap resulted in increasing amounts of noise, i.e., predicted pockets of only 1-6 grid vertices. To provide a balance between the noise and the size of the predicted pockets, we selected an initial overlap of 8 and included a refinement step to remove pockets with fewer than 8 grid vertices.

Unfortunately, this algorithm has worst-case run time of $O(n^m)$, where n is the number of residues and m is the degree of overlap. By grouping non-overlapping residue partitions together into a single object, we can typically reduce n by a factor somewhere between four to eight, depending on the size of the protein. A careful analysis and redesign of the algorithm could significantly improve the worst-case run time, but such an effort is beyond the scope of the project at this time.

PickPocket was initially created to see if we could identify the buried pockets in a particularly difficult test case: Rec A (PDB code 3cmx) is a particularly large DNA-binding protein consisting of two chains with ten buried, adenosine-5'-diphosphate (ADP) binding pockets between them (Figure 4.5). With an overlap of 8, all ten buried pockets were partially identified (Figure 4.6A,B); however, the defined pocket consistently missed the region where adenine sits. Additionally, the Rec A DNA binding groove was also found (Figure 4.6C). These results show that our GridDock overlap algorithm is effective for identifying pockets and grooves on or near a protein surface; however, it is not yet a useful tool. As mentioned,

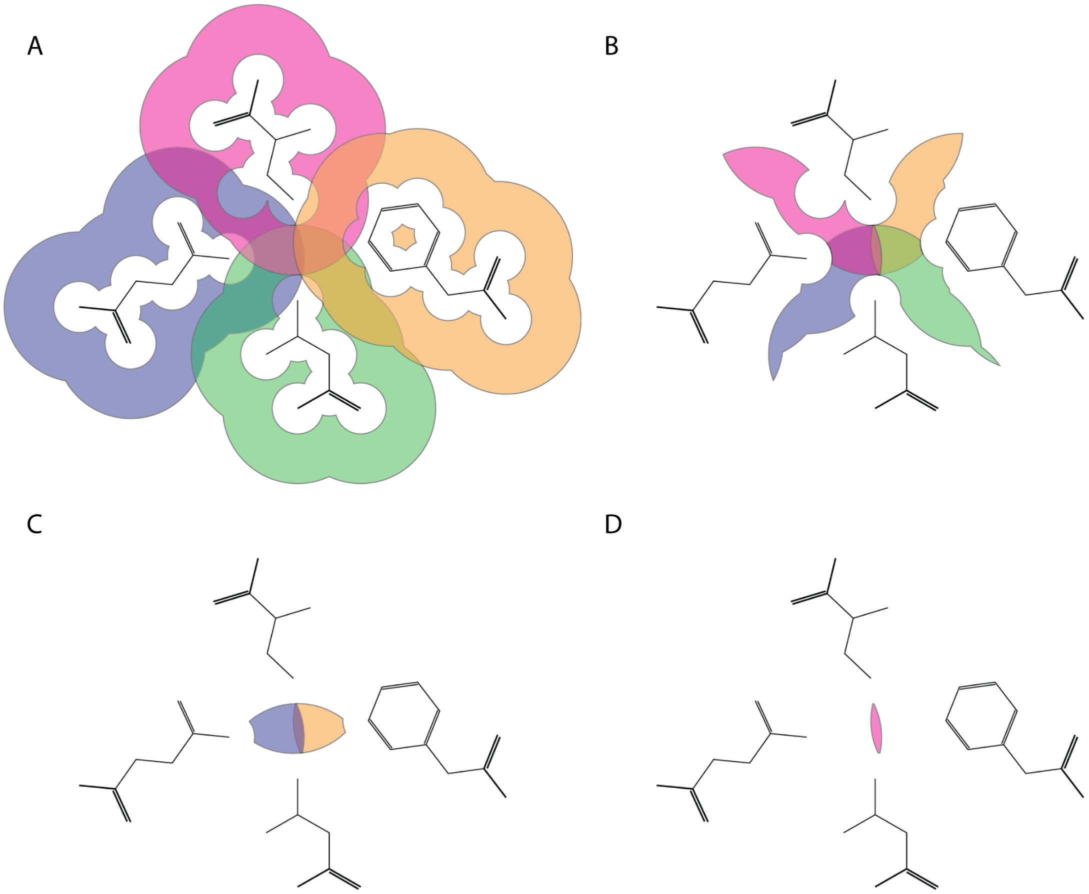

Figure 4.4: A visual representation of the GridDock overlap approach utilized by PickPocket. (A) An overlap degree of 1, i.e., the interaction space defined for each residue. (B-D) The predicted ”pocket” for an overlap degree of 2, 3, and 4.

the none of the ADP pockets completely included the adenine group. We suspect this is due to tighter packing of the adenine group with the surrounding residues, and we hypothesize that creation of a dynamic `fit` variable that depends on the atom type would allow the binding pocket to be more clearly defined. The much larger problem is a lack of specificity. Figure 4.5B clear shows that although PickPocket does correctly identify the pits and grooves on the protein surface, far too much of the surface is predicted. We do not believe that GridDock can be used to rank or select functional pockets, but we hypothesize that we could use CRACLe to filter out pockets that are purely structural. Although SNAPP was designed and developed solely for proteins, there is some evidence that small molecules may bind at protein-protein or protein-peptide hot spots [84].

4.4 Conclusion

Once completed, PoPP will provide a computationally efficient solution for rapidly predicting protein-peptide interactions or protein-protein interactions for ligands with a continuous epitope. Like the CRACLe algorithm, PoPP requires minimal *a priori* data—a three-dimensional structure of the protein receptor and the sequence of the peptide ligand. Furthermore, the grid- and geometry-based pose sampling algorithm allows generation and refinement of ten-thousand new protein-peptide conformations in five to ten minutes, greatly improving on other energy-based docking algorithms. Although initial comparisons with other algorithms are favorable, we cannot compare PoPP against other docking algorithms without additional benchmarking. However, we do not expect PoPP to compete against existing high-resolution docking algorithms such as HADDOCK or RosettaDock, partially due to PoPP's current limitation of a peptide ligand, but largely due to the coarse-grained amino acid representation. Instead, we expect PoPP to be most useful to generate low-resolution bound structures that may be submitted to existing docking and structure refinement programs without the computational overhead associated with *ab initio* docking.

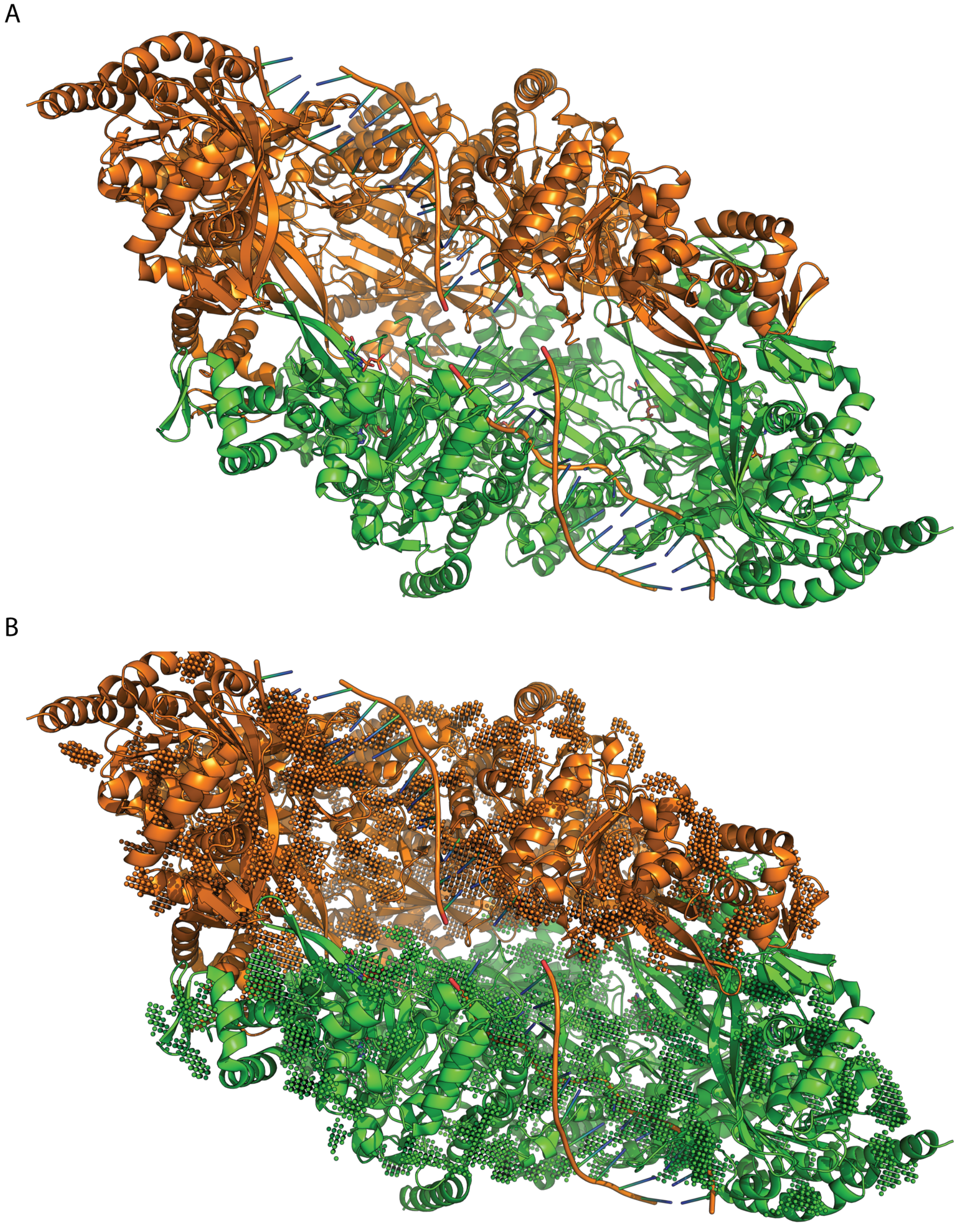

Figure 4.5: (A) The DNA-binding protein Rec A. Each chain has five buried ADP binding pockets. (B) Rec A showing the pockets predicted by PickPocket. The pockets accurately represent clefts, grooves, and pockets for the protein, but show no specificity for ligand binding.

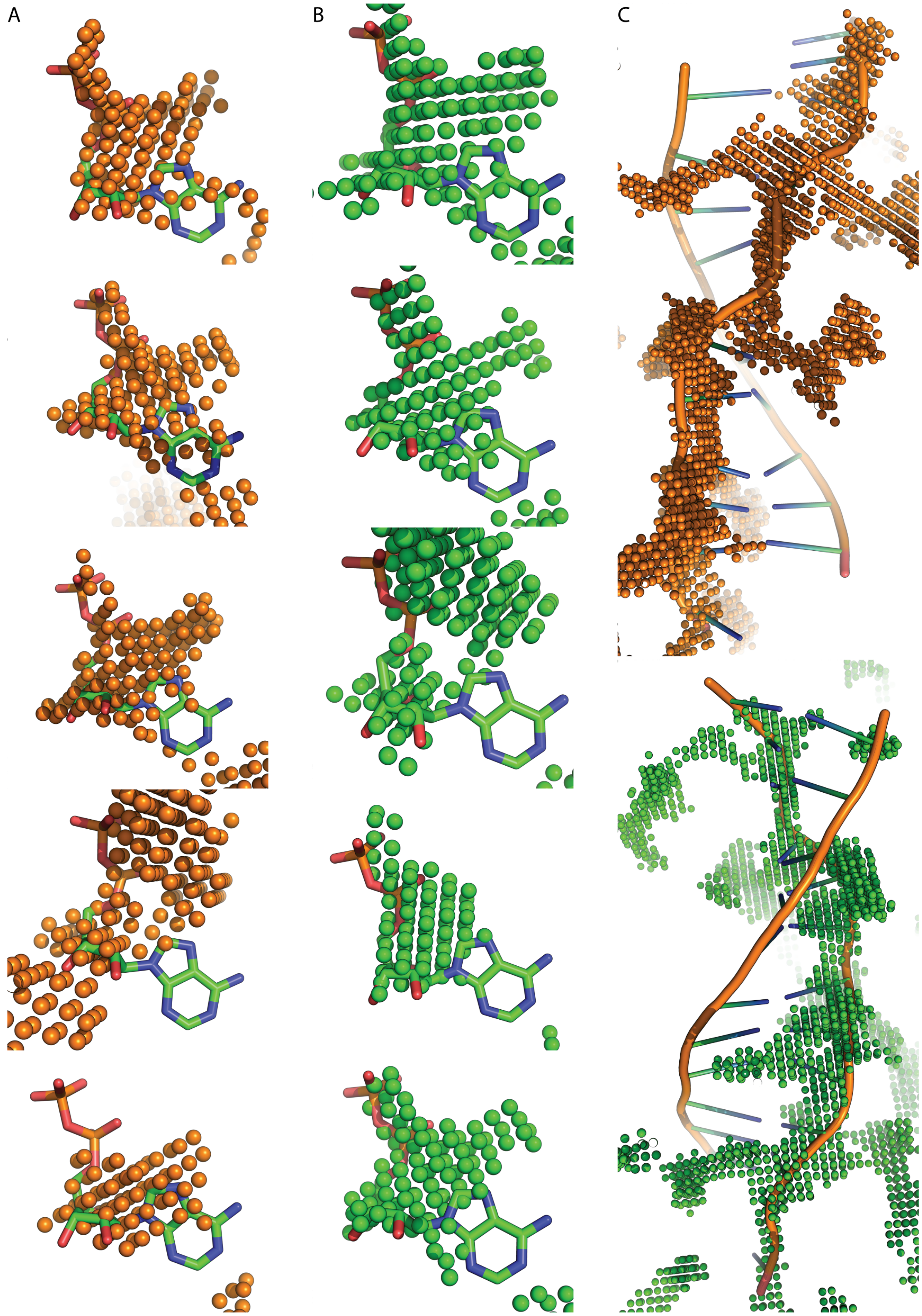

Figure 4.6: (A) The predicted binding pockets for ADP from Rec A chain A. (B) The predicted pockets from Rec A chain B. (C) The DNA-binding grooves from both Rec A chains.

CHAPTER 5

Conclusions and Future Directions

We have described the development of a set of tools for the analysis and prediction of protein interactions. Similar tools currently available to the scientific community require the researcher have some knowledge regarding the interaction, such as where the interface occurs, and were not intended for high-throughput analysis, i.e., they can analyze only a single protein or interaction at a time. Our tools provide researchers a stepping stone for when they know nothing about a particular protein save its three-dimensional structure. SNAPP, CRACLe, and PoPP are all computationally efficient and were developed with high-throughput analysis in mind.

5.1 Further Definition of the SNAPP Applicability Domain

In order to properly define the AD for SNAPP, we could quantify the range of SNAPP descriptor values found within the dataset and evaluate whether a new structure was within the applicability domain based on its own SNAPP descriptor values. We hypothesize that a separate AD could be defined for each of the four inter-chain types of protein-protein interactions defined by Ofran and Rost, along with two additional ADs for protein-peptide and protein-small molecule interactions. Definition of these ADs would be useful not only for identifying PPI that SNAPP cannot predict, but also for classifying PPI into one of the eight categories and providing further discrimination of decoy docking conformations. One potential issue is over-fitting the data due to a lack of statistical power or a specificity of the problem — disregarding the types of amino acids involved may be just as harmful as using the full 20 amino acid alphabet. Using a reduced alphabet such as the 10-letter alphabet set forth by Li et al. [121] could lower the total number of independent variables without a significant loss of data. Although the software includes the ability to use such a reduced alphabet, none of the studies conducted thus far have tested the efficacy over a full 20-letter alphabet.

5.2 Future Work for CRACLe

Our next step for CRACLe is to complete both development and testing of the Complementarity Likelihood algorithm to evaluate the interaction potential between two proteins, i.e., estimate the likelihood that two given proteins will interact. To our knowledge, no algorithm currently exists to perform this task; instead, two proteins must first be docked, and their binding conformation evaluated.

In an effort to further the usability of CRACLe in existing scoring functions, we plan to create two additional SNAPP scoring functions specifically for evaluating a protein-protein interface. The new scoring functions discussed in this paper were designed to evaluate the likelihood of an interaction on a protein surface; the new scoring functions will be tailored to evaluate a particular binding conformation. To this end, we are also developing a convexity index to further improve complementarity predictions.

We are also looking into CRACLe as a tool for evaluating binding site promiscuity. PPI are often characterized by both hydrophobic and polar interactions. The former lend stability without specificity while the latter provide both stability and ligand specificity. We have noticed that hydrophobic residues tend to score well for several possibilities of the fourth residue X, and that polar residues tend to score well for only a couple of specific residues. We hope to analyze these trends and develop a promiscuity index for additional evaluation of predicted binding sites.

5.3 Future Work for PoPP

The PoPP algorithm was originally designed for protein-protein docking, although the requirement of ligand flexibility currently limits PoPP to protein-peptide interactions. In order to extend the algorithm to handle protein ligands, we would need to incorporate (i) a global search algorithm to match likely binding sites, (ii) a surface perturbation algorithm to handle the movement of protein vertices, and (iii) some method to quantify how much any given vertex may be moved based on its characteristics and those of neighboring vertices. The first is fulfilled in CRACLe, especially once the Complementarity Likelihood algorithm is implemented; however, we must keep in mind that CRACLe is not intended to define the interface, but define likely binding sites. As such, CRACLe should be used as a guide, not a hard and fast solution. The second addition is non-trivial: Although the position of a single peptide residue will certainly affect the position of neighboring residues, the effect is largely linear, whereas movement of a residue on a protein surface will affect all neighboring residues. The use of an SPPR model simplifies this movement considerably, and if we are concerned only with the movements of surface residues until the final stages of refinement, we can further simply the calculations. Furthermore, the edge-length descriptors can be used to help constrain the distances between neighboring residues, and we can assume the movement will have a negligible effect at a certain distance. Third, we could quantify the allowed movement of any particular receptor vertex by calculating flexibility descriptors for each residue type. Using the ZDock Protein-Protein Benchmark 4.0, we could calculate a typical range of movement for each surface residue from the bound to the unbound state. Further considerations could include whether the residue was in an α-helix, β-sheet, or loop in both the bound and unbound forms or include basic information concerning neighboring residues, such as polar or non-polar. Such descriptors would allow the creation of a distance distribution that could be sampled using a Monte Carlo algorithm. Although non-trivial, the majority of the code needed to handle protein flexibility already exists in the current PoPP implementation, and once implemented, the same code could handle both protein ligand and receptor flexibility.

In theory, GridDock could be used in a protein-folding algorithm, either as the basis of an algorithm or as an aid in another. The exclusion and interaction spheres could provide constraints on the protein structure, and the residue packing could be scored using SNAPP-Fold. However, we would need to calculate additional residue descriptors such as average angle between descriptors and likelihood of a residue to participate in a secondary structure. As the code stands now, such an application of GridDock is merely hand-waving, but the design and efficacy of the implementation lends itself toward such a task.

