DEVELOPMENT OF NOVEL METHODS FOR COMPUTATIONAL PROTEIN DESIGN AND PROTEIN-LIGAND DOCKING.

By

Sam DeLuca

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Chemical and Physical Biology

August, 2015

Nashville, Tennessee

Approved:

Terry Lybrand, Ph.D.

Brian Bachmann, Ph.D.

Heidi Hamm, Ph.D.

Stephen Fesik, Ph.D.

Jens Meiler, Ph.D.

ACKNOWLEDGEMENTS

Much of the work presented here was funded by a pre-doctoral fellowship in informatics from the PhRMA foundation. I would like to thank my advisor, Dr. Jens Meiler, for his guidance, and mentoring over the course of my PhD. Jens' commitment to pushing the boundaries of molecular modeling methodology has been an inspiration to me, and his insight and support were critical to the success of my work presented here. I would also like to thank the members of my thesis committee, Dr. Terry Lybrand, Dr. Brian Bachmann, Dr. Heidi Hamm, and Dr. Stephen Fesik for their excellent advice and guidance throughout my time as a graduate student.

Rosetta is a collaborative project, and this work would not have been possible without the help of a number of people in the Meiler lab and the RosettaCommons as a whole. Gordon Lemmon and Kristian Kaufmann were instrumental in helping me get started with my work, and I thank them for their advice, friendship and mentoring. In addition to coping with sitting next to me for 6 years, I would like to thank Steven Combs for being a great friend and colleague, and for helping me deal with science, graduate school, and life in general.

RosettaHTS could not exist in its present form without the ability to rapidly store information in a database. Developing the infrastructure necessary to accomplish this was a tremendous undertaking, and would have been impossible to complete without the collaboration of Matt O'Meara and Tim Jacobs at UNC North Carolina, and Chris Miles at the University of Washington.

Will Lowe, Mariusz Butkiewicz, and Jeff Mendenhall kindly took the time to teach me the fundamentals of Artificial Neural Networking, answer my endless questions, and brainstorm ideas about incorporating machine learning methods and ligand docking. Their help was instrumental to the work described in Chapter IV. Darwin Fu, Brittany Allison, Thomas Austin, Jordan Willis, and Amanda Duran have been vital to the development of

the work described in this thesis both through their direct contributions, and through their feedback as they used the software tools described here.

Graduate school is a long and arduous process, and I would not have been able to complete it without friends. In addition to everyone mentioned above, I'd like to acknowledge Thuy Nguyen, Liz Nguyen, Sten Heinze, Kate Mittendorf, Mert Karakas, Julia Koehler, Andrew Morin, and Ralf Mueller at Vanderbilt University.I would also like to acknowledge Vanderbilt, Andrew Leaver-Fay at UNC Chapel Hill, Doug Renfrew at New York University, Brian Weitzner, Jason Labonte, and Jared Adolf-Bryfogle at John Hopkins. I will be forever grateful for their friendship.

Throughout my time at Vanderbilt, Stephanie, my wife, friend and colleague, has been there with me. Her support and encouragement cannot be understated. Alice and Ed, my parents, and Marie, my sister, are the best family one could ask for. I would not be where I am right now if not for their endless support and encouragement.

My path to this point in my education has been a long one, and I credit several of my early teachers for helping set me along this road. Jannine, Keith and Ken Baker taught me the joy of computer programming, and Dr. David Form and Dr. Mary Jane Kurtz introducing me to computational biology many years ago.

I would also like to acknowledge my cat, Callisto, for her boundless enthusiasm in all things.

Finally, I would like to dedicate this thesis in memory of my grandfather, Eldon Boling. Eldon was a doctor, scientist, and inventor, and his gift for asking the next question has been an inspiration to me.

SUMMARY

Progress in computational drug discovery depends on our ability to process and exclude large numbers of candidate compounds. This thesis describes the development of new tools that improve the ability of Rosetta to design the surfaces of small globular proteins (Chapter II), and to more rapidly and efficiently screen libraries of small molecules for protein ligand docking (chapters III-IV). A set of appendices (chapters A-F) address the tools and techniques that are necessary for RosettaLigand to smoothly and efficiently perform High Throughput Screening (HTS) studies. Additionally, we discuss the development of several experimental energy functions, and discuss the current issues with these energy functions and ways to move forward.

Chapter I is an introductory chapter outlining the background and significance of the research described in the dissertation. Part of this chapter, specifically Section I.1, was partially based on text originally published as “Practically useful: what the Rosetta protein modeling suite can do for you” (Kaufmann et al., 2010), To which I was a contributing co-first author. The remainder of Chapter I is original to this dissertation.

Chapter II was originally published as “Design of Native-like Proteins through an Exposure-Dependent Environment Potential” (DeLuca et al., 2011). This chapter describes the implementation, optimization, and benchmarking of a novel energy term for protein design. This energy term is a Knowledge-Based Potential (KBP) which computes a score based on the propensity of an amino acid existing at a given degree of burial within a protein. A new metric for assessing the quality of designed proteins based on their Position Specific Scoring Matrix (PSSM) score was introduced. The weights of the RosettaDesign energy function were then optimized using Particle Swarm Optimization. The resulting optimized energy function incorporating the new KBP showed significantly improved protein designs using the metrics of protein sequence recovery, sequence composition bias, and the new PSSM recovery metric.

Chapter III is a draft of a manuscript submitted to PLoS ONE and currently review on which I will be the sole first author. This manuscript describes improvements in the ability of RosettaLigand to identify correct ligand binding poses, as well as improvements in the speed and efficiency of the simulation convergance. Specifically, the initial placement step of the RosettaLigand algorithm was rewritten to more efficiently sample the binding site space using a Metropolis Monte Carlo algorithm that simultaneously translates and rotates the ligand. Code used to translate and rotate the ligand within the binding site was carefully optimized to improve computational efficiency. The code modifications resulted in a 10-15% increase in the number of protein targets to which ligands can be successfully docked. The number of models needed to produce a high quality binding dropped from 1000 to 150, the compute time for the model dropped from 45-90s to 5-15s. With these improvements we can now use RosettaLigand for virtual High Throughput Screening (vHTS).

Chapter IV describes a vHTS protocol which uses predictions generated by the RosettaLigand protocol described in Chapter III as input into an Artificial Neural Network (ANN) trained to predict ligand activity and binding affinity. This chapter describes the design of a set of data for training the ANN model which is both diverse in protein and chemical space and also balanced in chemical space between active and inactive compounds. A set of Radial Distribution Function (RDF) based fingerprint descriptors representing the geometric and chemical properties of the protein-ligand interface are introduced. The goal is to produce a model which is capable of accurately classifying compounds based on activity. While the ANN models described in this chapter do not significantly improve upon the performance of RosettaLigand alone, the pattern recognition approach described here has potential. Potential future research in training set curation, network training methodology, and descriptor design is described.

Chapter V outlines the findings and future directions of the research presented in the prior chapters.

Appendix A is an appendix describing the design and usage of a software processing pipeline I developed to rapidly parameterize large numbers of ligands for input into RosettaLigand. This pipeline makes it possible to prepare hundreds of thousands of ligands in a matter of hours in a semiautomated fashion, and is a critical part of the work performed in III and IV.

Appendix B is an appendix describing the design of a software system for storing and retrieving protein structural information generated by Rosetta using a Structured Query Language (SQL) database. This system was developed in collaboration between myself and three other members of the RosettaCommons, and decreases the disk space requirements of RosettaLigand by approximately 99% in a vHTS protocol.

Appendix C is an appendix describing the development of novel scoring grids for use with the RosettaLigand initial placement algorithm. These scoring grids did not significantly improve upon the performance of RosettaLigand, and the development of effective KBP based scoring functions for initial placement remains an active area of research. Several scoring grids are described here: A set of two dimensional scoring grids designed to model shape complementarity and hydrogen bonding interactions, and a set of 3 dimensional grids designed to model favorable areas of ligand occupancy in the regions surrounding the 20 canonical amino acids. While these scoring grids did not result in improved performance beyond the results seen in Chapter III, the general techniques may be useful.

Appendix D is a protocol capture document describing the method by which the experiments in Chapter II can be reproduced. This document was originally published as a supplement to the manuscript (DeLuca et al., 2011).

Appendix E is a protocol capture document describing the method by which the experiments in Chapter III can be reproduced. It will be published along side the manuscript as supplemental information.

Appendix F is a protocol capture document describing the method by which the experiments in Chapter IV can be reproduced.

Appendix G describes a ligand docking study performed in 2012 in collaboration with the Fesik and Chazin labs. In this study, a set of small molecules vHTS hits from Quantitative Structure Activity Relationship (QSAR) studies of Kirsten Rat Sarcoma Virus (KRAS) and Replication Protein A 70 (RPA70) were docked into crystal structures to predict potential binding modes. In the course of this study, a predicted binding mode that differed substantially from the binding modes previously experimentally observed. This prediction is compared to a set of more recent X-Ray crystal structures obtained in 2014.

Appendix H is a discussion of the limitations of the Rosetta atom-typing system when applied to small drug-like molecules. This appendix includes a discussion of a brief, preliminary investigation performed in 2014 in which molecular orbitals computed with Density Functional Theory (DFT) were qualitatively compared to the orbital positions assigned using the method developed by Combs et al.(Combs, 2013)

TABLE OF CONTENTS

Page

Appendix

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

3DA	3D Autocorrelation	115
AMBER	Assisted Model Building with Energy Refinement	44
ANN	Artificial Neural Network	v
AUC	Area Under the Curve	96
BCL	BioChemical Library	71
BLAST	Basic Local Alignment Search Tool	25
CASP	Critical Assessment of protein Structure Prediction	14
CPU	Central Processing Unit	58
CSAR	Community Structure-Activity Resource	53
CSV	Comma Separated Value	184
DBN	Deep Belief Network	116
DFT	Density Functional Theory	vii
DUD	Directory of Useful Decoys	82
DUD-E	Directory of Useful Decoys Enhanced	83
FPR	False Positive Rate	96
GOLD	Genetic Optimization of Ligand Docking	11
GPCR	G-Protein Coupled Receptor	14
GPU	Graphical Processing Unit	16
hERG	human Ether-a-go-go-Related Gene	78
HOMO	Highest Occupied Molecular Orbital	xxxi
HSE	Half Sphere Exposure	23
HTS	High Throughput Screening	iv
JSON	JavaScript Object Notation	126
KBP	Knowledge-Based Potential	iv
KRAS	Kirsten Rat Sarcoma Virus	vii
LGA	Lamarckian Genetic Algorithm	44
MC	Monte Carlo	52
MCM	Monte Carlo Minimization	xvii
mGluR5	metabotropic Glutamate Receptor 5	78
MIN	Minimization	53
MD	Molecular Dynamics	226

MM	Molecular Mechanics	228
MOE	Molecular Operating Environment	210
NCR	Neighbor Count	21
NMR	Nuclear Magnetic Resonance	209
NR	Non-Redundant	26
NV	Neighbor Vector	x
OSR1	Odd-Skipped Related 1	78
ORM	Object-Relational Mapping	129
PBS	Portable Batch System	183
PDB	Protein DataBank	1
PPV	Positive Predictive Value	95
PSSM	Position Specific Scoring Matrix	iv
QSAR	Quantitative Structure Activity Relationship	vii
RDF	Radial Distribution Function	v
REU	Rosetta Energy Units	149
RMS	Root Mean Square	30
RMSD	Root Mean Square Deviation	xxix
RPA70	Replication Protein A 70	vii
ROC	Receiver Operating Characteristic	95
ROC-AUC	Receiver Operating Characteristic Area Under Curve	95
rSASA	relative Solvent Accessible Surface Area	23
SAR	Structure Activity Relationship	209
SASA	Solvent Accessible Surface Area	21
SCOP	Structural Classification of Proteins	24
SDF	Structure Data File	121
SiFT	Structural Interaction Fingerprint	116
SQL	Structured Query Language	vi
TPR	True Positive Rate	96
TPSA	Total Polarizable Surface Area	168
VDW	Van Der Waals	71
VSEPR	Valence Shell Electron Pair Repulsion	231
vHTS	virtual High Throughput Screening	v
WNCR	Weighted Neighbor Count	xix
XML	eXtensible Markup Language	51

CHAPTER I

Introduction

I.1 Sampling and scoring methods used by the Rosetta software suite

Rosetta is a software package for protein structure prediction and functional design. It has been applied to predict protein structures with and without the aid of sparse experimental data, perform protein-protein and protein-small molecule docking, design novel proteins, and redesign existing proteins for altered function. Rosetta allows for rapid tests of hypotheses in biomedical research which would be impossible or exorbitantly expensive to perform via traditional experimental methods. As a result, Rosetta methods have gained increasing importance in the interpretation of biological findings from genome projects, the engineering of therapeutics, probe molecules, and model systems in biomedical research.

While the Rosetta suite is capable of performing a wide range of modeling tasks, it uses a core set of sampling and scoring strategies to accomplish most of these. The majority of conformational sampling protocols in Rosetta use the Metropolis Monte Carlo algorithm to guide sampling. Gradient based minimization is often employed for last step refinement of initial models. Since each Rosetta protocol allows degrees of freedom specific for the task, Rosetta can perform a diverse set of protein modeling tasks (Wang et al., 2007).

I.1.1 Sampling strategies for backbone degrees of freedom

Rosetta separates large backbone conformational sampling from local backbone refinement. Large backbone conformational changes are modeled by exchanging the backbone conformations of 9 or 3 amino acid peptide fragments. Peptide conformations are collected from the Protein DataBank (PDB) for homologous stretches of sequence (Simons et al., 1997) which capture the structural bias of the local sequence (Bystroff et al., 1996). For local refinement of protein models, Rosetta utilizes Metropolis Monte Carlo sampling of phi and psi angles calculated not to disturb the global fold of the protein. Rohl (Rohl et al.,

2004) provides a review of the fragment selection and backbone refinement algorithms in Rosetta.

I.1.2 Sampling strategies for side-chain degrees of freedom

Systematic sampling of side-chain degrees of freedom of even short peptides quickly becomes intractable (Levinthal, 1968). Rosetta drastically reduces the number of conformations sampled by usage of discrete conformations of side-chains observed in the PDB (Kuhlman and Baker, 2000; Dunbrack Jr and Karplus, 1993). These "rotamers" capture allowed combinations between side-chain torsion angles as well as the backbone phi and psi angles and thereby reduce the conformational space (Dunbrack Jr and Karplus, 1993). A Metropolis Monte Carlo simulated annealing run is used to search for the combination of rotamers occupying the global minimum in the energy function (Kuhlman and Baker, 2000; Leaver-Fay et al., 2005). This general approach is extended to protein design by replacing a rotamer of amino acid *A* with a rotamer of amino acid *B* in the Monte Carlo step.

I.1.3 Rosetta energy function

Simulations with Rosetta can be classified based on whether amino acid side-chains are represented by super atoms or centroids in the low-resolution mode or at atomic detail in the high-resolution mode. Both modes have optimized energy functions that have been reviewed previously by Rohl (Rohl et al., 2004).

I.1.3.1 Knowledge based centroid energy function

The Rosetta low-resolution energy function treats the side-chains as centroids (Simons et al., 1997, 1999). This energy function models solvation, electrostatics, hydrogen bonding between beta strands, and steric clashes. Solvation effects are modeled as the probability of seeing a particular amino acid residue with a given number of alpha carbons within an amino acid dependent cutoff distance. Electrostatic interactions are modeled as the probability of observing a given distance between centroids of amino acids. Hydrogen bonding between beta strands is evaluated based on the relative geometric arrangement of strand fragments. Backbone atom and side-chain centroid overlap is penalized providing the repulsive component to a van der Waals force. A radius of gyration term is used to model the effect of van der Waals attraction. All probability profiles are derived using Bayesian statistics on crystal structures from the PDB. The lower resolution of this centroid-based energy function smoothes the energy landscape at the expense of its accuracy. This smoother energy landscape allows structures which are close to the true global minima to maintain a low energy even with structural defects that a full atom energy function would penalize harshly.

I.1.3.2 Knowledge based all atom energy function

The all-atom high-resolution energy function used by Rosetta was originally developed for protein design (Kuhlman and Baker, 2000; Kuhlman, 2003). It combines the 6-12 Lennard Jones potential for van der Waals forces, a solvation approximation (Lazaridis and Karplus, 1999), an orientation dependent hydrogen bonding potential (Kortemme et al., 2003), a knowledge based electrostatics term, and a knowledge based conformation dependent amino acid internal free energy term (Dunbrack Jr and Karplus, 1993). An important consideration when constructing this potential was that all energy terms are pairwise decomposable. The pairwise decomposition of each of the terms limits the total number of energy calculations to $\frac{1}{2}N(N-1)$ where N is the number of atoms within the system. This limitation allows pre-computation and storage of many of these energy contributions in the computer memory which is necessary for rapid execution of the Metropolis Monte Carlo sampling strategies employed by Rosetta during protein design and atomic-detail protein structure prediction.

I.2 Protein design using Rosetta

Protein design methods seek to determine an amino acid sequence that folds into a given protein structure or performs a given function. The protein design problem of finding a sequence that folds into a given tertiary structure is also known as the "inverse protein folding problem". The RosettaDesign (Kuhlman, 2003) algorithm is an iterative process that energetically optimizes both the structure and sequence of a protein. RosettaDesign alternates between rounds of fixed backbone sequence optimization and flexible backbone energy minimization (Kuhlman, 2003). During the sequence optimization step, a Monte Carlo simulated annealing search is used to sample the sequence space. Every amino acid is considered at every position in the sequence, and rotamer positions are constrained using the Dunbrack Library (Dunbrack Jr and Karplus, 1993). After each round of Monte Carlo sequence optimization, the backbone is relaxed to accommodate the designed amino acids (Kuhlman, 2003). The practical uses of RosettaDesign can be divided into five basic categories: Design of novel folds (Kuhlman, 2003), redesign of existing proteins (Korkegian, 2005), protein interface design, enzyme design (Jiang et al., 2008), and prediction of fibril forming regions in proteins (Thompson et al., 2006).

I.2.1 De novo protein design

The RosettaDesign method has been used for the *de novo* design of a fold that was not (yet) represented in the PDB. A starting backbone model consisting of a five stranded beta-sheet and two packed alpha-helices was constructed with the Rosetta *de novo* protocol using distance constraints derived from a two-dimensional sketch (Rohl et al., 2004). The sequence was iteratively designed with five simulation trials of 15 cycles each. The final sequence was expressed and the structure was determined using X-ray crystallography. The experimental structure has an RMSD to the computational design of less than 1.1 Å (Kuhlman, 2003).

Similarly, a molecular switch which folded into a trimeric coiled coil in the absence of zinc, and a monomeric zinc finger in the presence of zinc was designed by extending RosettaDesign to simultaneously optimize a sequence in two different folds. The sequence of an existing zinc finger domain was aligned with a coiled-coil hemaglutinin domain. During the design protocol the sequence was optimized to fold into both tertiary structures (Ambroggio and Kuhlman, 2006).

I.2.2 Redesign of existing proteins

When nine globular proteins were stripped of all side-chains and then redesigned using RosettaDesign the average sequence recovery was 35% for all residues (Kuhlman, 2003). In four out of nine cases the stability of the proteins improved as measured by chemical denaturation. The structure of a redesigned human procarboxypeptidase (PDB 1AYE) (García-Sáez et al., 1997) was determined experimentally. RosettaDesign was then used to systematically identify mutations of procarboxypeptidase that would improve the stability of the proteins. All of the tested mutants were more stable than the wildtype protein with the top scoring mutant having a reduction of free energy of 5.2 kcal/mol (Dantas et al., 2007).

RosettaDesign has also been used to modify the structure of existing proteins. In one study, the HisF TIM Barrel protein was selected as the basis for the design of a novel symmetric protein. The backbone structure of half the barrel was duplicated, and Rosetta was used to redesign the new structure to have both symmetric sequence and structure. The new protein, named FLR, was expressed and crystallized. The two resulting crystal structures had RMSDs of 0.49 Å and 0.87 Å to the computational prediction, demonstrating the ability of RosettaDesign to make accurate predictions of side-chain conformation and energies (Fortenberry et al., 2011).

I.3 Existing challenges with protein surface design

I.3.1 Electrostatic energy is insufficient to predict the impact of protein surface mutations

While protein design has had frequent successes, there are outstanding challenges, particularly with respect to the design of the surfaces of soluble proteins. Solvent interactions are critical to accurately measuring the electrostatics and stability of protein surfaces (Park et al., 2004). However, due to the computational complexity associated with explicit solvent modeling, implicit models are frequently used, and may not be sufficiently detailed to make accurate energetic predictions. Furthermore, the network of interactions between protein surface residues and the overall stability of the proteins are highly complex. Xiao (Xiao et al., 2013) demonstrated that while electrostatic surface interactions are important for stability, the impact of a single mutation on experimentally determined stability frequently cannot be explained by the impact on computed electrostatic energy.

I.3.2 Computationally designed proteins frequently aggregate unless "supercharged"

In addition to issues with stability, computationally designed proteins often have issues with aggregation. A study was performed in which RosettaDesign was used to fully redesign 10 proteins (Dantas et al., 2003). They found that 4 of the 10 designed proteins formed insoluble aggregates at 1 mM concentration. Aggregation appears to be a general phenomenon affecting protein design, and it has been repeatedly demonstrated that "supercharging" proteins by introducing large numbers of charged surface residues (Simeonov et al., 2011; Kurnik et al., 2012; Lawrence et al., 2007) can reduce aggregation in designed proteins. However, "supercharged" proteins are infrequently seen in nature, suggesting that evolved mechanisms for retaining solubility and avoiding aggregation are more complex. Observation of the folding properties of supercharged proteins suggest that excessive charging can inhibit folding (Lawrence et al., 2007), which may have acted as an evolutionary barrier to natural supercharging.

I.4 The history of ligand docking

I.4.1 Early attempts at hand-docking ligands using physical models

Attempts to model and predict protein-drug interactions date began shortly after the publication of the X-ray structure of hemoglobin. To this end, Beddell et. al. published a proof of concept method for structure based drug discovery in 1976 (Beddell et al., 1976). The method developed by Beddell relied on the manual placement of physical molecular models into a scale model of the hemoglobin electron density, which allowed the authors to identify novel compounds with millimolar activity. While the identified compounds were relatively poor by modern standards, and the method of manual placement into physical models did not provide a means of postulating mechanism of action, the authors recognized the value of the new technique, saying:

> It has been common practice to design new drugs by modifying the chemical structure of a known substance which has the desired biological properties, and this procedure has imposed severe restraints on the choice. However, it is not necessary for the novel compounds to be related to the original substance when the structure of the receptor site is already known.

It is remarkable that this observation on the state of rational drug discovery continues to be relevant, nearly 40 years after it was originally made.

I.4.2 Energy functions for protein-ligand docking

Over the past 40 years, a wide range of energy functions have been developed or adapted for use in protein-ligand docking. While numerous scoring functions have been developed, they can be generally categorized into 3 groups: Knowledge-Based, Empirical, and physics based (Sliwoski et al., 2014). Knowledge-Based scoring functions are derived from sets of experimental data, and are based on the idea that structures seen more frequently in nature have a more favorable energy relative to those seen less frequently. Using this concept, it is possible to model complex physical interactions in a computationally tractable way, even if the underlying physics and chemistry driving these interactions is poorly understood.

Physics based functions exist at the other end of the spectrum. Physics based functions are based on newtonian approximations of the quantum-physical interactions that govern chemical interactions. Typically, these approximations are parameterized using either experimental information, or information derived from high accuracy quantum mechanical calculations. Well parametrized physics based scoring functions are capable of making highly accurate predictions, but rely on detailed knowledge of the underlying physical system being modeled. Physics based functions are frequently used for molecular dynamics calculations, and have been successfully employed in protein-ligand docking as the energy function used by tools such as DOCK (Kuntz et al., 1982).

Like KBPs, empirical scoring functions are derived from experimental data. However, rather than attempting to broadly model complex systems, empirical functions are derived from statistical regressions of specific chemical measurements, such as hydrophobic contacts. FlexX (Rarey et al., 1996) is an example of a ligand docking tool utilizing and empirical scoring function.

In 2004, a comparative benchmark of nine commonly used scoring functions from all three categories mentioned above provides some insight into the strengths and weaknesses of these methods (Ferrara et al., 2004). In the Ferrara et al study, A set of 189 protein-ligand complexes was used to study the ability of the nine tested scoring functions to correctly distinguish between correctly and incorrectly docked ligands. Of the tested scoring functions, CHARMm, DOCK, DrugScore, ChemScore and AutoDock were most effective at correctly distinguishing between correctly an incorrectly docked models. Of these, DrugScore is knowledge based, CHARMm and DOCK are physics based, and ChemScore and AutoDock are empirical. The success of these methods, which had 80-90% success rates at distinguishing between correctly and incorrectly docked position, indicates that all three categories of scoring function can be successful when self-docking. DrugScore, as a knowledge based potential, is relatively insensitive to small changes in protein atom positions. This insensitivity proved to be advantageous in many cases when distinguishing between poses in cross-docked ligands. As the effect of cross-docking is to essentially introduce structural noise, the insensitivity of the knowledge based potential to very small perturbations means that these perturbations do not negatively impact the overall ability of the system to correctly score the ligand.

I.4.3 An overview of influential protein-ligand docking methods

After the advent of relatively inexpensive general purpose computers in the early 1980s, the promise of accurate and rapid computational design of novel small molecules has driven a wide array of research into improved methods for predicting protein-ligand interfaces. A successful protein-ligand docking tool must solve two basic problems: sampling and scoring. To effectively solve the sampling problem, the software must be able to efficiently explore both the rigid space of the protein binding site, as well as the conformational space of both the protein and the ligand. To effectively solve the scoring problem, a score function must be developed which can rapidly distinguish between energy favorable and unfavorable conformations. Solving both of these problems has proven highly challenging, although great progress has been made.

I.4.3.1 DOCK

In 1982, Kuntz et al. published DOCK, one of the earliest computational tools for modeling protein-ligand interactions (Kuntz et al., 1982). DOCK used a relatively simple energy function which modeled repulsive forces as hard spheres, and a rough approximation of hydrogen bonding which favored binding positions in which hydrogen bond donor groups on the ligand were within 3-5 Å of acceptor nitrogens and oxygens on the protein backbone. In concept, the DOCK algorithm is similar to the manual placement method described by Beddell et al. above. The program uses the van der Waals radii of the protein and ligand atoms to create “space filled” representations of both the receptor pocket and the ligand. Pairs of protein and ligand spheres are then considered systematically, and the set of pairings which minimizes sphere overlap is selected. This algorithm is driven almost entirely by shape complementarity, and effectively models the "lock and key" hypothesis of protein-ligand binding, in which a rigid protein is matched with a rigid ligand.

I.4.3.2 GRID

In 1984, Goodford et al published GRID, a computational method for predicting energetically favorable protein-ligand binding conformations (Goodford, 1985). GRID differed from previous attempts structure based drug discovery in that it used chemical information rather than relying entirely on receptor fit. Specifically, it assessed the protein-ligand interaction using an empirical energy function consisting of a Lennard-Jones term, electrostatic term, and hydrogen bonding term. This energy function was precomputed as a 3-dimensional grid overlaid on the ligand binding site. Thus, the total score of the ligand could be rapidly assessed as the sum of the grid squares the atoms are located in. Precomputation of the scoring grid enabled many ligand conformations and compositions to be rapidly assessed, and the addition of chemical information in addition to shape proved more effective than simply evaluating shape complementarity.

While the GRID method proved reasonably effective, several shortcomings which limited the effectiveness of the method. The physics based force field used was relatively rudimentary, and the limited set of chemical probes used to create the grid. Additionally, accurate docking into a full-atom grid based model requires a high degree of precision in the position of the protein atoms, which limits the effectiveness of such a model in cases where the accuracy of the protein structure is lower.

I.4.3.3 The importance of protein and ligand flexibility

In the years following the publication of DOCK and GRID, additional experimental study of protein structure began to indicate that the rigid body lock and key model was not adequate for the modeling of protein-ligand interactions. It had long been suspected (Koshland,

1958) that enzymes and receptors may be flexible to accommodate the fit of small molecules (the so called “induced fit” hypothesis), however in 1995, Nicklaus et al. (Nicklaus et al., 1995) published work suggesting that small molecules also undergo substantial conformational shift on binding. This conclusion was arrived at by comparing the geometry of flexible small molecules observed bound to proteins with the geometry of the same small molecules when crystallized in the absence of a protein, or when computationally minimized using molecular mechanics. The results of this study indicated that while the conformations of rigid structures typically differed by < 0.1 Å RMSD between the bound and unbound context, flexible ligands typically differed significantly, frequently by several angstroms. Furthermore, the difference in RMSD between bound and unbound ligands was strongly correlated with the number of rotatable bonds in the ligand, with an R^2 correlation of 0.82. In response to this research, the development of newer protein-ligand docking methods began to focus on the flexibility of the system. While flexibility had previously been avoided due to the inherent increase in computational complexity associated with modeling it, these findings made it clear that flexibility was a critical component of protein-ligand interaction.