The current GridDock implementation uses a constant `fit` to create the exclusion spheres around residue vertices. Although the `fit` parameter may be modified by the user, a constant value does not accurately describe the system. Instead, we propose two possible solutions: (1) use a dynamic `fit` for each vertex based on the average length of each edge the given vertex participates in; and (2) define a set minimum distance for each residue type based on a distribution calculated from a dataset of protein-protein and protein-peptide interactions. The former method would increase the computational overhead required to create the interaction space, including calculation of the minimum distance and creation of exclusion and interaction spheres for each surface residue; however, each would only need to be calculated once and could define a topology much more suited to the surface of the protein. The latter approach provides an empirical solution with minor overhead costs, but does not take the neighboring residues into account. In either case, the difference in the `fit` parameter could be inconsequential considering the minimum distance between two residues, although conserved [122], will change depending on the composition of the second residue. In such case, it would serve to define the minimum distance based on empirical data from glycine-glycine interactions and constrain the distance between particular ligand and receptor vertices pairs during pose generation rather than hard coding the distance using grid vertices. By implementing this constraint during pose generation rather than grid creation, we increase the computational overhead, but we allow the receptor-ligand pose to be more dynamic based on the surface residues involved.

APPENDIX A: SNAPP DATASETS

PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES
119L	✓			1AXN	✓			1C44		✓		1DOZ	✓		
153L	✓			1AY7	✓			1C52	✓			1DP7	✓		
16PK	✓			1B0X		✓		1C5E	✓			1DPS	✓		
19HC	✓			1B0Y	✓			1C75	✓		✓	1DPT	✓		
1A12	✓			1B16	✓			1C7K			✓	1DQ0		✓	
1A1I			✓	1B3A	✓			1C7S			✓	1DQG			✓
1A1X		✓	✓	1B4K	✓			1C90	✓			1DS1			✓
1A1Y	✓			1B5E	✓			1C96			✓	1DSL		✓	
1A28	✓			1B8E		✓		1CC8	✓		✓	1DUA		✓	
1A2P	✓			1B8O	✓			1CCZ	✓		✓	1DUN		✓	
1A2Z	✓			1B8P		✓		1CDY		✓		1DUP		✓	
1A34			✓	1BBH	✓			1CEI		✓		1DVJ	✓		
1A3A	✓			1BD0	✓			1CEM	✓	✓		1DVO		✓	✓
1A3H		✓		1BD8		✓		1CEW			✓	1DZF		✓	✓
1A4I	✓			1BDO	✓			1CEX	✓	✓		1E29			✓
1A6M	✓			1BEA		✓	✓	1CF9	✓			1E4C			✓
1A73	✓			1BEH	✓			1CFB			✓	1E5M		✓	
1A8D	✓			1BF2			✓	1CHD	✓	✓	✓	1E6U			✓
1A8E	✓			1BF4	✓			1CIP	✓			1EB6			✓
1A8L			✓	1BF6	✓			1CJW	✓			1EDG	✓	✓	
1A8Q		✓		1BFD	✓			1CL8	✓		✓	1EDQ		✓	
1A92	✓			1BFG	✓	✓		1CMB	✓			1EDT		✓	
1AAC	✓			1BGF	✓	✓	✓	1CNZ	✓			1EG3		✓	✓
1AAJ		✓		1BHE		✓		1CPQ	✓			1ELK	✓		
1AAY	✓			1BJ7	✓	✓		1CQY		✓	✓	1EOK			✓
1ABA	✓		✓	1BK7	✓			1CRU	✓			1EP0		✓	
1ACF		✓		1BKR	✓	✓	✓	1CTF	✓		✓	1ERV	✓		
1AGJ	✓			1BM8	✓	✓	✓	1CTJ	✓			1ERX	✓		
1AH7	✓		✓	1BOL		✓		1CV8	✓		✓	1ES5	✓	✓	✓
1AIL		✓		1BPI	✓			1CVR			✓	1ES9		✓	✓
1AJS	✓			1BQK	✓			1CWY		✓		1EUR		✓	
1AK0	✓		✓	1BRT	✓		✓	1CYD	✓			1EUW	✓		✓
1AK1		✓		1BS0	✓			1CYO	✓			1EW4		✓	✓
1AKO	✓	✓		1BS9	✓			1CZF	✓			1EY4		✓	
1AKR	✓			1BSM	✓			1CZP	✓			1EYH		✓	✓
1AL3			✓	1BU7	✓			1D2N	✓		✓	1EZJ			✓
1ALY		✓		1BW9	✓			1D3V	✓			1EZW			✓
1AMF	✓			1BXO	✓			1D4O			✓	1F00		✓	✓
1AMM	✓			1BYI	✓	✓	✓	1D7P	✓			1FAS	✓		
1AMP	✓			1BYQ	✓			1DCI	✓			1FC9		✓	
1AMX		✓		1BYR		✓	✓	1DFU	✓		✓	1FK5			✓
1AQB	✓			1BZ4		✓		1DGF	✓			1FKJ	✓		
1ARB	✓	✓	✓	1C02	✓			1DHN	✓	✓		1FL0		✓	✓
1ARL		✓		1C1K	✓		✓	1DIF	✓			1FLM	✓		
1ARU	✓			1C1L	✓		✓	1DK8			✓	1FLP	✓		
1ATZ	✓			1C3D	✓			1DLW			✓	1FLT	✓		
1AUO	✓			1C3K		✓		1DOS	✓			1FNA	✓		

Table A.1: The SNAPP-Fold Training Set.

PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES
1FNC	✓			1HMT	✓			1JQ5			✓	1MIX		✓	✓
1FNF		✓		1HOE		✓		1JUV			✓	1MJC		✓	
1FOB			✓	1HQ0			✓	1JVW		✓		1MK0			✓
1FP2			✓	1HUF		✓	✓	1JX6			✓	1ML4			✓
1FS7			✓	1HX0			✓	1JYE			✓	1MLA	✓		
1FSF		✓		1HYP		✓	✓	1JYH		✓	✓	1MML	✓	✓	
1FT5			✓	1HZ4			✓	1K04			✓	1MOF	✓		
1FTR	✓			1HZT		✓	✓	1K1B		✓		1MOL	✓		
1FXD	✓			1I1W			✓	1K4I			✓	1MOQ	✓		
1G12			✓	1I27			✓	1K7C			✓	1MRJ	✓		
1G2R			✓	1I2A			✓	1KCM		✓	✓	1MRO	✓		
1G5A			✓	1I2T		✓	✓	1KF5		✓		1MSC		✓	✓
1G5T			✓	1I71			✓	1KFN		✓		1MSI	✓		
1G66			✓	1I8O			✓	1KHI		✓		1MSK	✓		
1G6H			✓	1IAB	✓			1KLX		✓	✓	1MUG	✓		
1G6X			✓	1IAP		✓	✓	1KMO			✓	1MUN	✓		✓
1G8A		✓		1ID0			✓	1KN3		✓		1MUW			✓
1G9G			✓	1IDO	✓			1KOE	✓	✓	✓	1MW7		✓	
1GAK		✓	✓	1IFC	✓			1KP6	✓		✓	1MWP		✓	✓
1GBS		✓	✓	1IFR			✓	1KPF	✓			1MZL		✓	
1GCA	✓			1IIB	✓			1KPT	✓			1N5U			✓
1GCU		✓		1IO0			✓	1KQR			✓	1N67			✓
1GDJ	✓			1IO1		✓		1KS8			✓	1NAR	✓	✓	
1GNY			✓	1IQZ			✓	1KT6			✓	1NBC	✓		
1GOF	✓			1ISU	✓			1KUH	✓			1NC5		✓	✓
1GP0			✓	1ITX			✓	1KWB		✓		1NDD	✓		
1GPE	✓			1IXH	✓			1L2P		✓		1NEP			✓
1GPR		✓	✓	1IZC			✓	1L9L			✓	1NF9			✓
1GQN		✓		1J0P			✓	1LAM	✓			1NG2		✓	
1GS5			✓	1J1T			✓	1LC0			✓	1NG6		✓	✓
1GS9		✓	✓	1J23		✓		1LIT		✓		1NIF	✓		
1GSA			✓	1J24			✓	1LKI		✓	✓	1NIJ		✓	✓
1GSM		✓		1J27		✓	✓	1LMB	✓			1NKD	✓	✓	✓
1GUQ	✓			1J2L			✓	1LN4		✓		1NKG			✓
1GVD			✓	1J33		✓		1LO7			✓	1NKR	✓	✓	✓
1GVP	✓	✓	✓	1J74		✓		1LPL		✓		1NLS	✓		✓
1GWE			✓	1J7G		✓		1LSL			✓	1NOA		✓	
1GWM			✓	1J85		✓		1LSY		✓		1NOG		✓	
1GXN		✓		1JB3		✓		1LTU		✓		1NOX	✓		
1GXQ		✓		1JDW			✓	1LWB		✓	✓	1NPK	✓		
1GXU			✓	1JF3			✓	1LYV			✓	1NSJ			✓
1H16			✓	1JF8			✓	1LZL		✓	✓	1NTH			✓
1H2R	✓			1JFB			✓	1M15			✓	1NTY		✓	✓
1H6T		✓		1JHC		✓		1M1H		✓	✓	1NWA		✓	✓
1H72			✓	1JHJ			✓	1M4L			✓	1NWP	✓		
1H75		✓		1JHS		✓	✓	1M9Z			✓	1NWZ			✓
1HCR	✓			1JID			✓	1MAI			✓	1NZY	✓		
1HDO			✓	1JIX			✓	1MD6		✓		1O0X		✓	
1HF8		✓		1JL1		✓	✓	1MF7		✓	✓	1OA4		✓	
1HFC	✓			1JMW		✓		1MFI	✓			1OAA	✓		
1HH8			✓	1JOS		✓	✓	1MG4			✓	1OCY			✓
1HKA	✓	✓		1JOV			✓	1MGT	✓			1OFL			✓
1HLW		✓		1JPE		✓		1MHN		✓		1OGM		✓	

(Continued—2 of 7) The SNAPP-Fold Training Set.

PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES
1OGO			✓	1QNR			✓	1SAU		✓	✓	1UAE	✓		
1OGQ			✓	1QOY			✓	1SDI			✓	1UAI		✓	
1OK0			✓	1QQ4	✓			1SFP		✓	✓	1UAS			✓
1OPC			✓	1QQ5	✓			1SFS			✓	1UCD			✓
1OPD	✓			1QQF			✓	1SJW			✓	1UCS		✓	✓
1OSA	✓			1QR0			✓	1SKZ			✓	1UEK		✓	✓
1OTM		✓		1QS1	✓			1SLL		✓		1UG6			✓
1OUV		✓		1QSA	✓			1SMD	✓			1UGI	✓		
1OX3		✓		1QSG	✓			1SML	✓			1UH4			✓
1OXJ		✓		1QST			✓	1SQW		✓	✓	1UI0			✓
1OZ9		✓	✓	1QTO		✓		1SRA			✓	1UJ8		✓	✓
1P1L		✓		1QTS	✓	✓		1SRV		✓		1UK8			✓
1P1M			✓	1QTW	✓		✓	1STN	✓			1ULN		✓	
1P3C		✓	✓	1QWK		✓		1SU8			✓	1ULR		✓	
1P4P		✓		1QZM		✓	✓	1SUR		✓	✓	1UNP		✓	
1P7S		✓		1QZN		✓		1SYY			✓	1UOH		✓	
1PB1			✓	1R1H			✓	1T1U		✓	✓	1UOK		✓	
1PBN		✓		1R29		✓	✓	1T2D			✓	1UOY		✓	✓
1PBV		✓		1R69		✓		1T2I		✓		1URO	✓		
1PCF	✓			1R6D			✓	1T6C			✓	1UTG		✓	✓
1PDO	✓	✓	✓	1R6J			✓	1T8K			✓	1UWF			✓
1PGV		✓		1R6X			✓	1TAX	✓			1UXY	✓		
1PLC	✓			1R7J		✓	✓	1TCA	✓			1V05		✓	✓
1PMI	✓			1R8O			✓	1TFE	✓	✓	✓	1V2X			✓
1PO5			✓	1R9H		✓		1TFU		✓		1V30			✓
1POA	✓			1R9L			✓	1TG0		✓		1V33			✓
1POC			✓	1RA0			✓	1TGX	✓			1V77		✓	✓
1PSR	✓			1RA9	✓			1THV	✓			1V7Q		✓	
1PUC			✓	1RB9	✓			1TIF	✓		✓	1V8E		✓	
1PZ4			✓	1RC9		✓		1TIG		✓	✓	1V9F			✓
1PZC		✓		1RCF	✓			1TJE		✓		1VBW			✓
1PZW			✓	1RGE	✓			1TJY			✓	1VCA	✓		
1Q1F			✓	1RI6		✓		1TKE			✓	1VCC	✓	✓	✓
1Q2Y		✓		1RIE	✓			1TL2			✓	1VE1			✓
1Q5Z		✓	✓	1RIS		✓		1TM2		✓		1VF8		✓	
1Q6Z			✓	1RJ1		✓		1TML	✓			1VFR	✓		
1QAZ			✓	1RL0		✓		1TOA	✓			1VFY	✓		
1QB7	✓			1RL6		✓	✓	1TP6		✓	✓	1VHH	✓		
1QCX	✓	✓		1RMG			✓	1TPH	✓			1VIE	✓		
1QD1	✓			1RO2			✓	1TQG		✓	✓	1VIN		✓	
1QFT	✓			1ROA		✓		1TT8			✓	1VLS			✓
1QGI	✓		✓	1ROC			✓	1TTB	✓			1VSR	✓		✓
1QH4	✓			1RTQ			✓	1TU9			✓	1VYI			✓
1QH5	✓			1RTT			✓	1TUA		✓	✓	1VYR			✓
1QHD			✓	1RU4			✓	1TUK			✓	1W0N			✓
1QHF	✓			1RV9			✓	1TWU		✓	✓	1W4S			✓
1QHV	✓			1RW7		✓	✓	1TX4	✓			1W53			✓
1QIP	✓			1RWH			✓	1TXL			✓	1W66			✓
1QJD	✓			1RWZ		✓	✓	1TZV		✓		1W7B		✓	
1QKS	✓			1RYO			✓	1U53		✓		1WAB	✓		
1QL0	✓			1RZL	✓			1U5H			✓	1WAP	✓		
1QLM			✓	1S3C			✓	1U5P			✓	1WC2			✓
1QNF	✓			1S7I		✓	✓	1U84			✓	1WCW			✓

(Continued—3 of 7) The SNAPP-Fold Training Set.

PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES
1WD3			✓	1YQE			✓	2BOP	✓			2EAQ			✓
1WER		✓	✓	1YQS			✓	2BUE			✓	2ECE		✓	✓
1WHI	✓	✓	✓	1YT3			✓	2BV9		✓		2EEY		✓	
1WHO		✓		1YTB	✓			2C0H			✓	2EFV			✓
1WKA		✓		1YTQ		✓		2C2Q			✓	2EHG		✓	
1WLU			✓	1YU0			✓	2C2U			✓	2EHZ			✓
1WN2			✓	1YU5		✓	✓	2C71			✓	2EJX		✓	✓
1WNH		✓		1YVE	✓			2C78			✓	2END	✓	✓	✓
1WOL			✓	1YW5		✓		2C7P			✓	2ERF		✓	✓
1WOS		✓		1YZV			✓	2CBP	✓			2ERW		✓	
1WOU		✓	✓	1Z2N			✓	2CC6			✓	2ESK		✓	
1WPA		✓		1ZCJ		✓		2CCQ			✓	2ET1			✓
1WUB			✓	1ZD8		✓	✓	2CE2			✓	2EVB		✓	
1WVF			✓	1ZDY			✓	2CG7		✓	✓	2EX2			✓
1WWC		✓		1ZEQ		✓		2CGQ		✓		2EYI		✓	
1X0T			✓	1ZHX			✓	2CHH			✓	2F15		✓	
1X1E		✓		1ZI8			✓	2CI2		✓		2F5T			✓
1X1N			✓	1ZIN	✓			2CI3		✓		2F60			✓
1X38			✓	1ZJC			✓	2CIW			✓	2F6E		✓	✓
1X3K			✓	1ZK4			✓	2CKK			✓	2F7F			✓
1X3O		✓		1ZLB		✓		2CKX		✓	✓	2FBA			✓
1X54			✓	1ZLM		✓		2CM4			✓	2FBQ		✓	
1X6J		✓		1ZPW		✓		2CPL	✓			2FC3		✓	
1X8Q			✓	1ZSQ			✓	2CTC	✓			2FCB		✓	
1X91		✓	✓	1ZT3			✓	2CUA	✓			2FD5		✓	
1XAK		✓	✓	1ZVA		✓	✓	2CVE			✓	2FDN	✓		✓
1XAU			✓	1ZXX			✓	2CWR		✓	✓	2FGQ			✓
1XAW		✓	✓	1ZZK		✓	✓	2CWS			✓	2FHF			✓
1XBI			✓	256B	✓			2CXC		✓		2FI1			✓
1XFK		✓	✓	2A14			✓	2CYG		✓	✓	2FI9		✓	✓
1XGW		✓		2A1I			✓	2CYJ			✓	2FJ8		✓	✓
1XIK	✓			2A4D		✓		2D48			✓	2FJZ		✓	
1XIX		✓		2A6Z		✓	✓	2D4P		✓	✓	2FK9		✓	
1XKR		✓	✓	2ACY	✓			2D4X		✓	✓	2FL4		✓	
1XMK			✓	2AHN		✓		2D59		✓	✓	2FMA			✓
1XMT			✓	2ASB			✓	2D5B			✓	2FPH		✓	✓
1XNB	✓			2AYD			✓	2D80			✓	2FQ3		✓	✓
1XOV			✓	2AYH	✓			2D8E		✓		2FQX			✓
1XQO		✓	✓	2B0A		✓	✓	2DDX			✓	2FRG		✓	✓
1XQW			✓	2B0J		✓		2DJI			✓	2FUJ		✓	
1XUB			✓	2B0T			✓	2DP9		✓	✓	2FUK		✓	
1XW3			✓	2B1K		✓		2DRI	✓			2FVY			✓
1XWL	✓			2B2H		✓		2DSX			✓	2FWH			✓
1Y8A			✓	2B3M		✓		2DYI		✓	✓	2FYG			✓
1Y93			✓	2B5W			✓	2E0T		✓		2FZP		✓	✓
1YAC	✓			2B8I		✓	✓	2E1F			✓	2G3R			✓
1YD0			✓	2BAA	✓	✓		2E2C		✓		2G5X		✓	
1YFQ			✓	2BBE			✓	2E2O			✓	2G69		✓	
1YGE	✓			2BBK	✓			2E4T			✓	2G7O		✓	✓
1YGT			✓	2BJF			✓	2E56			✓	2GDM			✓
1YHH		✓		2BJQ		✓	✓	2E7Z			✓	2GGC			✓
1YIB		✓		2BK8		✓		2E8E			✓	2GGO		✓	
1YP5		✓		2BKF			✓	2E8F		✓		2GJL			✓

(Continued—4 of 7) The SNAPP-Fold Training Set.

PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES
2GKE			✓	2NAC	✓			2PWQ		✓		2VFO			✓
2GKT		✓		2NLR	✓		✓	2Q35			✓	2VFR			✓
2GQV		✓		2NSN		✓		2Q3T			✓	2VGA		✓	
2GU3			✓	2NUH			✓	2Q88			✓	2VH7		✓	
2GUY			✓	2NVH			✓	2QBV		✓		2VHK			✓
2GWM			✓	2NWD		✓		2QCP			✓	2VPA			✓
2GXG		✓		2NX2		✓	✓	2QDX			✓	2VQ2			✓
2GYZ		✓		2O36			✓	2QED			✓	2VXN			✓
2H1V			✓	2O37		✓		2QHF			✓	2W0G		✓	✓
2H2Z		✓	✓	2O4A			✓	2QHT		✓		2W15			✓
2HBG	✓			2O90			✓	2QIA			✓	2W1R		✓	✓
2HD9			✓	2O9S			✓	2QIM			✓	2W2E			✓
2HDZ		✓		2O9U			✓	2QJL			✓	2W39			✓
2HEW			✓	2OBI		✓		2QRL			✓	2W5Q			✓
2HY7		✓	✓	2OCH		✓		2QSA			✓	2W86			✓
2HYK			✓	2OEB		✓	✓	2QSK			✓	2WAG			✓
2HZC			✓	2OF3		✓	✓	2QT4		✓		2WAO			✓
2I1U		✓		2OFZ			✓	2QVO		✓		2WCJ			✓
2I49		✓	✓	2OG4		✓	✓	2QY9		✓		2WDC			✓
2I53			✓	2OH5			✓	2R0B			✓	2WDS			✓
2I5V			✓	2OIT			✓	2R2Y		✓	✓	2WF7			✓
2I6V		✓		2OKT		✓	✓	2R6Q		✓		2WFO			✓
2I9I		✓	✓	2OP6		✓		2R75			✓	2WJ5		✓	✓
2IBL		✓	✓	2OPC			✓	2R9F			✓	2WNP			✓
2IE8		✓		2OSA		✓		2RB8		✓	✓	2WOL			✓
2IGD	✓	✓		2OSX			✓	2RBK			✓	2WW5			✓
2IGP			✓	2OUJ		✓		2RDQ			✓	2WY4			✓
2II2			✓	2OV0			✓	2RER		✓	✓	2WZO			✓
2IIH			✓	2OVG			✓	2RFA		✓		2X0C		✓	
2IMF			✓	2OY7		✓		2RH3		✓	✓	2X35		✓	
2IMQ			✓	2P09			✓	2RHE		✓		2X3M		✓	✓
2IQY			✓	2P14			✓	2RIK		✓		2X49			✓
2IVN			✓	2P51			✓	2RJ2			✓	2X4L		✓	✓
2IXM		✓	✓	2P52		✓		2RK5		✓		2X5X			✓
2J6A			✓	2P6W			✓	2RKN			✓	2X5Y		✓	✓
2J6B		✓	✓	2P84		✓		2RN2	✓			2XEU			✓
2J70		✓		2PBO		✓		2SAK	✓		✓	2XFG			✓
2J71		✓		2PCY		✓		2SIL		✓		2XJ4		✓	✓
2J8B		✓	✓	2PET		✓		2SN3	✓			2XKI			✓
2J8K			✓	2PFZ			✓	2SPC	✓			2XMZ		✓	✓
2J9V		✓		2PKO		✓		2TGI	✓	✓	✓	2XOD			✓
2JCP		✓		2PLC		✓		2TPS	✓			2XOM			✓
2JDC			✓	2PLQ		✓		2TRX	✓			2XQH			✓
2JEK			✓	2PMR		✓	✓	2UVJ			✓	2XRH			✓
2JFR			✓	2PN6			✓	2UYT			✓	2XU3			✓
2JGP			✓	2PND		✓	✓	2V2P			✓	2XVY			✓
2JHY		✓		2PNE		✓	✓	2V3I			✓	2XWS		✓	✓
2JIC		✓		2PO4		✓		2V84			✓	2XWV			✓
2LIS	✓	✓	✓	2POR	✓		✓	2V8I			✓	2XXP			✓
2MBR			✓	2PPN		✓		2V9V			✓	2XZH			✓
2MCM	✓		✓	2PTD		✓		2VB1			✓	2Y1B			✓
2MHR	✓		✓	2PTH	✓	✓	✓	2VBU			✓	2Y24			✓
2MYR	✓			2PVQ			✓	2VC8		✓	✓	2Y39			✓

(Continued—5 of 7) The SNAPP-Fold Training Set.

PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES
2Y6H			✓	3ADO		✓		3CTG		✓		3FWU		✓	
2Y6X			✓	3AFV			✓	3CTZ			✓	3FY3		✓	
2Y78			✓	3AG7		✓		3CU9			✓	3G21			✓
2Y88			✓	3AGN			✓	3CYR	✓			3G2B			✓
2Y9F		✓	✓	3AHC			✓	3D2A		✓		3G63			✓
2Y9U			✓	3AJ7			✓	3D30			✓	3G6L		✓	
2YFO			✓	3AKS			✓	3D79		✓		3G91			✓
2YGS		✓		3AMR			✓	3DCM			✓	3GA4			✓
2YH5			✓	3APP		✓		3DFG		✓	✓	3GD6			✓
2YL6			✓	3AS8			✓	3DG6			✓	3GHA			✓
2YLH		✓		3ATR			✓	3DHA			✓	3GKJ			✓
2YLN			✓	3ATV			✓	3DJ9		✓		3GKR			✓
2YSK			✓	3AU2			✓	3DNZ			✓	3GMI			✓
2YVI			✓	3AWM			✓	3DSH		✓		3GOE			✓
2YVN		✓		3B02		✓		3DSO			✓	3GON			✓
2YVT			✓	3B0G			✓	3DU1		✓		3GRH		✓	
2YWJ		✓		3B34			✓	3DXT			✓	3GRS	✓		
2YXF		✓		3B7H		✓		3E4G			✓	3GW3		✓	
2YXN			✓	3B9W			✓	3E7P		✓		3GWI			✓
2YZT			✓	3BA1		✓		3E8T			✓	3H04		✓	
2Z0M		✓		3BC9			✓	3EAZ		✓		3H0O			✓
2Z0X			✓	3BCI		✓		3EB5			✓	3H4X			✓
2Z51			✓	3BHS		✓		3EBX	✓			3H6J		✓	✓
2Z5W			✓	3BOD			✓	3EE4			✓	3H6Q		✓	✓
2Z6O			✓	3BOE			✓	3EEH			✓	3H79			✓
2Z72			✓	3BPV		✓	✓	3EIN			✓	3H7I			✓
2Z84		✓	✓	3BQE		✓		3EIP	✓			3H9C			✓
2Z8Z			✓	3BS1			✓	3EJC		✓		3HAK		✓	
2ZA7		✓		3BTO	✓			3EJF		✓		3HHY			✓
2ZB4			✓	3BV4			✓	3EJG		✓		3HJH			✓
2ZCO		✓		3BWH			✓	3EKI			✓	3HLF			✓
2ZGR		✓		3BWZ			✓	3ELN			✓	3HMS			✓
2ZHJ			✓	3BY8			✓	3EMV			✓	3HNX		✓	
2ZJ3			✓	3BZM			✓	3ENU		✓	✓	3HNY		✓	✓
2ZK9			✓	3BZT		✓		3ERS		✓		3HPC			✓
2ZNR			✓	3C70			✓	3EVF			✓	3HR8		✓	✓
2ZPT			✓	3CA7		✓	✓	3EXV		✓		3HRN		✓	
2ZQE		✓	✓	3CHB	✓			3EYE		✓	✓	3HSU			✓
2ZWU			✓	3CHJ			✓	3EZM	✓			3HVV		✓	
2ZXY			✓	3CHM			✓	3F47			✓	3I0W			✓
2ZZJ			✓	3CHY	✓			3F4M			✓	3I2K			✓
3A09			✓	3CIV		✓		3F4S			✓	3I31			✓
3A0X		✓		3CKF		✓		3F6F		✓		3I45			✓
3A2Z		✓	✓	3CL5			✓	3F7L			✓	3I47		✓	
3A38			✓	3CM3			✓	3F7M		✓		3I8Z		✓	
3A4C		✓	✓	3CNU		✓	✓	3FAP			✓	3I94			✓
3A57		✓	✓	3CO1		✓		3FBL			✓	3IB7			✓
3A72			✓	3COU		✓		3FCI			✓	3ID1		✓	✓
3A7L		✓		3CQT		✓		3FH2		✓		3ID4		✓	
3ACH			✓	3CSG		✓		3FO8			✓	3II2			✓
3ACP		✓	✓	3CSP		✓		3FRR		✓		3IIS			✓
3ACX			✓	3CSR		✓		3FTD		✓		3IL8		✓	
3ADG		✓		3CT5			✓	3FW9			✓	3ILS		✓	✓

(Continued—6 of 7) The SNAPP-Fold Training Set.

PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES	PDB Code	Richardson Top 500	PDB Subset	PICES
3IM1		✓		3M66		✓	✓	3PSH			✓	3T3L			✓
3IM3			✓	3M7K			✓	3PTE	✓	✓		3TBD			✓
3IOH			✓	3MBR			✓	3PVH			✓	3TBN			✓
3IOX			✓	3MDM			✓	3PVI	✓			3TCH		✓	
3IP0			✓	3MDU			✓	3PWZ		✓		3TGL		✓	
3IPC			✓	3MSH			✓	3PYP	✓			3THG			✓
3IRP			✓	3MU7			✓	3PYW			✓	3THI			✓
3JS8			✓	3MX7		✓	✓	3PZ9		✓		3TOW		✓	✓
3JU4			✓	3N0K		✓	✓	3Q4O			✓	3TP3		✓	
3JV1		✓	✓	3N11		✓		3Q6B		✓	✓	3TPA		✓	
3JYO			✓	3N17			✓	3Q6L		✓		3TSS		✓	✓
3JZZ		✓		3N2T		✓		3QC7		✓	✓	3TTC			✓
3K3V		✓	✓	3N4J			✓	3QEX			✓	3TUA		✓	
3K4K			✓	3N90			✓	3QM9			✓	3U01			✓
3K6U		✓		3N9K			✓	3QNS			✓	3U0V		✓	
3K7I			✓	3NDI			✓	3QP4			✓	3U81			✓
3K8U		✓	✓	3NDQ			✓	3QSQ		✓	✓	3UJC			✓
3K8W		✓	✓	3NE0		✓	✓	3QUW		✓		3UMH			✓
3KB5		✓		3NE3		✓		3QVP			✓	3UQ8			✓
3KB9			✓	3NE4		✓		3QVX			✓	3US6		✓	✓
3KCW		✓		3NJM		✓		3QY7			✓	3V39			✓
3KFF			✓	3NM6			✓	3QZX			✓	3V46		✓	✓
3KJT		✓		3NOJ			✓	3R26			✓	3VEN			✓
3KLK			✓	3NPH		✓	✓	3R2K			✓	3VGL			✓
3KLR			✓	3NVS			✓	3R5T			✓	3VMN			✓
3KNV			✓	3NYC			✓	3R6U			✓	3VNY			✓
3KP8		✓		3NZM			✓	3R9M			✓	3VUB	✓		✓
3KQ0			✓	3O1C			✓	3RC1			✓	3WRP		✓	
3KR9		✓		3O1Z		✓		3RDJ		✓		3ZR8			✓
3KSX			✓	3O48		✓	✓	3RGA			✓	3ZSL		✓	
3KT9		✓		3O8M			✓	3RHB			✓	3ZT9			✓
3KVD		✓		3OD3			✓	3RLK			✓	3ZUC			✓
3KWE			✓	3OG2			✓	3RNV		✓	✓	3ZUD			✓
3KXT			✓	3OHS			✓	3RT2		✓	✓	451C	✓		
3L8W			✓	3OIG			✓	3RVC		✓	✓	4A02		✓	✓
3L9A			✓	3OO8			✓	3RZN			✓	4AEQ			✓
3L9S		✓		3ORY			✓	3RZY		✓		4D8L		✓	✓
3L9U		✓		3OV8			✓	3S0A		✓	✓	4EUG	✓	✓	
3LDC			✓	3OZP			✓	3S2J			✓	4LZT	✓		
3LE4		✓	✓	3P0F			✓	3S60		✓		4PGA	✓		
3LFP		✓		3P0K			✓	3SBM			✓	4PTI		✓	
3LHC			✓	3P3C			✓	3SC0			✓	5NUL	✓		
3LIG		✓	✓	3P6D			✓	3SDH	✓			5P21	✓		
3LLB		✓	✓	3PAC		✓		3SH4		✓		6CEL	✓		
3LMO		✓		3PB6			✓	3SHS			✓	6GSV	✓		
3LP5			✓	3PBF			✓	3SIL	✓			6XIA		✓	
3LQE		✓		3PFG			✓	3SMZ			✓	7A3H	✓		✓
3LS0		✓	✓	3PG4		✓		3SNY			✓	7ATJ	✓		
3LTJ		✓	✓	3PIW		✓	✓	3SY1			✓	7FD1	✓		
3LX3			✓	3PMS			✓	3SZ7		✓		7RSA	✓		
3LY7		✓		3PO0			✓	3SZY			✓				
3M3G			✓	3PP5			✓	3T2C			✓				
3M5Q			✓	3PR9		✓		3T3K		✓					

(Continued—7 of 7) The SNAPP-Fold Training Set.

PDB Code	Baker Decoys	Rosetta 62	Richardson Top 500	PDB Subset	PICES	PDB Code	Baker Decoys	Rosetta 62	Richardson Top 500	PDB Subset	PICES	PDB Code	Baker Decoys	Rosetta 62	Richardson Top 500	PDB Subset	PICES
1A19		✓				1FM9	✓					1TMQ	✓				
1A2K	✓					1FNA		✓	✓			1TUL		✓			
1A2Y	✓					1G20	✓					1TX6	✓				
1A32		✓				1G6V	✓					1U7F	✓				
1A68		✓				1GPQ	✓					1UBI		✓			
1ACF		✓		✓		1GPW	✓					1UEX	✓				
1AIL		✓		✓		1GVP		✓	✓	✓	✓	1UGH	✓	✓			
1AIU		✓				1HE1	✓					1URN		✓			
1AKJ	✓					1HE8	✓					1UTG		✓		✓	✓
1AVW	✓					1HXY	✓					1VCC		✓	✓	✓	✓
1B3A		✓	✓			1HZ6		✓				1VIE		✓	✓		
1BGF		✓	✓	✓	✓	1IG5		✓				1VLS		✓			✓
1BK2		✓				1IIB		✓	✓			1W1I	✓				
1BKR		✓	✓	✓	✓	1JPS	✓					1WEJ	✓				
1BM8		✓	✓	✓	✓	1KPE		✓				1WHO		✓		✓	
1BQ9		✓				1KU6	✓					1WQ1	✓				
1BTH	✓					1L9B	✓					1XD3	✓				
1BUI	✓					1LIS		✓				1XX9	✓				
1BVN	✓					1LOU		✓				1YVB	✓				
1C8C		✓				1MA9	✓					1ZY8	✓				
1C9O		✓				1NBF	✓					256B		✓	✓		
1CC8		✓	✓		✓	1NPS		✓				2A5T	✓				
1CEI		✓		✓		1OOK	✓					2ACY		✓	✓		
1CG5		✓				1OPD		✓	✓			2BKR	✓				
1CHO	✓					1OPH	✓					2BNQ	✓				
1CTF		✓	✓		✓	1P7Q	✓					2BTF	✓				
1DFJ	✓					1PGX		✓				2CHF		✓			
1DHN		✓	✓	✓		1PPF	✓					2CI2		✓		✓	
1E6I		✓				1PTQ		✓				2CKH	✓				
1E96	✓					1R0R	✓					2FI4	✓				
1ELW		✓				1R4M	✓					2GOO	✓				
1ENH		✓				1R69		✓		✓		2KAI	✓				
1EW4		✓		✓	✓	1RNB		✓				2SNI	✓				
1EWY	✓					1S6V	✓					2TIF		✓			
1EYV		✓				1SCJ		✓				3FAP	✓				✓
1EZU	✓					1SHF		✓				3PRO	✓				
1F51	✓					1T6G	✓					3SIC	✓				
1F6M	✓					1TEN		✓				4UBP		✓			
1FKB		✓				1TIG		✓		✓	✓	5CRO		✓			

Table A.2: The Baker and Rosetta 62 Decoy Sets.

PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero
12AS	✓		1BPL		✓	1DPG	✓		1F2T		✓	1H03	✓	
137L	✓		1BQU	✓		1DQE	✓		1F3U		✓	1H2B	✓	
1A19	✓		1BR1		✓	1DQP	✓		1F46	✓		1H2D	✓	
1A22		✓	1BRS		✓	1DQZ	✓		1F5M	✓		1H2V		✓
1A25	✓		1BT6	✓		1DU5	✓		1F5Q		✓	1H3F	✓	
1A2X		✓	1BTK	✓		1DVK	✓		1F60		✓	1H3L	✓	
1A3A	✓		1BUH		✓	1DXG	✓		1F9W	✓		1H4G	✓	
1A4U	✓		1BVN		✓	1DYS	✓		1FBV		✓	1H4R	✓	
1A73	✓		1BXG	✓		1DYT	✓		1FD3	✓		1H6C	✓	
1A99	✓		1BXT	✓		1DZP	✓		1FGU	✓		1H6P	✓	
1A9N		✓	1BYF	✓		1E05	✓		1FIE	✓		1H7S	✓	
1AA7	✓		1BYU	✓		1E2K	✓		1FJH	✓		1H97	✓	
1AAP	✓		1C02	✓		1E51	✓		1FLG	✓		1H9S	✓	
1ACB		✓	1C8U	✓		1E5R	✓		1FM0		✓	1HCI	✓	
1AD3	✓		1C94	✓		1E6F	✓		1FM6		✓	1HDH	✓	
1AIH	✓		1CDC	✓		1E9G	✓		1FN9	✓		1HDM	✓	
1AKH		✓	1CI6		✓	1E9Y		✓	1FON	✓		1HEI	✓	
1ALL	✓		1CI9	✓		1EAJ	✓		1FP3	✓		1HGX	✓	
1AN9	✓		1CKI	✓		1EAY		✓	1FQK		✓	1HLG	✓	
1AOC	✓		1CLV		✓	1EBF	✓		1FR8	✓		1HR6		✓
1AOH	✓		1CLX	✓		1ECS	✓		1FSY	✓		1HRH	✓	
1AOR	✓		1CM5	✓		1ECX	✓		1FV1		✓	1HRK	✓	
1AOZ	✓		1CQ3	✓		1EDM	✓		1FVK	✓		1HSL	✓	
1AQ0	✓		1CSE		✓	1EE8	✓		1FWK	✓		1HSS	✓	
1AQU	✓		1CSG	✓		1EEJ	✓		1G60	✓		1HW1	✓	
1ATL	✓		1CT9	✓		1EF0	✓		1G6G	✓		1HX1		✓
1AU1	✓		1CXZ		✓	1EGA	✓		1G6V		✓	1HXM		✓
1AUO	✓		1CY9	✓		1EI6	✓		1G6W	✓		1HYN	✓	
1AVA		✓	1D0N	✓		1EK6	✓		1G73		✓	1I1C	✓	
1AYF	✓		1D0Q	✓		1EKE	✓		1G8T	✓		1I2M		✓
1AYO	✓		1D2O	✓		1EO6	✓		1GAN	✓		1I2S	✓	
1AZ3	✓		1D2Z		✓	1EPA	✓		1GD2	✓		1I49	✓	
1AZT	✓		1D3Y	✓		1EPF	✓		1GGG	✓		1I4J	✓	
1AZW	✓		1D7F	✓		1EQT	✓		1GL4		✓	1I58	✓	
1B0N		✓	1D7M	✓		1ERN	✓		1GMV	✓		1I7N	✓	
1B2K	✓		1D8H	✓		1ESG	✓		1GOU	✓		1I8L		✓
1B34		✓	1DBQ	✓		1ET1	✓		1GQ1	✓		1IAJ	✓	
1B41		✓	1DEK	✓		1ETA	✓		1GQI	✓		1IAR		✓
1B49	✓		1DFN	✓		1ETE	✓		1GQP	✓		1IAZ	✓	
1B67	✓		1DG1	✓		1ETH		✓	1GQY	✓		1IBR		✓
1B8Z	✓		1DHF	✓		1EV7	✓		1GT6	✓		1IC2	✓	
1B9N	✓		1DJ0	✓		1EVL	✓		1GT9	✓		1ICI	✓	
1BAY	✓		1DJ7		✓	1EX2	✓		1GU2	✓		1ID1	✓	
1BCM	✓		1DJ8	✓		1EXT	✓		1GUD	✓		1IGQ	✓	
1BD9	✓		1DJN	✓		1EYM	✓		1GVE	✓		1II2	✓	
1BDY	✓		1DJT	✓		1EYV	✓		1GVF	✓		1II7	✓	
1BH5	✓		1DKF		✓	1EZ0	✓		1GWI	✓		1IJY	✓	
1BH8		✓	1DKT	✓		1EZG	✓		1GX2	✓		1IK9	✓	
1BHH	✓		1DL5	✓		1F0K	✓		1GXR	✓		1ILR	✓	
1BK5	✓		1DML		✓	1F14	✓		1GXY	✓		1IPS	✓	
1BLX		✓	1DOK	✓		1F1M	✓		1GY2	✓		1IQ8	✓	
1BMO	✓		1DOS	✓		1F2D	✓		1GYJ	✓		1IRD	✓	
1BP3		✓	1DP4	✓		1F2I	✓		1GZJ	✓		1ITB		✓

Table A.3: The Dockground Training Set.

PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero
1ITV	✓		1KO6	✓		1MXR	✓		1OKJ	✓		1QGK		✓
1IU8	✓		1KP0	✓		1MY7	✓		1OLO	✓		1QH3	✓	
1IVU	✓		1KPT	✓		1MZG	✓		1OLZ	✓		1QH4	✓	
1IX9	✓		1KSH		✓	1MZW		✓	1OMO	✓		1QJS	✓	
1IXM	✓		1KTJ	✓		1N1B	✓		1ON2	✓		1QLT	✓	
1IXS		✓	1KTN	✓		1N1J	✓		1OO0		✓	1QLW	✓	
1IYB	✓		1KU2	✓		1N46	✓		1OOE	✓		1QO8	✓	
1IYJ		✓	1KU6		✓	1N71	✓		1OQM		✓	1QPO	✓	
1J0H	✓		1KU7	✓		1N7S		✓	1OR7		✓	1QPP	✓	
1J2J		✓	1KXQ		✓	1N80	✓		1ORY		✓	1QQ5	✓	
1J30	✓		1KZH	✓		1NA6	✓		1OSY	✓		1QQG	✓	
1J3M	✓		1KZQ	✓		1NA8	✓		1OTJ	✓		1QQJ	✓	
1J7N	✓		1L0W	✓		1NBQ	✓		1OV9	✓		1QRD	✓	
1JAT		✓	1L4D		✓	1NF3		✓	1OVN	✓		1QSD	✓	
1JC5	✓		1L4I	✓		1NGM		✓	1P22		✓	1QSJ	✓	
1JDH		✓	1L6R	✓		1NMU		✓	1P35	✓		1QUP	✓	
1JE5	✓		1L6X		✓	1NO4	✓		1P4K	✓		1QXM	✓	
1JEQ		✓	1L7D	✓		1NOY	✓		1P65	✓		1QXR	✓	
1JI3	✓		1L8D	✓		1NPE		✓	1P6A		✓	1QYD	✓	
1JIW		✓	1LB1		✓	1NQD	✓		1PC6	✓		1R0M	✓	
1JK0		✓	1LGQ	✓		1NQL		✓	1PCF	✓		1R0R		✓
1JK6	✓		1LHP	✓		1NRV	✓		1PFQ	✓		1R0V	✓	
1JKE	✓		1LLF	✓		1NSX	✓		1PH5		✓	1R1D	✓	
1JKG		✓	1LM5	✓		1NW9		✓	1PIX	✓		1R4C	✓	
1JKX	✓		1LM7	✓		1NXM	✓		1PL5	✓		1R5P	✓	
1JL0	✓		1LP1	✓		1O0W	✓		1PN0	✓		1R61	✓	
1JL9	✓		1LQ9	✓		1O1H	✓		1PN4	✓		1R7A	✓	
1JLY	✓		1LT1	✓		1O4T	✓		1PNV	✓		1R8D	✓	
1JMA		✓	1LUC	✓		1O4Z	✓		1PPV	✓		1R8O		✓
1JME	✓		1LWJ	✓		1O62	✓		1PQ1		✓	1R9D	✓	
1JMK	✓		1M0W	✓		1O64	✓		1PQW	✓		1RDL	✓	
1JMV	✓		1M1F	✓		1O7I	✓		1PS6	✓		1RJC		✓
1JOC	✓		1M27		✓	1O7N		✓	1PSA	✓		1RK4	✓	
1JOE	✓		1M2D	✓		1O7Z	✓		1PSR	✓		1RKE		✓
1JQL		✓	1M2V		✓	1O9P	✓		1PUG	✓		1RKU	✓	
1JU9	✓		1M4I	✓		1O9S	✓		1PVC		✓	1RQ2	✓	
1JXH	✓		1M4R	✓		1OBB	✓		1PVH		✓	1RW0	✓	
1K0Z	✓		1M9X		✓	1OBQ	✓		1PVM	✓		1RY9	✓	
1K55	✓		1MA9		✓	1OC0		✓	1PXV		✓	1S0P	✓	
1K5N		✓	1MBY	✓		1OC2	✓		1PY1	✓		1S4C	✓	
1K66	✓		1MDT	✓		1OD5	✓		1PYB	✓		1S6B	✓	
1K8R		✓	1MI3	✓		1ODT	✓		1Q08	✓		1S7H	✓	
1K94	✓		1MIU		✓	1ODZ	✓		1Q1E	✓		1SAW	✓	
1KA8	✓		1MIY	✓		1OF3	✓		1Q2H	✓		1SB2	✓	
1KA9		✓	1MJF	✓		1OF5		✓	1Q2W	✓		1SEI	✓	
1KCF	✓		1MJH	✓		1OFP	✓		1Q3O	✓		1SGH		✓
1KCX	✓		1MJV	✓		1OFZ	✓		1Q67	✓		1SGM	✓	
1KFI	✓		1MK4	✓		1OG5	✓		1QA9		✓	1SH5	✓	
1KI1		✓	1MKB	✓		1OH0	✓		1QB2	✓		1SH8	✓	
1KJN	✓		1MKF	✓		1OI2	✓		1QBI	✓		1SHY		✓
1KLF		✓	1MKZ	✓		1OIA	✓		1QC7	✓		1SJ1	✓	
1KMM	✓		1MQK	✓		1OIO	✓		1QD1	✓		1SJ5	✓	
1KNQ	✓		1MVF		✓	1OKI	✓		1QFH	✓		1SMX	✓	

(Continued—2 of 7) The Dockground Training Set.

PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero
1SPH	✓		1U5K	✓		1VIX	✓		1XB4	✓		1YT5	✓	
1SQ0		✓	1U5W	✓		1VKC	✓		1XBY	✓		1YUZ	✓	
1SQ5	✓		1U60	✓		1VL7	✓		1XCC	✓		1YXY	✓	
1SQU	✓		1U6Z	✓		1VL8	✓		1XCG		✓	1YYQ	✓	
1SR7	✓		1U8S	✓		1VLJ	✓		1XDP	✓		1YZ3	✓	
1SRQ	✓		1U9D	✓		1VP6	✓		1XE7	✓		1Z0J		✓
1SV0	✓		1UAD		✓	1VQ0	✓		1XFF	✓		1Z3E		✓
1SVP	✓		1UC2	✓		1VQU	✓		1XG2		✓	1Z40	✓	
1SVX		✓	1UC8	✓		1VSC	✓		1XG7	✓		1Z6B	✓	
1SYX		✓	1UCR	✓		1VSG	✓		1XGS	✓		1Z85	✓	
1T06	✓		1UDU	✓		1VYB	✓		1XHM		✓	1Z8L	✓	
1T08		✓	1UDV	✓		1VZ0	✓		1XI3	✓		1Z8U		✓
1T0F		✓	1UEF	✓		1VZ6	✓		1XKF	✓		1Z92		✓
1T0P		✓	1UFB	✓		1VZI	✓		1XKO	✓		1Z9H	✓	
1T11	✓		1UGH		✓	1W2Y	✓		1XKZ	✓		1ZC3		✓
1T3C	✓		1UGS		✓	1W3Z	✓		1XLY	✓		1ZC6	✓	
1T4B	✓		1UIU	✓		1W5R	✓		1XTG		✓	1ZGR	✓	
1T6B		✓	1UJN	✓		1W61	✓		1XUV	✓		1ZH1	✓	
1T6F	✓		1UJW		✓	1W7I		✓	1XV8	✓		1ZHQ	✓	
1T6G		✓	1ULK	✓		1W9C	✓		1XVP		✓	1ZI0	✓	
1T6S	✓		1US7		✓	1W9E	✓		1XXO	✓		1ZJJ	✓	
1T6Z	✓		1USU		✓	1WDZ	✓		1XZP		✓	1ZK8	✓	
1T70	✓		1UT7	✓		1WIW	✓		1Y0Z	✓		1ZKE	✓	
1T92	✓		1UTC	✓		1WKQ	✓		1Y3T	✓		1ZKR	✓	
1TA3		✓	1UUZ		✓	1WLE	✓		1Y4J	✓		1ZLH		✓
1TC1	✓		1UV7	✓		1WLG	✓		1Y64		✓	1ZM1	✓	
1TC5	✓		1UW4		✓	1WLT	✓		1Y6H	✓		1ZOQ		✓
1TD9	✓		1UWK	✓		1WMH		✓	1Y6Z	✓		1ZOR	✓	
1TDQ		✓	1UZ3	✓		1WMW	✓		1Y71	✓		1ZPS	✓	
1TE1		✓	1V00	✓		1WMX	✓		1Y8Q		✓	1ZQ9	✓	
1TE5	✓		1V13	✓		1WNF	✓		1Y96		✓	1ZRS	✓	
1TEE	✓		1V25	✓		1WPN	✓		1Y9W	✓		1ZSV	✓	
1THT	✓		1V3E	✓		1WPX		✓	1YAV	✓		1ZTD	✓	
1TLJ	✓		1V4E	✓		1WR8	✓		1YBE	✓		1ZUY	✓	
1TLL	✓		1V4V	✓		1WRD		✓	1YCO	✓		1ZV1	✓	
1TLU	✓		1V5V	✓		1WTJ	✓		1YCS		✓	1ZVP	✓	
1TMQ		✓	1V6Z	✓		1WU9	✓		1YD8		✓	1ZW0	✓	
1TNR		✓	1V74		✓	1WV2	✓		1YDY	✓		1ZWW	✓	
1TO6	✓		1V8C	✓		1WV9	✓		1YHC	✓		1ZYM	✓	
1TQY		✓	1V8H	✓		1WW7	✓		1YKD	✓		1ZZG	✓	
1TT5		✓	1V96	✓		1WWS	✓		1YKH		✓	2A1S	✓	
1TU1	✓		1VB5	✓		1WX1	✓		1YKW	✓		2A2J	✓	
1TUE		✓	1VBK	✓		1WY5	✓		1YLA	✓		2A42		✓
1TVN	✓		1VC1	✓		1WYW		✓	1YLK	✓		2A4N	✓	
1TW9	✓		1VCQ	✓		1WYX	✓		1YLM	✓		2A72	✓	
1TX9	✓		1VDW	✓		1WZ3	✓		1YNF	✓		2A8F	✓	
1TXG	✓		1VE2	✓		1WZD	✓		1YO3	✓		2A8J	✓	
1TY0	✓		1VET		✓	1X24	✓		1YO6	✓		2A9D	✓	
1TZP	✓		1VF6		✓	1X6I	✓		1YOV		✓	2A9F	✓	
1U07	✓		1VH4	✓		1X6M	✓		1YOZ	✓		2A9S	✓	
1U0S		✓	1VH5	✓		1X7I	✓		1YPT	✓		2AA4	✓	
1U58		✓	1VHI	✓		1X8D	✓		1YRT		✓	2ACV	✓	
1U5E	✓		1VI6	✓		1X9M		✓	1YSB	✓		2ADV		✓

(Continued—3 of 7) The Dockground Training Set.

PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero
2AJF		✓	2CCA	✓		2E31		✓	2G04	✓		2HRK		✓
2AKA		✓	2CH4		✓	2E5F	✓		2G0T	✓		2HTH		✓
2AKZ	✓		2CH7	✓		2E5Y	✓		2G38		✓	2HZG	✓	
2AMX	✓		2CH8	✓		2E67	✓		2G3P	✓		2I00	✓	
2ANV	✓		2CJP	✓		2E7Y	✓		2G42	✓		2I25		✓
2AQ1		✓	2CJT	✓		2EB1	✓		2G59	✓		2I26		✓
2AQ2		✓	2CK2	✓		2EBY	✓		2G67	✓		2I2W	✓	
2ASH	✓		2CKL	✓		2ECR	✓		2G95	✓		2I4L	✓	
2ATP		✓	2CMG	✓		2ECS	✓		2GAO	✓		2I5G	✓	
2AVT	✓		2CO7		✓	2EFD		✓	2GCU	✓		2I6H	✓	
2AVW	✓		2CUY	✓		2EGO	✓		2GE7	✓		2I7S	✓	
2AW2		✓	2CW6	✓		2EK0	✓		2GEC	✓		2I9B		✓
2AW6	✓		2CWK	✓		2EKG	✓		2GEY	✓		2I9U	✓	
2AXP	✓		2D0J	✓		2EQ5	✓		2GF2	✓		2IA2	✓	
2AYI	✓		2D13	✓		2ERE	✓		2GHV	✓		2IAB	✓	
2AZJ	✓		2D1G	✓		2ETX	✓		2GHW		✓	2IB0	✓	
2B0R	✓		2D2X	✓		2EUL	✓		2GI7	✓		2IBP	✓	
2B30	✓		2D4G	✓		2EV0	✓		2GIB	✓		2ID3	✓	
2B59		✓	2D4Q	✓		2EX3		✓	2GIY	✓		2IDL	✓	
2B5A	✓		2D5R		✓	2F02	✓		2GJX	✓		2IE4		✓
2B5L		✓	2D68	✓		2F1F	✓		2GK9	✓		2IGQ	✓	
2B5U		✓	2D73	✓		2F4M		✓	2GL7		✓	2IK8		✓
2B7L	✓		2D7D		✓	2F5J	✓		2GOM	✓		2IKC	✓	
2BAY	✓		2D8D	✓		2F6U	✓		2GOP	✓		2IO4		✓
2BDU	✓		2DBB	✓		2F9J		✓	2GPE	✓		2IO8	✓	
2BE1	✓		2DC0	✓		2F9Z		✓	2GRR		✓	2IP2	✓	
2BEX		✓	2DC1	✓		2FA1	✓		2GRX		✓	2IPB	✓	
2BH1		✓	2DC4	✓		2FBK	✓		2GSK		✓	2IQQ	✓	
2BJD	✓		2DDC	✓		2FBL	✓		2GTD	✓		2ITJ	✓	
2BJI	✓		2DFK		✓	2FBN	✓		2GTP		✓	2ITM	✓	
2BJN	✓		2DJX	✓		2FCO	✓		2GU9	✓		2IU5	✓	
2BKK		✓	2DM9	✓		2FDB		✓	2GUZ		✓	2IW2	✓	
2BM5	✓		2DOU	✓		2FDS	✓		2GW1	✓		2IWK	✓	
2BMI	✓		2DPL	✓		2FE3	✓		2GWF		✓	2IXP	✓	
2BNK	✓		2DQL	✓		2FHZ		✓	2GX5	✓		2IYC	✓	
2BOV		✓	2DQR	✓		2FIP	✓		2GY7		✓	2J04		✓
2BSJ	✓		2DQW	✓		2FJR	✓		2GZ1	✓		2J3L	✓	
2BTU	✓		2DSJ	✓		2FJU		✓	2GZ4	✓		2J4H	✓	
2BYK		✓	2DST	✓		2FMT	✓		2H2N	✓		2J4R	✓	
2BYW	✓		2DU8	✓		2FN0	✓		2H4M	✓		2J5B	✓	
2C0L		✓	2DVW		✓	2FN6	✓		2H63	✓		2J5V	✓	
2C1M		✓	2DW6	✓		2FP1	✓		2H98	✓		2J6I	✓	
2C2X	✓		2DWC	✓		2FPG		✓	2HBV	✓		2J6Y	✓	
2C35		✓	2DWU	✓		2FPR	✓		2HD0	✓		2J73	✓	
2C3N	✓		2DX8	✓		2FQ1	✓		2HDI		✓	2J7N	✓	
2C5E	✓		2DXU	✓		2FQM	✓		2HE0	✓		2J96	✓	
2C5J	✓		2DY0	✓		2FSF	✓		2HFK	✓		2J98	✓	
2C61	✓		2DZN		✓	2FT0	✓		2HIQ	✓		2J9U		✓
2C7N		✓	2E0A	✓		2FV2	✓		2HO3	✓		2J9Y		✓
2C8U	✓		2E0K	✓		2FVU	✓		2HP4	✓		2JA3	✓	
2C9N	✓		2E11	✓		2FXM	✓		2HPA	✓		2JAQ	✓	
2CAR	✓		2E2D		✓	2FYI	✓		2HQX	✓		2JBA	✓	
2CC0	✓		2E2E	✓		2FZF	✓		2HQY	✓		2JEE	✓	

(Continued—4 of 7) The Dockground Training Set.

PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero
2JEM	✓		2OZA		✓	2QHO		✓	2UY1	✓		2W3G	✓	
2JG3	✓		2OZN		✓	2QIY	✓		2UY7		✓	2W3P	✓	
2JGD	✓		2P0R	✓		2QJF	✓		2UZQ	✓		2W40	✓	
2JGT	✓		2P19	✓		2QJW	✓		2V29	✓		2W4E	✓	
2JHH	✓		2P1M		✓	2QKH		✓	2V3E	✓		2W4S	✓	
2JI5	✓		2P1R	✓		2QKL		✓	2V3M	✓		2W6A	✓	
2JJB	✓		2P3E	✓		2QKP	✓		2V42	✓		2W73	✓	
2JKI		✓	2P49		✓	2QL2	✓		2V55		✓	2W7R	✓	
2NOG	✓		2P50	✓		2QN5		✓	2V57	✓		2W80		✓
2NQL	✓		2P5L	✓		2QN6		✓	2V5K	✓		2W82	✓	
2NQT	✓		2P7J	✓		2QQQ	✓		2V5Q		✓	2W8B		✓
2NRF	✓		2P7O	✓		2QR4	✓		2V5R	✓		2W9Z		✓
2NRH	✓		2P7V		✓	2QRZ	✓		2V62	✓		2WBW		✓
2NTS		✓	2PA8		✓	2QSD	✓		2V7B	✓		2WD5		✓
2NVW	✓		2PAR	✓		2QSF		✓	2V87	✓		2WE5	✓	
2NWI	✓		2PBD		✓	2QSJ	✓		2V8P	✓		2WEU	✓	
2NX4	✓		2PBK	✓		2QSU	✓		2V8S		✓	2WG4		✓
2NX9	✓		2PCJ	✓		2QT7	✓		2V8Y	✓		2WGK	✓	
2NXO	✓		2PEQ	✓		2QTE	✓		2V9B	✓		2WGQ	✓	
2NXX		✓	2PEZ	✓		2QU7	✓		2VBL	✓		2WJV		✓
2NZ8		✓	2PF4		✓	2QUL	✓		2VDW		✓	2WK7	✓	
2NZW	✓		2PIG	✓		2QXY	✓		2VEO	✓		2WLB	✓	
2O27	✓		2PJU	✓		2QYA	✓		2VFD	✓		2WLV	✓	
2O2A	✓		2PK3	✓		2QYP	✓		2VG0	✓		2WMM	✓	
2O2E	✓		2PKD	✓		2R1J	✓		2VHA	✓		2WMP		✓
2O2K	✓		2PL2	✓		2R25		✓	2VHB	✓		2WNS	✓	
2O3A	✓		2PL7	✓		2R33	✓		2VID	✓		2WP4	✓	
2O4C	✓		2PLG	✓		2R40		✓	2VJP	✓		2WPV		✓
2O70	✓		2PM9		✓	2R5O	✓		2VL1	✓		2WSM	✓	
2O7G	✓		2PMI		✓	2R5Y	✓		2VLG	✓		2WT7		✓
2O8M	✓		2PPT	✓		2R7G	✓		2VLM		✓	2WTO	✓	
2O96	✓		2PQA		✓	2R8Q	✓		2VN6		✓	2WTY	✓	
2O9A	✓		2PQV	✓		2RA6	✓		2VNS	✓		2WU9	✓	
2OB3	✓		2PR1	✓		2RAG	✓		2VOK	✓		2WUK	✓	
2ODF	✓		2PR8	✓		2RAW		✓	2VPH	✓		2WUS		✓
2OFC	✓		2PRV	✓		2RB9	✓		2VPQ	✓		2WV0	✓	
2OFY	✓		2PRZ	✓		2RBE	✓		2VRW		✓	2WVL	✓	
2OGY	✓		2PUY	✓		2RBG	✓		2VSG	✓		2WVQ	✓	
2OIF	✓		2PV2	✓		2RBL	✓		2VSI	✓		2WWX		✓
2OMZ		✓	2PW3	✓		2RC8	✓		2VSK		✓	2WWY	✓	
2ON3	✓		2PW9	✓		2RDH	✓		2VUX	✓		2WXB	✓	
2ONG	✓		2PWJ	✓		2RDJ	✓		2VVM	✓		2WY3		✓
2OOB		✓	2PYW	✓		2REQ		✓	2VVT	✓		2WYT	✓	
2OQQ	✓		2Q1Z		✓	2REX		✓	2VVW	✓		2WZ1	✓	
2OR2	✓		2Q3A	✓		2RJZ	✓		2VX8	✓		2WZI	✓	
2ORI	✓		2Q6Q	✓		2RKL	✓		2VXB	✓		2X0G		✓
2OTN	✓		2Q7N		✓	2RL8	✓		2W01	✓		2X2S	✓	
2OV2		✓	2Q8O	✓		2SCP	✓		2W07		✓	2X3B	✓	
2OVI	✓		2Q8V	✓		2SPC	✓		2W1V	✓		2X3V	✓	
2OWL	✓		2QBY	✓		2UUE		✓	2W2A	✓		2X4I	✓	
2OX1	✓		2QDQ	✓		2UUY		✓	2W2G	✓		2X5Q	✓	
2OX6	✓		2QG7	✓		2UUZ	✓		2W2K	✓		2X65	✓	
2OXG		✓	2QGY	✓		2UW1	✓		2W2X		✓	2X6H	✓	

(Continued—5 of 7) The Dockground Training Set.

PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero
2X78	✓		2YVS	✓		3A5P	✓		3BX1		✓	3D34	✓	
2X7X	✓		2YXZ	✓		3A5Y	✓		3BXF	✓		3D3C		✓
2X9A		✓	2YY0	✓		3A7O	✓		3BXW	✓		3D3K	✓	
2X9M	✓		2YYY	✓		3A7P	✓		3BY4		✓	3D5R		✓
2XB6		✓	2Z0D		✓	3A7Q		✓	3BY6	✓		3D72	✓	
2XBU	✓		2Z0J	✓		3A8R	✓		3BYP	✓		3D78	✓	
2XCC	✓		2Z26	✓		3A9L	✓		3C1T	✓		3D7A	✓	
2XCJ	✓		2Z34	✓		3AB8	✓		3C25	✓		3DA4	✓	
2XDG	✓		2Z3O	✓		3ABI	✓		3C2X	✓		3DA5	✓	
2XDN	✓		2Z3Q		✓	3ADD	✓		3C3K	✓		3DAL	✓	
2XEC	✓		2Z4V	✓		3AEV		✓	3C3Y	✓		3DAW		✓
2XG4		✓	2Z5E	✓		3AGC	✓		3C4N	✓		3DAX	✓	
2XGG	✓		2Z69	✓		3AGJ		✓	3C6K	✓		3DBO		✓
2XGU	✓		2Z6E	✓		3AGX	✓		3C7K		✓	3DBX		✓
2XHE		✓	2Z8V		✓	3AIN	✓		3C9A		✓	3DCG		✓
2XHF	✓		2Z9O	✓		3AKJ	✓		3CAI	✓		3DD9	✓	
2XHY	✓		2ZA4		✓	3AL5	✓		3CAW	✓		3DDT	✓	
2XIT	✓		2ZAE		✓	3AL9	✓		3CEQ	✓		3DEM	✓	
2XLA	✓		2ZB9	✓		3AMI	✓		3CES	✓		3DER	✓	
2XME	✓		2ZCI	✓		3AN1	✓		3CG6	✓		3DEX	✓	
2XMJ	✓		2ZCN	✓		3ANW		✓	3CG7	✓		3DGC		✓
2XOL	✓		2ZD7	✓		3AP1	✓		3CHH	✓		3DGQ	✓	
2XOT	✓		2ZEW	✓		3APT	✓		3CJP	✓		3DHX	✓	
2XPI	✓		2ZFD		✓	3AQB		✓	3CKA	✓		3DI2		✓
2XPL	✓		2ZIG	✓		3AQL	✓		3CKI		✓	3DJW	✓	
2XQN		✓	2ZIU		✓	3ASZ	✓		3CNM	✓		3DLB	✓	
2XR1	✓		2ZIX		✓	3AU4		✓	3COO	✓		3DLK		✓
2XR4	✓		2ZJS		✓	3AV0		✓	3CP7	✓		3DO8	✓	
2XT2	✓		2ZKT	✓		3AYH		✓	3CQ4	✓		3DOR	✓	
2XTY	✓		2ZL7	✓		3AZD	✓		3CQC		✓	3DP7	✓	
2XVE	✓		2ZNJ	✓		3B0F	✓		3CQJ	✓		3DPT	✓	
2XVO	✓		2ZOD	✓		3B4R	✓		3CQR	✓		3DQG	✓	
2XWB		✓	2ZOY	✓		3B4U	✓		3CRY	✓		3DRA		✓
2XWL	✓		2ZSG	✓		3B5M	✓		3CS5	✓		3DRW	✓	
2XWU		✓	2ZSJ	✓		3B82		✓	3CSN		✓	3DS4	✓	
2XZ9	✓		2ZSK	✓		3BA3	✓		3CSX	✓		3DSL	✓	
2Y1E	✓		2ZUV	✓		3BBJ	✓		3CT6	✓		3DTN	✓	
2Y1H	✓		2ZVI	✓		3BC1		✓	3CTP	✓		3DVO	✓	
2Y1X	✓		2ZVR	✓		3BCV	✓		3CTW	✓		3DZY		✓
2Y3W	✓		2ZVT	✓		3BCX	✓		3CU5	✓		3E0J		✓
2Y43	✓		2ZVY	✓		3BEJ	✓		3CUO	✓		3E1H	✓	
2Y4I		✓	2ZW5	✓		3BIL	✓		3CW9	✓		3E1R	✓	
2Y4J	✓		2ZX2	✓		3BIX	✓		3CWF	✓		3E1Y		✓
2Y4O	✓		2ZXH	✓		3BLH		✓	3CWN	✓		3E20		✓
2Y7K	✓		2ZYQ	✓		3BMZ	✓		3CWW	✓		3E33		✓
2Y9M		✓	2ZZ8	✓		3BNY	✓		3CXK	✓		3E3C	✓	
2Y9W		✓	2ZZV	✓		3BOF	✓		3CYP	✓		3E3M	✓	
2YAL	✓		3A0C	✓		3BOX	✓		3CZB	✓		3E3R	✓	
2YC2		✓	3A1N	✓		3BPJ	✓		3CZZ	✓		3E54	✓	
2YH6	✓		3A36	✓		3BS7	✓		3D0R	✓		3E57	✓	
2YHO		✓	3A3D	✓		3BT3	✓		3D0T	✓		3E5Q	✓	
2YV9	✓		3A4K	✓		3BUZ		✓	3D1B	✓		3E7D	✓	
2YVR	✓		3A4M	✓		3BWG	✓		3D1G	✓		3E7J	✓	

(Continued—6 of 7) The Dockground Training Set.

PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero	PDB Code	Dockground Homo	Dockground Hetero
3E7L	✓		3EVI	✓		3G9W		✓	3LCP		✓	3PHF		✓
3E7Q	✓		3EW1	✓		3GB8		✓	3LL8		✓	3PHX		✓
3E96	✓		3EWE		✓	3GCG		✓	3LPE		✓	3PTF		✓
3E9M	✓		3EZ1	✓		3GFU		✓	3LQC		✓	3PV6		✓
3EAB		✓	3F08	✓		3GQB		✓	3LXR		✓	3QBT		✓
3ED4	✓		3F13	✓		3GTY		✓	3M18		✓	3QF7		✓
3EDJ	✓		3F1C	✓		3H11		✓	3M1C		✓	3QHY		✓
3EDV	✓		3F1I		✓	3H3B		✓	3M7F		✓	3QKU		✓
3EEA	✓		3F1L	✓		3H5C		✓	3M7Q		✓	3QLU		✓
3EED	✓		3F1P	✓		3H7H		✓	3MC0		✓	3QML		✓
3EEY	✓		3F1R	✓		3HCT		✓	3MCB		✓	3QQ8		✓
3EFE	✓		3F31	✓		3HE5		✓	3MDY		✓	3R4D		✓
3EFO		✓	3F6C	✓		3HEI		✓	3MFF		✓	3RCZ		✓
3EFY	✓		3F6H	✓		3HJY		✓	3MJ7		✓	3REA		✓
3EFZ	✓		3F6O	✓		3IEY		✓	3MLQ		✓	3REP		✓
3EGG		✓	3F6Q		✓	3ILP		✓	3MP7		✓	3RGF		✓
3EGO	✓		3F70	✓		3IXS		✓	3MZW		✓	3RL0		✓
3EI3		✓	3F84	✓		3JUA		✓	3N06		✓	3RNK		✓
3EI7	✓		3F89	✓		3K1I		✓	3N1F		✓	3RNQ		✓
3EIK	✓		3FAV		✓	3K1R		✓	3N4I		✓	3S4W		✓
3EIP	✓		3FB9	✓		3K2M		✓	3NQU		✓	3S97		✓
3EN0	✓		3FBG	✓		3K6S		✓	3NVM		✓	3S9D		✓
3ENH		✓	3FBK	✓		3K9O		✓	3NW0		✓	3SOH		✓
3ENK	✓		3FBN		✓	3K9P		✓	3O0G		✓	3SXU		✓
3ENT	✓		3FBT	✓		3KCP		✓	3O2Q		✓	3TAC		✓
3EO9	✓		3FBU	✓		3KF8		✓	3OED		✓	3TDU		✓
3EPO	✓		3FCX	✓		3KJL		✓	3OG4		✓	3U1J		✓
3EPW	✓		3FDG	✓		3KLD		✓	3OJM		✓	3YGS		✓
3ER6	✓		3FJU		✓	3KLS		✓	3ONA		✓	3ZWL		✓
3ER9		✓	3FLO		✓	3KNB		✓	3OQ3		✓	3ZYI		✓
3ERR	✓		3FMO		✓	3KXC		✓	3OSS		✓	3ZYJ		✓
3ESG	✓		3FOE		✓	3KYJ		✓	3OUR		✓	4CPA		✓
3ETH	✓		3FPN		✓	3KZ1		✓	3OXU		✓			
3EUP	✓		3FRU		✓	3LB6		✓	3P0G		✓			
3EUS	✓		3FXD		✓	3LBX		✓	3PH0		✓			

(Continued—7 of 7) The Dockground Training Set.

PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero
1A1M	✓			1D3P	✓			1GG6	✓			1JEK	✓		
1A1N	✓			1D4P	✓			1GGD	✓			1JF1	✓		
1A1O	✓			1D4T	✓			1GHA	✓			1JGD	✓		
1A1R	✓			1D5D	✓			1GL1	✓			1JGE	✓		
1A2X	✓		✓	1D5H	✓			1GMC	✓			1JHT	✓		
1A8K	✓			1D8D	✓			1GMD	✓			1JK4	✓		
1A94	✓			1DD3	✓			1GMH	✓			1JMT	✓		
1A9E	✓			1DD4	✓			1GUX	✓			1JOT	✓		
1AB9	✓			1DDV	✓			1GWQ	✓			1JPF	✓		
1ABO	✓			1DKD	✓			1GWR	✓			1JPG	✓		
1AFQ	✓			1DKX	✓			1GYB	✓			1JPL	✓		
1AGB	✓			1DKZ	✓			1GZL	✓			1JQ8	✓		
1AGC	✓			1DLK	✓			1H24	✓			1JQ9	✓		
1AGD	✓			1DOW	✓			1H25	✓			1JRR	✓		
1AGE	✓			1DUY	✓			1H26	✓			1JUF	✓		
1AGF	✓			1DUZ	✓			1H27	✓			1JUQ	✓		
1AIK	✓			1DXP	✓			1HC9	✓			1JW6	✓		
1APM	✓			1DY8	✓			1HHI	✓			1JWG	✓		
1AQC	✓			1DY9	✓			1HHJ	✓			1JWY	✓		
1ATP	✓			1E27	✓			1HHK	✓			1JX2	✓		
1AW8	✓			1E54	✓			1HJA	✓			1JXP	✓		
1AWI	✓			1E8N	✓			1HOC	✓			1K4W	✓		
1AWQ	✓			1EE4	✓			1HSA	✓			1K5N	✓		✓
1AWU	✓			1EE5	✓			1HTM	✓			1K74	✓		
1B0G	✓			1EEY	✓			1HXL	✓			1K7L	✓		
1BAI	✓			1EEZ	✓			1HXZ	✓			1K8D	✓		
1BBZ	✓			1EG4	✓			1HY2	✓			1KCS	✓		
1BC5	✓			1EGP	✓			1I1Y	✓			1KD8	✓		
1BE9	✓			1EHK	✓			1I31	✓			1KD9	✓		
1BII	✓			1EJ4	✓			1I4F	✓			1KJ3	✓		
1BJR	✓			1EJH	✓			1I7R	✓			1KJ7	✓		
1BT6	✓	✓		1EJO	✓			1I7U	✓			1KJF	✓		
1CA0	✓			1ELR	✓			1I8I	✓			1KJG	✓		
1CA9	✓			1ELW	✓			1I8K	✓			1KJH	✓		
1CDK	✓			1EMU	✓			1IHJ	✓			1KJM	✓		
1CDM	✓			1EVH	✓			1IID	✓			1KJV	✓		
1CE0	✓			1EYX	✓			1IK9	✓	✓		1KLU	✓		
1CE1	✓			1F47	✓			1INQ	✓			1KPU	✓		
1CF0	✓			1F4V	✓			1IQ5	✓			1KPV	✓		
1CHO	✓			1F7A	✓			1IR3	✓			1KU8	✓		
1CIQ	✓			1FCH	✓			1ISQ	✓			1KUJ	✓		
1CJF	✓			1FM6	✓		✓	1IWQ	✓			1KY7	✓		
1CJR	✓			1FM9	✓			1J19	✓			1KYD	✓		
1CKA	✓			1FMO	✓			1J7Z	✓			1KYF	✓		
1CKB	✓			1FV1	✓		✓	1J80	✓			1KYU	✓		
1CLV	✓		✓	1FYN	✓			1J81	✓			1KZO	✓		
1CM1	✓			1FZJ	✓			1J82	✓			1KZP	✓		
1CM4	✓			1FZK	✓			1J8H	✓			1L2I	✓		
1CMI	✓			1FZM	✓			1JAC	✓			1L3R	✓		
1CN3	✓			1FZO	✓			1JBP	✓			1L6O	✓		
1CQ4	✓			1G7P	✓			1JCS	✓			1L6X	✓		✓
1CZY	✓			1G7Q	✓			1JD5	✓			1L7Z	✓		
1D3D	✓			1GCT	✓			1JDP	✓			1LD9	✓		

Table A.4: The PepX Test Set.

PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero
1LEG	✓			1OF2	✓			1RBE	✓			1SYS	✓		
1LEK	✓			1OGA	✓			1RBF	✓			1SYV	✓		
1LEW	✓			1OGT	✓			1RBG	✓			1T01	✓		
1LK2	✓			1OJ5	✓			1RBH	✓			1T08	✓		✓
1LQ8	✓			1OKV	✓			1RBI	✓			1T0M	✓		
1LVB	✓			1OM9	✓			1RDQ	✓			1T0N	✓		
1LVM	✓			1OQN	✓			1RDT	✓			1T1W	✓		
1M26	✓			1OQO	✓			1RIW	✓			1T1X	✓		
1M2Z	✓			1OSV	✓			1RJK	✓			1T1Y	✓		
1M45	✓			1OSZ	✓			1RJY	✓			1T1Z	✓		
1M46	✓			1OU8	✓			1RK1	✓			1T20	✓		
1M6O	✓			1OV3	✓			1RK3	✓			1T21	✓		
1M7E	✓			1OW6	✓			1RKG	✓			1T22	✓		
1MF4	✓			1OXG	✓			1RKH	✓			1T3L	✓		
1MFG	✓			1P4U	✓			1RST	✓			1T4F	✓		
1MFL	✓			1P7V	✓			1RSU	✓			1T4U	✓		
1MHC	✓			1P7W	✓			1RXZ	✓			1T4V	✓		
1MIZ	✓			1P9U	✓			1RZX	✓			1T5W	✓		
1MK7	✓			1PCX	✓			1S6C	✓			1T5Z	✓		
1MT7	✓			1PFG	✓			1S7Q	✓			1T65	✓		
1MT8	✓			1PIP	✓			1S7S	✓			1T6O	✓		
1MT9	✓			1PJ8	✓			1S7T	✓			1T73	✓		
1MTP	✓			1PJM	✓			1S7V	✓			1T74	✓		
1MV9	✓			1PJN	✓			1S8D	✓			1T76	✓		
1MVC	✓			1PQ1	✓		✓	1S9W	✓			1T79	✓		
1MVU	✓			1PU9	✓			1S9X	✓			1T7F	✓		
1MWA	✓			1PXD	✓			1S9Y	✓			1T7M	✓		
1MXE	✓			1PYO	✓			1SDZ	✓			1T7R	✓		
1MZN	✓			1PZL	✓			1SE0	✓			1TDV	✓		
1MZW	✓		✓	1Q1S	✓			1SEM	✓			1TFC	✓		
1N12	✓			1Q1T	✓			1SJE	✓			1TG4	✓		
1N2R	✓			1Q2D	✓			1SJH	✓			1TJK	✓		
1N4H	✓			1Q3P	✓			1SKG	✓			1TMC	✓		
1N4M	✓			1Q61	✓			1SLD	✓			1TN6	✓		
1N7F	✓			1Q62	✓			1SLE	✓			1TN7	✓		
1N8O	✓			1Q8T	✓			1SLG	✓			1TN8	✓		
1NAN	✓			1Q8U	✓			1SMH	✓			1TOQ	✓		
1NIW	✓			1Q8W	✓			1SP5	✓			1TP3	✓		
1NLN	✓			1Q94	✓			1SQK	✓			1TP5	✓		
1NQ7	✓			1QD6	✓			1SRN	✓			1TSQ	✓		
1NRL	✓			1QEW	✓			1SSA	✓			1TSU	✓		
1NTV	✓			1QLS	✓			1SSB	✓			1TVB	✓		
1NU2	✓			1QMZ	✓			1SSC	✓			1TVH	✓		
1NVQ	✓			1QO3	✓			1SSH	✓			1TW6	✓		
1NVR	✓			1QR1	✓			1STC	✓			1TWB	✓		
1NVS	✓			1QSC	✓			1STR	✓			1U00	✓		
1NX0	✓			1QTX	✓			1STS	✓			1U3R	✓		
1NX1	✓			1QVO	✓			1SVE	✓			1U3S	✓		
1O6K	✓			1R17	✓			1SVF	✓			1U6H	✓		
1O6L	✓			1R2B	✓			1SVG	✓			1U7B	✓		
1O9U	✓			1R5V	✓			1SVH	✓			1U8T	✓		
1OAI	✓			1R9N	✓			1SVZ	✓			1U9E	✓		
1OEB	✓			1RBC	✓			1SYQ	✓			1U9L	✓		

(Continued—2 of 6) The PepX Test Set.

PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero
1UEF	✓	✓		1XH6	✓			1ZW2	✓			2C1A	✓		
1UGX	✓			1XH7	✓			1ZYS	✓			2C1B	✓		
1UHD	✓			1XH8	✓			2A25	✓			2C3I	✓		
1UHE	✓			1XH9	✓			2A3I	✓			2C5I	✓		
1UJ0	✓			1XHA	✓			2A3Z	✓			2C5K	✓		
1UK4	✓			1XIU	✓			2A40	✓			2C7U	✓		
1UKH	✓			1XME	✓			2A4G	✓			2CCH	✓		
1UTC	✓	✓		1XOC	✓			2A4Q	✓			2CE8	✓		
1UTI	✓			1XOW	✓			2A4R	✓			2CE9	✓		
1UVQ	✓			1XR8	✓			2A6I	✓			2CHA	✓		
1UXS	✓			1XR9	✓			2A83	✓			2CIK	✓		
1UXW	✓			1XU2	✓			2ADV	✓		✓	2CK3	✓		
1V1T	✓			1Y2A	✓			2AI4	✓			2CLR	✓		
1VAC	✓			1Y3A	✓			2AIJ	✓			2CLV	✓		
1VAD	✓			1Y43	✓			2AIK	✓			2CLZ	✓		
1VC3	✓			1YBO	✓			2AK5	✓			2CNY	✓		
1VGC	✓			1YCQ	✓			2AO6	✓			2CNZ	✓		
1VGK	✓			1YDI	✓			2AQ9	✓			2CO0	✓		
1VWA	✓			1YDP	✓			2ARQ	✓			2CO1	✓		
1VWB	✓			1YDR	✓			2ARR	✓			2CO2	✓		
1VWC	✓			1YDS	✓			2ATP	✓		✓	2CO4	✓		
1VWD	✓			1YDT	✓			2AV1	✓			2CVY	✓		
1VWE	✓			1YFN	✓			2AV7	✓			2CWG	✓		
1VWF	✓			1YK0	✓			2AXF	✓			2D0N	✓		
1VWG	✓			1YMT	✓			2AXG	✓			2D10	✓		
1VWH	✓			1YN6	✓			2B1J	✓			2D1K	✓		
1VWM	✓			1YN7	✓			2B1N	✓			2D1X	✓		
1VWN	✓			1YOK	✓			2B1V	✓			2D3G	✓		
1VWO	✓			1YP0	✓			2B1Z	✓			2D5W	✓		
1VWP	✓			1YPH	✓			2B23	✓			2DEW	✓		
1W0V	✓			1YUC	✓			2B3G	✓			2DEX	✓		
1W0W	✓			1YWO	✓			2B9H	✓			2DEY	✓		
1W3C	✓			1YY6	✓			2B9I	✓			2DF6	✓		
1W70	✓			1YYE	✓			2B9J	✓			2DRK	✓		
1W80	✓			1YYP	✓			2BBA	✓			2DRM	✓		
1W9O	✓			1Z96	✓			2BCX	✓			2DS2	✓		
1WBP	✓			1ZAF	✓			2BE6	✓			2DS8	✓		
1WBX	✓			1ZAV	✓			2BFY	✓			2DUJ	✓		
1WBY	✓			1ZAW	✓			2BJ4	✓			2DYH	✓		
1WBZ	✓			1ZAX	✓			2BP3	✓			2DYP	✓		
1WKW	✓			1ZDT	✓			2BR8	✓			2DZE	✓		
1X11	✓			1ZGX	✓			2BRQ	✓			2E7L	✓		
1X2R	✓			1ZGY	✓			2BSR	✓			2EEO	✓		
1X76	✓			1ZH7	✓			2BSS	✓			2ERZ	✓		
1X78	✓			1ZHK	✓			2BST	✓			2F31	✓		
1X7B	✓			1ZHL	✓			2BUO	✓			2F3Y	✓		
1X7J	✓			1ZKK	✓			2BVO	✓			2F3Z	✓		
1X7Q	✓			1ZKY	✓			2BVP	✓			2F53	✓		
1X7R	✓			1ZSD	✓			2BVQ	✓			2F7E	✓		
1XB7	✓			1ZT1	✓			2BYP	✓			2F7X	✓		
1XH3	✓			1ZUK	✓			2BZ8	✓			2FAI	✓		
1XH4	✓			1ZV7	✓			2BZK	✓			2FF6	✓		
1XH5	✓			1ZVZ	✓			2BZW	✓			2FFF	✓		

(Continued—3 of 6) The PepX Test Set.

PDB Code	PepX	Dockground Homo	Dockground Hetero
2FFU	✓		
2FGE	✓		
2FGR	✓		
2FKA	✓		
2FLK	✓		
2FLU	✓		
2FLW	✓		
2FMF	✓		
2FMH	✓		
2FMI	✓		
2FMK	✓		
2FNS	✓		
2FNT	✓		
2FOJ	✓		
2FOO	✓		
2FOP	✓		
2FOT	✓		
2FVJ	✓		
2FYM	✓		
2FYS	✓		
2FYZ	✓		
2FZ3	✓		
2G1T	✓		
2G30	✓		
2G5L	✓		
2G5O	✓		
2G9H	✓		
2GCH	✓		
2GCT	✓		
2GFC	✓		
2GGM	✓		
2GIT	✓		
2GMT	✓		
2GNF	✓		
2GNG	✓		
2GNH	✓		
2GNI	✓		
2GNJ	✓		
2GNS	✓		
2GPH	✓		
2GPO	✓		
2GT9	✓		
2GTK	✓		
2GTW	✓		
2GTZ	✓		
2GU8	✓		
2GUO	✓		
2GVF	✓		
2H1C	✓		
2H1P	✓		
2H2F	✓		
2H4J	✓		
2H4P	✓		
2H4Q	✓		
2H59	✓		
2H6H	✓		
2H6P	✓		
2HC4	✓		
2HCJ	✓		
2HD4	✓		
2HFP	✓		
2HI8	✓		
2HJK	✓		
2HJL	✓		
2HKF	✓		
2HLB	✓		
2HN7	✓		
2HPL	✓		
2HPZ	✓		
2HQW	✓		
2HT9	✓		
2I04	✓		
2I0L	✓		
2ILM	✓		
2IV9	✓		
2IVZ	✓		
2IZX	✓		
2J6F	✓		
2J6O	✓		
2J7X	✓		
2J7Y	✓		
2JAM	✓		
2JBY	✓		
2JDI	✓		
2JDL	✓		
2JDO	✓		
2JDR	✓		
2JDS	✓		
2JDT	✓		
2JDV	✓		
2JET	✓		
2JF1	✓		
2JF9	✓		
2JGB	✓		
2JGC	✓		
2JK9	✓		
2JKG	✓		
2MHA	✓		
2MIP	✓		
2NM1	✓		
2NNU	✓		
2NPA	✓		
2NPH	✓		
2NUD	✓		
2NV7	✓		
2NW3	✓		
2NXD	✓		
2NXL	✓		
2NXM	✓		
2O02	✓		
2O4J	✓		
2O4R	✓		
2O5G	✓		
2O60	✓		
2O88	✓		
2O8M	✓	✓	
2O9Q	✓		
2O9V	✓		
2OC0	✓		
2ODB	✓		
2OEI	✓		
2OH0	✓		
2OI9	✓		
2OIN	✓		
2OJF	✓		
2OKR	✓		
2OTW	✓		
2OVH	✓		
2P0W	✓		
2P15	✓		
2P1L	✓		
2P1N	✓		
2P1O	✓		
2P1Q	✓		
2P1T	✓		
2P1U	✓		
2P1V	✓		
2P4R	✓		
2P54	✓		
2P5E	✓		
2P5W	✓		
2P6B	✓		
2P8O	✓		
2PAV	✓		
2PEH	✓		
2PKS	✓		
2PQ2	✓		
2PQK	✓		
2PUY	✓	✓	
2PV1	✓		
2PV2	✓	✓	
2PYE	✓		
2Q0N	✓		
2Q3Y	✓		
2Q6G	✓		
2Q6W	✓		
2Q7I	✓		
2Q7J	✓		
2Q7K	✓		
2Q7L	✓		
2QA8	✓		
2QAB	✓		
2QAC	✓		
2QBW	✓		
2QBX	✓		
2QGT	✓		
2QGW	✓		
2QKH	✓		✓
2QKI	✓		
2QL5	✓		
2QL7	✓		
2QL9	✓		
2QME	✓		
2QN6	✓		✓
2QOS	✓		
2QPY	✓		
2QR9	✓		
2QSE	✓		
2QV1	✓		
2QXM	✓		
2QXV	✓		
2QYF	✓		
2QZO	✓		
2R28	✓		
2R2M	✓		
2R7G	✓	✓	
2RFX	✓		
2RIV	✓		
2RIW	✓		
2RKY	✓		
2SIV	✓		
2UVX	✓		
2UVY	✓		
2UVZ	✓		
2UW0	✓		
2UW3	✓		
2UW4	✓		
2UW5	✓		
2UW6	✓		
2UW7	✓		
2UW8	✓		
2UW9	✓		
2UWJ	✓		
2UZT	✓		
2UZU	✓		
2UZV	✓		
2UZW	✓		
2V17	✓		
2V1R	✓		
2V1T	✓		
2V2F	✓		
2V2W	✓		

(Continued—4 of 6) The PepX Test Set.

PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero
2V2X	✓			2ZMH	✓			3CVN	✓			3ECH	✓		
2V3S	✓			2ZMI	✓			3CVP	✓			3EDQ	✓		
2V8C	✓			2ZMJ	✓			3CWD	✓			3EG6	✓		
2V8W	✓			2ZNE	✓			3CY2	✓			3EHU	✓		
2V8X	✓			2ZPY	✓			3CY3	✓			3EMH	✓		
2V8Y	✓	✓		2ZVM	✓			3CYY	✓			3EMW	✓		
2VAA	✓			2ZVV	✓			3D18	✓			3EQS	✓		
2VAB	✓			2ZXN	✓			3D1E	✓			3EQY	✓		
2VAY	✓			3BEJ	✓	✓		3D1F	✓			3ERD	✓		
2VDR	✓			3BEV	✓			3D24	✓			3ERY	✓		
2VGC	✓			3BFQ	✓			3D25	✓			3ESK	✓		
2VJ0	✓			3BFW	✓			3D2U	✓			3ET1	✓		
2VKN	✓			3BG4	✓			3D32	✓			3ET3	✓		
2VLJ	✓			3BIN	✓			3D8C	✓			3EYD	✓		
2VLK	✓			3BL2	✓			3D9O	✓			3EYF	✓		
2VLL	✓			3BO8	✓			3D9T	✓			3F02	✓		
2VLR	✓			3BP4	✓			3D9U	✓			3F2O	✓		
2VM6	✓			3BP7	✓			3DA9	✓			3F7D	✓		
2VNW	✓			3BQD	✓			3DAB	✓			3F9W	✓		
2VNY	✓			3BQO	✓			3DAC	✓			3F9Z	✓		
2VO0	✓			3BRH	✓			3DCG	✓		✓	3FDL	✓		
2VO3	✓			3BRL	✓			3DCT	✓			3FDO	✓		
2VO6	✓			3BU3	✓			3DD7	✓			3FIE	✓		
2VO7	✓			3BU5	✓			3DDA	✓			3FII	✓		
2VOI	✓			3BU8	✓			3DDB	✓			3FQT	✓		
2VPE	✓			3BUA	✓			3DIW	✓			3FQW	✓		
2VPG	✓			3BW9	✓			3DND	✓			3FT3	✓		
2VR3	✓			3BWA	✓			3DNE	✓			3FT4	✓		
2VWF	✓			3BXL	✓			3DOW	✓			3FUG	✓		
2VZD	✓			3BXN	✓			3DRF	✓			3FUR	✓		
2VZG	✓			3BYA	✓			3DRG	✓			3FWV	✓		
2VZI	✓			3BZF	✓			3DRH	✓			3FXV	✓		
2W0P	✓			3C27	✓			3DRI	✓			3FY2	✓		
2W0Z	✓			3C2G	✓			3DRJ	✓			3G03	✓		
2W10	✓			3C3O	✓			3DRK	✓			3G8I	✓		
2W73	✓	✓		3C3Q	✓			3DS0	✓			3G94	✓		
2W9R	✓			3C3R	✓			3DS1	✓			3G9E	✓		
2WA8	✓			3C4M	✓			3DS4	✓	✓		3GCH	✓		
2WAX	✓			3C5J	✓			3DVE	✓			3GCI	✓		
2WAY	✓			3C9N	✓			3DVK	✓			3GCM	✓		
2Z32	✓			3CAL	✓			3DVP	✓			3GCT	✓		
2Z34	✓	✓		3CBL	✓			3DVU	✓			3GIV	✓		
2Z3N	✓			3CC5	✓			3DX6	✓			3GJF	✓		
2Z5S	✓			3CD3	✓			3DX7	✓			3GME	✓		
2Z5T	✓			3CDW	✓			3DX8	✓			3GYT	✓		
2Z7X	✓			3CH8	✓			3DXC	✓			3GYU	✓		
2ZFX	✓			3CHW	✓			3DXD	✓			3GZ1	✓		
2ZGH	✓			3CPL	✓			3DXE	✓			3GZE	✓		
2ZGJ	✓			3CQU	✓			3E0M	✓			3H0A	✓		
2ZJD	✓			3CQW	✓			3E1R	✓	✓		3H0T	✓		
2ZL9	✓			3CS8	✓			3E2B	✓			3H5R	✓		
2ZLA	✓			3CV0	✓			3E7C	✓			3H9G	✓		
2ZLC	✓			3CVL	✓			3E94	✓			3H9J	✓		

(Continued—5 of 6) The PepX Test Set.

PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero	PDB Code	PepX	Dockground Homo	Dockground Hetero
3HBV	✓			4CHA	✓			5CHA	✓			8GCH	✓		
3HDA	✓			4GCH	✓			6CHA	✓						
3SRN	✓			4SRN	✓			6GCH	✓						
3VGC	✓			4VGC	✓			7GCH	✓						

(Continued—6 of 6) The PepX Test Set.

PDB Code	ZDock Bound	ZDock Unbound	Dockground Homo	Dockground Hetero	PDB Code	ZDock Bound	ZDock Unbound	Dockground Homo	Dockground Hetero	PDB Code	ZDock Bound	ZDock Unbound	Dockground Homo	Dockground Hetero
1A2K	✓	✓			1IJK		✓			1XU1	✓			
1ACB	✓	✓		✓	1IRA	✓	✓			1Y64	✓	✓		✓
1AHW		✓			1J2J	✓			✓	1YVB		✓		
1AK4	✓	✓			1JIW	✓	✓		✓	1Z0K	✓	✓		
1ATN	✓	✓			1JK9	✓	✓			1Z5Y		✓		
1AZS		✓			1JPS	✓				1ZHH	✓	✓		
1B6C	✓	✓			1JTG	✓	✓			1ZHI	✓	✓		
1BJ1	✓	✓			1JWH	✓	✓			1ZLI	✓			
1BKD	✓	✓			1JZD	✓	✓			1ZM4	✓	✓		
1BUH	✓	✓		✓	1K4C	✓	✓			2A5T	✓	✓		
1BVK	✓	✓			1K5D		✓			2A9K	✓	✓		
1BVN	✓	✓		✓	1K74	✓	✓			2ABZ	✓	✓		
1CGI		✓			1KAC	✓	✓			2AJF	✓	✓		✓
1CLV	✓	✓		✓	1KKL		✓			2B42	✓	✓		
1DE4		✓			1KLU	✓	✓			2B4J	✓	✓		
1DFJ	✓				1KTZ	✓	✓			2BTF	✓	✓		
1DQJ	✓	✓			1KXP	✓	✓			2C0L	✓	✓		✓
1E4K	✓	✓			1KXQ	✓	✓		✓	2CFH	✓			
1E6E	✓	✓			1LFD	✓	✓			2FJU	✓			✓
1E6J	✓	✓			1M10	✓	✓			2G77	✓	✓		
1E96	✓				1MAH	✓	✓			2HLE		✓		
1EER	✓	✓			1ML0	✓	✓			2HMI	✓	✓		
1EFN	✓	✓			1MLC	✓	✓			2HQS	✓	✓		
1EWY	✓	✓			1MQ8	✓	✓			2HRK	✓	✓		✓
1F34	✓	✓			1NW9	✓			✓	2I25	✓	✓		✓
1F51	✓	✓			1OC0	✓	✓		✓	2I9B	✓	✓		✓
1F6M	✓	✓			1OFU	✓	✓			2IDO		✓		
1FC2	✓				1OYV	✓	✓			2J0T	✓	✓		
1FCC	✓	✓			1PVH	✓	✓		✓	2J7P	✓	✓		
1FFW	✓	✓			1PXV	✓	✓		✓	2NZ8	✓			✓
1FQ1	✓	✓			1QA9	✓	✓		✓	2O3B	✓	✓		
1FQJ	✓				1QFW	✓	✓			2O8V	✓	✓		
1FSK	✓	✓			1R0R	✓	✓		✓	2OOB	✓	✓		✓
1GCQ	✓	✓			1R6Q	✓	✓			2OT3		✓		
1GHQ	✓	✓			1R8S	✓	✓			2OUL	✓	✓		
1GL1	✓	✓			1RLB	✓	✓			2OZA	✓	✓		✓
1GLA	✓	✓			1RV6	✓	✓			2PCC	✓	✓		
1GP2	✓	✓			1S1Q	✓	✓			2SIC	✓	✓		
1GPW	✓	✓			1SBB	✓				2SNI	✓	✓		
1GRN	✓	✓			1SYX	✓	✓		✓	2UUY	✓			✓
1GXD	✓	✓			1T6B	✓	✓		✓	2VDB	✓	✓		
1H1V	✓	✓			1TMQ	✓	✓		✓	2VIS	✓			
1HCF	✓				1UDI	✓	✓			2Z0E		✓		
1HE1	✓				1US7	✓	✓		✓	3BP8	✓	✓		
1HE8	✓				1VFB	✓	✓			3CPH	✓	✓		
1I2M	✓	✓		✓	1WDW	✓	✓			3D5S	✓	✓		
1I4D	✓				1WEJ	✓	✓			4CPA	✓	✓		✓
1I9R		✓			1WQ1	✓	✓			7CEI	✓	✓		
1IB1	✓				1XD3	✓	✓			BOYV	✓	✓		
1IBR	✓	✓		✓	1XQS	✓	✓							

Table A.5: The ZDock Benchmark 4.0 Test Set.