I.4.3.4 FlexX and GOLD

FlexX (Rarey et al., 1996) and Genetic Optimization of Ligand Docking (GOLD) (Jones et al., 1997), are two of the early methods which attempted to model ligand flexibility. FlexX represents the ligand binding site using a set of interaction sites, which are defined as surfaces surrounding hydrogen bond donors and acceptors, metals and metal acceptors, aromatic rings, methyls and amides. An empirical scoring function is used to score ligand conformations based on the distance and angle between defined protein and ligand interaction sites. FlexX uses an incremental construction algorithm to model ligand flexibility. An initial central fragment of the ligand is placed in the binding site using an incremental construction algorithm, and the additional fragments necessary to build the entire ligand are then placed such that they can connect to the initial fragment and minimize the energy function score. GOLD, on the other hand, relies on the user providing a reasonable initial position for the ligand inside the protein binding site. From that initial position, a genetic algorithm (Jones et al., 1995) was used to optimize the rotation angles of both the ligand and the interacting protein side-chains. The genetic algorithm makes it possible to rapidly find a high quality local minimum without the exhaustive sampling of bond angles that had made the problem previously intractable. As a result of this new sampling technique, GOLD was able to successfully recover the correct binding conformation in 71 out of 100 X-ray crystal structures in a benchmarking study.

I.4.3.5 Glide

In 2004, Glide (Friesner et al., 2004) was published as a novel method for protein-ligand docking aimed at the screening of large libraries of small molecules. To improve the speed of the algorithm, Glide models the receptor site using a set of cartesian scoring grids, and keeps the receptor atoms fixed. This allows the ligand to be rapidly scored, making it possible for a large number of ligand positions to be evaluated. Glide performs a set of exhaustive searches along at cartesian grid overlaid on the receptor binding site. To reduce the amount of sampling required, the step size of the grid is reduced over the course of the search process, beginning with a 2.0 Å pitch grid. Additionally, a set of filters based on the empirical ChemScore (Eldridge et al., 1997) energy function are used to progressively filter the set of allowable binding orientations using increasingly detailed metrics. After an initial starting position is accepted, The conformational space of the ligand is exhaustively searched, and the final pose is energy minimized. The use of a grid representation for the energy function makes it possible to to screen large numbers of compounds very rapidly, making Glide a popular choice for virtual screening studies (Yilmaz et al., 2013; Bauer et al., 2013).

I.5 The history of RosettaLigand

RosettaLigand was originally published in 2006 (Meiler and Baker, 2006) as a protein-ligand docking tool based off of the previously published RosettaDock (Gray et al., 2003) protein-protein docking tool. The original RosettaLigand docking algorithm took advantage of the knowledge based energy function used by RosettaDock. The use of a knowledge based potential rather than a physics based potential is advantageous as knowledge based potentials are capable of indirectly modeling effects that are difficult to model directly. Additionally, the ability of RosettaLigand to rapidly optimize protein side-chain geometry (Barth et al., 2007) made it possible to model protein-ligand interactions with full atomic detail. While RosettaLigand was frequently able to accurately predict the binding orientation ligands (Meiler and Baker, 2006), it was unable to model backbone or ligand flexibility, which have long been suspected to be critical for protein-ligand binding (Yang et al., 2014; Koshland, 1958). To rectify this situation, further extensions were made to RosettaLigand by Davis et al (Davis and Baker, 2009) which allowed RosettaLigand to fully consider the flexibility of all parts of both the protein and the ligand. A blind benchmarking study comparing the pose recovery performance of the 2009 version of RosettaLigand suggested that overall it performed similarly to other major ligand docking tools (Davis et al., 2009). A notable conclusion of this study is that while most of the tools studied have a similar performance overall, the performance in predicting docking pose for individual protein targets varies wildly. This inconstant performance between protein targets and protein docking tools is seem in other studies as well.

I.5.1 RosettaLigand is capable of successfully predicting binding based on comparative models

One of the advantages of a knowledge based energy function is the ability to accurately model complex physical effects without a direct physical model. In principle, this, combined with the ability to model both backbone and side-chain flexibility would make RosettaLigand well suited to the docking of ligands into comparative models or other low resolution protein structures. To assess this, a benchmarking study was performed in which small molecules with known binding positions were docked into homology models generated in the Critical Assessment of protein Structure Prediction (CASP) experiment (Kaufmann and Meiler, 2012). The results of this benchmark demonstrated that in most of the tested cases, Rosetta was able to generate low energy binding positions within 2.0Å of the crystallographic binding site.

I.5.2 Applications of RosettaLigand to drug discovery

In addition to benchmarking studies, Rosetta has been used to develop models of ligand binding in G-Protein Coupled Receptor (GPCR)s. A comparative model of hSERT was created based on the dSERT crystal structure. S- and R-citalopram were docked into this comparative model using RosettaLigand, and the resulting predicted binding poses were used to design mutational studies to identify residues critical for S-citalopram binding. Rosetta was able to correctly predict that Y95 and E444 formed protein-ligand interactions critical to binding (Combs et al., 2011). Similarly, RosettaLigand was used to model the binding of Positive Allosteric Modulators in a comparative model of mGlu$_5$ (Turlington et al., 2013). In this case, the predictions made by RosettaLigand were used to guide mutation and radioligand binding studies, the results of which were used to further refine models. These models made it possible to map out critical interactions between Positive Allosteric Modulators and the mGlu$_5$ binding site even in the absence of crystal structure information.

I.6 Computational ligand docking has inconsistent predictive power

A common thread running through the ligand docking research described above is the difficulty of docking ligands into some proteins. For every protein-ligand method developed, some percentage of protein-ligand interfaces cannot be effectively predicted. While the predictions generated by protein-ligand docking has made some major scientific contributions to drug discovery and molecular modeling, the unreliability of the method has historically constrained its usefulness.

In 2006, a diverse set of 81 protein targets, each with a diverse set of known active and predicted inactive ligands was assembled as the DEKOIS 2.0 dataset (Bauer et al., 2013). Glide, GOLD and Autodock Vina were used to screen this dataset, and the pROC AUC enrichment for each target and each screening method was computed. The results of this benchmark showed a wide range in the predictive ability of the three screening methods. While all three docking methods had strong predictive power against some protein targets (COX2, KIF11), there were several cases in which no method had predictive power (HSP90, QPCT), and more cases in which some methods were able to make accurate predictions while others were not (COX1, ROCK-1). Furthermore, it was not possible for the authors to identify straightforward patterns to predict which protein targets could be successfully screened against and which could not. The phenomenon of structure based vHTS methods having inconsistent performance depending on the protein target has been replicated in other studies. For example, the Directory of Useful Decoys: Enhanced (DUD-E) benchmark set was screened using DOCK, and the resulting predictions exhibited similar inconsistencies to those seen in the DEKOIS 2.0 study (Mysinger et al., 2012).

I.7 Artificial Neural Network techniques have proven valuable for extracting complex signals

Since the publication of the perceptron as a method of machine learning (Rosenblatt, 1958), ANN techniques have become an area of great interest to the machine learning community. While there was much initial optimism regarding the use of ANNs to learn complex tasks, early perceptron based models proved limited in their abilities (Gallant, 1990), and the state of computational hardware at the time prevented ANN based techniques from living up to the early optimism. In more recent years, the availability of large clusters of low cost computer hardware as lead to a renaissance in both the development and application of ANN based machine learning techniques. ANNs have been used for tasks such as face recognition (Zhao et al., 2003), cancer cell identification (Zhou et al., 2002), and drug activity classification (Gohlke and Klebe, 2002). ANNs are popular choices for these tasks due to their ability to extract the signal from complex patterns.

I.8 Over-training is a common pitfalls in the use of ANNs for pattern recognition

While ANN based approaches have been valuable to many fields, they are often difficult to use in practice. Due to the very large number of free parameters in a neural network, they are very prone to over-fitting. In over-fitting, the neural network effectively "memorizes" the dataset, and becomes a model that exactly implements the set of data used for training (Tetko et al., 1995). The consequence of overfitting is that the model will be capable of exactly reproducing the training data set, but will have no ability to make predictions beyond that. The standard method for addressing this is to use as small a network as possible, and to perform a "cross-validation", in which part of the training dataset is withheld from training and used to keep track of the network performance as training proceeds. cross-validation makes it possible to determine when over-fitting is occurring and halt training, resulting in a model that is well trained but still general.

I.8.1 Deep networks and node dropout as novel methods for improving network generalizability

Very recently, new methods in network training have been developed to improve the generalizability of neural networks and prevent over-training. The development of inexpensive General Purpose Graphical Processing Unit (GPU) hardware has made it possible for extremely large networks to be efficiently trained. Additionally, development of new training methodologies (Hinton et al., 2006) has made it possible to train networks with very large numbers of nodes, and more than 2 layers of hidden nodes. These so-called "deep networks" appear to be capable of learning abstract features and concepts in an un-supervised fashion (Le, 2013), and appear to exhibit the kinds of learning behaviors that were originally envisioned by the developers of early perceptron methods. Another promising and broadly applicable new method in the training of neural networks is the so-called “node dropout” method. In this method, every time a new training case is provided to the network, 50% of the nodes in the network are excluded. This has the effect of preventing nodes from becoming dependent on each other, which leads to over-training. By using node dropout, it has been possible to both conventional shallow networks and deep networks using a larger number of nodes than would normally be allowable, increasing the generalizability and performance of the models (Hinton et al., 2012).

I.9 Using ANNs to make predictions regarding drug activity is a major area of current research

As a result of their properties to model complex interactions in natural systems, ANN based methods are a popular choice for constructing models of drug activity and binding. In many ways, drug activity is a harder problem to solve than image recognition. Unlike images, The activity of a drug depends in large part on its conformation (Nicklaus et al., 1995). A cat does not become a bobsled if it folds its legs, but active small molecules can become inactive in certain geometric conformations. To sidestep this, ANN based methods are often used to make 2D ligand-based QSAR models which are trained using the 2 dimensional structures of known active and inactive small molecules without including protein structure information (Myint et al., 2012). While 2D descriptors do frequently outperform 3D descriptors, 3D descriptors can be made useful. By encoding 3D information in the form of a RDF, the 3D geometry of the small molecule is described in a way that is rotationally and translationally independent. Additionally, RDFs encode 3 dimensional protein data as a one dimensional fingerprint, making them ideally suited as ANN descriptors. RDF based descriptors, in conjunction with 2D descriptors, have been used to build a QSAR model capable of predicting novel active compounds (Mueller et al., 2010), demonstrating the value of the technique as a whole.

While ligand-based QSAR methods have proven valuable for predicting the activity of drugs against specific targets, these models have a fundamental limitation in that they can only be applied to the target they were trained against. Techniques exist to define and maximize the domain of applicability of these models (Sahigara et al., 2012), but they are fundamentally tuned to a specific target or subset of targets. Furthermore, the training of a ligand based QSAR model requires that a set of experimentally known active and inactive compounds exist, which limits the use of ligand based methods to targets which have already been experimentally evaluated. As a result of this limitation, some recent research has focused on using ANN based models to score protein-ligand docking positions. Because of the ability of modern ANN based methods to recognize very complex and noisy signals, the potential exists to develop an ANN model which is capable of distinguishing between active and inactive small molecule poses even in cases where the scoring function of the docking system is unable to do so. A number of methods have been developed to do this (Durrant et al., 2013), and while they have in general been successful (Durrant and Mccammon, 2011), the dream of a vHTS method that acts as a generally applicable model of protein-ligand binding affinity has not yet been realized, and research in the area continues.

CHAPTER II

Design of Native-Like Proteins through an Exposure-Dependent Environment Potential

II.1 Introduction

II.1.1 Computational design of proteins is an active area of research

The design of protein surfaces with proper amino acid composition is critical to preventing aggregation and allowing for correct protein folding (Chandler, 2005). Thermostabilization of enzymes and design of proteins with novel folds are two possible applications of the present research.

II.1.2 The current Rosetta solvation model does not penalize the burial of apolar atoms

As there are relatively few explicit interactions of amino acids on the protein surface, the total energy of a residue is dominated by Rosetta's implicit solvation model. The solvation model currently used by Rosetta is a function developed by Lazaridis and Karplus (Lazaridis and Karplus, 1999). This potential estimates the solvation free energy of an atom from a reference free energy, where the atom is essentially fully solvent-exposed. For every nearby atom, a cost of "desolvation" is added in a pairwise decomposable and distance-dependent manner. This procedure aligns with the protein folding process, where amino acids move from a completely exposed location into varying degrees of burial. While the model is parameterized for all amino acid atom types, it is driven by high desolvation penalties of polar atoms. For this reason, it is quite insensitive to the burial of apolar atoms because desolvation energies are small.

II.1.3 RosettaDesign currently has difficulty designing protein surfaces

This paradigm of desolvation is useful for determining energy changes in the folding of a monomeric protein. However, hydrophobic patches on the surface of a *de novo* designed protein are hardly penalized, as the environment of these amino acids did not change in the folding process. At present, RosettaDesign excels in the design of tightly packed protein cores, while the protein surface is often poorly composed and requires manual adjustment (Dantas et al., 2003). We hypothesize that native proteins have evolved to minimize unspecific aggregation, a fact that is ignored by the desolvation potential. Evolutionary pressures exerted on protein sequence composition by the requirement of protein solubility are difficult to model with a typical physics-based model, but can be modeled effectively with a knowledge-based energy potential.

II.1.4 Description of the RosettaDesign energy function

The RosettaDesign energy function is a weighted composite of the Lazaridis-Karplus solvation free energy potential, attractive and repulsive interactions, an action center pairwise potential to approximate electrostatic interactions, an orientation-dependent hydrogen bonding potential (Kortemme et al., 2003), and reference energies for amino acid type and conformation (Dantas et al., 2003). Amino acid reference energies and scoring function weights are optimized to maximize sequence recovery in a protein design benchmark. Reference energies can be viewed as the ground state energy of an amino acid in an essentially fully exposed, unfolded peptide chain. Hence, these reference energies can disfavor apolar amino acids on the surface, thereby representing some of the evolutionary pressure to prevent aggregation. However, the same reference energies are also fitted to reflect amino acid propensities in nature independently of burial. In addition, the reference energies are fitted to maximize sequence recovery and thereby counterweight other inaccuracies in the RosettaDesign energy function. As a result, the reference energies form a container term that combines multiple effects that can be difficult to disentangle, and it provides a corrective power against exposed hydrophobic amino acids on the surface.

II.1.5 A knowledge based environment potential was developed to improve the quality of protein surface designs

To improve upon the above shortcomings of RosettaDesign, we implemented the Neighbor Vector (NV)-based KBP previously described by Durham et al. (Durham et al., 2009). This neighbor vector environment KBP converts the likelihood to see an amino acid at a given level of exposure into an environment energy. The NV environment KBP encapsulates both desolvation energy and evolutionary biases against apolar amino acids at the protein surface with amino acid level resolution.

II.1.6 The development of a more accurate approximation of SASA

The usefulness of an environment potential based on burial is contingent on an accurate measure of burial. Solvent Accessible Surface Area (SASA) is the most accurate means of calculating amino acid burial but is generally time-consuming to compute, limiting its usefulness in protein design. RosettaDesign currently uses a Neighbor Count (NCR) for estimating solvent accessibility in the pair potential. While the NCR method correlates with residue burial, high inaccuracies are common in surface and partially exposed positions (Figure II.1).

II.1.6.1 The NV SASA approximation was previously developed by Durham et al.

To overcome the limitations of the NCR burial approximation, an NV approximation of residue burial was implemented. For a schematic representation of the NV algorithm, see Figure II.2. The NV algorithm and KBP generated and described by Durham et al (Durham et al., 2009) was used in our implementation. Proteins selected for deriving the KBP were monomeric, globular proteins, which do not engage in obligate, and therefore strong, protein-protein interactions. It is expected that some of these proteins will engage in transient interactions with other proteins, however, these interactions will be weaker. As

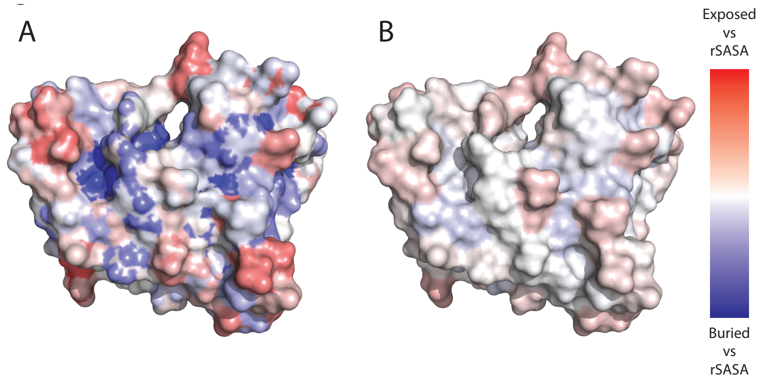


Figure II.1: Comparison of NV and NCR measures to rSASA. In both panels, a color map plots the difference between a surface approximation method and the normalized rSASA value. A residue for which the SASA approximation matches rSASA exactly would have a score of 0.0 and be colored white. Regions of the surface in red are categorized as more solvent exposed than by rSASA, while regions in blue are categorized as less solvent exposed than by rSASA. (A) protein 7DFR colored by the NCR approximation of surface accessibility as used in Rosetta. (B) protein 7DFR colored using the NV approximation. The NV measure has significantly smaller deviations from the rSASA standard with a mean of 0.14 compared to the mean deviation of 0.20 seen with the NCR measure. Additionally, the NV measure is more consistent, with a standard deviation of 0.11 compared to the standard deviation of 0.46 seen with the NCR metric. Panels A and B illustrate the improvement in consistency, as areas of score deviation in Panel B are smaller and generally less "patchy" in their appearance.

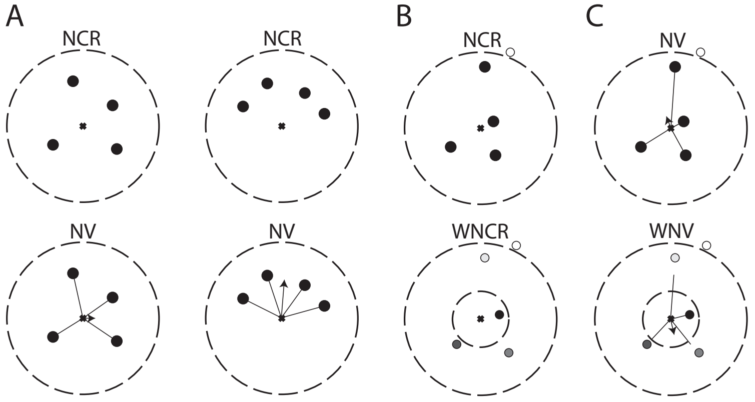


Figure II.2: (A) The left and right panels both have the same Rosetta neighbor count (Dantas et al., 2003) but very different degrees of burial. The neighbor vector method is able to distinguish between these cases by calculating the vectors between the query residue and its neighbors. The length of the vector indicates the degree of burial, with shorter vectors representing more buried residues. (B) The Weighted Neighbor Count (WNCR) method gives a higher weight to neighbors near the query residue, smoothing the effect of small changes in composition on the measured degree of burial. (C) The combination of the NV and WNCR methods results in a more accurate measure of residue distribution. In all panels, dotted lines represent lower and upper bounds for counts, the X marks the query residue, and circles represent residues surrounding the query residue.

a result, the noise added to the KBP by these interactions will be of low magnitude and uniform.

II.1.6.2 NV is a more accurate approximation of SASA than other methods

The half sphere approximation method developed by Hamelryck in 2005 (Hamelryck, 2005) approximates surface accessibility by counting the number of residues in a half sphere below the side chain of each amino acid. The half sphere count is directly related to residue burial. Half Sphere Exposure (HSE) is implemented in the freely available BioPython library, and this library was used to compare performance of HSE and NV to relative Solvent Accessible Surface Area (rSASA) (calculated using NACCESS). The per residue exposure was calculated for each of the proteins in the 42 protein benchmark set, and adjusted R^2 values were calculated for the correlation of each measure to rSASA. The adjusted correlation factor R^2 value for HSE to rSASA was 0.68, while the adjusted correlation factor R^2 for the NV method was 0.86. This suggests that while HSE is conceptually simpler, it does not perform as accurately as NV for proteins in our benchmark set.

II.1.6.3 Mapping NV results to rSASA results

A linear regression modeling the correlation between rSASA values (in the range of 0 to 1 calculated using NACCESS) and NV score (range 0 to 1) was generated based on all proteins in the 42 protein benchmark set. The equation for the resulting linear regression model was $rSASA = 1.29(NV) - 0.11$ and had an R^2 of 0.86. Based on this model, residues with NV scores between 0.00 to 0.24 will have an approximate rSASA value of 0.0 to 0.19, residues with NV scores from 0.25 to 0.39 will have an approximate rSASA value of 0.21 to 0.39, and residues with NV scores between 0.40 to 1.00 will have an approximate rSASA value of 0.40 to 1.10.

II.1.7 Overview of RosettaLigand scoring term architecture

Terms in the RosettaDesign energy function can take the form of either single-body or two-body terms. Two-body terms describe energies that pertain to the interaction between residues, such as the energy associated with hydrogen bonding, while single-body terms describe energies that pertain only to a single residue. The resulting NV environment KBP was implemented as a single-body term in the RosettaDesign energy function. RosettaDesign revision 39040 was used in all calculations.

II.1.8 Sequence recovery is insufficient as a metric for assessing protein design

Computationally assessing the performance of a protein design algorithm is inherently challenging. Historically, percent sequence recovery has been used as a metric for the quality of a protein design, as it has been observed that protein sequences are frequently close to optimal for a given fold (Kuhlman and Baker, 2000). However, many protein folds having large variations in sequence are frequently seen in nature (Chothia and Lesk, 1986). Of the 74,608 protein chains present in the Structural Classification of Proteins (SCOP) database as of 2009, only 1,280 individual folds are observed (Schaeffer et al., 2010). In many positions, particularly on the surface of proteins, multiple residues can be tolerated with similar energies. This finding limits sequence recovery as a measure for successful protein design because the design of a different but tolerated amino acid is counted as a failure. To resolve this problem, we introduce a metric based on sequence homology. A PSSM is derived from a Basic Local Alignment Search Tool (BLAST) query of the native sequence of a protein. The percent recovery of amino acids with positive values in the PSSM determines recovery of evolutionarily tolerated amino acids.

II.2 Experimental Procedures

II.2.1 Optimization of the weight of the new energy term

The RosettaDesign energy function is a linear combination of individual energy terms. As a result, the addition of a new energy term will impact the energy function as a whole. To address this, each energy term is multiplied by a weight, and these weights must be carefully optimized following the introduction of a new term. In most cases, it is not necessary to optimize the entire scoring function when a new term is added. Instead, only the terms that describe similar information as the new term are optimized. In the case of the NV environment KBP, the solvation free energy potential and the reference energies must also be optimized.

II.2.1.1 Development of a training data set

To ensure that the optimized weights would apply to a wide range of proteins, a set of 100 soluble protein crystal structures from the PDB were used in optimization. Structures were selected to have a sequence homology of less than 25%, a length of 67-179 amino acids, and a resolution better than 2.0 Å. The optimization was conducted using a five-way cross validation protocol. In this protocol, the 100 crystal structures described above were split into 5 groups of 20 structures each. In each component of the five-way validation, 80 proteins were used during optimization, and the remaining 20 were used to benchmark the resulting weights. In statistics generated from the benchmarking phase of the optimization process, results from all 5 sets of 20 proteins are combined, resulting in a total benchmark set of 100 proteins.

II.2.1.2 Particle Swarm Optimization scheme

An iterative particle swarm approach (Chen et al., 2007) was used to optimize the weights. The RosettaDesign standard energy function was used as an initial point for optimization, and the weight of the NV environment KBP was arbitrarily given an initial weight of 1.0. Twenty rounds of particle swarm optimization were performed for each component of the five-way cross validation described above. The weights were optimized to maximize the PSSM score of proteins designed using the energy function (Table II.1). The PSSM for each protein was generated from a PSI-PRED BLAST query of the protein structure sequence using an *e* threshold of 0.001 and 3 iterations. The Non-Redundant (NR) sequence database was used. The average sequence identity between the query sequence and all other sequences in the generated PSSMs was 30% for both benchmark sets.

II.2.1.3 Reference energies were optimized in addition to the NV environment KBP term

Because the standard deviation of the averaged reference averaged reference energies was relatively high, the reference energies of the averaged energy function are optimized to reduce the overall sequence composition biases introduced during design (Table II.2).

II.2.1.4 Two separate optimization experiments were performed

In the first experiment, the reference energies, solvation free energy potential, and the NV environment KBP were optimized. In the second experiment, the NV environment KBP was excluded from the energy function, and only the reference energies were optimized. This second experiment acts as a control and makes it possible to distinguish between design improvements due to reference energy optimization and design improvements due

			Five way cross validation sets						
	Rosetta Scoring term	**Scoring term description**	**1**	**2**	**3**	**4**	**5**	**Mean**	**Standard Deviation**
Free Weights	fa_sol	Solvation Free Energy Potential	0.558	0.569	0.563	0.547	0.585	0.564	0.014
	neigh_vect	NV environment KBP	1.025	1.013	0.996	1.059	0.978	1.014	0.030
Fixed Weights	fa_atr	Attractive force	0.8					0.8	0.0
	fa_rep	Repulsive force	0.44					0.44	0.0
	fa_intra_rep	Intra-residue repulsive force	0.004					0.004	0.0
	pro_close	Proline closure bonus	1.0					1.0	0.0
	fa_pair	Pair energy	0.49					0.49	0.0
	hbond_sr_bb	Hydrogen bonding: short range backbone	0.585					0.585	0.0
	hbond_lr_bb	Hydrogen bonding: long range backbone	1.17					1.17	0.0
	hbond_bb_sc	Hydrogren bonding: backbone-sidechain	1.17					1.17	0.0
	hbond_sc	Hydrogen bonding: sidechain-sidechain	1.1					1.1	0.0
	dslf_ss_dst	Disulfide sidechain distance	1					1	0.0
	dslf_cs_ang	Disulfide cystine sulfur angle	1					1	0.0
	dslf_ss_dih	Disulfide sidechain-sidechain dihederal	1					1	0.0
	dslf_ca_dih	Disulfide Cα-sidechain dihederal	1					1	0.0
	rama	Ramachandran score	0.2					0.2	0.0
	omega	Omega angle score	0.5					0.5	0.0
	p_aa_pp	Probability of an AA given phi/psi angle	0.32					0.32	0.0
	fa_dun	dunbrack rotamer library	0.56					0.56	0.0

Table II.1: A table showing the individual weights included in the optimization, and their values in each of the five cross validation sets. The mean and standard deviation of each free weight is also shown.

AA name	Five way cross validation sets							
	1	2	3	4	5	Mean	Standard Deviation	Re-optimized Energies
A	-0.313519	-0.345577	-0.332996	-0.299376	-0.311145	-0.3205226	0.018489708	-0.280778
C	-0.186065	-0.267899	-0.251004	-0.203121	-0.273443	-0.2363064	0.039429392	-0.191836
D	-0.0516543	-0.152027	-0.117976	-0.04581	-0.158867	-0.10526686	0.053922242	-0.0894836
E	-0.116794	-0.225243	-0.221177	-0.13447	-0.230275	-0.1855918	0.055185343	-0.163316
F	0.979346	1.00881	1.09867	0.976394	1.03573	1.01979	0.05028824	1.0029
G	0.187346	0.862973	0.681024	0.322192	0.952133	0.6011336	0.334354423	0.318222
H	0.773425	0.727736	0.727848	0.755789	0.683065	0.7335726	0.034276949	0.738805
I	-0.0879415	-0.063614	-0.0662562	-0.147069	-0.0349584	-0.07996782	0.041974551	-0.0892347
K	-0.0356543	-0.113405	-0.125976	-0.0465226	-0.104256	-0.08516278	0.041146221	-0.0565743
L	-0.288888	-0.27654	-0.302976	-0.350292	-0.286631	-0.3010654	0.029090569	-0.295543
M	-0.475654	-0.472027	-0.532784	-0.514173	-0.478867	-0.494701	0.027189556	-0.488778
N	-0.523683	-0.559673	-0.549976	-0.500635	-0.582867	-0.5433668	0.031950345	-0.532584
P	-0.486622	-0.556899	-0.579705	-0.488828	-0.551983	-0.5328074	0.042469908	-0.494263
Q	-0.481516	-0.58252	-0.554953	-0.491584	-0.569756	-0.5360658	0.046379036	-0.497717
R	-0.280894	-0.364054	-0.324666	-0.255467	-0.389405	-0.3228972	0.055748092	-0.294276
S	-0.396296	-0.436734	-0.438423	-0.37399	-0.437601	-0.4166088	0.029793151	-0.393299
T	-0.312536	-0.375408	-0.373141	-0.307255	-0.368256	-0.3473192	0.034311466	-0.332279
V	-0.175465	-0.166283	-0.166976	-0.217204	-0.134509	-0.1720874	0.029660029	-0.176609
W	1.43079	1.50528	1.52902	1.46971	1.42618	1.472196	0.045170863	1.47413
Y	0.842276	0.853102	0.902422	0.851712	0.81571	0.8530444	0.031423486	0.842514

Table II.2: A table showing the optimized weights of the reference energies for each amino acid.

to the addition of the NV environment KBP itself. While both NV environment KBP and the solvation free energy potential describe overlapping but different phenomena at different levels of resolution: The NV environment KBP is an indirect measure of solvation free energy and evolutionary biases against aggregation. This energy term functions at amino acid resolution and will be independent of side chain conformation. In contrast, the solvation free energy potential is at atomic resolution incorporating a specific model of solvation. While the solvation free energy potential does an inadequate job of accounting for biases against aggregation on the protein surface, it is highly accurate in avoiding burial of polar atoms, and is important to determine side chain conformation.

II.2.2 Analysis of the optimization experiments

The optimization experiments described above produce five individual energy functions, each generated from one Section of the five-way cross validation. To produce a single optimized scoring function for general use, the weights from the five optimized energy functions are averaged together, and the reference energies of the averaged energy function are optimized using the set of 100 proteins used in the initial cross validation. The averaged energy function is benchmarked on an independent set of 42 protein crystal structures, in which the proteins have a sequence homology of less than 15%, a size range of 150-225, and a resolution less than 1.5 Å. Note that these proteins are larger and more complex than the proteins used in the more time-consuming weight optimization procedure. As a result, this benchmark poses a formidable challenge for the RosettaDesign fixed backbone design algorithm. Several different metrics were used during benchmarking to assess the quality of designed proteins. Percent PSSM recovery was the primary benchmarking metric used in the study. Percent PSSM recovery was calculated as the percentage of residues that were designed as residues with a positive score in the PSSM of the native protein. In addition, the percent sequence recovery was measured as the percentage of residues that remained in as the native residue after design.

II.2.2.1 Percent PSSM recovery calculation

The percent PSSM recovery per residue, percent sequence recovery per residue, and the change in overall sequence composition were also calculated for each designed protein. Percent PSSM recovery per residue is calculated as $\frac{\textit{num pssm recovered}}{\textit{num designed}}$ where *num pssm recovered* is the number of residues with a given identity which were designed to a residue with a positive PSSM score, and *num designed* is the total number of residues designed. Percent sequence recovery per residue was calculated as $\frac{\textit{num recovered}}{\textit{num designed}}$ where *num recovered* is the number of residues with a given identity which were designed to an identical residue. In addition to calculating overall percent sequence recovery, sequence recovery by chemical group was also calculated. In this metric, residues were grouped into the categories polar (Ser, Thr, Asn, Gln), non-polar (Ala, Val, Leu, Met, Ile), aromatic (Phe, Tyr, Trp), charged (Lys, Arg, His, Asp, Glu), and other (Cys, Pro, Gly). A residue was counted as recovered if it was mutated to another residue within the same group.

II.2.2.2 Percent sequence recovery calculation

Percent sequence composition change per amino acid type was calculated as $\frac{d-n}{num\ designed}$ where d is the count of designed residues of a given type, and n is the count of native residues. To compute the change in overall sequence composition, a Root Mean Square (RMS) method was used. RMS percent sequence composition change was calculated as shown in equation II.1, where *statistical metric* is one of the metrics described in Section II.2.2.1 (shown as black bars in figures II.3 and II.5).

$$RMS = \sqrt{\frac{1}{20}\sum_{i=1..20}^{20}(statistical\ metric)^2} \tag{II.1}$$

II.2.2.3 Calculation of metrics by degree of burial

All of the above metrics were calculated for the entire protein, as well as the deeply buried region, surface region, and a boundary layer between the two. For this study, the buried region is defined as all residues with a NV score between 0.00 and 0.24, the boundary is defined as 0.25 to 0.39, and the surface region is defined as 0.40 to 1.00. The performance of the optimized energy functions via these benchmarks was compared to the performance of the standard RosettaDesign energy function.

The benchmarks described above are intended as a measure of how well RosettaDesign is accomplishing its goal of generating low-energy, native-like protein sequences. In a well-optimized energy function, we expect that the percent PSSM recovery will increase compared to the standard RosettaDesign energy function. We also expect that the percent sequence recovery will remain similar to that obtained with the standard energy function. Finally, we expect that a well-optimized energy function will exhibit smaller biases in sequence composition compared to proteins designed with the standard energy function.

	Percent PSSM Recovery			Percent Sequence Recovery		
	Standard	**Reference**	**NV environment KBP**	**Standard**	**Reference**	**NV environment KBP**
Buried	73.4%	77.1%	78.9%	64.9%	66.5%	65.5%
Boundary	72.1%	75.3%	77.3%	44.3%	46.6%	45.5%
Surface	70.4%	74.4%	75.9%	32.8%	35.9%	35.5%
Overall	72.0%	75.6%	77.2%	45.7%	48.1%	47.3%

Table II.3: Percent PSSM recovery and percent sequence recovery by degree of burial for 100 proteins used in optimization. “Standard” refers to the standard energy function, “Reference” refers to the modified standard energy function in which the reference energies were re-optimized, and “NV environment KBP” refers to the optimized energy function incorporating the NV environment energy term.

II.3 Results

The percent PSSM recovery and percent sequence recovery calculated for the 100 proteins used in the five-way cross validation are shown in Table II.3. The results of PSSM recovery and sequence recovery analysis show that the optimized NV environment KBP energy function exhibits a 5.2% improvement in percent PSSM recovery compared to the standard energy function and that 3.6% of this improvement was a result of reference energy optimization. The NV environment KBP energy function showed a 1.6% improvement in percent sequence recovery compared to the standard RosettaDesign energy function and a 2.4% improvement if only reference energies are optimized.

II.3.1 The NV environment KBP improves PSSM recovery and reduces sequence composition bias

The percent change in composition between native and designed sequences for the 100 proteins used in the five-fold cross-validation is shown in Figure II.3A. Proteins designed with the NV environment KBP energy function show a decrease in the average magnitude of sequence composition biases introduced during design compared to proteins designed with the standard energy function. Proteins designed with the standard energy function exhibit an RMS percent change in sequence composition of 2.0%, while proteins designed with the NV environment KBP show an RMS percent change in sequence composition of 1.0%. Figure II.3B shows that RMS per residue PSSM recovery increased from 3.8%

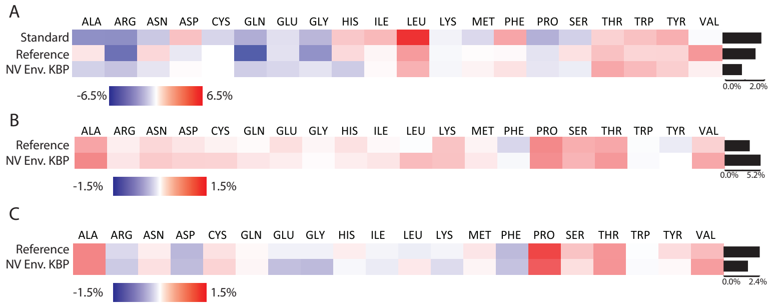


Figure II.3: A) shows the percent change in overall sequence composition between native and designed proteins for all 100 structures in the five-way cross-validation set. The black bars show the RMS percent composition change. B) shows the percent PSSM recovery for all 100 structures in the five-way cross-validation set. The black bars show RMS percent PSSM recovery. C) percent sequence recovery for all 100 structures in the five-way cross-validation set. The black bars show RMS percent sequence recovery.

with the standard energy function to 4.2% with the NV environment KBP, and II.3C shows that RMS per residue sequence recovery remained relatively constant between the standard energy function and NV environment KBP.

II.3.2 Independent benchmarking of a single averaged energy function shows improved performance

The energy functions produced with the five-way validation were averaged to produce a single energy function, the reference energies of this averaged function were optimized, and the benchmarking analysis used above was repeated using the averaged energy function. In this case, the independent benchmark set of 42 proteins was used. Table II.4 shows the percent PSSM recovery and percent sequence recovery calculated for the 42 proteins designed using the averaged energy function. The NV environment KBP showed an 8.8% improvement in PSSM recovery compared to the standard energy function and that 3.8% of this improvement was a result of the reference energy optimization. The NV environment KBP showed a 3.2% overall improvement in sequence recovery of which 1.9% was due to the reference energy optimization.

	Percent PSSM Recovery			Percent Sequence Recovery		
	Standard	**Reference**	**NV environment KBP**	**Standard**	**Reference**	**NV environment KBP**
Buried	68.0%	70.7%	76.0%	49.5%	49.8%	51.5%
Boundary	66.1%	70.6%	75.7%	32.1%	34.1%	35.3%
Surface	67.2%	72.2%	75.8%	22.7%	26.3%	27.3%
Overall	67.4%	71.2%	75.9%	35.7%	37.6%	38.9%

Table II.4: Percent PSSM recovery and percent sequence recovery by degree of burial for 42 proteins used in benchmarking. "Standard" refers to the standard energy function, "Reference" refers to the modified standard energy function in which the reference energies were reoptimized, and "NV environment KBP" refers to the optimized energy function incorporating the NV environment energy term.

II.3.3 Improvement in the quality of both buried and surface designs is seen

When sequence recovery is broken down by group (Figure II.4), a large improvement (decrease) in the percentage of unrecovered buried charged residues is observed, from 8.78% to 1.96% unrecovered residues function in the 100 protein benchmark set. Additionally, a decrease from 8.77% to 3.46% unrecovered non-polar residues on the surface is observed. Additionally, Figure II.4 reveals a fundamental difference in the two datasets. The lower recovery values seen in all categories in the 42-protein benchmark set suggest that it is a much more challenging target for design than the 100-protein benchmark set used in optimization. The proteins of the 42-protein benchmark set are substantially larger (average length of 207 residues) than those in the 100-protein benchmark set (average length of 120 residues). Each additional residue drastically increases the number of possible sequences to consider, decreasing the probability of a high quality design. Despite this more challenging independent benchmark, improvement was still observed.

II.4 Discussion

The results of both the 100-protein five-way cross-validation and the 42-protein independent benchmark set are consistent. In both cases, introduction of the NV environment KBP into the energy function and optimization of the energy function weights lead to an overall improvement in the quality of designed sequences. As the independent benchmark set tests an averaged scoring function that would be generally useful, the remaining analysis will

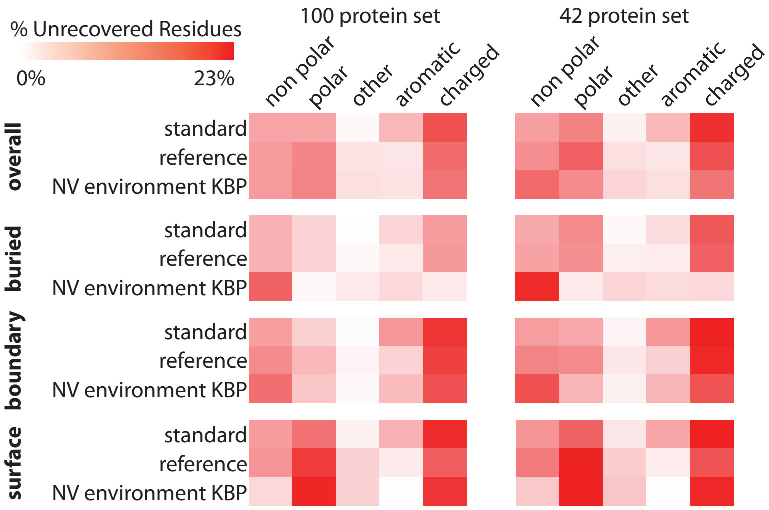


Figure II.4: Percentage of unrecovered residues (number of recovered residues divided by total number of residues in the benchmark set) by amino acid category in the 100 and 42 protein benchmark sets. The color scale ranges from white (small number of mistakes) to red (large number of mistakes). In this metric, residues were grouped into the categories polar (Ser, Thr, Asn, Gln), non-polar (Ala, Val, Leu, Met, Ile), aromatic (Phe, Tyr, Trp), charged (Lys, Arg, His, Asp, Glu), and other (Cys, Pro, Gly). A residue was counted as recovered if it was mutated to another residue within the same group.

focus on this benchmark set.

II.4.1 Explanation for source of observed improvements in design

The results of the benchmarking show that, in general, structures designed using the NV environment KBP exhibit smaller conformation biases and more evolutionarily favorable mutations. A detailed analysis of these results also provides some insight into the behavior of the RosettaDesign scoring function.

II.4.1.1 The NV environment KBP energy counteracts limitations in the RosettaDesign solvation potential

Due to the lack of an explicit water model in RosettaDesign, the standard RosettaDesign energy function is dominated by the solvation free energy potential. As a result, there are few constraints on amino acid mutations on the protein surface. Due to this lack of constraints, proteins designed with the standard energy function exhibit large biases in sequence composition on the protein surface. Proteins designed with the standard energy function show large numbers of aromatic residues on the protein surface. Specifically, there is a 3.1% increase in phenylalanines, a 1.9% increase in tryptophans, and a 2.5% increase in tyrosines on the protein surface in the benchmark set designed with the standard RosettaDesign energy function compared to the native structure. Proteins designed with the NV environment KBP show a large reduction in these biases. Proteins designed with the NV environment KBP showed a 1.4% increase in phenylalanines, 0.8% increase in tryptophans, and 1.5% increase in tyrosines compared to native proteins. While still large, these biases are much smaller than the biases observed with the standard energy function.

II.4.2 The NV environment KBP term improves design of buried residues as well as surface residues

It was expected that improvements in the quality of surface sequence design would be the primary benefit of the NV environment KBP. However, an analysis of the overall PSSM recovery, sequence recovery, and sequence composition biases suggest that the improvement given by the NV environment KBP implementation occurred across the board rather than merely at the protein surface. Table II.4 shows the overall impact of the NV environment KBP at various levels of burial. The percent PSSM recovery improved using the NV environment KBP by 8.0% in the buried region, 9.6% in the boundary region, and 8.5% on the surface region compared to the standard energy function. The percent sequence recovery improved by 2.0% in the buried region, 3.2% in the boundary region, and 4.6% in the surface region compared to the standard energy function.

When percent sequence recovery is broken down by group (Figure II.4), a large increase in the recovery of buried charged residues is observed, with a 6.2% increase compared to the standard energy function. Additionally, a 3.8% increase in recovery of non-polar residues is observed on the surface. Not all groups show improvement, and this is expected, as the scoring function was not directly optimized for percent sequence recovery.

	Percent PSSM Recovery			**Percent Sequence Recovery**		
	Standard	**Reference**	**NV environment KBP**	**Standard**	**Reference**	**NV environment KBP**
Buried	8.9%	8.9%	8.4%	11.6%	10.8%	10.1%
Boundary	9.6%	11.7%	9.2%	10.7%	11.3%	11.2%
Surface	8.1%	6.6%	6.9%	7.0%	8.0%	7.9%
Overall	6.3%	6.0%	5.2%	7.0%	7.1%	6.5%

Table II.5: Standard deviations for 100 protein benchmark set data. shown in table II.3

	Percent PSSM Recovery			**Percent Sequence Recovery**		
	Standard	**Reference**	**NV environment KBP**	**Standard**	**Reference**	**NV environment KBP**
Buried	7.1%	6.7%	5.9%	8.1%	8.9%	7.7%
Boundary	9.2%	9.1%	8.7%	8.1%	8.0%	8.6%
Surface	6.9%	6.6%	7.4%	4.7%	6.8%	6.0%
Overall	5.5%	5.7%	5.7%	4.6%	5.5%	5.3%

Table II.6: Standard deviations for 42 protein benchmark set data. shown in table II.4

While the overall percent changes are relatively small, these changes are both statistically and scientifically significant. To assess the statistical significance of the data, standard deviations were calculated in Tables II.5 and II.6. The standard deviations were calculated for both percent PSSM recovery and percent sequence recovery. Each of the five scoring functions generated during the five-way cross validation weight optimization using 100 proteins was used to design the independent set of 42 proteins. The standard deviations of PSSM and sequence recovery are listed in supplementary Tables II.5 and II.6. The standard deviations range from 0.1-1.2%. The average error is 0.4% and therefore smaller that the improvements in recovery rates observed.

II.4.3 Sequence recovery values are near the expected maximum

It is important to consider not only the absolute change in percent recovery, but also the change relative to the maximum possible recovery value. In the case of sequence recovery, the maximum possible sequence recovery can be estimated by analyzing the amino acids tolerated in each position in BLAST derived PSSMs. In this case, the average percentage of time that the native residue is seen in the PSSM is used as an estimate for expected sequence recovery. For the 100 protein benchmark set, the average was 34%, with a standard deviation of 12%, while in the 42 protein benchmark set, the average was 34% with a standard deviation of 7%. While the achievable sequence recovery is somewhat higher due to the correlation between individual positions, these values suggest that obtaining sequence recovery rates of 40-50% would approach the maximum. Tables II.3 and II.4 show that for the 100 protein benchmark set, total overall sequence recovery is 45.7% with the standard energy function and 47.0% with the NV environment KBP. For the 42 protein benchmark set, total overall sequence recovery is 35.7% with the standard energy function and 38.9% with the NV environment KBP. This explains the relatively small increases in sequence recovery, as current recovery values are approaching the practical maximum. For that reason we introduce the PSSM recovery metric. In this context it is important to note that the scoring functions were not directly optimized for sequence recovery but rather PSSM recovery. As a result, it is not surprising that the sequence recovery is not necessarily maximized during optimization.

II.4.4 PSSM recovery values improve substantially overall

In the case of PSSM recovery, it is reasonable to expect that 100% PSSM recovery is unreachable as evolution might not have sampled all amino acids tolerated in a sequence position. A more realistic value for maximum possible PSSM recovery is between 80% and 90%, though the exact value of this upper bound is difficult to estimate. PSSM recovery with the standard energy function was 72.0%. The observed increase to 77.2% with the NV environment KBP represents a substantial increase relative to the 80-90% maximum and the 72% starting point. Generally, improvements in sequence recovery rates have been moderate when altering the energy function (Kortemme et al., 2003), as the major contributors to the overall energy are already fine-tuned and remain unaltered.

Comparison of PSSM and sequence recovery results between the 42 protein benchmark set and the 100 protein set illustrates that the performance of the RosettaDesign algorithm varies based on the characteristics of the protein being designed. For example, Table II.3 and II.4 show the overall sequence recovery for proteins designed with the standard energy function. The overall recovery for the 42 protein benchmark set was 35.7%, while the overall recovery for the 100 protein benchmark set was 45.7%. This substantial difference is likely a result of the different criteria used to select the proteins in each set. The proteins in the 42 protein benchmark set are larger than those in the 100 protein set, and will therefore have a larger total surface area and thus be more challenging targets for design.

II.4.5 The sequence recovery values observed are realistic, given previous values reported in the literature

Despite the differences inherent to different design targets, these values are similar to those obtained in the literature. Schneider et al. designed a set of proteins of size between 89-223 amino acids based on high resolution crystal structures. They observed surface sequence recovery rates of 22% ± 11%, and buried recovery rates of 56% ± 13.7% when designing with RosettaDesign (Schneider et al., 2009). These values are similar to those seen in Tables II.3 and II.4. Additionally, Sharabi et al. reported overall sequence recovery values of between 40% and 70% depending on the weights of the scoring function used during their design (Sharabi et al., 2011). These numbers are within the range of the sequence recovery values obtained during the experiments described in this manuscript.

II.4.6 NV environment KBP term reduces sequence bias

In addition to improvements in PSSM and sequence recovery, the degree of sequence bias seen in the buried and boundary regions of designs made using the NV environment KBP decreased. When all residues in the benchmark set are considered, proteins designed with the NV environment KBP have an RMS percent composition change of 2.8% compared to the native protein, while proteins designed with the standard energy function have an RMS percent composition change of 2.9% (Figure II.5A). When this overall value is broken down by region, the buried region designed with the NV environment KBP shows an increase in RMS percent composition change compared to the standard energy function from 4.2% to 4.5%, the boundary region shows a decrease from 3.4% to 2.7%, and the

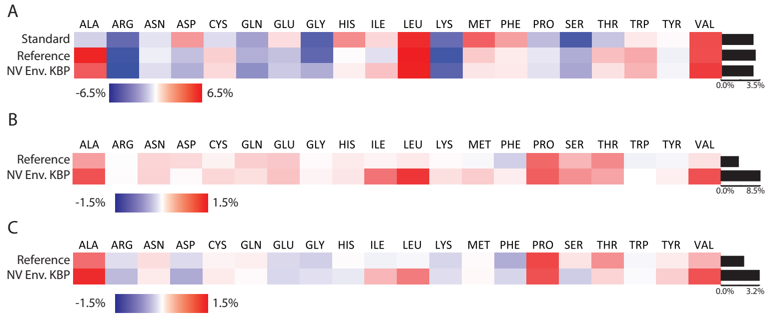


Figure II.5: A) shows the percent change in overall sequence composition between native and designed proteins for all 42 structures in the independent benchmark set. The black bars show the RMS percent composition change. B) shows the percent PSSM recovery for all 42 structures in the independent benchmark set. The black bars show RMS percent PSSM recovery. C) percent sequence recovery for all 42 structures in the independent benchmark set. The black bars show RMS percent sequence recovery.

surface region shows a reduction from 2.9% to 2.4%. While the improvements in sequence composition bias are minimal, Figure II.5B shows an increase in RMS per residue PSSM recovery from 3.8% with the standard energy function to 4.2% with the NV environment KBP. Additionally, Figure II.5C shows an increase in RMS per residue sequence recovery from 2.4% with the standard energy function to 2.6% with the NV environment KBP energy function, which is expected given the optimization of the scoring function towards PSSM improvement.

II.4.7 Addition of the NV environment KBP term results in reduction of solvation term weight

An investigation of the optimized weights lends some insight into the cause of the improvements in sequence design. Table II.1 and II.2 show the scoring function and reference energy weights of the standard energy function and the optimized NV environment KBP. When the NV environment KBP term is added to the energy function, the weight of the free energy solvation potential decreases from 0.65 in the standard energy function to 0.56 in the NV environment KBP. The NV environment KBP term has a value of 1.01. As discussed earlier, in the standard energy function, the reference energies and solvation free

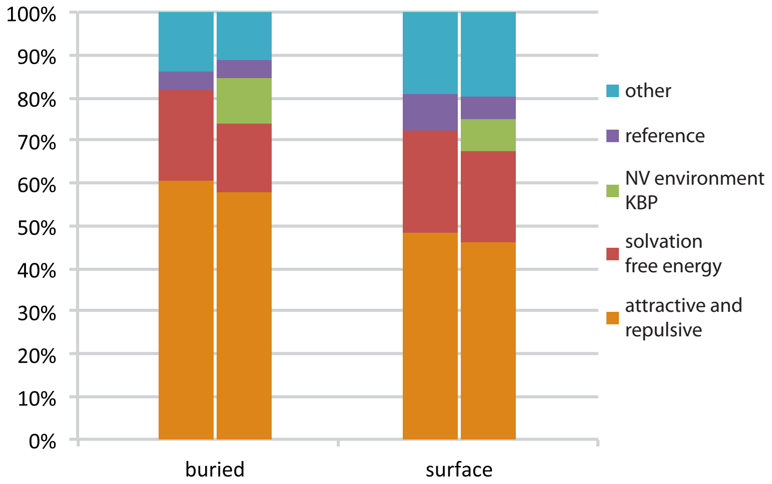


Figure II.6: The contribution of individual scoring terms towards the overall score of buried and surface residues. The introduction of the NV environment KBP reduces the reliance on solvation free energy and the attractive/repulsive forces at both levels of exposure.

energy potential are the dominant forces on surface residues due to the lack of explicit inter-residue interactions. Because the penalty given by the solvation free energy potential for apolar residues on the surface is relatively weak, the weight of this potential will need to be increased for it to adequately effect surface residues. However, because the energy function is applied evenly, regardless of degree of burial, the increase in weight necessary to maintain a reasonable protein surface may cause the solvation free energy potential to apply too strongly to the boundary region. As the burial level increases, the number of inter-residue interactions will also increase, which explains the decrease in improvement in sequence bias seen at more highly buried regions of the protein. This idea is supported by the decrease in free energy solvation potential weight observed in the NV environment KBP energy function. The NV environment KBP provides additional information about protein surface composition, reducing the dominance of the free energy solvation potential.

II.4.8 The NV environment KBP term reduces the influence of the solvation term on total score

Figure II.6 shows the effect of the NV environment KBP on the overall scoring function. All proteins used in the 100 protein benchmark set were scored using both the standard RosettaDesign energy function and the optimized NV environment KBP energy function. The average magnitude of each scoring term for each buried and surface residue was calculated, and converted to percentage of the total energy for each residue to measure the influence of each scoring term. We observe that the addition of the NV environment KBP term decreases the influence of the reference energies, solvation free energy term, and the attractive and repulsive terms throughout all degrees of burial. Specifically, the influence of the solvation free energy decreases from 21% to 16% for buried residues, and from 24% to 21% for surface residues. Additionally, the influence of the reference energy decreases from 8% to 5% on the surface, though it remains relatively unchanged for buried residues. The attractive and repulsive forces also change somewhat, with a decrease in influence from 60% to 57% in buried residues, and 48 to 46% in surface residues. This change in influence is significantly less than the change in influence seen in the reference and solvation free energy functions. The NV environment KBP was designed to address shortcomings in the design of the protein surface. These shortcomings are the result of the energy function failing to model aspects of the protein surface that are not completely described through the solvation and reference energies. To achieve reasonably good performance despite these inaccuracies, both energy terms are overweighted in the standard energy function. As expected, addition of the NV environment KBP term reduces the impact of solvation and reference energies on the surface. As these adjustments apply throughout all degrees of burial, the artificially inflated weight of the solvation and reference energies can be decreased, improving performance also in the buried regions of the protein as well.

II.4.9 The NV environment KBP term may encode environmental effects reducing aggregation potential

In addition to providing information about solvation effects, the NV environment potential also sheds light on the evolutionary and environmental forces on protein composition. Soluble proteins have evolved to be non-aggregative and generally stable in the environment of a cell. These properties are difficult to model via physics-based methods, as they arise from numerous inter-protein interactions that are difficult to explicitly model. The implicit modeling of these environmental effects accounts in part for the improvements in native-like sequence design seen during design with the NV environment KBP. By optimizing the NV environment KBP energy function to maximize PSSM score rather than sequence recovery, the energy function is optimized to design proteins similar to those which are favored evolutionarily, rather than to merely reproduce the native sequence.

CHAPTER III

Improving RosettaLigand Speed and Sampling Efficiency through the Development of a Novel Sampling Algorithm

III.1 Abstract

RosettaLigand has been successfully used to predict protein-ligand binding positions in a number of cases (Turlington et al., 2013; Davis et al., 2009; Combs et al., 2013). However, the RosettaLigand docking protocol is relatively inefficient at sampling the ligand binding site space, making it unfeasibly slow for use as a vHTS tool. We show here that the development of a new sampling algorithm for initially placing the ligand in the protein binding site dramatically improves both the overall success rate of small molecule docking as well as speed of RosettaLigand. The new algorithm improves the docking success rate by 10-15% in a 43 protein benchmarking set, reduces the average time to generate a model from 50 seconds to 10 seconds, and reduces the necessary number of models to generate from 1000 to 150 resulting in an effective 10-fold speed increase. We also demonstrate that accurate initial placement of the ligand prior to full atom refinement is critical to successful prediction of an accurate binding position.

III.2 Introduction

III.2.1 Ligand docking background

Computational ligand docking has been a historically successful method for predicting the binding position of small molecules to a protein. Beginning with PJ Goodford's work in computational drug design (Goodford, 1985), many methods have been developed to predict the interactions between proteins and small molecules. Early tools focused primarily on rigid body goodness of fit between a small molecule and a protein crystal structure. However, further study of the changes observed in protein conformation upon the binding of a small molecule (Bystroff and Kraut, 1991) suggested that modeling of protein and ligand flexibility was important to correctly model protein-ligand interactions.

III.2.1.1 An overview of popular ligand docking tools

Over the past several decades, numerous tools have been developed to attempt to better address the ligand docking problem. DOCK (Ewing et al., 2001), FlexX (Hindle et al., 2002), AutoDock (Morris et al., 1998), and Glide (Friesner et al., 2004) are currently among the most popular tools. They utilize a wide range of protein representations, sampling algorithms and scoring functions in order to accurately predict protein-ligand binding positions. Approximations in scoring and sampling must be made in order to allow ligand binding predictions to be made in a reasonable time. To accomplish this, most ligand docking tools operate in stages, so that the size of the search space is limited as the complexity of the scoring function and sampling density within the search space is increased.

III.2.1.2 Summary of popularly used docking algorithms

Docking methods differ in their means of accomplishing this step-wise increase in sampling resolution coupled with the reduction of search space. For example, the DOCK algorithm creates a "negative space" model of the binding site created by placing spheres inside the solvent accessible area of the binding site, and uses this model to guide docking of the ligand, while an Assisted Model Building with Energy Refinement (AMBER) based molecular mechanics force field is used to score the resulting binding positions (Moustakas et al., 2006). FlexX, on the other hand, represents the protein by "interaction centers" consisting of surfaces surrounding common ligand interaction groups (hydrogen bond donors and acceptors, metals, aromatic rings, etc.). Atoms in a based fragment of the ligand are then matched to the interaction centers to provide an ensemble of potential initial placements (Rarey et al., 1996). AutoDock represents the receptor using a cartesian scoring grid populated with information from an empirically derived energy function. A Lamarckian Genetic Algorithm (LGA) in combination with simulated annealing is then used to optimize both the ligand conformation and position (Morris et al., 1998). Glide uses a grid based representation of the protein binding site. A rapid exhaustive search is first performed to find generally favorable areas for ligand placement. A size filter is then used to exclude areas without sufficient space for ligand placement. Finally, MCM of the binding position using the grid based scoring function is performed. The scoring girds themselves are generated using a scoring function derived from ChemScore (Friesner et al., 2004).