APPENDIX B: PROTEIN DESCRIPTORS

Type	Name	Vertex	Triplet	Simplex	Tessellation	Equation
Geometric	Volume			✓	✓	$V = \frac{1}{3}A_0 h$
	Surface Area		✓	✓	✓	$T = \sqrt{s(s-a)(s-b)(s-c)}, s = \frac{a+b+c}{2}$
	Edge Length		✓	✓	✓	$d_{xy} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
	Optimality[1]		✓	✓		$o = \sum_{i>j} \frac{(l_i - l_j)^2}{15\bar{l}^2}$
Topological	Graph Distance					$g_{xy} = \min(\lvert e_{x \to y} \rvert)$
	Degree		✓	✓	✓	$d_i = \lvert e_i \rvert$
	Randic				✓	$R = \frac{1}{(d_i d_j)^{\frac{1}{2}}}$
	Wiener				✓	$W = \frac{1}{2}\sum_{x=1}^{n}\sum_{y=1}^{n} g_{xy}$
	Convexity	✓				$cvx = 1 - \frac{6d_i(non-surface)}{5d_i}$
	Estrada				✓	$EE = \sum_{j=1}^{n} e^{\lambda_j}$
	Balaban				✓	$J = \frac{m}{\gamma+1}\sum_{i=1}^{n}\sum_{j=1}^{n}(D_i D_j)^{-1/2}$
						$\gamma = m - n + 1$
						$D_i = \sum g_{iy}$
Inherent	N Vertex		✓	✓	✓	
	Chirality			✓		–
	onSurface			✓	✓	
	atInterface		✓	✓	✓	
	inHelix	✓				
	inSheet	✓				
	Residue	✓	✓	✓		

Table B.1: A list of the calculated protein descriptors.

[1] Evaluates how similar a given triangle or tetrahedra is to an optimal, or equilateral, shape. Also called tetrahedrality.

APPENDIX C: HARDWARE AND SOFTWARE

Hardware

A custom desktop PC was used throughout the course of this project for both the development and testing of the software described in this paper. The technical specifications of the computer are provided in Table C.1.

CPU	Intel core i7-950 3.06 GHz LGA 1366 130W Quad-Core
Motherboard	ASUS P6T Deluxe V2 ATX
RAM	6 GB DDR3 1600 (PC3 12800)
HDD	2x 500 GB 3.0 Gb/s
Graphics	nVidia GeForce 9500 GT 01G-P3-N959-TR
Power	650W

Table C.1: The hardware specifications for the computer used throughout the project.

Software

All software described in chapters above was developed by Stephen J. Bush, with the exception of the Mersenne Twister algorithm [123]. The code was written in C++ and generates three binaries: `dtess`, `cracle`, and `popp`. The inputs used by each program are listed in Table C.2. Each program outputs a series of space delimited files containing the data and a Python script for visualization in PyMol (Table C.3). The options used by each of the programs are given in Table C.2. The scripts used to parse and analyze the data were written in Perl and Matlab.

All programming was done on the computer mentioned above running Ubuntu 10.10 until a slight disagreement occured between program and programmer, at which point the system was upgraded to Linux Mint 15 Cinnamon.

required	
-p <file>	`-p ./1A1M.pdb` A PDB file name. *Either* `-p` *or* `-d` *is required.*
-d <dir>	`-d /Dropbox/pdbs/` A directory containing PDB files. *Either* `-p` *or* `-d` *is required.*
recommended	
`-b`	Batch option, for use only with `-d`. Tells the program that each *.pdb file in the directory should be handled individually.
`-c [chains]`	`-c AC` Tells program to ignore all chains in the PDB except those listed.
`-i [chains][chains]`	`-i AB C` Tells program to expect an interface. If no chains are given, looks for a file `00_chain_list.txt`* in the directory with the PDB file; otherwise, assume the interface is between the first two chains given in the PDB file.
`-peptide`	Flag to tell the program that there are peptide chains in the PDB file.*
`-v`	Flag to tell the program to create files used for visualization.
optional	
`-delim <punctuation>`	`-delim ,` Specify the delimiter to use in the descriptor files.
`-quick`	Flag to tell the program to only run new PDB files. Checks for *.smx.txt, *.tess.txt, *.vtx.txt, *.hsi.txt, and *.bs.txt files.
`-trim <%f>`	`-trim 11.5` Specify the threshold used to remove Delaunay edges (in Å).
popp Only	
`-f <%f>`	`-f 4.0` Sets the value of the `fit` parameter
`-r <%f>`	`-r 0.5` Sets the value of the `resolution` parameter
`-t <%f>`	`-t 6.0` Sets the value of the `thickness` parameter

Table C.2: Command line options for `dtess`, `cracle`, and `pop`.

* The `00_chain_list.txt` file should have the format `<PDB name>_<chains>_<chains>_<peptide chains>`, where the `peptide chains` argument is a list of the peptide chains that should be considered but not analyzed, e.g., `1A1M A C` for a file with two chains, one of which is a peptide (note the triple space between the A and the C).

dtess	The executable used to calculate the Delaunay tessellation of a protein or complex.
⋆.aa.txt	A count of the amino acid distribution for the protein.
⋆.aa.i.text	The amino acid distribution for only interfacial residues.
⋆.aa.s.text	The amino acid distribution for only surface residues.
⋆.ecm.text	A 20×20 matrix that lists the number of edges between each amino acid type.
⋆.ecm.i.text	The contact matrix using only edges between interfacial amino acids.
⋆.ecm.p.text	The contact matrix using only edges between sequentially adjacent amino acids.
⋆.ecm.s.text	The contact matrix using only edges between amino acids on the protein surface.
⋆.edge.text	A list of the physical distances between each residue vertex, given by the amino acid type, e.g., `L T 5.906`.
⋆.gdm.txt	A $n_v \times n_v$ matrix of the graph distances between each vertex residue pair.
⋆.hex.text	A list of binary strings for each vertex denoting edges between other residue vertices—can be used to recreate the Delaunay tessellation.
⋆.pdm.txt	A $n_v \times n_v$ matrix of the physical distances between each vertex residue pair.
⋆.py	A Python script for visualization in PyMol (Windows and Linux compatible).
⋆.smx.txt	A space-delimited file containing data pertaining to each Delaunay simplex from the tessellation, e.g., volume.
⋆.tess.txt	A space-delimited file containing data pertaining to the Delaunay tessellation, e.g., number of simplices.
⋆.trp.txt	A space-delimited file containing data pertaining to each surface triplet, e.g., surface area .
⋆.vtx.txt	A space-delimited file containing data pertaining to each residue vertex, e.g., coordinates. Also lists the absolute index of the vertex (numbered $0-n$ for all vertices), the relative index (numbered $0 - n$ within each chain), and the index as given by the PDB file.
head.smx.txt	The header file for *.smx.txt. Lists names of each column.
head.tess.txt	The header file for *.tess.txt.
head.trp.txt	The header file for *.trp.txt.
head.vtx.txt	The header file for *.vtx.txt.
cracle†	The executable used to run CRACLe.
⋆.bs.txt	A list of the predicted binding sites, broken down by residue.
⋆.hsi.txt	A list of the predicted hot spots.
⋆.hsi.py	A Python script for PyMol visualization of the predicted binding sites.
⋆.snap.<S>.trp.txt	A list of the SNAPP scores for each triplet. A separate file is created for each SNAPP scoring function S.
⋆.snap.<S>.vtx.txt	A list of the SNAPP pairing potentials for each residue vertex. A separate file is created for each SNAPP scoring function S.
⋆.vtx.txt	A space-delimited file containing data pertaining to each residue vertex, e.g., coordinates.
head.bs.txt	The header file for *.bs.txt.
head.hsi.txt	The header file for *.hsi.txt.
head.vtx.txt	The header file for *.vtx.txt.

Table C.3: A list of the files generated by the software. The software creates a folder called `<pdb>_<chains>_catalog` to store the output files, where `pdb` is the name of the PDB file, e.g., 1AWQ.pdb, and `chains` is the list of chains used in the calculation, e.g., `A` or `A_B`.

⋆ denotes the name as used to create the catalog directory, e.g., `1A1M_A_C`.

† CRACLe analyzes and outputs data for each protein separately so each file name only contains the chains used in the analysis, e.g. `1A1M_A`. Note that the PyMol file combines the visualization into a single file.

BIBLIOGRAPHY

[1] D. W. Ritchie, “Recent progress and future directions in protein-protein docking.,” *Current protein & peptide science*, vol. 9, pp. 1–15, Feb. 2008. 1, 2, 32, 48

[2] G. Fuentes, J. Oyarzabal, and A. M. Rojas, “Databases of protein-protein interactions and their use in drug discovery.,” *Current opinion in drug discovery & development*, vol. 12, pp. 358–66, May 2009. 1

[3] W. L. DeLano, “Unraveling hot spots in binding interfaces: progress and challenges.,” *Current opinion in structural biology*, vol. 12, pp. 14–20, Feb. 2002. 1, 2

[4] Q. Xu, E. W. Xiang, and Q. Yang, “Transferring Network Topological Knowledge for Predicting Protein-protein Interactions.,” *Proteomics*, pp. 1–23, July 2011. 1

[5] V. Lafont, M. Schaefer, R. Stote, D. Altschuh, and A. Dejaegere, “Proteinprotein recognition and interaction hot spots in an antigenantibody complex: Free energy decomposition identifies efficient amino acids,” *Proteins: Structure, Function, and Bioinformatics*, vol. 67, no. 2, pp. 418–434, 2007. 1

[6] P. Tuffery and P. Derreumaux, “Flexibility and binding affinity in protein-ligand, protein-protein and multi-component protein interactions: limitations of current computational approaches.,” *Journal of the Royal Society, Interface / the Royal Society*, Oct. 2011. 1, 5

[7] I. Moreira, P. Fernandes, and M. Ramos, “Proteinprotein docking dealing with the unknown,” *Journal of Computational Chemistry*, vol. 31, no. 2, pp. 317–342, 2010. 1, 2

[8] C. Dominguez, R. Boelens, and A. M. J. J. Bonvin, “HADDOCK: a protein-protein docking approach based on biochemical or biophysical information.,” *Journal of the American Chemical Society*, vol. 125, pp. 1731–7, Feb. 2003. 1

[9] S. de Vries, A. van Dijk, M. Krzeminski, M. van Dijk, A. Thureau, V. Hsu, T. Wassenaar, and A. Bonvin, “HADDOCK versus HADDOCK: new features and performance of HADDOCK2. 0 on the CAPRI targets,” *Proteins: structure, function, and bioinformatics*, vol. 69, no. 4, pp. 726–733, 2007. 1

[10] C. Wang, O. Schueler-Furman, I. Andre, N. London, S. Fleishman, P. Bradley, B. Qian, and D. Baker, “RosettaDock in CAPRI rounds 612,” *Proteins: Structure, Function, and Bioinformatics*, vol. 69, no. 4, pp. 758–763, 2007. 1, 2

[11] S. Lyskov and J. J. Gray, “The RosettaDock server for local protein-protein docking.,” *Nucleic acids research*, vol. 36, pp. W233–8, July 2008. 1

[12] J. Janin, “The targets of CAPRI rounds 3-5.,” *Proteins*, vol. 60, pp. 170–5, Aug. 2005. 1, 2

[13] J. Janin, “The targets of CAPRI rounds 612,” *Proteins: Structure, Function, and Bioinformatics*, vol. 69, no. 4, pp. 699–703, 2007. 1, 2

[14] J. Janin, “The targets of CAPRI Rounds 13-19.,” *Proteins*, vol. 78, pp. 3067–72, Nov. 2010. 1, 2

[15] T. Geppert, E. Proschak, and G. Schneider, “Protein-protein docking by shape-complementarity and property matching,” *Journal of Computational Chemistry*, vol. 31, no. 9, pp. 1919–1928, 2010. 2

[16] J. Fernandez-Recio, M. Totrov, C. Skorodumov, and R. Abagyan, “Optimal docking area: a new method for predicting protein-protein interaction sites.,” *Proteins*, vol. 58, pp. 134–43, Jan. 2005. 2

[17] X. Li, O. Keskin, B. Ma, R. Nussinov, and J. Liang, “Protein-protein interactions: hot spots and structurally conserved residues often locate in complemented pockets that pre-organized in the unbound states: implications for docking.,” *Journal of molecular biology*, vol. 344, pp. 781–95, Nov. 2004. 2

[18] E. Guney, N. Tuncbag, O. Keskin, and A. Gursoy, “HotSprint: database of computational hot spots in protein interfaces.,” *Nucleic acids research*, vol. 36, pp. D662–6, Jan. 2008. 2, 4

[19] O. Keskin, B. Ma, and R. Nussinov, “Hot regions in protein–protein interactions: the organization and contribution of structurally conserved hot spot residues.,” *Journal of molecular biology*, vol. 345, pp. 1281–94, Feb. 2005. 2

[20] M. R. Landon, R. L. Lieberman, Q. Q. Hoang, S. Ju, J. M. M. Caaveiro, S. D. Orwig, D. Kozakov, R. Brenke, G.-Y. Chuang, D. Beglov, S. Vajda, G. a. Petsko, and D. Ringe, “Detection of ligand binding hot spots on protein surfaces via fragment-based methods: application to DJ-1 and glucocerebrosidase.,” *Journal of computer-aided molecular design*, pp. 491–500, June 2009. 2

[21] J. J. Gray, “High-resolution protein-protein docking.,” *Current opinion in structural biology*, vol. 16, pp. 183–93, Apr. 2006. 2

[22] E. Noy and A. Goldblum, “Flexible protein-protein docking based on Best-First search algorithm,” *Journal of Computational Chemistry*, vol. 31, no. 9, pp. 1929–1943, 2010. 2

[23] J. Mintseris, K. Wiehe, B. Pierce, R. Anderson, R. Chen, J. Janin, and Z. Weng, “Protein-Protein Docking Benchmark 2.0: an update.,” *Proteins*, vol. 60, pp. 214–6, Aug. 2005. 2

[24] S. Vajda and D. Kozakov, “Convergence and combination of methods in protein-protein docking,” *Current opinion in structural biology*, vol. 19, pp. 164–70, Apr. 2009. 2

[25] T. Clackson and J. A. Wells, "A hot spot of binding energy in a hormone-receptor interface," *Science*, vol. 267, no. 5196, pp. 383–386, 1995. 2

[26] S. Grosdidier and J. Fernández-Recio, "Protein-protein docking and hot-spot prediction for drug discovery.," *Current pharmaceutical design*, vol. 18, pp. 4607–18, Jan. 2012. 2, 5

[27] H. a. Gabb, R. M. Jackson, and M. J. Sternberg, "Modelling protein docking using shape complementarity, electrostatics and biochemical information.," *Journal of molecular biology*, vol. 272, pp. 106–20, Sept. 1997. 4

[28] S. A. Assi, T. Tanaka, T. H. Rabbitts, and N. Fernandez-Fuentes, "PCRPi: Presaging Critical Residues in Protein interfaces, a new computational tool to chart hot spots in protein interfaces.," *Nucleic acids research*, vol. 38, p. e86, Apr. 2010. 4

[29] Q. Nguyen, R. Fablet, and D. Pastor, "Protein interaction hotspot identification using sequence-based frequency-derived features," *IEEE Transactions on Biomedical Engineering*, vol. PP, no. 99, pp. 1–10, 2011. 4

[30] Q. C. Zhang, L. Deng, M. Fisher, J. Guan, B. Honig, and D. Petrey, "PredUs: a web server for predicting protein interfaces using structural neighbors.," *Nucleic acids research*, vol. 39, pp. W283–7, July 2011. 4, 5, 39, 48, 56

[31] B. a. Shoemaker, D. Zhang, M. Tyagi, R. R. Thangudu, J. H. Fong, A. Marchler-Bauer, S. H. Bryant, T. Madej, and A. R. Panchenko, "IBIS (Inferred Biomolecular Interaction Server) reports, predicts and integrates multiple types of conserved interactions for proteins.," *Nucleic acids research*, vol. 40, pp. D834–40, Jan. 2012. 4

[32] N. Tuncbag, G. Kar, O. Keskin, A. Gursoy, and R. Nussinov, "A survey of available tools and web servers for analysis of protein-protein interactions and interfaces.," *Briefings in bioinformatics*, vol. 10, pp. 217–32, May 2009. 5

[33] R. K. Singh, A. Tropsha, and I. I. Vaisman, "Delaunay tessellation of proteins: four body nearest-neighbor propensities of amino acid residues.," *Journal of computational biology : a journal of computational molecular cell biology*, vol. 3, pp. 213–21, Jan. 1996. 6, 7, 9, 18

[34] Delaunay.B, "Sur la sphere vide. A la memoire de Georges Voronoi.," *Izv Akad Nauk SSSR, Otdelenie Matematicheskih i Estestvennyh Nauk*, vol. 7, pp. 793–800, 1934. 6

[35] F. M. Richards, "Areas, volumes, packing and protein structure.," *Annual review of biophysics and bioengineering*, vol. 6, pp. 151–76, Jan. 1977. 6

[36] H. Edelsbrunner, "Weighted alpha shapes," tech. rep., Comp. Sci. Dept., Univ. Illinois, Urbana, Illinois, 1992. 6, 41

[37] W. Zheng, S. Cho, I. I. Vaisman, and A. Tropsha, "A new approach to protein fold recognition based on Delaunay tessellation of protein structure," in *Pac Symp Biocomput*, pp. 486–497, 1997. 6