III.2.2 Algorithm details

III.2.2.1 Performance of ligand docking tools is inconsistent

Despite the large differences between scoring and sampling algorithm implementations across the different ligand docking tools, a blind study of ligand docking performance conducted by Davis et al. (Davis et al., 2009) suggested that while certain methods of docking perform better than others for a given protein target, in the aggregate the commonly used systems have a similar range of performance. Interestingly, while some protein systems appear to be relatively successful (Chk1 kinase) or difficult (Hepatitis C RNA Polymerase) for most ligand docking tools to predict, the results for most systems vary depending on the ligand docking tool. The difficulty of predicting whether a given protein-ligand docking system will be easily docked by a given docking tool has been previously established (Mysinger et al., 2012; Bauer et al., 2013), and there is a clear need for more reliable protein-ligand docking tools.

III.2.3 Limitations of RosettaLigand low resolution docking

III.2.3.1 Description of the binary scoring grid and Translate step

The work presented in this manuscript consists of a set of improvements to the previously implemented RosettaLigand docking algorithm. As previously implemented, RosettaLigand used a two stage docking process consisting of an initial placement stage followed by a refinement stage. The overall effect of the initial placement algorithm is to place the ligand in a non-clashing position at random. The initial placement algorithm consists of three steps, which were initially described in Davis et al. (Davis and Baker, 2009) The algorithm uses a scoring grid to identify non-clashing regions of the protein. Two sets of scoring grids were evaluated. The binary scoring grid consists of "attractive" rings between 2.25 and 4.75 Å of every heavy atom, and "repulsive" spheres between 0 and 2.25 Å of each backbone heavy atom.

The first step of the initial placement algorithm ("Translate") consists of up to 50 random translations within 5.0 Å of the starting position. After each translation, the heavy atom closest to the geometric center of the ligand, termed the "neighbor atom", is scored using the binary scoring grid. If the score is -1 or 0 (attractive or neutral) the move is accepted and the translation step terminates. The aim of the Translate step is to place the ligand in a region of the binding site that does not result in a complete clash with the protein.

III.2.3.2 Description of the Rotate step

The second step in the previously implemented RosettaLigand initial placement algorithm is the Rotate step. The Rotate step consists of up to 500 random rotations with a maximum of 360° from the starting orientation. The Rotate step accumulates a set of diverse non-clashing ligand orientations, and then selects one of these orientations at random for further refinement. The size of the set of diverse orientations is either 5 or 5 times the number of rotatable bonds in the ligand, whichever is larger. The ligand is randomly reoriented and then accepted into the set of diverse orientations if the following conditions are met: No atoms are located in repulsive squares, 85% of the atoms are located on attractive squares, and the RMSD of the new orientation with respect to all previously accepted conformations is greater than $0.65\sqrt{number\ of\ heavy\ atoms}$. After either 500 orientations have been created or the maximum set size has been achieved, a random orientation from the set is selected, and the Rotate step terminates.

III.2.3.3 Description of the Slide Together step

The third and final step in the previously implemented RosettaLigand initial placement algorithm is the Slide Together step. Due to the relatively small amount of information provided by the binary scoring grid, it is possible for the ligand to be placed in a region where it does not contact any protein atoms at the end of the translate and rotate steps. In this case the apparent interaction energy at the beginning of the refinement stage would be 0, reducing the efficiency and sometimes causing failure in the following Monte Carlo refinement stage. To avoid this situation, the ligand must be brought in contact with the protein. The Slide Together step moves the ligand towards the center of mass of the protein until the full atom repulsive score increases. Following this initial placement, a refinement stage is carried out in which small perturbations of the ligand and repacking of the protein side-chains are performed using MCM. Finally, all atoms in the binding site are minimized using gradient minimization, and the final structure is scored.

III.2.3.4 Possible limitations of the low resolution placement in RosettaLigand

We hypothesize that independent sampling of translation and rotation will complicate sampling of all favorable initial placements, particularly if the ligand is not globular. For example, a rod-shaped ligand would easily enter a rod-shaped pocket but only if it is brought into the correct orientation first. A ligand with a bent shape might require reorientation while entering the binding pocket in order to avoid clashes. Therefore, RosettaLigand will miss out on favorable initial placements for other ligands but spend substantial time performing refinement and minimization moves on ligands placed in unfavorable initial positions. Figure III.1 schematically illustrates this hypothesis. The Translate step described in Section III.2.3.1 only takes into account the geometric center of the ligand. As a result, a ligand with a narrow binding pocket is likely to be initially translated unfavorably (Figure III.1B). Once the ligand has been translated into an undesirable locations the rotational sampling (Figure III.1C) has no possibility of arriving at a high quality binding position, and the mistake in translation can only be corrected by completing refinement and beginning a new binding position. The result of this inefficiency would be an increased failure rate as some ligands are never placed in favorable starting positions, and for other ligands an effectively

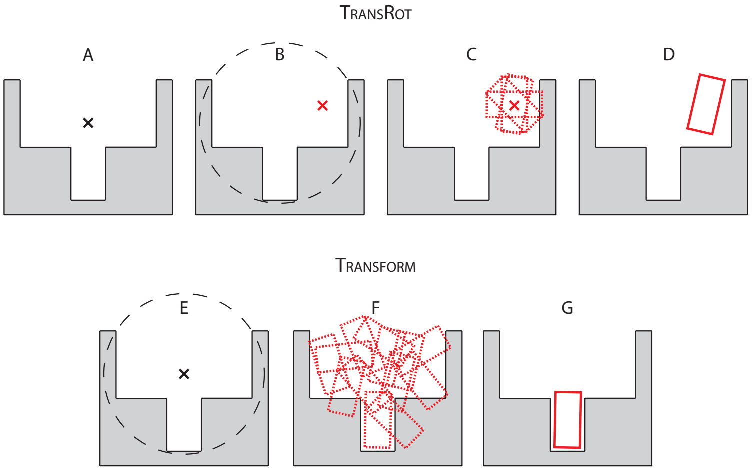


Figure III.1: A schematic indicating the hypothetical mechanism by which the TRANSFORM algorithm exhibits improved performance compared to the TRANSROT algorithm. A) When the TRANSROT algorithm is used, a cartesian starting coordinate is specified as the starting position for the ligand. B) This starting point is then translated to a random location which does not overlap with the protein backbone. C) The ligand is centered at the new random location within a user specified starting radius, and a set of diverse, minimally-clashing rotational binding positions are selected. D) A single random binding position is selected for refinement. E) When the TRANSFORM algorithm is used, the starting cartesian coordinate is specified as the starting position for the ligand. F) The simultaneous translations and rotations within a user specified radius is sampled using a MCM algorithm. G) The best scoring model is selected from step (F) for refinement.

increased runtime as the number of ligand binding positions which must be generated to reliably produce a high quality binding position is increased. Lemmon et al. (Lemmon and Meiler, 2013) determined that as many as 1000 models may be necessary to produce at least one high quality binding position in a challenging docking case. Given this, improving the efficiency of protein binding site sampling by starting from more favorable initial placements has the potential to drastically reduce the computational cost of RosettaLigand, allowing for a larger number of predictions to be made given a fixed amount of computing resources. The new TRANSFORM algorithm samples both translation and rotation simultaneously (Figure III.1F), and increases the likelihood of arriving at a reasonable binding conformation prior to refinement relative to the separate translation and rotation steps of the previously published TRANSROT algorithm.

III.3 Results

The improved initial placement algorithm, here referred to as the TRANSFORM algorithm has two components: A modular grid based scoring function and a MCM based sampling algorithm. Both components are fully independent. The implementation described here allows for the rapid implementation of new score terms and sampling methodologies, and the easy integration of these methods into the existing RosettaLigand pipeline.

III.3.1 Score function development

III.3.1.1 Description of scoring grids and manager

Scoring of ligand binding positions in the new TRANSFORM algorithm is handled using a set of scoring grids which are controlled by a scoring manager (Figure III.2). Each scoring grid is responsible for computing a single term in the energy function. In this study, a single scoring grid, identical to that used by the TRANSROT algorithm, and described in Section III.2.3.1, was used for scoring. The scoring manager consists of a 3D tensor of floating point values representing cartesian space and software functions to populate the tensor, and score ligands positioned in it. The scoring manager is responsible for keeping every scoring grid up to date with respect to the protein binding position, and for making sure that the ligand is scored in every grid. Additionally, the scoring manager is responsible for handling the weighting of the individual scoring terms to compute the total score. For this study, the tensor is a 30 Å^3 cube, with a spacing of 0.25 Å between grid points. While the size and density of the grid does not need to be rigorously optimized, there are some guidelines for setting the parameters. The size of the grid must be large enough to accommodate the perturbation of the ligand, that is, if the ligand is translated as far as possible within the docking algorithm settings (5.0 Å in this study) the grid must be large enough for every ligand atom to exist within the grid. RosettaLigand will reject any move that results in ligand atoms placed outside the grid, so a grid which is too small will artificially constrain ligand sampling. On the other hand, the amount of memory required to store a scoring grid

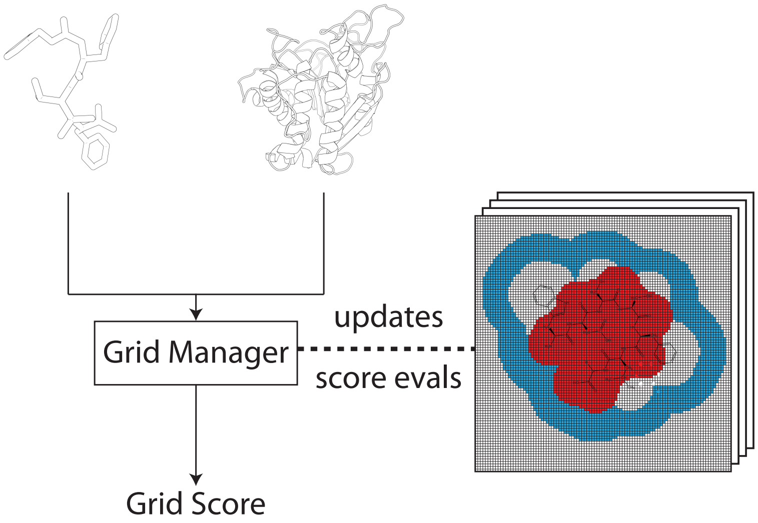


Figure III.2: A schematic showing the architecture of the scoring grid manager. The grid manager takes as input protein and ligand models and computes a score based on these scoring grids. Additionally, the grid manager is responsible for generating and updating the information encoded in the scoring grids.

increases with the cube of the grid side length. Smaller scoring grids can be handled more efficiently by the CPU, so making the grid too large may result in a substantial decrease in algorithm speed. Similarly, the spacing between grid points must be small enough to capture the differences between nearby atoms, but not so small that the grid is too large to be efficiently handled. The overall guideline then is that a scoring grid should be large enough to encompass the entire protein-ligand binding site, but no larger.

III.3.2 Sampling architecture

III.3.2.1 Description of grid MCM

An MCM algorithm is used to compute the initial binding position for the ligand. Figure III.3 shows a flow chart of the overall steps in the sampling process. At each step in the sampling process the ligand is either randomly perturbed in the binding site, or the conformation of the ligand is changed. Ligand perturbation is performed as a combination of a random translation and rotation, and the conformation of the ligand is perturbed by selecting a random conformation from a library of pre-computed conformers. After

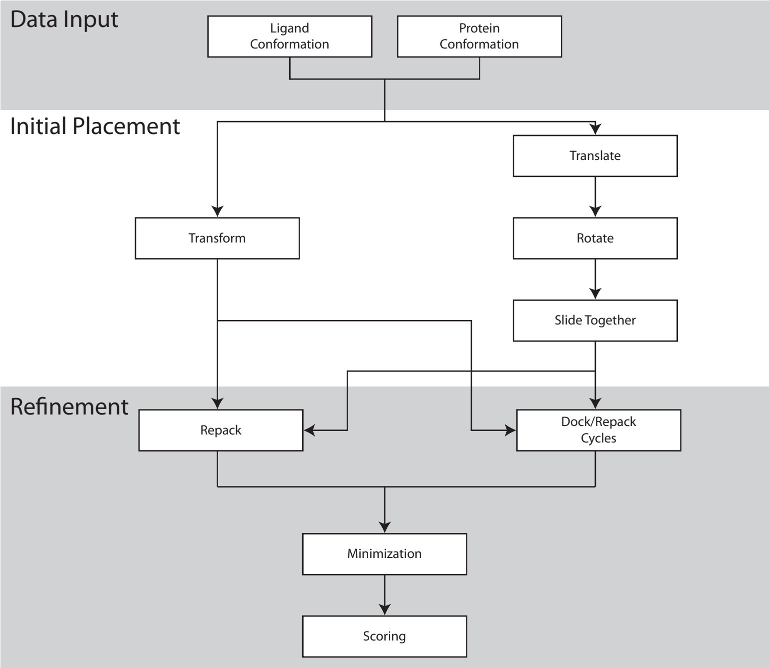


Figure III.3: A general schematic of the docking protocols described in this paper. Because the initial placement and refinement steps are independent, the two initial placement algorithms can be alternatively selected to produce a total of four ligand docking algorithms.

the perturbation, the ligand is scored using the scoring grids described above, and, the Metropolis criterion is applied to either accept or reject the new binding position. After 500 cycles of sampling are performed, the best scoring ligand binding position is saved. During sampling, only the scoring grids are used to provide scoring information, and the protein is therefore rigid. By only using scoring grid information, it is possible to perform 500 cycles of sampling in roughly 1-3 seconds.

III.3.3 Docking Protocol

III.3.3.1 Introduction to the docking process

The overall docking protocol is illustrated in the schematic flowchart in Figure III.3. The study compares different configurations of both the initial placement step and the refinement step, described below. Complete RosettaScripts eXtensible Markup Language (XML) files for each experiment can be found in Chapter E. As a baseline we use the TRANSROT initial placement algorithm, in which translation and rotation moves are performed separately using the binary scoring grids summarized in Section III.2.3.1. The specific TransRot protocol used here is originally described in Fleishman et al. (Fleishman et al., 2011), and is functionally identical to the process described in the Davis paper, though the user interface is different.

III.3.3.2 Description of the TRANSFORM based initial placement algorithm

In the new TRANSFORM based initial placement algorithm, translation and rotation are performed simultaneously. Five hundred Monte Carlo (MC) steps are carried out. At each step, the ligand position is either transformed or the ligand conformation is changed using the ligand conformers described in Section III.6.2.1. When a transformation move is selected, the ligand is randomly translated within 0.1 Å of its current position, and rotated within 20°. The degree of rotation and translation selected is based on a random gaussian around the current position. The ligand is constrained to only move within 5 Å of the starting position. After each move, the score is evaluated as the sum of the values at each grid square occupied by a ligand atom. The move is then accepted or rejected using the Metropolis criterion, and the best scoring accepted move after all 500 steps is returned.

III.3.3.3 Description of the MCM refinement algorithm

During MCM refinement, the full atom Rosetta energy function is used for energy computation rather than the scoring grids. MCM refinement consists of two steps: high resolution docking and gradient-based minimization. Six steps of high resolution docking are performed. Steps 1, 3 and 6 consist of repacking followed by minimization, while steps 2 and 5 consist of small perturbations of the ligand. In the repacking and minimization step, the side-chain positions are optimized using side-chain rotamers from the Dunbrack rotamer library (Shapovalov and Dunbrack, 2011), and the ligand is allowed to change conformation using the pre-computed ligand conformers. Following repacking, a gradient based minimization is applied to minimize the energy of the side-chain and ligand atoms. In the perturbation step, the ligand is randomly perturbed within a range of 0.1 Å and rotated within a range of 20° per move. These perturbation and rotation ranges have not been rigorously optimized but are sufficient to provide full coverage of the rotational and translational space. The two moves are alternated such that the first, third and final moves are repacking and minimization, while the remaining are ligand perturbations. The six moves are performed using a MCM algorithm, and the best scoring binding position of the six moves is selected. After the high resolution docking step, a final minimization is carried out in which the protein side-chain and backbone atoms in the binding site, as well as the ligand atoms, are all minimized using a gradient based minimization algorithm.

III.3.3.4 Description of MIN refinement

Minimization (MIN) refinement is carried out similarly to MCM refinement, except that in the case of MIN refinement only a single round of repacking is performed prior to the final minimization. Because no ligand perturbation is performed during MIN refinement, the ligand binding position generated during the initial placement stage is more important to the final binding position and score of the ligand.

III.3.4 Benchmarking setup

III.3.4.1 Overview of the CSAR benchmarking scheme

To benchmark the performance of the new initial placement algorithm, a docking benchmark derived from the Community Structure-Activity Resource (CSAR) (Dunbar Jr. et al., 2011) dataset was used. A subset of 43 proteins from the CSAR data set was used (table III.1). This subset omits protein/ligand complexes with co-factors, metal ions, or water molecules that bridge ligand and protein. While Rosetta has successfully been used in such cases (Lemmon and Meiler, 2013), the inclusion of critical waters, co-factors or metal ions greatly increases the number of degrees of freedom in the docking simulation, which in turn would make the results of benchmarking more complex to interpret.

PDB ID	Ligand ID	Ligand Formula	Protein name	Ligand Name	$log(K_i)$	Weight	AA Count	Citation
2q89	6CS	$C_6H_{10}N_2O_3$	putative abc transporter amino acid-binding protein	(4s,5s)-5-hydroxy-2-methyl-1,4,5,6-tetrahydropyrimidine-4-carboxylic acid	6.3	158.155	257	(Hanekop et al., 2007)
3bgz	VX3	$C_{21}H_{15}NO_2$	proto-oncogene serine/threonine-protein kinase pim-1	2,3-diphenyl-1h-indole-7-carboxylic acid	6.26	313.349	333	(Pierce et al., 2008)
1ukb	BEZ	$C_7H_6O_2$	2-hydroxy-6-oxo-7-methylocta-2,4-dienoate hydrolase	benzoic acid	3.19	122.121	282	(Fushinobu et al., 2005)
1vot	HUP	$C_{15}H_{18}N_2O$	acetylcholinesterase	huperzine a	6.6	242.316	537	(Raves et al., 1997)
2p7g	2OH	$C_{15}H_{16}O_2$	estrogen-related receptor gamma	4,4'-propane-2,2-diyldiphenol	6.53	228.286	251	(Abad et al., 2008)
2rca	GLY	$C_2H_5NO_2$	glutamate [nmda] receptor subunit 3b	Glycine	7.79	75.066	584	(Yao et al., 2008)
2vuk	P83	$C_{16}H_{18}N_2$	cellular tumor antigen p53	1-(9-ethyl-9h-carbazol-3-yl)-n-methylmethanamine	3.82	238.328	438	(Boeckler et al., 2008)
2nta	521	$C_{12}H_9ClN_2O_3S_2$	tyrosine-protein phosphatase non-receptor type 1	5-(4-chloro-5-phenyl-3-thienyl)-1,2,5-thiadiazolidin-3-one 1,1-dioxide	4.8	328.794	299	(Wan et al., 2007)
2are	MAN	$C_6H_{12}O_6$	lectin	alpha-d-mannose	3.28	180.156	504	(Buts et al., 2006)
2z4b	DC8	$C_{18}H_{16}F_2O_3$	estrogen receptor beta	(3as,4r,9br)-2,2-difluoro-4-(4-hydroxyphenyl)-1,2,3,3a,4,9b-hexahydrocyclopenta[c]chromen-8-ol	9.36	318.315	514	(Richardson et al., 2007)
2c94	TSF	$C_{15}H_{23}F_2N_4O_{10}P$	6,7-dimethyl-8-ribityllumazine synthase	3-(1,3,7-trihydro-9-d-ribityl-2,6,8-purinetrione-7-yl) 1,1 difluoropentane-1-phosphate	6.82	488.334	800	(Morgunova et al., 2006)
2b3f	GAL	$C_6H_{12}O_6$	glucose-binding protein	beta-d-galactose	6.03	180.156	2400	(Cuneo et al., 2006)
2bbf	344	$C_9H_7N_5O$	trna guanine transglycosylase	6-amino-3,7-dihydro-imidazo[4,5-g]quinazolin-8-one	5.1	201.185	386	(Stengl et al., 2007)
2pwg	CTS	$C_8H_{15}NO_4$	sucrose isomerase	castanospermine	4.82	189.209	1112	(Ravaud et al., 2007)
2otz	1MR	C_7H_9N	lysozyme	n-methylaniline	3.63	107.153	162	(Mobley et al., 2007)
2ou0	MR3	C_5H_7N	lysozyme	1-methyl-1h-pyrrole	3.23	81.116	162	(Mobley et al., 2007)
2pzv	IPH	C_6H_6O	steroid delta-isomerase	phenol	3.87	94.111	524	(Kraut et al., 2006)
2q88	4CS	$C_6H_{10}N_2O_2$	putative abc transporter amino acid-binding protein	(4s)-2-methyl-1,4,5,6-tetrahydropyrimidine-4-carboxylic acid	5.8	142.156	257	(Hanekop et al., 2007)
1ui0	URA	$C_4H_4N_2O_2$	uracil-dna glycosylase	uracil	7.06	112.087	205	(Hoseki et al., 2003)
1uz1	IFL	$C_6H_{11}NO_4$	beta-glucosidase a	(3s,4r,5r)-3,4-dihydroxy-5-(hydroxymethyl)piperidin-2-one	6.89	161.156	936	(Vincent et al., 2004)
1uz4	IFL	$C_6H_{11}NO_4$	man5a	(3s,4r,5r)-3,4-dihydroxy-5-(hydroxymethyl)piperidin-2-one	3.4	161.156	440	(Vincent et al., 2004)
1v0l	XIF-XYP	$C_{10}H_{21}NO_7$	endo-1,4-beta-xylanase a	piperidine-3,4-diol-beta-d-xylopyranose	6.32	267.276	313	(Gloster et al., 2004)
1ws4	AMG	$C_7H_{14}O_6$	agglutinin alpha chain	alpha-methyl-d-galactoside	3	194.182	612	(Arockia Jeyaprakash et al., 2005)
1y20	1AC	$C_4H_7NO_2$	glutamate [nmda] receptor subunit zeta 1	1-aminocyclopropanecarboxylic acid	5.32	101.104	292	(Inanobe et al., 2005)
2fai	459	$C_{17}H_{24}O_3$	estrogen receptor	4-[(1s,2s,5s,9r)-5-(hydroxymethyl)-8,9-dimethyl-3-oxabicyclo[3.3.1]non-7-en-2-yl]phenol	6.24	276.371	540	(Hsieh et al., 2006)
2fqw	NOS	$C_{10}H_{12}N_4O_5$	membrane lipoprotein tmpc	inosine	6.68	268.226	318	(Deka et al., 2006)
2j78	GOX	$C_6H_{12}N_2O_5$	beta-glucosidase a	(2s,3s,4r,5r)-6-(hydroxyamino)-2-(hydroxymethyl)-2,3,4,5-tetrahydropyridine-3,4,5-triol	6.42	192.17	936	(Gloster et al., 2007)
1bky	1MC	$C_5H_7N_3O$	vp39	1-methylcytosine	3.84	125.129	307	(Hu et al., 1999)
1q4w	DQU	$C_8H_8N_4O$	queuine trna-ribosyltransferase	2,6-diamino-3h-quinazolin-4-one	6.46	176.175	386	(Brenk et al., 2004)
1fcx	184	$C_{26}H_{28}O_3$	retinoic acid receptor gamma-1	6-[hydroxy-(5,5,8,8-tetramethyl-5,6,7,8-tetrahydro-naphtalen-2-yl)-methyl]-naphtalene-2-carboxylic acid	7.12	388.499	235	(Klaholz et al., 2000)
1fh8	XYP-XIF	$C_{10}H_{21}NO_7$	beta-1,4-xylanase	beta-d-xylopyranose-piperidine-3,4-diol	6.89	267.276	312	(Notenboom et al., 2000)
1y93	HAE	$C_2H_5NO_2$	macrophage metalloelastase	acetohydroxamic acid	2.1	75.067	159	(Bertini et al., 2005)
1lhw	ESM	$C_{19}H_{26}O_3$	sex hormone-binding globulin	1,3,5(10)-estratrien-2,3,17-beta-triol 2-methyl ether	8.16	302.408	189	(Avvakumov et al., 2002)
1fh9	XYP-LOX	$C_{10}H_{20}N_2O_9$	beta-1,4-xylanase	beta-d-xylopyranose-3,4,5-trihydoxy-piperidine-2-one-oxime	6.43	312.274	312	(Notenboom et al., 2000)
1s9t	QUS	$C_5H_7N_3O_5$	glutamate receptor, ionotropic kainate 2	(s)-2-amino-3-(3,5-dioxo-[1,2,4]oxadiazolidin-2-yl)-propionic acid	6.6	189.126	518	(Mayer, 2005)
1fhd	XYP-XIM	$C_{12}H_{20}N_2O_8$	beta-1,4-xylanase	beta-d-xylopyranose-5,6,7,8-tetrahydro-imidazo[1,2-a]pyridine-6,7,8-triol	6.82	320.296	312	(Notenboom et al., 2000)
1lnm	DTX	$C_{23}H_{34}O_4$	diga16	digitoxigenin	8.7	374.514	184	(Korndörfer et al., 2003)
1nli	ADE	$C_5H_5N_5$	ribosome-inactivating protein alpha-trichosanthin	adenine	3.59	135.127	248	(Shaw et al., 2003)
1ow4	2AN	$C_{16}H_{13}NO_3S$	pheromone binding protein	8-anilino-1-naphthalene sulfonate	5.68	299.344	258	(Lartigue, 2003)
1r5y	DQU	$C_8H_8N_4O$	queuine trna-ribosyltransferase	2,6-diamino-3h-quinazolin-4-one	6.46	176.175	386	(Lartigue, 2003)
1s38	MAQ	$C_9H_9N_3O$	trna guanine transglycosylase	2-amino-8-methylquinazolin-4(3h)-one	5.15	175.187	386	(Meyer et al., 2004)
1s39	AQO	$C_8H_7N_3O$	trna guanine transglycosylase	2-aminoquinazolin-4(3h)-one	7.7	161.161	386	(Meyer et al., 2004)
1sw1	PBE	$C_7H_{14}NO_2$	osmoprotection protein (prox)	1,1-dimethyl-prolinium	7.3	144.192	550	(Schiefner et al., 2004b)

Table III.1: A table showing the PDB IDs, gene names, protein and ligand information of the proteins in the 43 protein benchmark set derived from CSAR.

III.3.4.2 Description of the three sets of input models used in the CSAR based benchmark

Because the new initial placement algorithm relies on a pre-computed scoring grid, the initial positions of the protein atoms will have an impact on the quality of the generated binding positions. To assess the extent of this impact, three sets of input structures were used in docking: the crystal structures provided in the CSAR dataset, repacked structures in which the backbone was held fixed and the side-chains re-optimized without the co-crystallized ligand present, and relaxed structures in which both the side-chain and backbone atoms were minimized in absence of the small molecule. In the case of the crystal and repacked structures, only a single protein structure was used for docking. In the case of the relaxed structures, the ligand was docked into an ensemble of ten models.

III.3.4.3 Twelve benchmark experiments were performed

Each experiment is a combination of one set of input protein structures above (crystal, repacked, relaxed or homology model) and one docking protocol. Four docking protocols were selected to investigate the behaviors of each component of the docking algorithms. A docking protocol consists of an initial placement algorithm (TRANSROT or TRANSFORM), and a refinement algorithm (MCM or MIN). Figure III.3 is a schematic describing the overall docking process.

III.3.5 Summary of results

III.3.5.1 The TRANSFORM algorithm decreases the amount of time required to make one model

Figure III.4 shows the change in the average time necessary to generate a single model with each of the four tested algorithms. The average time needed to generate a model using the previously published TRANSROT/MCM protocol is 49.4 seconds per model. Changing the Refinement protocol from MCM to MIN reduces the time per model to 33.3 seconds, and changing both the refinement protocol to MIN and the initial placement model from TRANsRot to Transform further reduces the time per model to 9.3 seconds. The distribution of per-model timing is not uniform, and varies based on the docking protocol used. We see that the standard deviations of the time to generate models using Transform based algorithms are lower than those of the TransRot based algorithms. Specifically, the distribution time to generate Transform/MCM models has a standard deviation of 10.51 and the standard deviation of the distribution for Transform/MIN is 3.98, On the other hand, the timing distribution of the TransRot/MCM models has a standard deviation of 26.6, and the timing distribution of the TransRot/MIN models has a standard deviation of 20.96.

In addition to a narrower timing distribution, the choice of algorithm also appears to affect the skewness of the distribution. Specifically, the timing distribution for models generated using Transform algorithm exhibit a lower skewness value and therefore a more normal distribution than models generated using the TransRot algorithm. Specifically, we see skewness values of 1.67 and 0.67 for the Transform/MCM and Transform/MIN protocols, and skewness values of 2.81 and 2.42 for the TransRot/MCM and TransRot/MIN protocols.

From this timing data we can conclude that that the majority of computational time spent by the previously published TransRot/MCM algorithm is split roughly evenly between the initial placement stage and the refinement stage. A combination of the new Transform initial placement algorithm and MIN refinement is capable of consistently generating models approximately 5-10 times faster than the previously published docking algorithm.

III.3.5.2 Use of the Transform mover improves sampling efficiency

The use of the sampling mover increases the probability of sampling low RMSD binding positions. 150 initial placement docking trajectories for each protein in the 43 protein CSAR benchmark set were generated using the Transform and TransRot initial place-

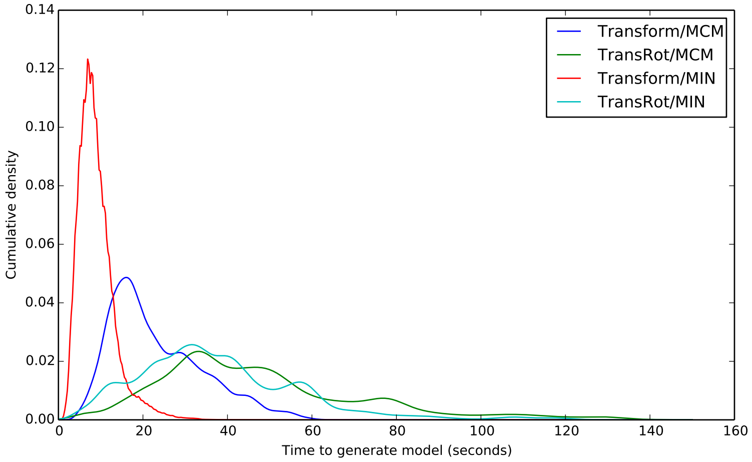


Figure III.4: Kernel Density Estimate curves showing the distribution of time necessary to generate a single model using various RosettaLigand protocols. TRANSROT/MCM is the protocol previously published by Davis et al. (Davis and Baker, 2009).

ment algorithms. After each sampled position, the RMSD to the crystallographic ligand position was computed. Figure III.5 illustrates the impact of the TRANSFORM algorithm on sampling efficiency. While the TRANSROT algorithm rarely samples models with RMSD to the crystal structure < 2.0Å, the TRANSFORM algorithm is far more frequently capable of sampling these native-like models. Specifically, the TRANSFORM algorithm samples native-like models 7.0% of the time, while the TRANSROT algorithm produces native-like structures 0.16% of the time.

III.3.5.3 Use of the MIN refinement algorithm improves consistency in run time compared to MCM refinement

Further timing consistency is seen when the MIN refinement stage is used in place of MCM refinement. Each round of repacking in MCM refinement requires that the interactions between atoms in the binding site be recomputed. As the computational complexity of this operation increases with the square of the number of atoms in the protein-ligand interface, the docking of ligands into larger binding pockets takes substantially longer when using MCM refinement compared to the docking of ligands into smaller binding pockets, which

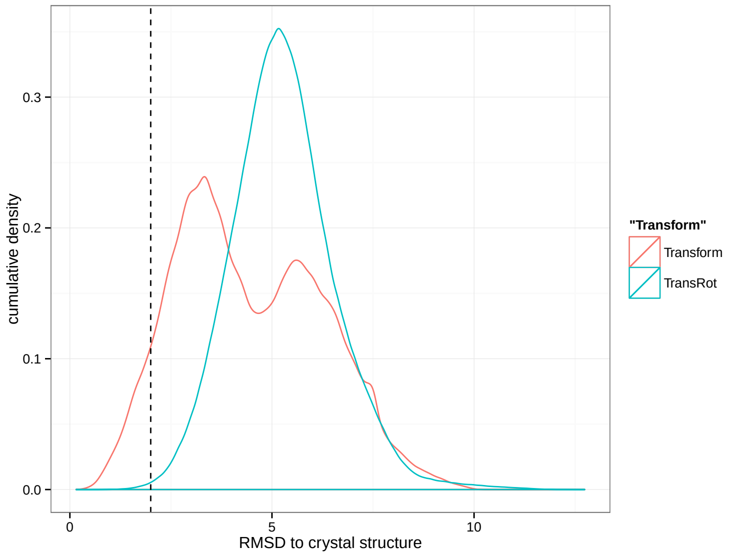


Figure III.5: A plot showing the distribution of RMSDs for ligand positions sampled using the TRANSROT and TRANSFORM initial placement algorithms. 2Å cutoff indicated with a vertical dotted line.

contributes to the observed changes in timing consistency.

III.3.5.4 The TRANSFORM algorithm improves docking success rate

Figure III.6 and Figure III.7 plot the fraction of protein-ligand systems for which the lowest scoring binding position is < 2.0 Å RMSD as a function of total Central Processing Unit (CPU) time and number of models generated, respectively. These figures indicate that the choice of the initial placement algorithm is far more important than choice of low resolution scoring method or refinement method. Docking protocols which make use of the TRANSFORM initial placement algorithm can reliably dock an additional 10-15% of models within roughly 15 minutes of CPU time, or 150 models, compared to protocols which use the previously published TRANSROT initial placement algorithm. The choice of refinement algorithm appears to play little role in the overall performance of the docking protocol, except in the case of the previously published algorithm (TRANSROT/MCM), in which case docking performance begins to approach the TRANSFORM based protocols after roughly 800-1000 models have been generated (Figure III.6). This observed behavior

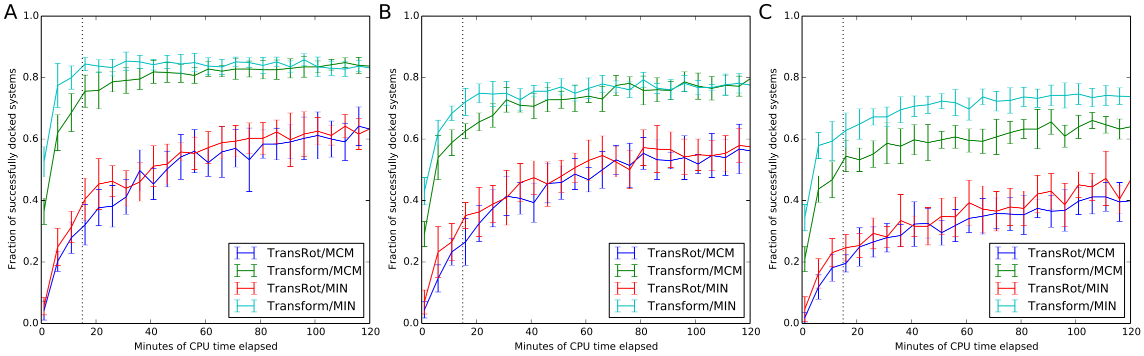


Figure III.6: The fraction of protein systems in which the lowest scoring model has an RMSD < 2.0 Å to the native structure as a function of CPU time using the 3 evaluated RosettaLigand docking algorithms when docked into A) Crystal structures, B) Repacked crystal structures, and C) Relaxed crystal structures. A large pool of models were generated, and random subsamples were taken corresponding to time points at 5 minute intervals. The number of structures included in each time point was based on the average time to generate a model for each algorithm. 20 random samples were taken for each time point, and the means are plotted, with the error bars representing the standard deviation. Docking protocols which make use of the TRANSFORM algorithm are reliably converged after approximately 15 minutes (dotted line).

is consistent with previously published studies of RosettaLigand performance using this protocol (Davis et al., 2009; Combs et al., 2013; Lemmon et al., 2012).

Figure III.8 shows illustrates the statistical significance of the apparent differences in performance between pairs of experiments illustrated in Figure III.7 and provides some insight into the relative effectiveness of the methods studied. Figure III.8 was generated using the Welches T-test to measure statistical significance. For each set of experiments plotted in Figure III.7, a Welches T-test was computed to determine the likelihood that the distribution of success rates between a pair of methods was statistically significant. The moving average of the T-Test *p*-value with a window of 5 was then plotted. A moving average was used to smooth the plot and aid in visualization. We consider *p*-values below 0.05 (indicated with a dotted line) to represent statistically significant differences in performance. From this analysis, several interesting conclusions can be drawn. There appears to be minimal difference between the performance of the TRANSFORM/MCM and TRANSFORM/MIN metrics when docking ligands into crystal structures or repacked mod-

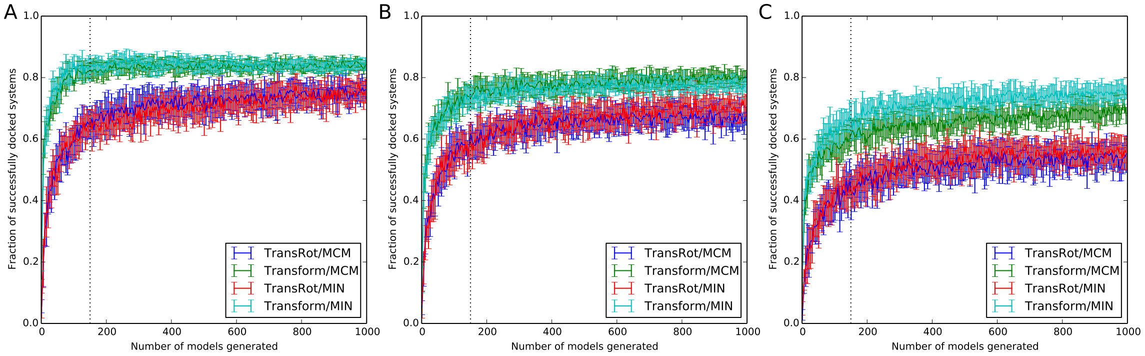


Figure III.7: The fraction of protein systems in which the lowest scoring model has an RMSD < 2.0 Å to the native structure as function of the total number of structures generated using the 3 evaluated RosettaLigand docking algorithms when docked into A) Crystal structures, B) Repacked crystal structures, and C) Relaxed crystal structures. A large pool of models were generated, and random subsamples were taken. 20 random samples were taken for each point, and the means are plotted, with the error bars representing the standard deviation. Docking protocols which make use of the Transform algorithm are reliably converged after approximately 150 models (dotted line).

els, suggesting that when the Transform sampling algorithm is used when self-docking, the impact of the refinement algorithm is minimal. The major conclusion that can be drawn from this data is that docking protocols using the Transform algorithm are significantly improved over the TransRot algorithm based protocols regardless of the number of models generated. Interestingly, there does appear to be a significant improvement in performance when TransRot/MCM is used rather than TransRot/MIN and more than 600 models are sampled. Given the very low sampling efficiency afforded by the TransRot algorithm, it is likely that this high number of models is necessary to generate an initial placement position capable of resulting in a good score after refinement. We also see that when applied to relaxed models, the Transform/MIN algorithm performs slightly better than Transform/MCM. The MCM refinement algorithm conducts alternating rounds of side-chain repacking and ligand perturbation. It is therefore possible that in the case of the relaxed protein model set, these small perturbations are adding noise, rather than improving the quality of binding prediction.

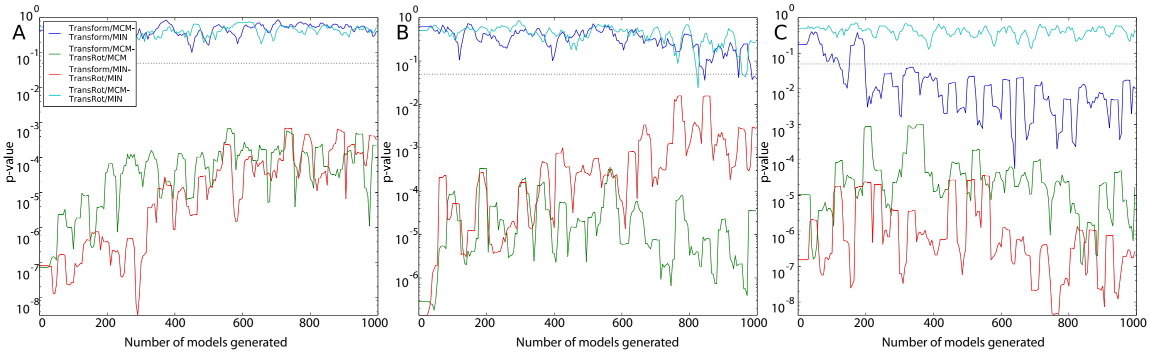


Figure III.8: A Welch's T-Test was computed comparing the success rates between pairs of protocols across a range of numbers of generated models. The T-Test values for Transform/MCM-Transform/MIN comparisons are in Blue, Transform/MCM-TransRot/MCM are in Green, Transform/MIN-TransRot/MIN are in red, and TransRot/MCM-TransRot-/MIN are in teal To reduce noise, a moving average of the T-Test *p*-value is plotted for each of the three sets of models (A) Crystal structures, B) Repacked crystal structures, and C) Relaxed crystal structures). The horizontal dotted line indicates the statistical significance threshold of 0.05.

III.3.5.5 The new Transform algorithm is still tolerant of backbone and side-chain perturbations while improving success rate

It is clear from Figure III.6 and Figure III.7 that despite using a pre-computed scoring grid during initial placement, RosettaLigand with the new initial placement algorithm is still tolerant of changes to the side-chain and backbone conformations of the protein binding site. In all tested protocols, the success rate of RosettaLigand decreases as the uncertainty associated with the protein side-chain and backbone atoms increases. In other words, after 1000 models have been generated docking ligands into crystal structures (Figure III.7A), The TransRot/MCM protocol has successfully docked 81% of models, while the Transform/MCM protocol has successfully docked 87%. When ligands are docked into relaxed models in which both backbone and side-chain atoms are perturbed (Figure III.7C), The TransRot/MCM protocol has successfully docked 60% of models, while the Transform/MCM protocol has successfully docked 75%. The reduction in success rate is expected because the addition of side-chain and backbone perturbation effectively adds noise to the protein structure. However, we see that the Transform/MCM protocol results in a 12% decrease in success rate between relaxed and crystal structures, rather than 21% for the TRANSROT/MCM protocol, so the new TRANSFORM protocol is more tolerant of inaccurate protein structures than the original protocol. Because the new initial placement algorithm is more likely to place the ligand in a high quality binding position, a greater percentage of total docking time is spent in proximity of the correct binding site and binding position. As a result, the sampling density increases and thereby the overall success rate of RosettaLigand increases relative to the TRANSROT/MCM algorithm.

III.4 Discussion

III.4.1 Explanation for time decrease

As the rotation step of the TRANSROT/MCM initial placement algorithm uses the number of rotatable bonds to determine how many rotations to perform, the amount of time required for the rotation step varies linearly with the number of rotatable bonds. Because the TRANSFORM initial placement algorithm uses an MCM algorithm with a fixed number of cycles, the time to complete a single model is more consistent compared to protocols that use the TRANSROT algorithm.

III.4.2 Details of performance optimization in the TRANSFORM algorithm

While the TRANSFORM initial placement algorithm performs roughly the same number of sampling moves during initial placement as the TRANSROT algorithm, the speed improvements seen are a result of differences in how those moves are computed. Rosetta uses a system called the “fold tree” to represent the relationships between rigid body regions of the protein system (Davis and Baker, 2009; Das and Baker, 2008). Since permutations of the protein structure made using the fold tree are performed in internal coordinate space, it is possible to rapidly modify a large system. In the case of ligand docking, however, the system being manipulated is quite small, and the computation of fold tree based permutations quickly becomes dominated by conversions between internal and cartesian coordinate space. As only the scoring grids are used for binding position evaluation during the initial placement step, the TRANSFORM algorithm can represent the ligand as a list of points, which are directly transformed using a rotation and translation matrix. This method of computing ligand permutations is substantially faster than the previous fold tree based method, and accounts for the majority of the observed speed improvement.

III.4.3 The new algorithm improves sampling efficiency and speed

Based on the results of the benchmarking studies described above, the overall effect of the new TRANSFORM sampling algorithm is two-fold. First, the quality of binding positions generated during the initial placement stage is improved, and second, the amount of time required to generate the initial placement is reduced. The improvement of the binding positions generated by the initial placement stage results in additional speed improvements by reducing the amount of sampling necessary to produce a high quality binding position. The improved sampling efficiency afforded by the new initial placement algorithm both reduces the time that must be spent in high resolution docking, and reduces the total number of models which must be created to reliably produce a correct predicted binding position.

III.4.4 The majority of performance improvement is driven by the improvements to the initial placement sampling algorithm

Figure III.9 compares the performance of several of the tested RosettaLigand protocols, and provides further insight into the impact of the various components of the protocol on overall performance. The RMSD vs. RMSD plots illustrate specific performance differences comparison between pairs of Rosetta protocols when 1000 models are generated. When the original TRANSROT initial placement algorithm is used, minimal improvement is observed when the MIN refinement algorithm is used as compared to MCM initial placement (Left). While 21 of the 43 proteins show improvement in RMSD, only 4 exhibit sufficient improvement to cross the 2.0 Å cutoff. Comparison of the TRANSROT and TRANSFORM initial placement (Center) shows substantial improvement when the TRANSFORM initial placement algorithm is used, with 24/43 proteins having improved RMSD, and 15/43 having

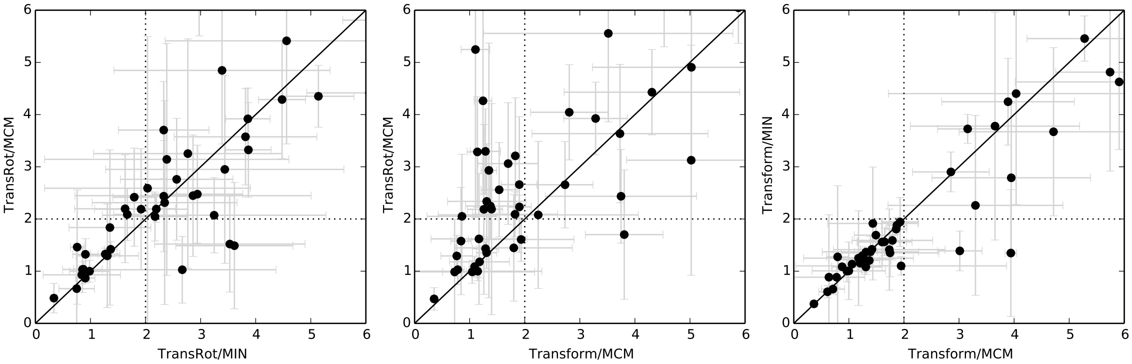


Figure III.9: RMSD vs. RMSD plots comparing the performance of various docking protocols when docking ligands into relaxed structures. 20 samples of 150 models were collected, and the average of the RMSD of the lowest scoring model is plotted for each protein-ligand system. The standard deviation of these 20 samples is shown with error bars. Dotted lines indicate the 2.0 Å RMSD cutoff used to classify correct vs incorrect binding positions.

enough improvement to cross the 2.0 Å threshold. Comparison of the MCM and MIN refinement algorithms when the TRANSFORM initial placement algorithm is used shows that in this context the two refinement algorithms have nearly identical performance (Right). From this data we can conclude that the improvements seen are driven primarily by the new initial placement algorithm.

III.4.5 Examination of the successes and failures of RosettaLigand illustrates the impact of the TRANSFORM algorithm

Figure III.10 illustrates several examples of the successes and failures of RosettaLigand. Figure III.10A illustrates a case in which the TRANSFORM/MCM protocol successfully docks a ligand that the TRANSROT/MCM algorithm cannot dock. The ligand in question is somewhat flexible and is capable of engaging in hydrogen bonding interactions from both sides. As a result, there are likely multiple possible binding positions with relatively low Rosetta energy scores, meaning that the more efficient sampling afforded by the TRANSFORM initial placement algorithm will increase the probability of sampling a correct binding position. In certain cases, the TRANSROTinitial placement algorithm results in improved results over TRANSFORM algorithm. Figure III.10B is one such case.

In this case, the ligand is extremely small, and can, as such, be placed in a number of similar positions with varying RMSDs. We have seen from previous studies(Combs et al., 2013) that it is difficult for the Rosetta energy function to distinguish between accurate and inaccurate binding positions of very small ligands. It is likely that in this case the TRANSROTinitial placement algorithm arrived at a nearly native binding position by chance, while the TRANSFORM algorithm did not. Figure III.10C is a case in which both protocols were successfully able to dock the ligand. This case represents a "best case scenario" from the point of view of the Rosetta energy function and and sampling algorithm. The ligand is asymmetric and 2 dimensional, with no rotatable bonds, and the ligand binding site is compact and deeply buried. As a result of this, the sampling space is sufficiently constrained that the additional initial placement sampling afforded by the TRANSFORM mover is unnecessary. Conversely, Figure III.10D is close to a worst case scenario. Here, a very small ligand is bound to a shallow pocket near the surface of the protein. Inspection of the crystal structure shows that the ligand is involved in a π-stacking interaction with two phenylalanine protein residues. This interaction is likely responsible for a substantial part of the total binding energy, but π-stacking interactions are not directly modeled by the Rosetta energy function and as a result will not be correctly recovered.

III.4.6 The TRANSFORM algorithm improves performance by improving sampling

The improvements yielded by the introduction of the TRANSFORM algorithm are likely a result of the types of moves performed during sampling. Because the original TRANSROT algorithm performs translation and rotation steps separately, it will be difficult to produce a move from the starting position that requires simultaneous translation and rotation. By simultaneously transforming the ligand in all dimensions, the space of the binding site can be more effectively sampled.

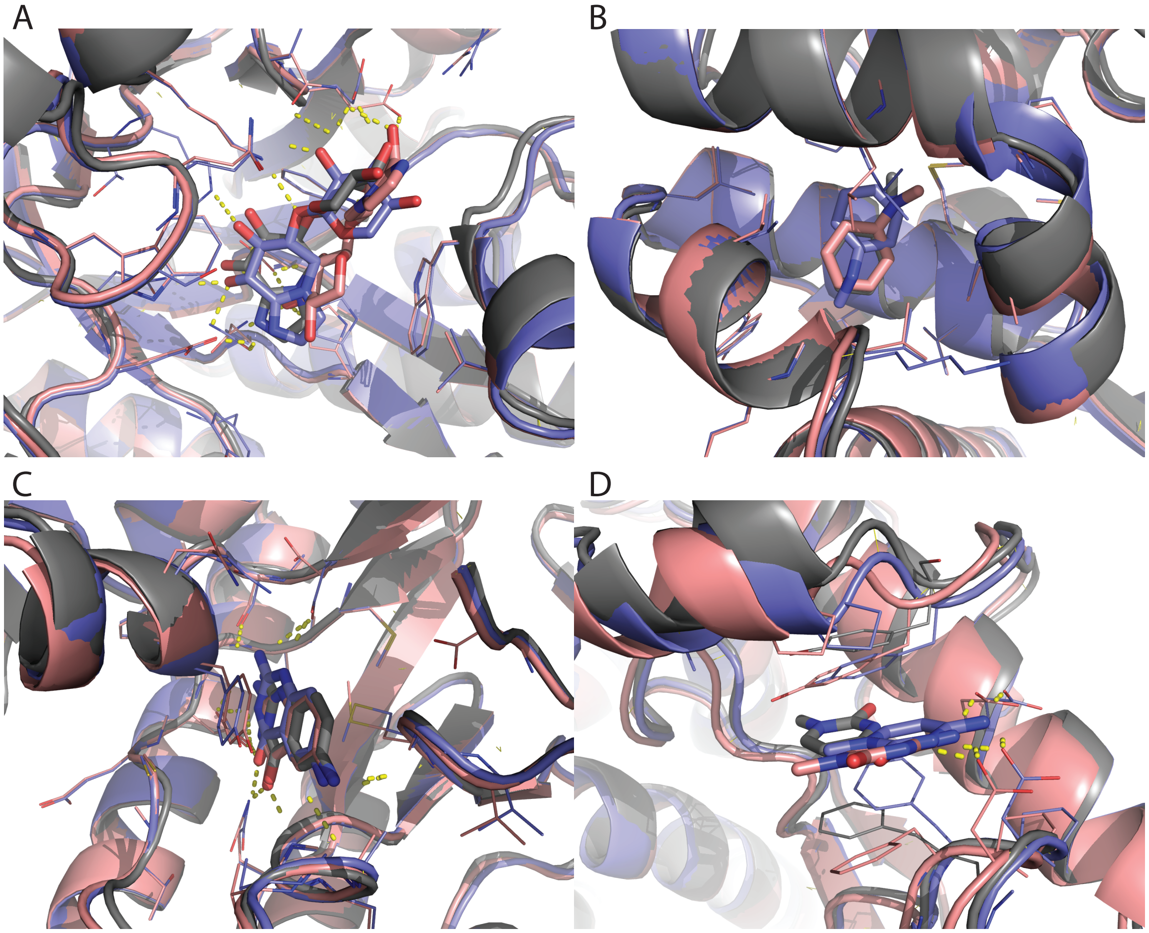


Figure III.10: Comparison of specific successes and failures between the RosettaLigand protocols. Native structures are in grey, lowest scoring models generated by the TRANSFORM/MCM protocol in blue, and lowest scoring models generated by TRANSROT/MCM in pink. A) A case in which the TRANSROT/MCM protocol was unsuccessful but the TRANSFORM/MCM protocol was successful (PDB ID: 1fhd). B) A case in which the TRANSFORM/MCM protocol was unsuccessful but the TRANSROT/MCM protocol was successful (PDB ID: 2otz). C) A case in which both methods were successful (PDB ID: 1bky). D) A case in which neither method was successful (PDB ID: 1q4w).

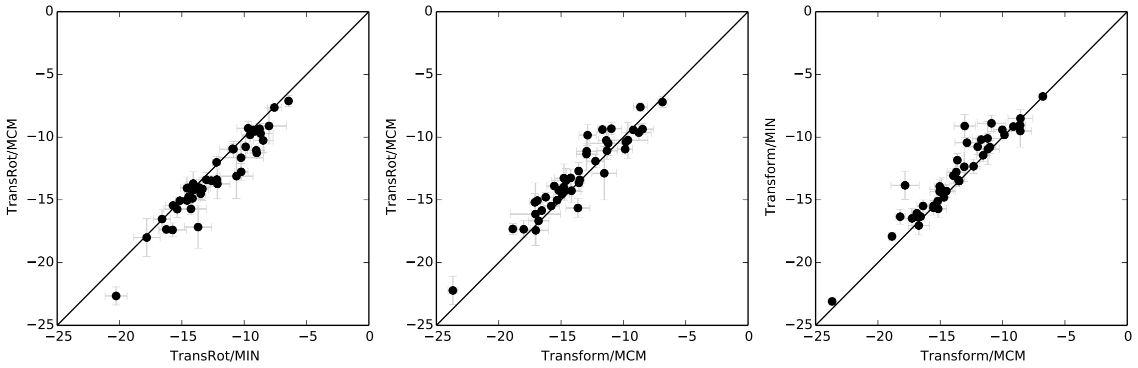


Figure III.11: Scatterplots showing the change in average all-atom Rosetta score of the lowest scoring model produced by several pairs of docking algorithms.