[38] C. W. Carter, B. LeFebvre, S. Cammer, A. Tropsha, and M. Edgell, "Four-body potentials reveal protein-specific correlations to stability changes caused by hydrophobic core mutations.," *Journal of Molecular Biology*, vol. 311, pp. 625–38, Aug. 2001. 6, 10

[39] H. H. Gan, A. Tropsha, and T. Schlick, "Lattice protein folding with two and four-body statistical potentials.," *Proteins*, vol. 43, pp. 161–74, May 2001. 6, 11

[40] S. Cammer, C. W. Carter, and A. Tropsha, "Identification of sequence-specific tertiary packing motifs in protein structures using Delaunay tessellation," *Computational Methods for*, 2002. 6, 8, 11

[41] B. Krishnamoorthy and A. Tropsha, "Development of a four-body statistical pseudo-potential to discriminate native from non-native protein conformations," *Bioinformatics (Oxford, England)*, vol. 19, pp. 1540–8, Aug. 2003. 6, 7, 12

[42] D. Bandyopadhyay, J. Huan, J. Prins, J. Snoeyink, W. Wang, and A. Tropsha, "Identification of family-specific residue packing motifs and their use for structure-based protein function prediction: I. Method development.," *Journal of computer-aided molecular design*, vol. 23, pp. 773–84, Nov. 2009. 6

[43] A. Tropsha, C. W. Carter, S. Cammer, and I. I. Vaisman, "Simplicial neighborhood analysis of protein packing (SNAPP): a computational geometry approach to studying proteins.," *Methods in enzymology*, vol. 374, pp. 509–44, Jan. 2003. 6, 8, 11

[44] K. Dill and J. MacCallum, "The protein-folding problem, 50 years on," *Science*, vol. 338, pp. 1042–6, Nov. 2012. 7

[45] D. Bandyopadhyay and J. Snoeyink, "Almost-Delaunay simplices: Nearest neighbor relations for imprecise points," *of the fifteenth annual ACM-SIAM*, pp. 410–419, 2004. 7, 13

[46] A. Bowyer, "Computing dirichlet tessellations," *The Computer Journal*, vol. 24, no. 2, p. 162, 1981. 7

[47] D. Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes," *The computer journal*, vol. 24, no. 2, p. 167, 1981. 7

[48] S. Cammer and C. W. Carter, "Six Rossmannoid folds, including the Class I aminoacyl-tRNA synthetases, share a partial core with the anti-codon-binding domain of a Class II aminoacyl-tRNA synthetase.," *Bioinformatics (Oxford, England)*, vol. 26, pp. 709–14, Mar. 2010. 11

[49] C. Deutsch and B. Krishnamoorthy, "Four-body scoring function for mutagenesis.," *Bioinformatics (Oxford, England)*, vol. 23, pp. 3009–15, Nov. 2007. 13

[50] D. C. Richardson and J. S. Richardson, "Top 500 Database ::: Kinemage Website," 2000. 14, 20

[51] G. Wang and R. L. Dunbrack, "PISCES: a protein sequence culling server.," *Bioinformatics (Oxford, England)*, vol. 19, pp. 1589–91, Aug. 2003. 14, 20

[52] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank.," *Nucleic acids research*, vol. 28, pp. 235–42, Jan. 2000. 14, 20

[53] I. M. KLOTZ, "The application of the law of mass action to binding by proteins; interactions with calcium.," *Archives of biochemistry*, vol. 9, pp. 109–17, Jan. 1946. 17

[54] I. Gutman and M. Randic, "Algebraic characterization of skeletal branching," *Chemical Physics Letters*, vol. 47, no. 1, pp. 15–19, 1977. 18

[55] L. Hall and L. Kier, "The E-state as the basis for molecular structure space definition and structure similarity," *Journal of chemical information and computer sciences*, vol. 40, pp. 784–91, May 2000. 18

[56] J. Huan, D. Bandyopadhyay, W. Wang, J. Snoeyink, J. Prins, and A. Tropsha, "Comparing graph representations of protein structure for mining family-specific residue-based packing motifs," *Journal of Computational Biology*, vol. 12, no. 6, pp. 657–671, 2005. 18

[57] T. Taylor and I. I. Vaisman, "PROPERTIES OF AMINO ACID CONTACT NETWORKS OF DELAUNAY TESSELLATED PROTEIN STRUCTURES," *binf.gmu.edu*, pp. 1–10, 2000. 18

[58] S. Vishveshwara, K. Brinda, and N. Kannan, "Protein structure: insights from graph theory," *Journal of Theoretical and Computational Chemistry*, vol. 1, no. 1, pp. 187–212, 2002. 18

[59] A. Canutescu, A. Shelenkov, and R. Dunbrack Jr, "A graph-theory algorithm for rapid protein side-chain prediction," *Protein science*, vol. 12, no. 9, pp. 2001–2014, 2003. 18

[60] D. J. Jacobs, a. J. Rader, L. a. Kuhn, and M. F. Thorpe, "Protein flexibility predictions using graph theory," *Proteins*, vol. 44, pp. 150–65, Aug. 2001. 18

[61] C. Langmead and H. Kamisetty, "Detecting Protein-Protein Interaction Decoys using Fast Free Energy Calculations," *Computer Science Department*, p. 1055, 2007. 18, 54

[62] A. M. J. J. Bonvin, "Flexible protein-protein docking.," *Current opinion in structural biology*, vol. 16, pp. 194–200, Apr. 2006. 18

[63] D. Schneidman-Duhovny, Y. Inbar, R. Nussinov, and H. J. Wolfson, "Geometry-based flexible and symmetric protein docking.," *Proteins*, vol. 60, pp. 224–31, Aug. 2005. 18

[64] S. Klavzar and I. Gutman, "Wiener number of vertex-weighted graphs and a chemical application," *Discrete Applied Mathematics*, vol. 80, pp. 73–81, Dec. 1997. 18

[65] M. Randic, "On characterization of molecular branching," *Journal of American Chemical Society*, vol. 97, no. 23, 1975. 19

[66] E. Estrada, “Characterization of 3D molecular structure,” *Chemical Physics Letters*, vol. 319, pp. 713–718, Mar. 2000. 19

[67] 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*, vol. 53, pp. 76–87, Oct. 2003. 21, 23

[68] D. Baker, “Protein Folding Decoys – Backbone.” 21

[69] Y. A. Arnautova, Y. N. Vorobjev, J. A. Vila, and H. A. Scheraga, “Identifying native-like protein structures with scoring functions based on all-atom ECEPP force fields, implicit solvent models and structure relaxation.,” *Proteins*, vol. 77, pp. 38–51, Oct. 2009. 24

[70] S. Liu, Y. Gao, and I. A. Vakser, “DOCKGROUND protein-protein docking decoy set.,” *Bioinformatics (Oxford, England)*, vol. 24, pp. 2634–5, Nov. 2008. 30

[71] B. Huang and M. Schroeder, “LIGSITEcsc: predicting ligand binding sites using the Connolly surface and degree of conservation.,” *BMC structural biology*, vol. 6, p. 19, Jan. 2006. 32

[72] J. a. Capra, R. a. Laskowski, J. M. Thornton, M. Singh, and T. a. Funkhouser, “Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure.,” *PLoS computational biology*, vol. 5, p. e1000585, Dec. 2009. 32

[73] T. Geppert, B. Hoy, S. Wessler, and G. Schneider, “Context-Based Identification of Protein-Protein Interfaces and Hot-Spot Residues,” *Chemistry & Biology*, vol. 18, pp. 344–353, Mar. 2011. 32

[74] A. Tropsha, “Best Practices for QSAR Model Development, Validation, and Exploitation,” *Molecular Informatics*, pp. n/a–n/a, July 2010. 35

[75] A. Tropsha and A. Golbraikh, “Predictive QSAR modeling workflow, model applicability domains, and virtual screening.,” *Current pharmaceutical design*, vol. 13, pp. 3494–504, Jan. 2007. 35

[76] Y. Ofran and B. Rost, “Analysing Six Types of ProteinProtein Interfaces,” *Journal of Molecular Biology*, vol. 325, pp. 377–387, Jan. 2003. 35, 51

[77] D. Douguet, H.-C. Chen, A. Tovchigrechko, and I. A. Vakser, “DOCKGROUND resource for studying protein-protein interfaces.,” *Bioinformatics (Oxford, England)*, vol. 22, pp. 2612–8, Nov. 2006. 37

[78] P. Vanhee, J. Reumers, F. Stricher, L. Baeten, L. Serrano, J. Schymkowitz, and F. Rousseau, “PepX: a structural database of non-redundant protein-peptide complexes.,” *Nucleic acids research*, vol. 38, pp. D545–51, Jan. 2010. 39, 46

[79] H. Hwang, T. Vreven, J. Janin, and Z. Weng, “Protein-protein docking benchmark version 4.0.,” *Proteins*, vol. 78, pp. 3111–4, Nov. 2010. 39, 47, 60

[80] R. Sinha, P. J. Kundrotas, and I. a. Vakser, "Docking by structural similarity at protein-protein interfaces.," *Proteins*, vol. 78, pp. 3235–41, Nov. 2010. 40

[81] M. Cohen, V. Potapov, and G. Schreiber, "Four distances between pairs of amino acids provide a precise description of their interaction.," *PLoS computational biology*, vol. 5, p. e1000470, Aug. 2009. 40

[82] S. Chaudhury, M. Berrondo, B. D. Weitzner, P. Muthu, H. Bergman, and J. J. Gray, "Benchmarking and Analysis of Protein Docking Performance in Rosetta v3.2.," *PloS one*, vol. 6, p. e22477, Jan. 2011. 48, 54

[83] S. J. de Vries and A. M. J. J. Bonvin, "CPORT: a consensus interface predictor and its performance in prediction-driven docking with HADDOCK.," *PloS one*, vol. 6, p. e17695, Jan. 2011. 48, 60, 61, 62

[84] M. R. Arkin and J. A. Wells, "Small-molecule inhibitors of protein-protein interactions: progressing towards the dream.," *Nature reviews. Drug discovery*, vol. 3, pp. 301–17, Apr. 2004. 48, 81

[85] G. a. Weiss, C. K. Watanabe, a. Zhong, a. Goddard, and S. S. Sidhu, "Rapid mapping of protein functional epitopes by combinatorial alanine scanning.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, pp. 8950–4, Aug. 2000. 48

[86] D. a. Erlanson, a. C. Braisted, D. R. Raphael, M. Randal, R. M. Stroud, E. M. Gordon, and J. a. Wells, "Site-directed ligand discovery.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, pp. 9367–72, Aug. 2000. 48

[87] T. Kortemme and D. Baker, "A simple physical model for binding energy hot spots in protein-protein complexes.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 14116–21, Oct. 2002. 48

[88] a. a. Bogan and K. S. Thorn, "Anatomy of hot spots in protein interfaces.," *Journal of molecular biology*, vol. 280, pp. 1–9, July 1998. 48, 59

[89] B. Nisius, F. Sha, and H. Gohlke, "Structure-based computational analysis of protein binding sites for function and druggability prediction.," *Journal of biotechnology*, pp. 1–12, Dec. 2011. 54

[90] N. London, D. Movshovitz-Attias, and O. Schueler-Furman, "The structural basis of peptide-protein binding strategies.," *Structure (London, England : 1993)*, vol. 18, pp. 188–99, Feb. 2010. 54, 66

[91] G. Kar, A. Gursoy, and O. Keskin, "Human cancer protein-protein interaction network: a structural perspective.," *PLoS computational biology*, vol. 5, p. e1000601, Dec. 2009. 54

[92] K. J. Smith, S. W. Reid, K. Harlos, a. J. McMichael, D. I. Stuart, J. I. Bell, and E. Y. Jones, "Bound water structure and polymorphic amino acids act together to allow the binding of different peptides to MHC class I HLA-B53.," *Immunity*, vol. 4, pp. 215–28, Mar. 1996. 54

[93] H.-X. Zhou and S. Qin, "Interaction-site prediction for protein complexes: a critical assessment.," *Bioinformatics (Oxford, England)*, vol. 23, pp. 2203–9, Sept. 2007. 60, 61

[94] H. Hwang, B. Pierce, J. Mintseris, J. Janin, and Z. Weng, "Protein-protein docking benchmark version 3.0.," *Proteins: Structure, Function, and Bioinformatics*, vol. 73, pp. 705–709, Nov. 2008. 60

[95] S. de Vries and A. Bonvin, "How proteins get in touch: interface prediction in the study of biomolecular complexes," *Current protein and peptide science*, pp. 394–406, 2008. 61

[96] I. Kufareva and L. Budagyan, "PIER: protein interface recognition for structural proteomics," *Proteins: Structure, . . .* , vol. 417, no. December 2005, pp. 400–417, 2007. 62

[97] N. J. Burgoyne and R. M. Jackson, "Predicting protein interaction sites: binding hotspots in protein-protein and protein-ligand interfaces.," *Bioinformatics (Oxford, England)*, vol. 22, pp. 1335–42, June 2006. 62

[98] K. J. Schmitz, F. Grabellus, R. Callies, F. Otterbach, J. Wohlschlaeger, B. Levkau, R. Kimmig, K. W. Schmid, and H. A. Baba, "Research article High expression of focal adhesion kinase (p125 FAK) in node-negative breast cancer is related to overexpression of HER-2 / neu and activated Akt kinase but does not predict outcome," *Breast Cancer Research*, vol. 7, no. 2, 2005. 63

[99] D. Lietha, X. Cai, D. Ceccarelli, and Y. Li, "Structural basis for the autoinhibition of focal adhesion kinase," *Cell*, vol. 129, no. 6, pp. 1177–1187, 2007. 63

[100] S.-Y. Chen and H.-C. Chen, "Direct interaction of focal adhesion kinase (FAK) with Met is required for FAK to promote hepatocyte growth factor-induced cell invasion.," *Molecular and cellular biology*, vol. 26, pp. 5155–67, July 2006. 63

[101] S. Liang, D. Zheng, C. Zhang, and M. Zacharias, "Prediction of antigenic epitopes on protein surfaces by consensus scoring.," *BMC bioinformatics*, vol. 10, p. 302, Jan. 2009. 64, 65

[102] K. Pan, J. Long, H. Sun, G. J. Tobin, P. L. Nara, and M. W. Deem, "Selective pressure to increase charge in immunodominant epitopes of the H3 hemagglutinin influenza protein.," *Journal of molecular evolution*, vol. 72, pp. 90–103, Jan. 2011. 64

[103] M. Blythe and D. Flower, "Benchmarking B cell epitope prediction: underperformance of existing methods," *Protein Science*, pp. 246–248, 2005. 65

[104] D. Fourches, E. Muratov, and A. Tropsha, "Trust, but verify: on the importance of chemical structure curation in cheminformatics and QSAR modeling research.," *Journal of chemical information and modeling*, vol. 50, pp. 1189–1204, July 2010. 66

[105] T. Walker, C. M. Grulke, D. Pozefsky, and A. Tropsha, "Chembench: a cheminformatics workbench.," *Bioinformatics (Oxford, England)*, vol. 26, pp. 3000–1, Dec. 2010. 67

[106] M. Nielsen and K. Museth, "Dynamic Tubular Grid: An efficient data structure and algorithms for high resolution level sets," *Journal of Scientific Computing*, vol. 26, no. 3, pp. 261–299, 2006. 72

[107] N. Metropolis, A. W. Rosenblush, M. N. Rosenbluth, and A. H. Teller, "Equation of state calculations by Fast Computing Machines," *The Journal of Chemical Physics*, vol. 21, pp. 1087–1092, Feb. 1953. 74

[108] V. B. Chen, W. B. Arendall, J. J. Headd, D. a. Keedy, R. M. Immormino, G. J. Kapral, L. W. Murray, J. S. Richardson, and D. C. Richardson, "MolProbity: all-atom structure validation for macromolecular crystallography.," *Acta crystallographica. Section D, Biological crystallography*, vol. 66, pp. 12–21, Jan. 2010. 76

[109] I. Davis, A. Leaver-Fay, and V. Chen, "MolProbity: all-atom contacts and structure validation for proteins and nucleic acids," *Nucleic acids …*, vol. 35, pp. W375–83, July 2007. 76

[110] T. Aita, K. Nishigaki, and Y. Husimi, "Toward the fast blind docking of a peptide to a target protein by using a four-body statistical pseudo-potential.," *Computational biology and chemistry*, vol. 34, pp. 53–62, Mar. 2010. 76

[111] F. F. Vajdos, S. Yoo, M. Houseweart, W. I. Sundquist, and C. P. Hill, "Crystal structure of cyclophilin A complexed with a binding site peptide from the HIV-1 capsid protein.," *Protein science : a publication of the Protein Society*, vol. 6, pp. 2297–307, Nov. 1997. 76

[112] O. Pornillos, S. L. Alam, D. R. Davis, and W. I. Sundquist, "Structure of the Tsg101 UEV domain in complex with the PTAP motif of the HIV-1 p6 protein.," *Nature structural biology*, vol. 9, pp. 812–7, Nov. 2002. 76

[113] V. J. Basus, G. Song, and E. Hawrot, "NMR solution structure of an alpha-bungarotoxin/nicotinic receptor peptide complex.," *Biochemistry*, vol. 32, pp. 12290–8, Nov. 1993. 76

[114] S. J. Campbell, N. D. Gold, R. M. Jackson, and D. R. Westhead, "Ligand binding: functional site location, similarity and docking," *Current Opinion in Structural Biology*, vol. 13, pp. 389–395, June 2003. 78

[115] F. Reisen, M. Weisel, J. M. Kriegl, and G. Schneider, "Self-organizing fuzzy graphs for structure-based comparison of protein pockets.," *Journal of proteome research*, vol. 9, pp. 6498–510, Dec. 2010. 78

[116] S. Pérot, O. Sperandio, M. A. Miteva, A.-C. Camproux, and B. O. Villoutreix, "Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery.," *Drug discovery today*, vol. 15, pp. 656–67, Aug. 2010. 78

[117] C. Hetényi and D. van der Spoel, "Toward prediction of functional protein pockets using blind docking and pocket search algorithms.," *Protein science : a publication of the Protein Society*, vol. 20, pp. 880–93, May 2011. 78

[118] S. Henrich, O. M. H. Salo-Ahen, B. Huang, F. F. Rippmann, G. Cruciani, and R. C. Wade, "Computational approaches to identifying and characterizing protein binding sites for ligand design.," *Journal of molecular recognition : JMR*, vol. 23, no. 2, pp. 209–19, 2010. 78

[119] R. G. Coleman and K. a. Sharp, "Protein pockets: inventory, shape, and comparison.," *Journal of chemical information and modeling*, vol. 50, pp. 589–603, Apr. 2010. 78

[120] J. C. Fuller, N. J. Burgoyne, and R. M. Jackson, "Predicting druggable binding sites at the protein-protein interface.," *Drug discovery today*, vol. 14, pp. 155–61, Feb. 2009. 78

[121] T. Li, K. Fan, J. Wang, and W. Wang, "Reduction of protein sequence complexity by residue grouping," *Protein Engineering Design and Selection*, vol. 16, pp. 323–330, May 2003. 85

[122] S. Liu and I. a. Vakser, "DECK: Distance and environment-dependent, coarse-grained, knowledge-based potentials for protein-protein docking.," *BMC bioinformatics*, vol. 12, p. 280, Jan. 2011. 87

[123] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, pp. 3–30, Jan. 1998. 111