III.4.7 The MCM initial placement algorithm improves the scores of generated models

Comparison of the scores of the lowest RMSD models generated by protocols using the Transform and TransRot models demonstrates that the use of the Transform initial placement algorithm results in models with slightly lower all-atom scores compared to those generated with the TransRot algorithm (Figure III.11). As the energy function which produces these scores is identical between the two protocols, these lower scores indicate that the lower RMSD models generated by the Transform based protocol are also more favorable according to the Rosetta energy function. Because the Transform initial placement algorithm is capable of more efficiently sampling the binding site, it is more likely to place the ligand in a favorable position prior to refinement and final scoring. Table III.2 summarizes the data seen in figures III.11 and III.9

III.4.8 Despite improved sampling efficiency, K_d prediction is difficult

While the Transform algorithm results in slightly lower scores, it has no impact on the correlation between Rosetta score and experimentally derived K_d (Figure III.12). The R^2 correlation between $\log(K_d)$ and the Rosetta energy of the models made with the TransRot/MCM protocol is 0.24, while the R^2 correlation for models made with Transform/MCM is 0.29. This observation is in line with previous published studies(Lemmon

PDB ID	Score of lowest scoring model		RMSD of lowest scoring model	
	TransRot/MCM	Transform/MCM	TransRot/MCM	Transform/MCM
2q89	-17.05645	-16.1337	1.02925	0.7834
3bgz	-15.78875	-15.48785	6.73195	4.53105
1ukb	-9.8015	-10.37875	4.26405	1.24235
1vot	-13.5644	-13.63745	4.04415	2.8081
2p7g	-16.54995	-15.8481	1.57715	0.84315
2rca	-14.78235	-13.98525	1.0863	1.08875
2vuk	-15.1821	-14.24355	2.1848	1.2578
2nta	-13.64955	-15.64855	0.9989	1.1451
2are	-11.6824	-9.3915	2.24865	1.3767
2z4b	-14.2266	-13.24475	2.33705	1.3088
2jj3	-18.87045	-17.31715	1.701	3.80675
2b3f	-14.77755	-13.2687	2.6562	2.7313
2bbf	-17.02835	-17.43075	1.60725	1.93445
2pwg	-11.53405	-12.87725	2.43515	3.74805
2otz	-9.8603	-10.957	0.9874	1.04625
2ou0	-9.20305	-9.4244	2.0506	0.86025
2pzv	-8.64085	-7.6027	3.12625	5.0221
2q88	-16.8035	-16.66835	1.43145	1.2881
1ui0	-8.7612	-9.63995	2.07845	2.2449
1uz1	-12.87105	-9.8404	3.29545	1.28635
1uz4	-11.37255	-10.25565	3.2094	1.82905
1v0l	-11.31775	-11.0844	2.55955	1.53555
1ws4	-9.63965	-10.2396	3.6352	3.72855
1y20	-14.73825	-14.3057	0.4692	0.35425
2fai	-14.9162	-14.53665	2.08935	1.82305
2fqw	-17.10135	-15.1967	2.1882	1.39745
2j78	-13.59105	-12.6966	3.92555	3.28615
1bky	-8.46035	-9.36075	4.429	4.30845
1q4w	-14.60255	-13.47	1.6188	1.16905
1fcx	-23.6791	-22.2225	1.35465	1.30715
1fh8	-10.96435	-9.32295	2.65895	1.9014
1y93	-6.86775	-7.1999	3.06125	1.6985
1lhw	-16.89405	-15.0563	3.2851	1.14035
1fh9	-12.94845	-11.12155	6.047	5.8786
1s9t	-11.2194	-10.4868	4.906	5.02605
1fhd	-12.96275	-11.3485	5.5551	3.51735
1lnm	-17.99695	-17.3451	5.2465	1.1052
1nli	-15.5466	-13.8951	0.9861	0.7264
1ow4	-16.2282	-14.7894	2.9311	1.3489
1r5y	-14.1599	-14.2713	1.17925	1.1798
1s38	-15.3163	-15.02655	1.2923	0.7658
1s39	-12.2389	-11.9154	1.44685	1.80105
1sw1	-13.4979	-13.4037	2.23155	1.90315

Table III.2: A table showing the the PDB IDs, scores and RMSDs for the lowest scoring models generated by the Transform/MCM and the TransRot/MCM protocols docking ligands into the set of 43 relaxed protein models. For each of the 43 protein-ligand pairs, 1000 models were generated.

et al., 2012) which indicates that the Rosetta energy function, as well as other popular energy functions (Bauer et al., 2013) are frequently unable to effectively predict binding affinity. Given that the binding affinity prediction problem appears to be driven by a lack of detail in the Rosetta refinement energy function, it would not be expected that improved sampling would have a significant impact on this correlation, and indeed we find this to be the case.

III.4.9 The benefit of the MIN refinement is primarily logistical, but important for vHTS studies

While Figure III.6 and Figure III.7 indicate that both the MIN and MCM refinement algorithms have a similar impact on sampling performance and average run time, the substantially reduced variability in run time of the MIN refinement algorithm illustrated in Figure III.4 provides a major practical advantage to using MIN rather than MCM for refinement. When docking a large number of ligands on a computing cluster, a protocol with a predictable run time is highly advantageous as it allows for more efficient utilization of the available resources of the cluster. For this reason, while the two refinement methods have similar scientific performance, we recommend using MIN refinement, rather than MCM refinement.

III.4.10 Improving the speed of RosettaLigand increases the number of compounds that can be feasibly screened, enabling vHTS studies

Given that RosettaLigand is an “Embarrassingly Parallel” application, and thus scales linearly with the amount of available CPU resources, a substantial reduction in required runtime per ligand is extremely valuable. By reducing the total processing time per ligand from several hours to approximately 15 minutes, it now becomes possible to screen large libraries of compounds. This development makes the use of RosettaLigand as a vHTS tool computationally feasible for the first time.

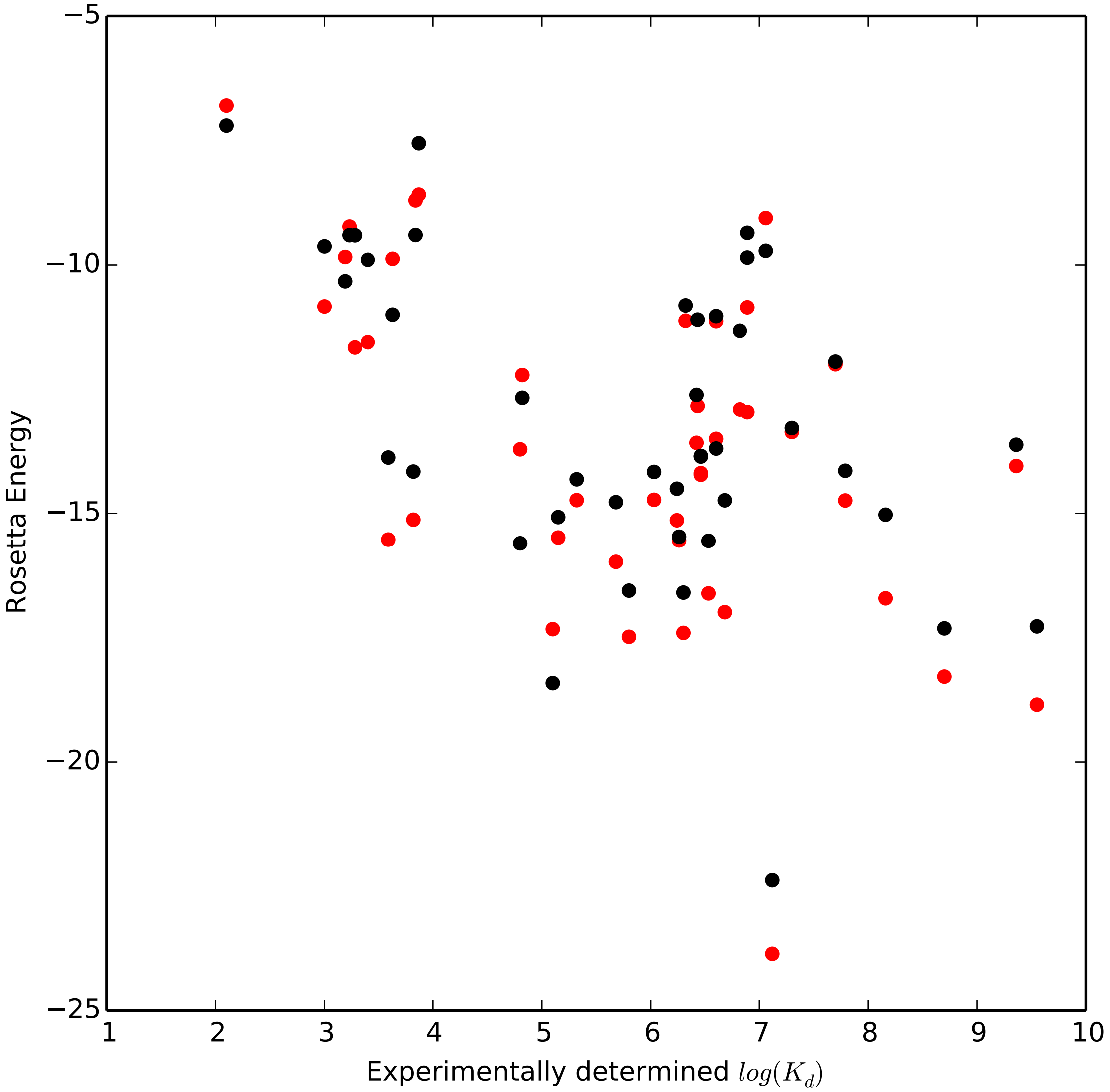


Figure III.12: Scatter plots showing the weak correlation between experimental log(K_d) and predicted Rosetta energy score for models in the 43 protein benchmark. Scores from models generated using the TRANSFORM/MCM protocol are in red while scores from models generated using the TRANSROT/MCM protocol are in black.

III.4.11 The TRANSFORM initial placement algorithm improves the ability of Rosetta to dock difficult protein-ligand pairs

A recurring theme in the development of protein-ligand docking tools is irregular performance of these tools in correctly predicting binding position(Kaufmann and Meiler, 2012; Lemmon and Meiler, 2013; Mysinger et al., 2012; Bauer et al., 2013; Davis et al., 2009). While the TRANSFORM mover appears to dramatically improve the ability of Rosetta to accurately predict ligand binding position, some ligands still cannot be correctly docked. The ability to predict *a priori* whether a ligand can be effectively docked, or at least develop some heuristics to aid in such a prediction, would be highly valuable. Figure III.13 plots the distribution of several ligand descriptors as a function of the ability of Rosetta to successfully dock the ligand. The number of atoms, rotatable bonds, stereo centers, hydrogen bond donors and acceptors are computed, as is the molecular weight, Van Der Waals (VDW) volume and surface area, and girth. Girth is computed as the longest distance between any pair of atoms in the small molecule. All ligand descriptors were computed using the BioChemical Library (BCL). Additionally Figure III.14 plots the distribution of several protein-ligand pair descriptors using the native crystal structure. Specifically, the ratio of Rosetta binding Energy to SASA, the total SASA, the Rosetta Hydrogen bonding energy, the number of residues in the complete protein and at the protein ligand interface, and the packing statistic(Sheffler and Baker, 2009). As before, a ligand is considered to be successfully docked if the lowest scoring model is within 2.0 Å of the crystal structure. Taken as a whole, figures III.13 and III.14 suggest that smaller, less flexible, and more deeply buried ligands are easier for both TRANSFORM and TRANSROT based docking protocols to handle, and that the TRANSFORM protocol is able to recover the binding mode in larger, more flexible, less deeply buried ligands that the TRANSROT protocol is unable to correctly model. Unfortunately, while the ligands that cannot be successfully docked by either model tend to be larger and more flexible, we can identify no hard rules for predicting reliably if a ligand will be be successfully docked by Rosetta, as the distributions of

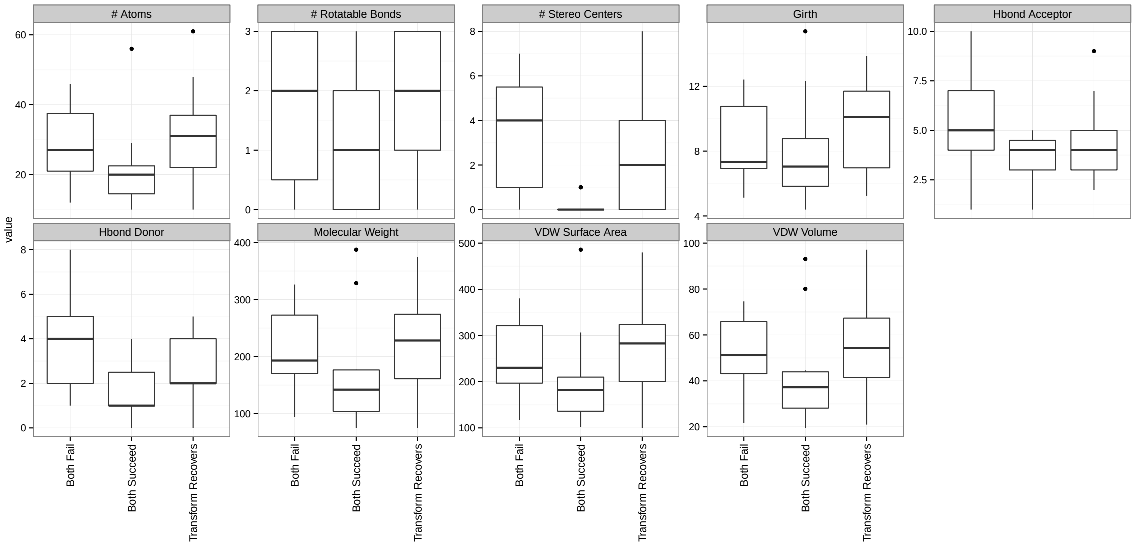


Figure III.13: Box and whisker plots showing the distribution of various ligand properties amongst subsets of protein-ligand pairs in the 34 protein binding set. “Both fail” is the set of pairs for which both TRANSFORM and TRANSROT protocols were unable to successfully dock a a ligand. “Both succeed” is the set of pairs in which both protocol are successful, and “Transform fix” is the set of pairs for which the TRANSROT protocol is successful and the TRANSFORM protocol is unsuccessful.

successful and unsuccessful ligands overlap for every descriptor evaluated.

III.5 Conclusion and Future Directions

III.5.1 The impact of improvements in low resolution sampling

Here we have shown that improvement in sampling efficiency can have a large impact on ligand docking performance, even in the absence of improvements to the energy function. Despite the relatively small number of degrees of freedom present in a preotein-ligand docking simulation compared to other types of protein simulations, the sampling space is still sufficiently complex that care must be taken to intelligently sample the binding space. We also demonstrate that even a highly simplistic scoring function such as the binary scoring grid described in Section III.2.3.1 contains more information than may be immediately apparent. While the binary scoring grid scores all clashing and non-clashing atomic positions as equally unfavorable and favorable respectively, we found that the addition of MC

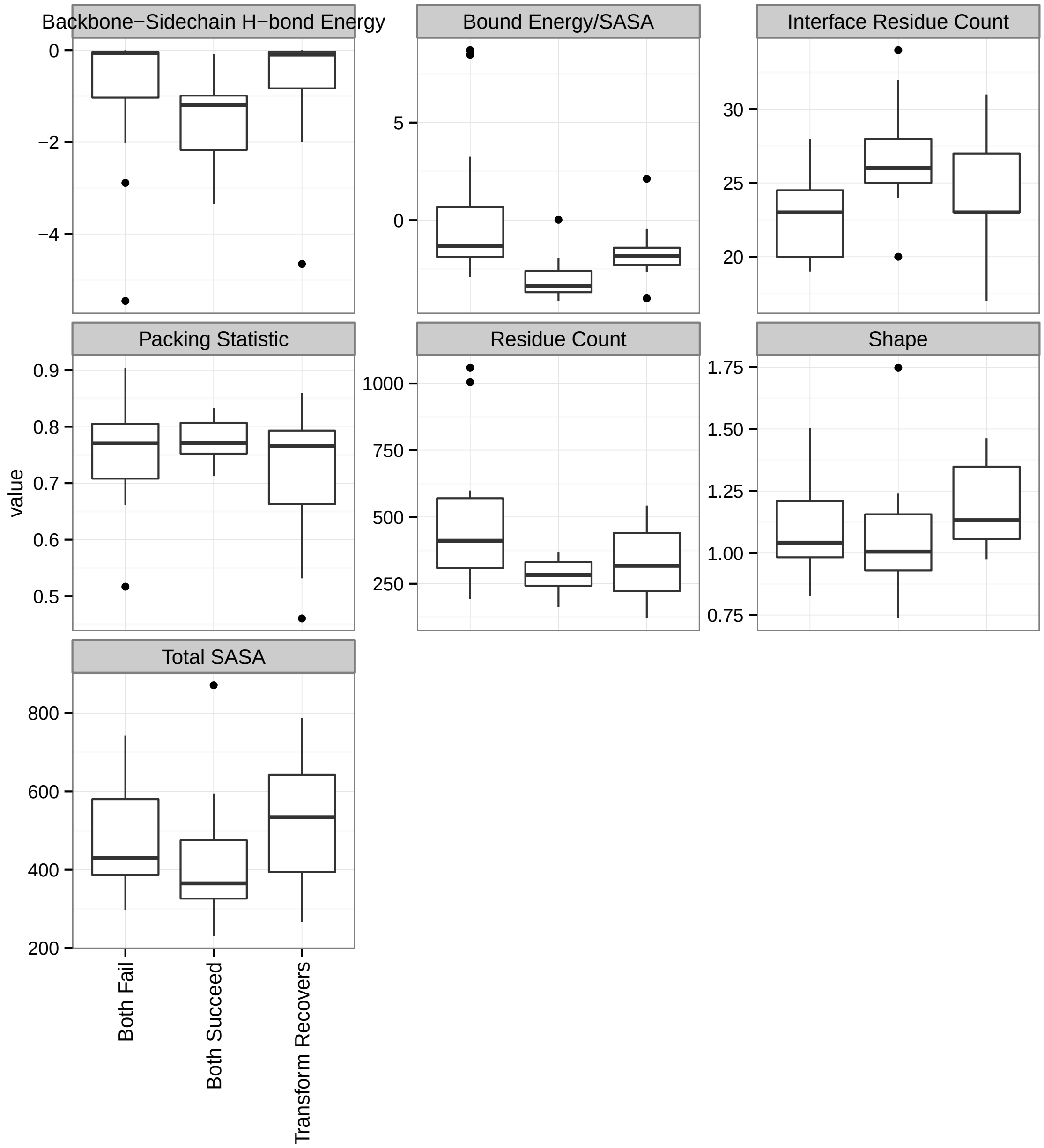


Figure III.14: Box and whisker plots showing the distribution of various protein properties amongst subsets of protein-ligand pairs in the 34 protein binding set. “Both fail” is the set of pairs for which both TRANSFORM and TRANSROT protocols were unable to successfully dock a a ligand. “Both suceed” is the set of pairs in which both protocol are successful, and “Transform fix” is the set of pairs for which the TRANSROT protocol is successful and the TRANSFORM protocol is unsuccessful.

sampling caused a dramatic improvement in the quality of ligand binding positions during the initial placement phase of docking. This suggests that while the energy function is very flat with respect to the position of individual atoms, it still contains enough information to distinguish between generally bad and generally good binding positions when the entire protein-ligand complex is taken into account.

III.5.2 The scientific value of increased algorithm speed

In addition to improvements in scientific performance, the TRANSFORM initial placement algorithm described here also results in a dramatic decrease in the total runtime of the ligand docking simulation relative to the TRANSROT algorithm. This improvement in speed has important scientific implications. First, increased speed increases the number of compounds which can be computationally tested given a fixed amount of CPU resources. As a result of the tremendous size of chemical space(Reymond et al., 2012), the probability of an active compound existing in a randomly selected subset of chemical space is small, therefore the size of the database screened should be as large as possible. The analysis described in this manuscript suggests that in a high throughput screening workflow, a two stage docking process would make the best use of availible CPU time. In this process, both the TRANSFORM/MIN and TRANSFORM/MCM docking protocols would be used. The primary advantage of the TRANSFORM/MIN protocol is the speed of computation. TRANSFORM/MIN takes an average of 9.3 seconds to generate a model, and requires approximately 150 models to reliably generate a high quality binding position, for a total of 1395 seconds per model (23.25 minutes). The previously published TRANSROT/MCM protocol takes an average of 49.4 seconds per model and requires approximately 1000 models to generate a high quality binding position for a total of 49400 seconds (823 minutes). This represents an approximately 35 fold decrease in CPU time, and allows a far larger number of models to be screened in a reasonable amount of time. For example, the ZINC(Irwin et al., 2012) “Clean Drug-Like” database currently contains 13 million compounds. Using the new TRANSFORM/MIN protocol, this database could be screened on a 10000 CPU cluster in approximately 20.7 days, as opbinding positiond to 742 days with the TRANSROT/MCM protocol. After screening the entire database, a smaller subset of compounds would be selected for further study. At this point, the TRANSFORM/MCM protocol would be used to re-dock the compounds, taking advantage of the improved scores obtained through the use of the slightly slower MCM refinement algorithm which were observed and described in Section III.4.7. This two stage approach would allow us to leverage the drastically increased speed of the TRANSFORM/MIN protocol with the improved refinement of the TRANSFORM/MCM protocol.

III.6 Methods

III.6.1 Complete docking protocol with high resolution refinement

III.6.1.1 Introduction to protein preparation

Complete command lines and instructions for the protein preparation are detailed in Chapter E.

III.6.1.2 Description of CSAR crystal structure preparation

The original crystal structures from the CSAR dataset were processed to remove existing water molecules, and hydrogens were added using Rosetta. The side-chains and protein backbone were left at the crystallographic positions.

III.6.1.3 Description of CSAR repacked structure preparation

The crystal structures prepared above were repacked in the absence of the ligand using the Rosetta fixbb application. The backbone was kept fixed, and all side-chain positions were allowed to repack (Kortemme et al., 2004).

III.6.1.4 Description of CSAR relaxed structure preparation

For each of the crystal structures prepared above, 10 relaxed models were produced using the Rosetta relax application. During the Rosetta relax protocol, cyclic repacking of the side-chains and gradient based minimization of the backbone are used to perform MCM of the entire protein structure. In this case, all CA atoms were restrained to within 0.3Å of the crystallographic coordinates, to prevent major conformational shifts. Relaxation was performed in the absence of the ligand.

III.6.2 Ligand Conformer preparation

III.6.2.1 Description of ligand conformer generation

Conformers were generated for each ligand using the BCL::ConformerGeneration application(unpublished). BCL::ConformerGeneration uses a stochastic fragment assembly approach to conformer generation, utilizing a database of fragment conformations derived from the Cambridge Structural Database. A maximum of 100 conformers were generated per ligand, though the actual number of generated conformers varies based on the structure of the ligand and the number of rotatable bonds. The generated conformers were used to produce params files and ligand rotamer libraries using the protocol detailed in Chapter E.

CHAPTER IV

RosettaHTS: A virtual High Throughput Screening tool integrating structure and ligand based information

IV.1 Introduction

IV.1.1 Ligand docking methods are inconsistently able to predict binding affinity

While protein-ligand docking tools are frequently capable of correctly predicting poses (Trott and Olson, 2009; Friesner et al., 2004; Ewing et al., 2001), these methods have proven limited in their ability to distinguish between active and inactive compounds (Bauer et al., 2013; Huang et al., 2006; Davis et al., 2009). The DEKOIS 2.0 benchmark (Bauer et al., 2013) and a blind study of protein-ligand docking tools (Davis et al., 2009) demonstrated that the majority of protein-targets can successfully be studied with protein-ligand docking tools, although the tool with the best performance varies based on the target. Here, success is defined as the ability for the protein-ligand docking tool to assign the lowest score to a model with a low RMSD structure. This suggests that in general, protein-ligand docking algorithms have the ability to be successful. However, the overall variance in ligand docking performance also suggests that while active and inactive ligands can be classified for most protein systems by at least one available tool, no tool is universally reliable. Furthermore, it has not been possible to easily predict which protein systems can be accurately targeted by which by which ligand docking tools.

IV.1.2 Target-specific models are fundamentally limited in modern drug discovery

Target specific models for vHTS screening have frequently been employed to identify novel hit compounds for drug development. These models are specific to a protein target of drug discovery and cannot be used to make effective predictions about drug activity outside of the narrow range of targets on which they are trained.

vHTS models can be broadly grouped into two categories: ligand-based models, and structure-based models. Ligand-based models are trained using only ligand information, and attempt to learn the differences between the chemistry of known active, and known inactive ligands as a means of predicting the activity of unknown compounds. Structure-based models, on the other hand, are trained using a combination of protein structure and ligand information and attempt to learn the differences in protein-ligand interactions between active and inactive compounds. Because ligand-based models include no protein structure information, it is extremely difficult to make a generic ligand-based model.

Despite the historic success of target specific models, there are downsides to the approach. First, a substantial volume of known active and inactive compounds must exist for the target molecule in order for a model to be trained. For this reason, successful models have been created for many popular targets, such as metabotropic Glutamate Receptor 5 (mGluR5) (Mueller et al., 2012), human Ether-a-go-go-Related Gene (hERG) (Kratz et al., 2014), and kinsases (Dranchak et al., 2013). Unfortunately, novel drug targets such as Odd-Skipped Related 1 (OSR1) (Austin et al., 2014; Villa et al., 2007) for which few (or no) existing inhibitors are known cannot make use of these techniques. This is problematic, as many critical areas of modern drug discovery, such as antibiotic discovery (Lewis, 2013), increasingly rely on novel protein targets to develop effective drugs.

Furthermore, vHTS methods that focus on protein-specific models are frequently biased to predict compounds similar to those that have already been predicted. In order to be truly successful, vHTS models must be capable of scaffold hopping (Böhm et al., 2004), or identifying compounds with substantially different structure than existing known compounds. While protein-specific models have been able to do this in the past (Butkiewicz et al., 2013; Gardiner et al., 2011), existing solutions exhibit high false positive rates. By combining pattern recognition techniques with ligand docking, it may be possible to create a generalized model capable of predicting ligand binding affinity independent of protein structure or ligand scaffold, potentially leading to a wider diversity of predicted compounds. This task will be aided by the increasing number of publicly available HTS datasets. Specifically, datasets such as PubChem, the PDB, and ChEMBL (Gaulton et al., 2012), provide increasingly large sets of public small molecule and protein structure information that can be used to train models.

IV.1.3 Neural network based methods can be used to re-score protein-ligand binding prediction

IV.1.3.1 NNScore

In recent years, tools have been developed to re-score protein-ligand binding predictions using neural networks. NNScore 1.0 (Durrant and Mccammon, 2010) and NNScore 2.0 (Durrant and Mccammon, 2011) are two such tools.

NNScore 1.0 is a neural network based scoring function intended to overcome the limitations of traditional knowledge and physics based scoring functions through the use of pattern recognition. The network itself is constructed as a feed forward network with a single hidden layer, 194 input nodes, and two outputs. The input descriptors to the NNScore 1.0 network consist of protein-ligand atom pair counts within 2.0 and 4.0 Å shells, an electrostatic-interaction energy between protein-ligand atom pairs within a 4.0 Å shell, the total count of ligand atom elements, and the number of rotatable bonds in the ligand being evaluated. In this way, the NNScore inputs contain information regarding both the protein-ligand environment and the ligand composition. NNScore 2.0 (Durrant and Mccammon, 2011) expands upon these descriptors by adding steric hinderance information, hydrophobic information, and hydrogen-bonding information computed by AutoDock Vina (Trott and Olson, 2009).

In both cases, the initial benchmarking of these models indicated their ability to distinguish accurately between well and poorly docked models, as well as their ability to act as a scoring function. (Durrant and Mccammon, 2011, 2010). However, a larger benchmark of NNScore 1.0 and 2.0 published in 2013 suggested that while these methods are frequently capable of improving activity classification and binding affinity prediction beyond protein-ligand docking, the success of the method remains highly dependent on the protein target. (Durrant et al., 2013)

IV.1.3.2 Predicting binding mode with ANNs

Another recent ANN-based approach to predicting ligand binding activity was described by Chupakhin, *et al.* (Chupakhin et al., 2013). Rather than using predicted binding mode as an input to a model predicting binding affinity, the Chupakhin method uses 3D fragment descriptors as an input and attempts to predict the interaction fingerprint of the protein-ligand pair. These predicted fingerprints are then used as the input to a similarity based virtual screening system to predict ligand activity. The advantage of this approach is that it is potentially possible to obtain results similar to those obtained by docking simulation and similarity based screening without actually performing computationally intensive docking studies.

In the initial implementation and benchmarking paper, Chupakhin, *et al.* report that they were able to accurately predict both interaction fingerprint and ligand activity classification for many protein targets. However, as with other methods, it appears that there remain many cases for which the method is unsuccessful.

IV.1.4 Using protein-ligand interaction similarity to predict binding affinity

In addition to the NNScore method, the properties of protein-ligand interactions can be used more directly as a predictor of small molecule affinity. The FEATURE method (Halperin et al., 2008; Tang and Altman, 2014) provides a means of describing the micro-environment of a small-molecule ligand interaction, and building a model of affinity based on that micro-environment. In this method, descriptor vectors are computed based on a set of chemical and physical properties computed for spherical shells surrounding each atom in the molecule. While spherical shells lack directional information, they are rapid to compute, and allow for a simple, orientation independent, description of the micro-environment of the small molecule. These simple vectors can then be used as the input to supervised machine learning methods to produce models of binding affinity.

IV.1.5 Existing limitations in data sources used for training and validation

One of the difficulties facing the successful implementation of any vHTS method is the selection of high quality datasets for model testing and validation. Having such a dataset is critical, so substantial existing literature is dedicated to the development of screening datasets, as well as characterization of the difficulties in curating them. Here, we discuss these limitations as they apply to the generation of a training dataset containing high quality active and inactive compounds.

IV.1.5.1 The limitations of existing sources of known active ligands

As the goal of RosettaHTS is to train a model capable of distinguishing between active and inactive small molecules across a range of protein targets and small molecule chemical space, the selection of a high quality training set was crucial. As discussed in Section IV.1.2, it may be possible to create models based on focused libraries of known active and inactive compounds for targets with substantial amounts of available vHTS data. However, if the goal is to create a model that can predict activity across a range of targets, or for targets with no currently known inhibitors, these focused libraries are insufficient for training purposes.

There are several additional factors that complicate the curation of a training set for a general classifier. First, compounds with known binding affinity must be located across a wide range of protein targets. IC_{50} and K_i are the two major means by which drug activity is measured. IC_{50} is defined as the concentration of an inhibitor necessary to cause a 50% reduction in biochemical function. On the other hand, K_i is a thermodynamic equilibrium constant which measures the ability of the protein-ligand complex to dissociate. As IC_{50} is a measurement of biochemical function, it is affected by a wide range of physical and chemical factors. For this reason, the IC_{50} of a single compound with respect to two different proteins cannot be easily compared. Because the goal is to compare the activity of ligands independent of the protein target, K_i must be used instead of IC_{50}.

Because IC_{50} values are often more challenging to obtain than K_i values, this substantially reduces the availability of data from public databases, such as ChEMBL or PubChem. Furthermore, the active compounds selected must bind to a wide and evenly distributed range of targets and have a wide and evenly distributed range of known activities.

Due to the Intellectual Property concerns associated with the drug discovery process, neither publicly nor privately available compound databases meet these requirements. For example, while ChEMBL contains 481,050 K_i value measurements across 164 distinct targets with known K_i values, 90% of these targets have fewer than 891 measurements each. In other words, nearly all of the available K_i values are confined to a small handful of protein targets.

IV.1.5.2 The limitations of existing sources of known inactive ligands

The difficulty of selecting a high quality training dataset is compounded by the availability of known inactive ligands. Binding affinity data of inactive ligands is less frequently published than active ligands. Additionally, most inactive ligands are measured as inactive by IC_{50} rather than K_i. Because IC_{50} and K_i are distinct properties, these experimental values are unusable for a model predicting K_i, as a compound with no measurable IC_{50} does not necessarily have measurable binding affinity.

The lack of known inactive compounds is most frequently addressed by the use of property matching techniques. In this process, compounds that have similar chemical properties to the known active ligands but dissimilar structures are selected and designated as presumed inactive compounds. Directory of Useful Decoys (DUD) (Huang et al., 2006) is one of the earlier attempts at a large scale library of inactive putative inactive compounds. The DUD dataset consists of a set of known active compounds with available crystal structural data, and unknown putative inactive compounds across a set of 40 protein targets. In the case of DUD, putative inactive decoy compounds were selected by a process of histogram matching across molecular weight, hydrogen-bond donor and acceptor weight, rotatable bond count, and logP, followed by topological dissimilarity, to select compounds with similar chemical properties but dissimilar physical properties.

While such a method can be useful for predicting inactivity, there is a substantial risk of inadvertently including active ligands among the set of predicted inactives. This concern was borne out in the case of the DUD dataset, as several of the decoys in the DUD dataset were found to have activity against the target ligand (Vogel et al., 2011). Additionally, the five parameters used for histogram matching in the original DUD dataset were determined to be insufficient for the purposes of dataset balancing. As an example, the original DUD dataset was found to have a charge imbalance, in which the set of active ligands was significantly more charged than the inactive decoys (Irwin, 2008). As an attempt to remedy these issues, the Directory of Useful Decoys Enhanced (DUD-E) (Mysinger et al., 2012), DEKOIS(Vogel et al., 2011), and later, DEKOIS 2.0 (Bauer et al., 2013) benchmarking sets were developed. These benchmarking sets take varying approaches to generating more balanced datasets.

DUD-E attempts to improve upon the original DUD dataset by increasing the size and scope of the benchmarking set, removing homology model based protein target structures, and improving the method of ligand clustering and decoy property matching. To reduce the 2D similarity of decoy compounds to known active compounds, the 2D similarity filtering method was refined to exclude a greater percentage of compounds. DUD-E thus represents an iterative refinement in the library design methods established by the original DUD publication rather than an entirely novel approach. Regardless, DUD-E effectively resolves the charge imbalance issue in the original dataset and improves the degree of dissimilarity between ligands and predicted decoys.

DEKOIS is an independent attempt to improve upon the DUD dataset. Like DUD, the DEKOIS protocol for identifying predicted inactive compound begins with the identification of a large pool of potential decoys through property matching. The potentially inactive ligands are then clustered to be similar in chemical property to the known active compounds and then filtered to be structurally dissimilar. However, DEKOIS adds a novel filtering function for removing latent actives from the decoy set while maintaining sufficient physiochemical similarity between the known actives and putative inactives. Latent activity is predicted using functional fingerprint bitstrings. Bitstrings for the known active compounds are analyzed to produce a set of substructure fingerprints associated with ligand activity, and ligands with a high degree of substructural similarity are removed from the set.

DEKOIS 2.0 expands upon the filtering algorithm presented in the original DEKOIS benchmarking set through the addition of descriptors representing the population of negatively and positively charged state and aromatic ring count. Incremental improvements were also made to the scoring function used to predict latent activity in the decoy set. Additionally, the DEKOIS 2.0 benchmarking set expands the size and breadth of the original DEKOIS set.

While property matching methods have been relatively effective for the development of ligand docking benchmarking tools, they are insufficient for training machine learning models, particularly ANNs. While ANNs are best known for their ability to model complex patterns, they are also generally capable of approximating arbitrary polynomial functions (Lindsey, 1997). This means that any ANN model trained using a set of algorithmically trained decoys may learn the decoy selection algorithm rather than the actual chemical and physical patterns distinguishing active and inactive compounds.

IV.2 Methods

IV.2.1 Development of a balanced training dataset

IV.2.1.1 Cross-docking a diverse set of ligands to create a balanced set of training dataset

To overcome the difficulties with synthetic benchmarking sets described in IV.1.2, the inactive component of the training dataset used for RosettaHTS is generated via cross-docking. In this approach, a diverse set of compounds with known binding affinity against a diverse set of protein targets is selected as a source of active ligands. Presumed inactive decoys are then generated by cross-docking each known active ligand into each protein. The advantage of this approach from an ANN training perspective is that it ensures that the distribution of active and inactive compounds is absolutely identical from a chemical standpoint. By using a diverse set of non-overlapping protein targets, the probability that cross-docked ligands in the inactive set actually have activity against the cross-docked targets is low.

PDBBind (Wang et al., 2004) is a collection of protein-ligand binding pairs for which known binding affinities and X-ray crystal structures exist. Since its original publication in 2004, The PDBBind database has been periodically updated. As of 2013, it contains 10,776 total complexes, of which 2,959 have known K_i values, are non-covalently bound, have only a single ligand in the binding site, and have crystal structures with a resolution of less than 2.5 Å. Compounds from this "refined" subset of the PDBBind database were used as the basis of the training dataset, which was further filtered as described in Section IV.2.1.2 to produce a training dataset.

IV.2.1.2 Additional filtering of PDBBind "refined" to produce a diverse set of high quality active compounds

For the purposes of this training set, all active ligands must have the following properties beyond those described in Section IV.2.1.1: The set of proteins must be diverse in that every protein in the set should come from a different family to avoid biasing training towards a particular class of receptors. Additionally, while RosettaLigand is capable of docking ligands into proteins of any size, extremely large proteins require the allocation of large amounts of memory, which makes screening more time consuming. As a result, all proteins in the training set were required to have fewer than 1,000 total residues across all chains. RosettaLigand must also be capable of predicting the pose of each complex such that the lowest Rosetta score has an RMSD of < 2.0 Å to the crystal structure. The rationale for this requirement is that the goal of the model is to predict ligand binding affinity. Including protein-ligand complexes which cannot be successfully docked will serve only to increase the noise in the model. As some of the compounds in PDBBind are large molecules, natural products, and peptides, the set of protein-ligand complexes considered was limited to those with ligands obeying Lipinski's rule of Five (Lipinski et al., 2001). The 2959 complexes in the PDBBind refined set were filtered based on the criteria described above and then docked using the RosettaLigand protocol described previously, resulting in a set of 120 unique protein-ligand complexes (summarized in Table IV.1). The resulting set of compounds represents a divers group of small, drug-like molecules. Specifically, the number of atoms ranges from 6 to 62, with a median of 25, and the number of rings ranges from 0 to 5 with a median of 1. The number of rotatable bonds ranges from 0 to 13 with a median of 2, The number of hydrogen-bond acceptors ranges from 1 to 5, with a median of 4, the number of hydrogen-bond donors ranges from 0 to 5 with a median of 2, and the log(K_i) ranges from 2.2 to 9.7 with a median of 4.38, the weight ranges from 87.06 to 496.51, with a median of 183.35. Figure IV.1 plots the overall distribution of these properties across the training dataset.

IV.2.1.3 Crossdocking active compounds to produce presumed inactive binding data

The active component of the training dataset was based on the lowest scoring of 200 RosettaLigand models produced for each known active protein-ligand pair described in Section IV.2.1.2. The specific protocol used for all RosettaLigand docking described in this study is

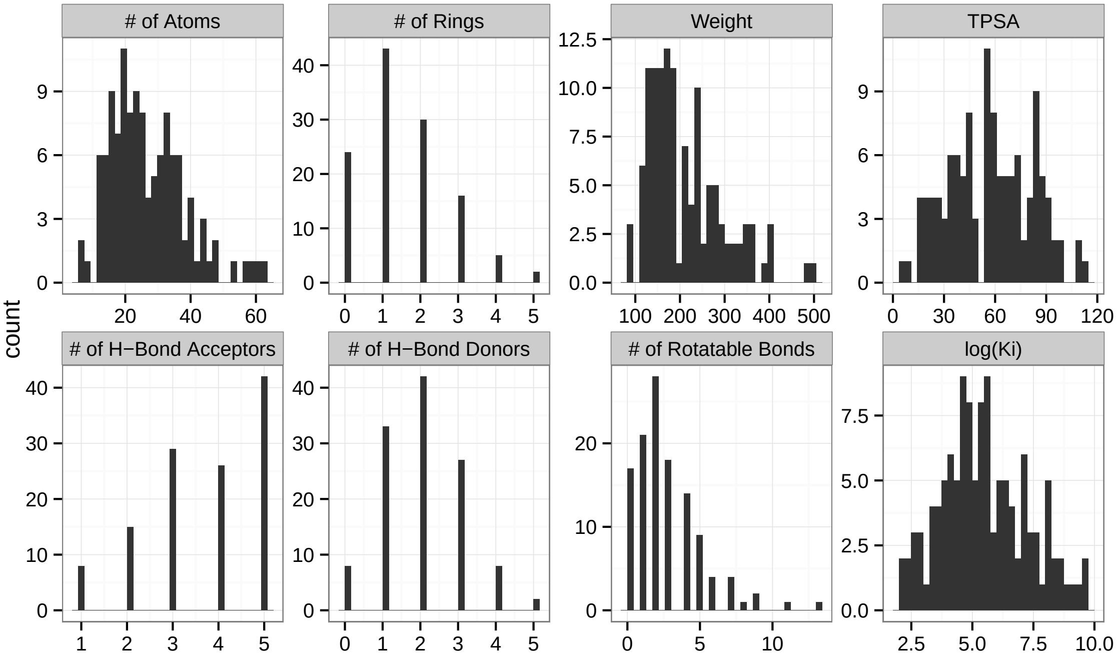


Figure IV.1: The basic property distribution of ligands in the training dataset. Histograms are plotted of the atom count, rotatable bond count, ring count, Topological Polar Surface Area, log(K_i), count of hydrogen-bond donors, acceptors, and molecular weight of the 120 ligands in the binding site.

PDB ID	Ligand ID	Ligand Formula	Protein name	$log(K_i)$	Weight	AA Count	Citation
3pwk	L14	$C_8H_{12}O_4$	trans-cyclohexane-1,4-dicarboxylic acid	3.35	172.178	732	(Pavlovsky et al., 2012)
3v7t	0GX	$C_{23}H_{28}BRFN_2O_2S$	(3-exo)-3-[5-(aminomethyl)-2-fluorophenyl]-8-azabicyclo[3.2.1]oct-8-yl(4-bromo-3-methyl-5-propoxythiophen-2-yl)methanone	7.21	495.448	980	(Liang et al., 2012b)
3i4y	35C	$C_6H_4CL_2O_2$	3,5-dichlorobenzene-1,2-diol	7.05	179.001	270	(Matera et al., 2010)
1n8v	BDD	$C_{12}H_{25}BRO$	bromo-dodecanol	6.03	265.23	224	(Campanacci et al., 2003)
2pql	TSS	$C_{10}H_{12}N_2$	2-(1h-indol-3-yl)ethanamine	7.28	160.216	145	(Becker et al., 1999)
3cyz	9OD	$C_{10}H_{16}O_3$	(2z)-9-oxodec-2-enoic acid	7.22185	184.232	238	(Pesenti et al., 2009)
3lka	M4S	$C_7H_9NO_3S$	4-methoxybenzenesulfonamide	2.82	187.216	158	(Borsi et al., 2010)
1d7i	DSS	$C_3H_8OS_2$	methyl methylsulfinylmethyl sulfide	3.6	124.225	214	(Burkhard et al., 2000)
2uxi	G50	$C_{15}H_{14}O_5$	3-(4-hydroxyphenyl)-1-(2,4,6-trihydroxyphenyl)propan-1-one	7.3	274.269	420	(Alguel et al., 2007)
2v25	ASP	$C_4H_8N_2O_3$	Asparagine	5.72	132.117	518	(Müller et al., 2007)
3uxk	BHO	$C_7H_7NO_2$	benzhydroxamic acid	4.93	137.136	1532	(Lietzan et al., 2012)
4dhl	0K7	$C_{11}H_9NO_2S$	2-(4-methylphenyl)-1,3-thiazole-4-carboxylic acid	4.48	219.26	1704	(Feder et al., 2012)
3kr8	XAV	$C_{14}H_{11}F_3N_2OS$	2-[4-(trifluoromethyl)phenyl]-7,8-dihydro-5h-thiopyrano[4,3-d]pyrimidin-4-ol	8.1	312.31	480	(Karlberg et al., 2010b)
1enu	APZ	$C_8H_7N_3O_2$	4-aminophthalhydrazide	5.08	177.16	386	(Grädler et al., 2001)
1ui0	URA	$C_4H_4N_2O_2$	uracil	7.06	112.087	205	(Hoseki et al., 2003)
1efy	BZC	$C_{15}H_{13}N_3O_2$	2-(3'-methoxyphenyl) benzimidazole-4-carboxamide	8.22	267.283	350	(White et al., 2000)
2hjb	PZM	$C_8H_{11}NO$	1-(4-methoxyphenyl)methanamine	4.39	137.179	992	(Hothi et al., 2007)
1qft	HSM	$C_5H_9N_3$	histamine	8.77	111.145	350	(Paesen et al., 1999)
2afw	AHN	$C_7H_{11}N_3O$	n-[2-(1h-imidazol-4-yl)ethyl]acetamide	4.77	153.182	658	(Huang et al., 2005)
1hsl	HIS	$C_6H_9N_3O_2$	Histidine	7.19	155.154	476	(Yao et al., 1994)
2j4g	NB1	$C_{10}H_{17}NO_4S$	(3ar,5r,6s,7r,7ar)-5-(hydroxymethyl)-2-propyl-5,6,7,7a-tetrahydro-3ah-pyrano[3,2-d][1,3]thiazole-6,7-diol	6.6	247.311	1430	(Whitworth et al., 2007)
2ojg	19A	$C_{16}H_{16}N_4O$	n,n-dimethyl-4-(4-phenyl-1h-pyrazol-3-yl)-1h-pyrrole-2-carboxamide	5.64	280.324	380	(Aronov et al., 2007)
1lgw	1AN	C_6H_6FN	2-fluoroaniline	4	111.117	164	(Wei et al., 2002)
3ebi	BEY	$C_{19}H_{24}NO_4P$	(2s)-3-[(r)-[(1s)-1-amino-3-phenylpropyl](hydroxy)phosphoryl]-2-benzylpropanoic acid	7.10237	361.372	890	(McGowan et al., 2009)
1pfu	MPJ	$C_4H_{12}NO_2PS$	(1-amino-3-methylsulfanyl-propyl)-phosphinic acid	2.72	169.182	551	(Crepin et al., 2003)
3p8p	LN6	$C_{10}H_{21}N_3O_2$	n 5 -[(1e)-pentanimidoyl]-l-ornithine	4.55	215.293	616	(Lluis et al., 2011)
1ai4	DHY	$C_8H_8O_4$	2-(3,4-DIHYDROXYPHENYL)ACETIC ACID	2.5	168.15	766	(Done et al., 1998)
3p7i	P7I	$C_2H_8NO_3P$	(2-aminoethyl)phosphonic acid	7.7	125.064	321	(Alicea et al., 2011)
1ceb	AMH	$C_8H_{15}NO_2$	trans-4-aminomethylcyclohexane-1-carboxylic acid	6	157.21	176	(Mathews et al., 1996)
1eoc	4NC	$C_6H_5NO_4$	4-nitrocatechol	6.05	155.108	450	(Vetting et al., 2000)
1a99	PUT	$C_4H_{12}N_2$	1,4-diaminobutane	5.7	88.151	1376	(Vassylyev et al., 1998)
2hzl	PYR	$C_3H_4O_3$	pyruvic acid	6.57	88.062	730	(Gonin et al., 2007)
4a6l	P43	$C_{25}H_{23}FN_2O_2$	1-3-[1-(5-[(2-fluorophenyl)ethynyl]furan-2-ylcarbonyl)piperidin-4-yl]phenylmethanamine	8	402.461	980	(Liang et al., 2012a)
4ai5	ADK	$C_6H_7N_5$	3-methyl-3h-purin-6-ylamine	2.92	149.153	940	(Zhu et al., 2012)
2vba	P4T	$C_{12}H_{12}N_2OS$	2-phenylamino-4-methyl-5-acetyl thiazole	4.6	232.301	1624	(Pappenberger et al., 2007)
3f6g	ILE	$C_6H_{13}NO_2$	Isoleucine	4.72	131.17	254	(Zhang et al., 2009b)
2q7q	C2B	C_7H_8ClN	1-(4-chlorophenyl)methanamine	5.3	141.598	970	(Hothi et al., 2007)
2vl4	MNM	$C_6H_{13}NO_4$	(2s,3s,4r,5r)-2,3,4-trihydroxy-5-hydroxymethyl-piperidine	6.011	163.172	1692	(Prehna et al., 2012)
3odu	ITD	$C_{21}H_{34}N_4S_2$	(6,6-dimethyl-5,6-dihydroimidazo[2,1-b][1,3]thiazol-3-yl)methyl n,n'-dicyclohexylimidothiocarbamate	8.14	406.651	1004	(Wu et al., 2010)
4agl	P84	$C_{14}H_{20}I_2N_2O$	2,4-bis(iodanyl)-6-[[methyl-(1-methylpiperidin-4-yl)amino]methyl]phenol	3.65	486.13	438	(Wilcken et al., 2012)
1r9l	BET	$C_5H_{12}NO_2$	trimethyl glycine	5.4	118.154	309	(Schiefner et al., 2004a)
2vo4	4NM	$C_7H_8NO_2S$	4-nitrophenyl methanethiol	4.93554	170.209	438	(Axarli et al., 2009)
1gyx	BEZ	$C_7H_6O_2$	benzoic acid	2.48	122.121	152	(Almrud et al., 2002)
1lst	LYS	$C_6H_{14}N_2O_2$	Lysine	7.85	146.19Ê	239	(Oh et al., 1993)
3iog	SDF	$C_7H_7CL_2O_3PS$	[(r)-(2,4-dichlorophenyl)(sulfanyl)methyl]phosphonic acid	5.52	273.073	227	(Lassaux et al., 2010)
3hmp	CX4	$C_{12}H_{12}CLN_3$	7-chloro-n-(cyclopropylmethyl)quinazolin-4-amine	5.49	233.697	342	(Chu et al., 2010)
2dua	OXL	C_2O_4	oxalate ion	4.77	88.019	290	(Chen et al., 2006)
3v78	ET	$C_{21}H_{20}N_3$	ethidium	5.54	314.404	416	(Bolla et al., 2012)
3rdq	DTB	$C_{10}H_{18}N_2O_3$	6-(5-methyl-2-oxo-imidazolidin-4-yl)-hexanoic acid	9.31	214.262	153	(Magalhães et al., 2011)
1pot	SPD	$C_7H_{19}N_3$	spermidine	5.49	145.246	325	(Sugiyama et al., 1996)
2vuk	P83	$C_{16}H_{18}N_2$	1-(9-ethyl-9h-carbazol-3-yl)-n-methylmethanamine	3.90309	238.328	438	(Boeckler et al., 2008)
1nli	ADE	$C_5H_5N_5$	adenine	3.59	135.127	248	(Shaw et al., 2003)
3sus	GNL	$C_8H_{13}NO_4S$	(3ar,5r,6r,7r,7ar)-5-(hydroxymethyl)-2-methyl-5,6,7,7a-tetrahydro-3ah-pyrano[3,2-d][1,3]thiazole-6,7-diol	4.15	219.258	525	(Sumida et al., 2012)
2v58	LZJ	$C_{13}H_9BR_2N_5$	6-(2,6-dibromophenyl)pyrido[2,3-d]pyrimidine-2,7-diamine	9.08619	395.052	898	(Miller et al., 2009)
3nq3	DKA	$C_{10}H_{20}O_2$	decanoic acid	3.78	172.265	162	(Loch et al., 2011)
2v95	PDN	$C_{21}H_{26}O_5$	17,21-dihydroxypregna-1,4-diene-3,11,20-trione	8.3	358.428	371	(Klieber et al., 2007)
2i80	G1L	$C_{12}H_{13}CLF_3NO$	3-chloro-2,2-dimethyl-n-[4-(trifluoromethyl)phenyl]propanamide	5.4	279.686	720	(Liu et al., 2006)
1nki	PPF	CH_3O_5P	phosphonoformic acid	6.7	126.005	270	(Rigsby et al., 2004)
3b7j	JUG	$C_{10}H_6O_3$	5-hydroxynaphthalene-1,4-dione	5.16749	174.153	954	(Kong et al., 2008)
3f78	ICF	$C_3H_2CLF_5O$	1-chloro-2,2,2-trifluoroethyl difluoromethyl ether	3.1	184.492	543	(Zhang et al., 2009a)
3imc	BZ3	C_9H_9NO	5-methoxy-1h-indole	2.95861	147.174	602	(Hung et al., 2009)
3uj9	PC	$C_5H_{15}NO_4P$	phosphocholine	4.63	184.151	266	(Lee et al., 2012)
2rin	ACH	$C_7H_{16}NO_2$	acetylcholine	4	146.207	596	(Oswald et al., 2008)
3acw	651	$C_{19}H_{21}NO$	(3r)-3-biphenyl-4-yl-1-azabicyclo[2.2.2]octan-3-ol	4.76	279.376	293	(Lin et al., 2010)
2xn3	ID8	$C_{15}H_{15}NO_2$	2-[(2,3-dimethylphenyl)amino]benzoic acid	5.92	241.285	383	(Qi et al., 2011)
4ts1	TYR	$C_9H_{11}NO_3$	Tyrosine	4.94	181.19	638	(Brick and Blow, 1987)
3kgt	GEN	$C_{15}H_{10}O_5$	genistein	5.51	270.237	254	(Trivella et al., 2010)
4b0b	54F	$C_{11}H_{10}N_2O$	3-(pyridin-2-yloxy)aniline	3.33	186.21	342	(Moynié et al., 2013)
3rf5	FUZ	$C_{12}H_{11}NO_3$	2-[(furan-2-ylmethyl)amino]benzoic acid	5.63	217.221	348	(Cho et al., 2011)
3kiv	ACA	$C_6H_{13}NO_2$	6-aminohexanoic acid	4.7	131.173	79	(Mochalkin et al., 1999)
1t5f	DHH	$C_7H_{15}NO_4$	(s)-2-amino-7,7-dihydroxyheptanoic acid	4.22	177.198	942	(Shin et al., 2004)
1lrh	NLA	$C_{12}H_{10}O_2$	naphthalen-1-yl-acetic acid	6.82	186.207	652	(Woo et al., 2002)
1hp5	NGT	$C_8H_{13}NO_4S$	3ar,5r,6s,7r,7ar-5-hydroxymethyl-2-methyl-5,6,7,7a-tetrahydro-3ah-pyrano[3,2-d]thiazole-6,7-diol	6.55	219.258	512	(Mark et al., 2001)
4dkp	0LL	$C_{17}H_{15}CLFN_3O_2$	n-[(1s,2s)-2-amino-2,3-dihydro-1h-inden-1-yl]-n'-(4-chloro-3-fluorophenyl)ethanediamide	5.72	347.771	706	(LaLonde et al., 2012)
1ikt	OXN	$C_{34}H_{62}O_{11}$	oxtoxynol-10	3.4	646.849	120	(Haapalainen et al., 2001)
1xk9	P34	$C_{17}H_{17}N_3O_2$	n 2 ,n 2 -dimethyl-n 1 -(6-oxo-5,6-dihydrophenanthridin-2-yl)glycinamide	6.85	295.336	430	(Yates et al., 2005)
2v3d	NBV	$C_{10}H_{21}NO_4$	(2r,3r,4r,5s)-1-butyl-2-(hydroxymethyl)piperidine-3,4,5-triol	3.94	219.278	1010	(Brumshtein et al., 2007)
5yas	FAC	$C_3H_2F_6O_2$	1,1,1,3,3,3-hexafluoropropanediol	3.26	184.037	257	(Zuegg et al., 1999)
2y7k	SAL	$C_7H_6O_3$	2-hydroxybenzoic acid	4.93	138.121	872	(Devesse et al., 2011)
1qy2	IPZ	$C_8H_{12}N_2O$	2-isopropyl-3-methoxypyrazine	5.74	152.194	174	(Bingham et al., 2004)
3g0e	B49	$C_{22}H_{27}FN_4O_2$	n-[2-(diethylamino)ethyl]-5-[(z)-(5-fluoro-2-oxo-1,2-dihydro-3h-indol-3-ylidene)methyl]-2,4-dimethyl-1h-pyrrole-3-carboxamide	7.69897	398.474	336	(Gajiwala et al., 2009)
3n7h	DE3	$C_{12}H_{17}NO$	n,n-diethyl-3-methylbenzamide	4.5	191.269	250	(Tsitsanou et al., 2012)
1hnn	SKF	$C_9H_{12}N_2O_2S$	1,2,3,4-tetrahydro-isoquinoline-7-sulfonic acid amide	6.24	212.269	564	(Martin et al., 2001)
3h78	BE2	$C_7H_7NO_2$	2-aminobenzoic acid	4.24413	137.136	718	(Bera et al., 2009)
3g0w	LGB	$C_{15}H_{13}CLF_3N_3O_2$	2-chloro-4-[(1r,3z,7s,7as)-7-hydroxy-1-(trifluoromethyl)tetrahydro-1h-pyrrolo[1,2-c][1,3]oxazol-3-ylidene]amino-3-methylbenzonitrile	9.52288	359.731	260	(Nirschl et al., 2009)
1drj	RIP	$C_5H_{10}O_5$	ribose(pyranose form)	7.4	150.13	271	(Björkman et al., 1994)
1df8	BTN	$C_{10}H_{16}N_2O_3S$	biotin	9.7	244.311	254	(Hyre et al., 2000)
2gkl	PD2	$C_7H_5NO_4$	pyridine-2,4-dicarboxylic acid	5.35	167.119	227	(Horsfall et al., 2007)
2buv	DHB	$C_7H_6O_4$	3,4-DIHYDROXYBENZOIC ACID	4	154.12	450	(Brown et al., 2004)
2cgf	P2N	$C_{17}H_{19}ClO5$	(5z)-13-chloro-14,16-dihydroxy-3,4,7,8,9,10-hexahydro-1h-2-benzoxacyclotetradecine-1,11(12h)-dione	6.41	338.783	225	(Proisy et al., 2006)
1ft7	PLU	$C_5H_{14}NO_3P$	leucine phosphonic acid	5.18	167.143	291	(Stamper et al., 2001)
1bju	GP6	$C_{14}H_{13}CLN_4O$	1-(4-amidinophenyl)-3-(4-chlorophenyl)urea	4.8	288.732	223	(Presnell et al., 1998)
3lk1	JKE	$C_7H_6O_2S$	2-sulfanylbenzoic acid	4.26	154.19Ê	90	(Wilder et al., 2010)
1cbx	BZS	$C_{11}H_{12}O_4$	l-benzylsuccinic acid	6.35	208.211	307	(Mangani et al., 1992)
3lc3	IYX	$C_{17}H_{18}N_2O_2S$	1-[5-(3,4-dimethoxyphenyl)-1-benzothiophen-2-yl]methanediamine	5.23	314.402	584	(Wang et al., 2010)
1x8d	RNS	$C_6H_{12}O_5$	l-rhamnose	2.21	164.156	416	(Ryu et al., 2005)
2pcp	1PC	$C_{17}H_{25}N$	1-(phenyl-1-cyclohexyl)piperidine	8.7	243.387	862	(Lim et al., 1998)
3uxd	0CU	$C_6H_3CL_2N_3$	5,7-dichloro-1h-benzotriazole	4.41	188.014	504	(Khersonsky et al., 2012)
1jys	ADE	$C_5H_5N_5$	adenine	3.52	135.127	484	(Lee et al., 2001)
3b4p	3B4	$C_{13}H_{17}NO_2$	2-(cyclohexylamino)benzoic acid	5.39794	219.28	370	(Ahuja et al., 2008)
3kjd	78P	$C_{13}H_{16}N_4O$	(2r)-2-(7-carbamoyl-1h-benzimidazol-2-yl)-2-methylpyrrolidinium	8.54	244.292	736	(Karlberg et al., 2010a)
1wm1	PTB	$C_{11}H_{17}N_3O_2$	(5-tert-butyl-1,3,4-oxadiazol-2-yl)[(2r)-pyrrolidin-2-yl]methanone	6.3	223.272	317	(Inoue et al., 2003)
2q88	4CS	$C_6H_{10}N_2O_2$	(4s)-2-methyl-1,4,5,6-tetrahydropyrimidine-4-carboxylic acid	5.79588	142.156	257	(Hanekop et al., 2007)
3jrs	A8S	$C_{15}H_{20}O_4$	(2z,4e)-5-[(1s)-1-hydroxy-2,6,6-trimethyl-4-oxocyclohex-2-en-1-yl]-3-methylpenta-2,4-dienoic acid	4.284	264.317	624	(Miyazono et al., 2009)
3ebl	GA4	$C_{19}H_{24}O_5$	gibberellin a4	6.30103	332.391	2190	(Shimada et al., 2008)
2ra6	ETY	$C_8H_{10}O$	4-ethylphenol	4.52288	122.164	664	(Watson et al., 2007)
1upf	URF	$C_4H_3FN_2O_2$	5-fluorouracil	4.6	130.077	896	(Schumacher et al., 1998)
1gpk	HUP	$C_{15}H_{18}N_2O$	huperzine a	5.37	242.316	537	(Dvir et al., 2002)
3b92	440	$C_{13}H_{16}O_3S_2$	3-[4-(but-2-yn-1-yloxy)phenyl]sulfonylpropane-1-thiol	8	284.394	259	(Bandarage et al., 2008)
3qqs	17C	$C_{14}H_{11}NO_4$	2,2'-iminodibenzoic acid	5.82	257.241	1508	(Castell et al., 2013)
2aac	FCB	$C_6H_{12}O_5$	beta-d-fucose	2.22	164.156	354	(Soisson et al., 1997)
3ip8	B85	$C_7H_9O_3P$	benzylphosphonic acid	2.284	172.118	248	(Okrasa et al., 2009)
1oif	IFM	$C_6H_{13}NO_3$	5-hydroxymethyl-3,4-dihydroxypiperidine	7.72	147.172	936	(Zechel et al., 2003)
1i7z	COC	$C_{17}H_{21}NO_4$	cocaine	6.4	303.353	878	(Larsen et al., 2001)
3hp9	CF1	$C_{15}H_{11}CLF_3NO_3$	2-[2-chloro-5-(trifluoromethyl)phenyl]amino-5-methoxybenzoic acid	4.59	345.701	482	(Lu et al., 2010)
1gcz	YZ9	$C_{12}H_{10}O_5$	7-hydroxy-2-oxo-chromene-3-carboxylic acid ethyl ester	5.13	234.205	366	(Orita et al., 2001)
2rkd	3PP	$C_3H_7O_5P$	3-phosphonopropanoic acid	2.72125	154.058	624	(Stiffin et al., 2008)
2gvv	DI9	$C_{10}H_{20}NO_3P$	dicyclopentyl phosphoramidate	3.9	233.244	314	(Blum et al., 2006)
1e4h	PBR	C_6HBr_5O	pentabromophenol	8.41	488.592	254	(Ghosh et al., 2000)
2fpz	270	$C_7H_7N_3$	2h-benzoimidazol-2-ylamine	3.96	133.151	980	(McGrath et al., 2006)

Table IV.1: The PDB IDs of the protein-ligand complexes selected for use in the training dataset.

laid out in detail in Chapter F. While crystal structures were available for the active protein-ligand pairs, high quality RosettaLigand models were used to expose the model to the types of noise introduced in the process of RosettaLigand docking.

As this set does not contain any known inactive compounds, the inactive component of the training set was generated through cross-docking. By using the same set of ligands for both the active and inactive datasets, it is impossible for the ANN to learn any algorithm used to select inactive compounds. For this reason, we can use this approach to avoid some of the challenges described in Section IV.1.5.2. Each ligand in the 120-compound set was docked into each of the proteins in the set except the one with measured activity data. Due to the size of chemical space (Reymond et al., 2012) and the fact that each ligand in the set of active ligands binds natively to a protein from a different family, we assume that every cross-docked complex has no (or minimal) binding affinity. The lowest scoring Rosetta model for each cross-docked complex was selected, and the resultant set of 59,295 protein-ligand complexes will comprise the inactive component of the training set.

IV.2.2 Development and description of ligand descriptors

IV.2.2.1 Sources of descriptor data

In addition to the design of a training model, the proper selection of descriptors is critical for the training of an effective model. It is frequently difficult to determine *a priori* which descriptors will be the most useful, so three classes of descriptor information were evaluated. Specifically: scalar scores and statistics describing the protein-ligand interface, RDF descriptors representing the arrangement of atoms in the protein-ligand interface, and scalar metrics describing the ligand chemistry.

IV.2.2.2 Scalar protein-ligand interface descriptors

The RosettaLigand energy function directly provides a number of metrics that can be used as scalar descriptors of the chemistry of the protein-ligand interface. In addition to these scores, Rosetta implements an “Interface Analyzer”, which generates additional descrip-

Rosetta energy descriptors	
Property name	**Description**
if_X_fa_atr	Attractive force of the protein-ligand interface
if_X_fa_rep	Repulsive force of the protein-ligand interface
if_X_fa_intra_rep	Repulsive force between atoms in each residue and the ligand
if_X_fa_elec	Electrostatic force of the protein-ligand interface
if_X_fa_pair	The value of the Rosetta pair energy between protein and ligand residues
if_X_fa_sol	The desolvation energy of the protein-ligand interface
if_X_hbond_bb_sc	The hydrogen bonding energy between protein backbone atoms and ligand atoms
if_X_hbond_sc	The hydrogen bonding energy between protein side chain atoms and ligand atoms
hbond_lr_bb	The long range hydrogen bonding energy between protein backbone atoms in the entire complex
hbond_sc	The hydrogen bonding energy between all side chain atoms in the entire complex
hbond_sr_bb	The long range hydrogen bonding energy between protein backbone atoms in the entire complex
interface_delta_X	The total Rosetta score associated with the protein-ligand interface
total_score/nres_all	The total complex score divided by the total number of protein residues
Rosetta Interface Analyzer descriptors	
Property name	**Description**
dSASA_hydrophobic	The hydrophobic SASA at the protein-ligand interface
dSASA_int	The total SASA at the protein-ligand interface
dSASA_polar	The polar SASA at the protein-ligand interface
delta_unsatHbonds	The number of unsaturated hydrogen bonds at the protein-ligand interface
hbond_E_fraction	The fraction of the interface energy associated with hydrogen bonding
nres_int	The number of protein residues at the protein-ligand interfaec
packstat	The protein-ligand interface packing statistic originally described by Sheffler (Sheffler and Baker, 2009)

Table IV.2: A summary of the names and definitions of the scalar descriptors generated by Rosetta. Originally described by Rohl (Rohl et al., 2004)

tors of the protein-ligand interface. Between these two descriptor sources, a set of 20 descriptors can be computed, describing the van der Waals, hydrogen-bonding, desolvation and electrostatic energy of the protein-ligand interface, as well as the size of the protein binding pocket, the number of unsatisfied hydrogen-bonds, and the SASA of the interface. Table IV.2 summarizes the specific scalar interface descriptors used.

IV.2.2.3 Scalar ligand descriptors

The scalar descriptors discussed in Section IV.2.2.2 encode only information related to the protein-ligand interface. To provide information about the chemistry of the ligand, an additional set of ligand descriptors was computed using the BCL (Butkiewicz et al., 2013). These descriptors provide information about the weight of the ligand, the number of hydrogen-bond donors and acceptors, predicted logP, number of rings, number of rotatable bonds and the circumference of the ligand around the widest dimension. Table IV.3 summarizes the scalar ligand descriptors. The number of descriptors used to describe the

Property name	**Description**
Weight	The molecular weight of the ligand
HbondDonor	The number of hydrogen bond donors in the ligand
HbondAcceptor	The number of hydrogenn bond acceptors in the ligand
LogP	The predicted logP (partition coefficient) of the ligand
TotalCharge	The total calculated charge of the ligand
NRotBond	The number of rotatable bonds of the ligand
NAromaticRings	The number of aromatic rings of the ligand
NRings	The total number of rings of the ligand
TopologicalPolarSurfaceArea	The topological polar surface area of the ligand
Girth	The circumference of the ligand around the widest dimension

Table IV.3: A summary of the names and definitions of the scalar descriptors generated by the BCL.

ligand is relatively small compared to previous machine learning studies performed using the BCL (Mueller et al., 2010). The rationale for including a smaller number of descriptors is that a relatively small number of ligands is being used to train the classifier. As a result, the descriptions of the ligand chemistry used should be broad enough that the range of the descriptor space is well covered.

IV.2.2.4 Protein-ligand fingerprint descriptors

In addition to the scalar descriptors discussed in Sections IV.2.2.2 and IV.2.2.3, a novel set of protein-ligand interface descriptors were added. These fingerprint descriptors are implemented as RDFs, which take the following form:

$$g(r) = \sum_{i,j} score_{ij} e^{-B(r-r_{ij})^2} \tag{IV.1}$$

Where i and j are a protein and ligand atom respectively, $score_{ij}$ is a score computed based on those two atoms, B is a smoothing factor, r is the radius of the sphere being currently considered, and r_{ij} is that distance between the two atoms. The function $g(r)$ computes the RDF for a single distance. To compute the complete fingerprint, $g(r)$ is computed for a range of values of of r. The resulting fingerprint represents the probability of two atoms existing within a sphere of radius r with some property. More broadly, these fingerprints can be interpreted as a 1-D representation of the 3-D distribution of geometric and chemical

Property name	Description
atr_interface_rdf	RDF using the Rosetta attractive score between pairs of atoms
solv_interface_rdf	RDF using the Rosetta desolvation score between pairs of atoms
elec_interface_rdf	RDF using the Rosetta electrostatic score between pairs of atoms
rep_interface_rdf	RDF using the Rosetta repulsive score between pairs of atoms
hbond_acceptor_interface_rdf	RDF using the Rosetta hydrogen bonding term between ligand acceptor atoms and protein donors
hbond_donor_interface_rdf	RDF using the Rosetta hydrogen bonding term between ligand donor atoms and protein acceptors
charge_minus_interface_rdf	RDF using the product of atom charges where the ligand atom is negatively charged and the protein atom positively charged
charge_plus_interface_rdf	RDF using the product of atom charges where the ligand atom is positively charged and the protein atom negatively charged
charge_unsigned_interface_rdf	RDF using the product of atom charges where the sign of both atom charges is matched
hbond_binary_acceptor_interface_rdf	RDF which returns 1.0 if the ligand atom is a hydrogen bond acceptor and the protein atom is a donor
hbond_binary_donor_interface_rdf	RDF which returns 1.0 if the ligand atom is a hydrogen bond donor and the protein atom is a acceptor
hbond_matching_pair_interface_rdf	RDF which returns 1.0 if the ligand and protein atom are both donors or both acceptors

Table IV.4: A summary of the names and definitions of the RDF fingerprint descriptors generated by Rosetta.

properties in the protein-ligand interface. A range of fingerprints were computed using this method, with various chemical properties used to compute $score_{ij}$.

Fingerprints are computed using the attractive, repulsive, electrostatic, solvation, and hydrogen-bonding scores used by Rosetta. Additionally, a charge-based function is implemented, in which $score_{ij}$ is computed as the product of the charges of each atom pair, and a hydrogen-bond count function is computed, in which $score_{ij}$ is 1.0 if the pair of atoms are a hydrogen-bond donor and acceptor, and 0.0 otherwise. The fingerprints are computed directly by Rosetta. Table IV.4 summarizes the RDF fingerprints computed by Rosetta. Based on previous experience with RDF fingerprints in the BCL, all fingerprints are computed using 24 evenly spaced distance steps between 0.0 and 6.0 Å. The smoothing factor B was set to 100. This fingerprint parameters (B, bin count, and overall fingerprint radius), were not rigorously optimized in this study. In future work, optimization of both the bin width and the smoothing factor should be performed. For the bin width of 0.25, the smoothing factor of 100 may be too high and result in loss of information. A smoothing factor of around 50 may be preferable, based on previous experience in the Meiler lab. The 12 RDF functions resulted in a total of 288 descriptor columns.

IV.2.3 ANN training protocol

IV.2.3.1 Cross validation scheme

In addition to selecting a set of input descriptors and a training dataset, a reasonable training mechanism and neural network architecture must be selected. As described in the introductory chapter, ANNs are prone to overtraining, and the proper design of a training protocol is critical to avoid this problem. Based on previous experience using ANNs to predict drug activity (Mueller et al., 2010, 2012; Butkiewicz et al., 2013), a 10-fold cross-validation was used. In this scheme, the combined set of active and inactive compounds described in Section IV.2.1 was randomized and split into 10 evenly sized blocks. In each round of cross validation, one block is selected as an "independent" set, one block is selected as a "monitoring" set, and the remaining blocks are selected for training. Figure IV.2 provides a schematic illustration of this cross validation scheme. Note that the cross validation is set up such that each block in the dataset plays a role as both an independent and a monitoring set. During training, the training dataset is used to train the ANN, and after every iteration of training, enrichment is calculated using the monitoring set. At the conclusion of the training process for each round of cross validation, the model with the best enrichment according to the monitoring dataset is output. The 10 fold cross-validation training scheme results in an ensemble of 90 models. When all rounds of cross validation are complete, the final enrichment of the ensemble of models is computed using the independent dataset from each round of cross validation. By using this type of cross validation, it is possible to create an ensemble of models that cover the complete range of training data while confirming that over-training is not taking place.

IV.2.3.2 Network architecture and training

For the purposes of this study, a feed-forward network with two hidden layers of 100 nodes each was used. The network was trained using a back-propagation algorithm with network dropout (Hinton et al., 2012). At each iteration, network dropout disabled 12.5% of input

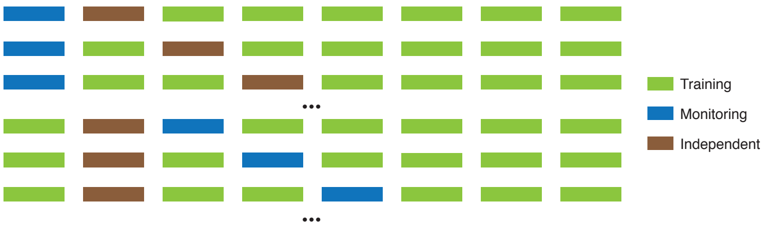


Figure IV.2: A schematic of the cross validation scheme used. The dataset is partitioned, and sufficient rounds of cross validation are performed such that every block in the partition is used for training, monitoring, and independent validation.

nodes and 50% of hidden nodes. The network dropout method described by Hinton *et al.* allows for larger numbers of hidden nodes to be used relative to a traditional ANN architecture. However, in future study, the size of the network should potentially be reduced, as the training dataset used in this study may not be large enough to support a network of this size. The purpose of network dropout is to prevent the neural network from becoming dependent on the relationships between specific input and hidden nodes in its representation of the model. This so-called "co-adaptation" can contribute to over-fitting, and thus network dropout makes it possible to conduct many more iterations of network training without over-fitting. The network was trained to classify active and inactive ligands, where activity is measured as log(K$_i$). A log(K$_i$) cutoff of 0.5 was used, and average enrichment was selected as a metric of classification. Here, we define enrichment as:

$$enrichment = \frac{TP}{TP+FP} / \frac{P}{P+N} \tag{IV.2}$$

Where TP and FP are the true positive and false positive rate, P is the total number of positives, and N is the total number of negatives. Enrichment is typically computed using a cutoff, and average enrichment is computed as the mean enrichment over a range of cutoffs. In this case the cutoff range used is between 0.0-1.0% of the total database. The goal of the average enrichment metric is to have as many true positives as possible relative to false positives within the first 1% of models selected. The network is trained for 800 iterations, and the model with the highest average enrichment according to the monitoring data block is selected. A strategy for more rigorous optimization of ANN network parameters is described in Section V.2.2.2.

IV.2.4 Summary of Results

IV.2.4.1 Summary of networks trained

Several networks were trained using a variety of input descriptors. The "Rosetta scalar" network was trained using only the Rosetta generated scalar descriptors in table IV.2, the "Rosetta and BCL scalar" network was trained using the Rosetta scalar descriptors combined with the BCL descriptors in table IV.3, and the "Rosetta fingerprint + scalar" network is trained using both the Rosetta scalar, and Rosetta fingerprint descriptors in table IV.4. As a control, the "BCL scalar" network is trained using only the BCL descriptors. Because the set of training data is balanced in chemical space, we expect that the BCL scalar network should not achieve any reasonable enrichment, as no signal should be available for classification.

IV.2.4.2 Results of network training

Because the networks described in Section IV.2.4.1 were trained using a cross-validation scheme, the performance of each of the 90 models generated can be evaluated using the independent dataset for each model. The 90 independent predictions produced by the ensemble of models were merged to produced a single set of independent predictions spanning the entire training dataset. These prediction sets were then compared to the classification performance obtained by using the RosettaLigand interface score. Three prediction performance metrics are presented here: Enrichment (Equation IV.2), Positive Predictive Value (PPV), Receiver Operating Characteristic Area Under Curve (ROC-AUC) and log(AUC). As described previously, enrichment provides a metric for the ability of the model to correctly make positive predictions early on. Receiver Operating Characteristic (ROC) are measurements of the overall performance of the classifier, which provide a convenient means of visualizing classification performance.

IV.2.4.3 Description of the ROC-AUC metric

To compute a ROC curve, the True Positive Rate (TPR) is computed as $TPR = TP/P$ where TP is the number of true positive predictions, and P is the number of total positive values in the given dataset, and the False Positive Rate (FPR) is computed as $FPR = FP/N$ where FP is the number of false positive predictions, and N is the total number of negative values in the dataset. The predictions made by each model are sorted by predicted score, with the best scores first, and the TPR and FPR values are computed for each cumulative fraction of the sorted dataset. The resulting curve provides a metric of the overall classification performance. The area under the curve can be computed by integration, resulting in a value between 0.0 and 1.0. A ROC-AUC value of 1.0 indicates a perfect classifier, a value of 0.5 indicates a classifier with a performance equivalent to a coin-toss, and a value of 0.0 indicates a classifier which is always incorrect.

Additionally, we can compute the log(Area Under the Curve (AUC)). By taking the log of the false positive rate, we effectively compute the stringency of the model (Clark and Webster-Clark, 2008). In other words, the log(AUC) corresponds to the average negative logarithm of the false positive rate. Thus, higher values of log(AUC) are more desirable for our purposes. In the analysis of these models we will use both ROC-AUC and log(AUC) as methods for evaluating model performance. Additionally, we take the log of the FPR when plotting ROC curves, to more effectively illustrate the differences in performance between the models during the early stages of screening.

IV.2.4.4 Description of the PPV metric

PPV is a measure of the accuracy of a classifier. PPV is computed as $PPV = TP/TP + FP$, and can be interpreted as the fraction of positive predictions that are actually positive. Thus, higher PPV indicates a more accurate classifier.

Classifier	Average Enrichment	ROC-AUC	log(AUC)
Rosetta Interface scores	27.88	0.83	1.34
ANN: BCL Scalar descriptors	0.04	0.54	0.49
ANN: Rosetta and BCL Scalar descriptors	54.28	0.850	1.55
ANN: Rosetta Fingerprint and Scalar descriptors	42.62	0.849	1.48
ANN: Rosetta Scalar descriptors	54.69	0.845	1.51

Table IV.5: ROC-AUC and average enrichment for the classification models being evaluated. The Rosetta Interface Scores classifier uses only the sorted RosettaLigand interface scores for classification. all "ANN" classifiers are neural nets using the specified descriptors. ROC-AUC is the area under the ROC curve generated from each descriptor (Figure IV.3) Average enrichment is the average enrichment within the first 1% of the each dataset.

IV.2.4.5 Summary of classifier performance

The ROC and PPV metrics can be used to produce a concise visual comparison of the performance of the various evaluated networks. Figure IV.3 plots ROC curves formed using the networks trained in Section IV.2.4.1, as well as a classifier which consists entirely of the RosettaLigand interface scores. In this experiment, the RosettaLigand interface score based classifier and the "BCL Scalar descriptor" network act as controls. We expect a successful ANN to have significant improvement compared to the RosettaLigand interface score classifier and that the BCL scalar descriptor network have performance roughly equal to a random coin toss. As shown in the figure, we see that this is indeed the case. The three networks trained using Rosetta interface information all exhibit similar ROC curve parameters, all of which are significantly improved over the RosettaLigand interface score classifier. As expected, the BCL scalar descriptor has no classification ability. Table IV.5 lists the ROC-AUC and average enrichment of each of the plotted classifiers. We see that the three networks trained with RosettaLigand interface score information have similar performance in terms of average enrichment, ROC-AUC, and log(AUC). The three non-control ANN classifiers evaluated show substantially improved enrichment over the RosettaLigand interface score classifier, and slightly improved ROC-AUC and log(AUC) performance. Specifically, the ROC-AUC of the ANN classifiers is increased by 0.026-0.036 over the RosettaLigand interface classifier, the log(ROC) is increased by 0.14-0.21 and the average Enrichment is increased by 14.74-26.81.

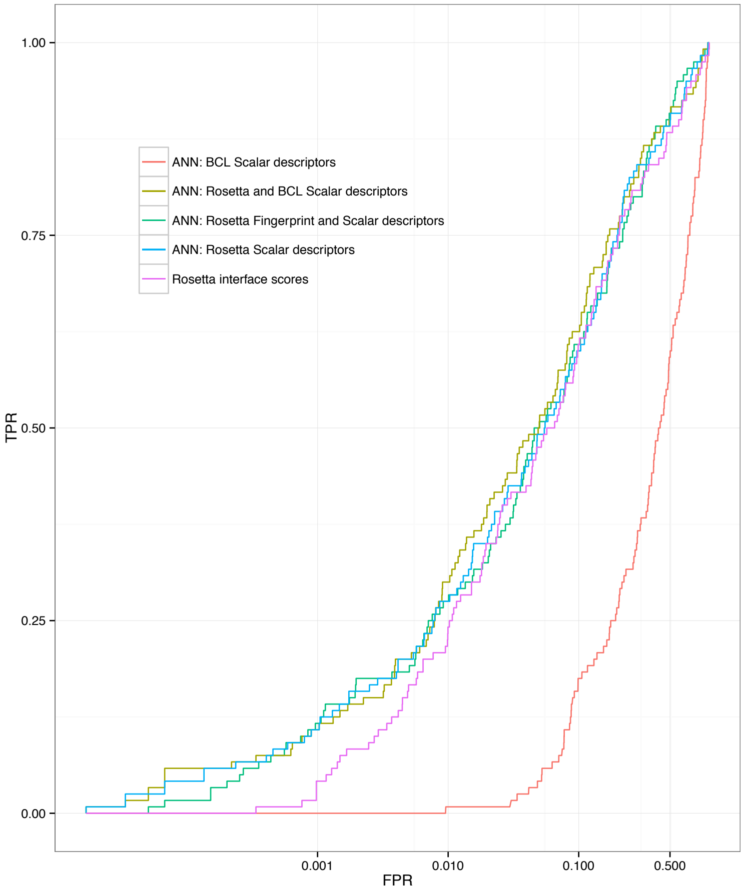


Figure IV.3: ROC curves showing the performance of the various networks trained using the 120 protein training set. Performance is plotted using the independent dataset from each of the 90 neural networks used. The ROC curve is plotted as the ratio of TPR to FPR. To accentuate the differences in early classification, the X axis is plotted on a log scale.

While the three ANN models show similar values for enrichment and AUC, inspection of the PPV of the models demonstrates significant performance differences. Figure IV.4 plots the PPV of each model as a function of the FPR. These plots provide a visual depiction of the accuracy of each classifier. We can see from these models that the models using Rosetta scalar descriptors or a combination of Rosetta and BCL scalar descriptors have significantly improved performance relative to the model using Rosetta scalar and fingerprint information. Specifically, the peak PPV drops from 0.50 to 0.25. This suggests that the introduction of fingerprint data results in a loss of model accuracy early in screening.

IV.2.4.6 Classification vs. regression models

The bulk of the analysis in this Chapter involves binary classification models. However, ANNs trained to predict K_i directly rather than simple binary classification were also investigated. As with the models described above, the filtered set of 120 protein-ligand pairs was used as a training set, and an ensemble of neural networks was trained using 90-fold cross-validation. However, rather than optimizing the networks to maximize classification enrichment, the RMSD between the predicted activity and experimental K_i was minimized instead. Figure IV.5 plots the correlation between the scores generated by ANNs trained to predict ligand activity, as well as Rosetta interface score alone. We see from these figures that there is minimal correlation between experimental K_i and model score. The Pearson correlations of the six classifiers depicted in Figure IV.5 are shown in Table IV.6, in no case is there a significant correlation. This lack of performance stands in contrast to the more successful classification models described above. The failure of these models to accurately predict K_i does not invalidate the overall concept of a regression model for ligand binding affinity. The lack of known experimental K_i values for non-binding ligands necessitated the use of 0.0 as a K_i value for the inactive ligands in the training data. As these ligands almost certainly do not have a binding affinity of exactly 0, it is likely that this practice introduces noise into the model, reducing the quality of the overall correlation. A useful

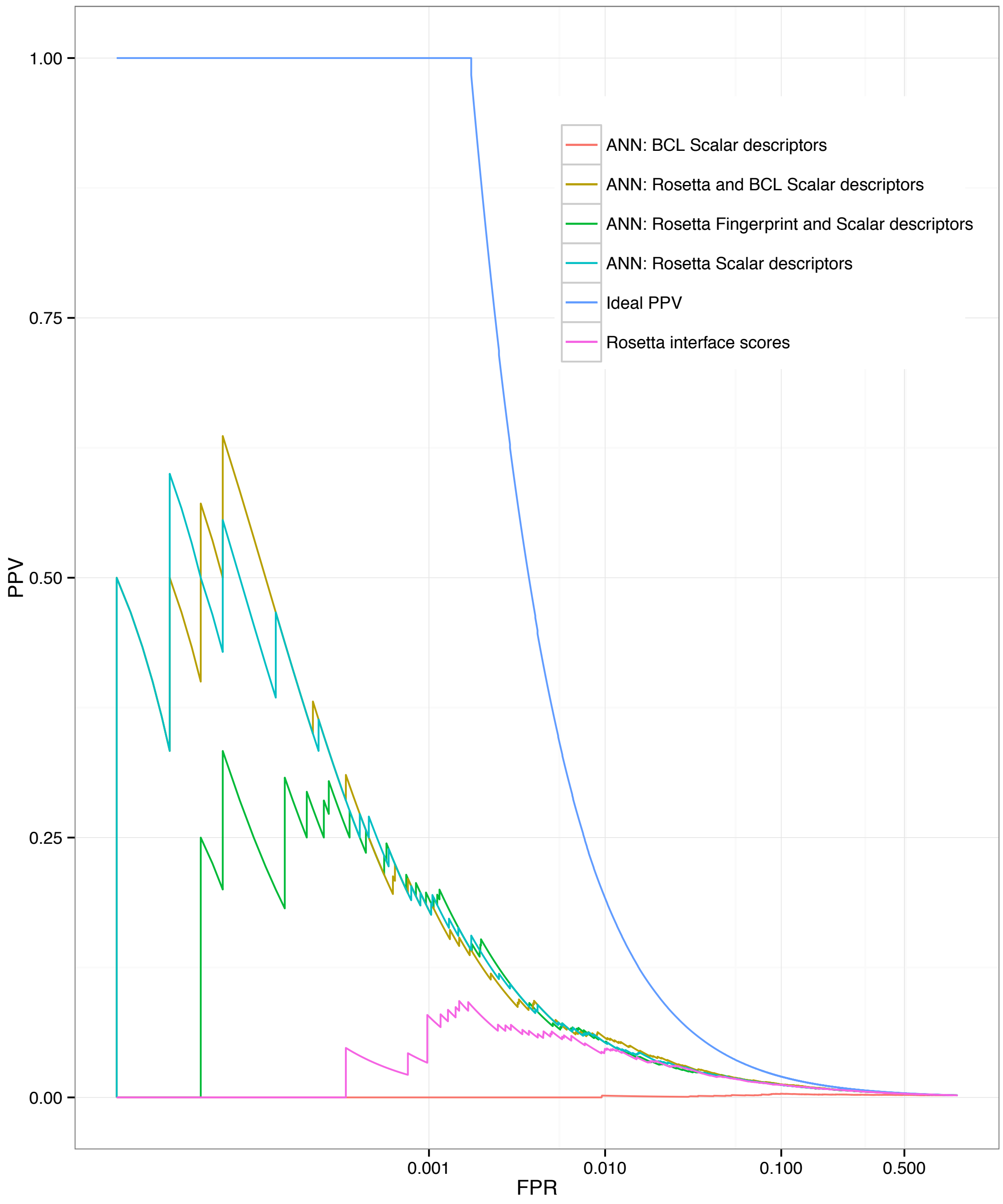


Figure IV.4: Plots showing the PPV as a function of FPR for the various networks trained using the 120 protein training set. Performance is plotted using the independent dataset from each of the 90 neural networks used. The PPV vs FPR curve of an ideal classifier is also plotted, for reference. To accentuate the differences in early classification, the X axis is plotted on a log scale.

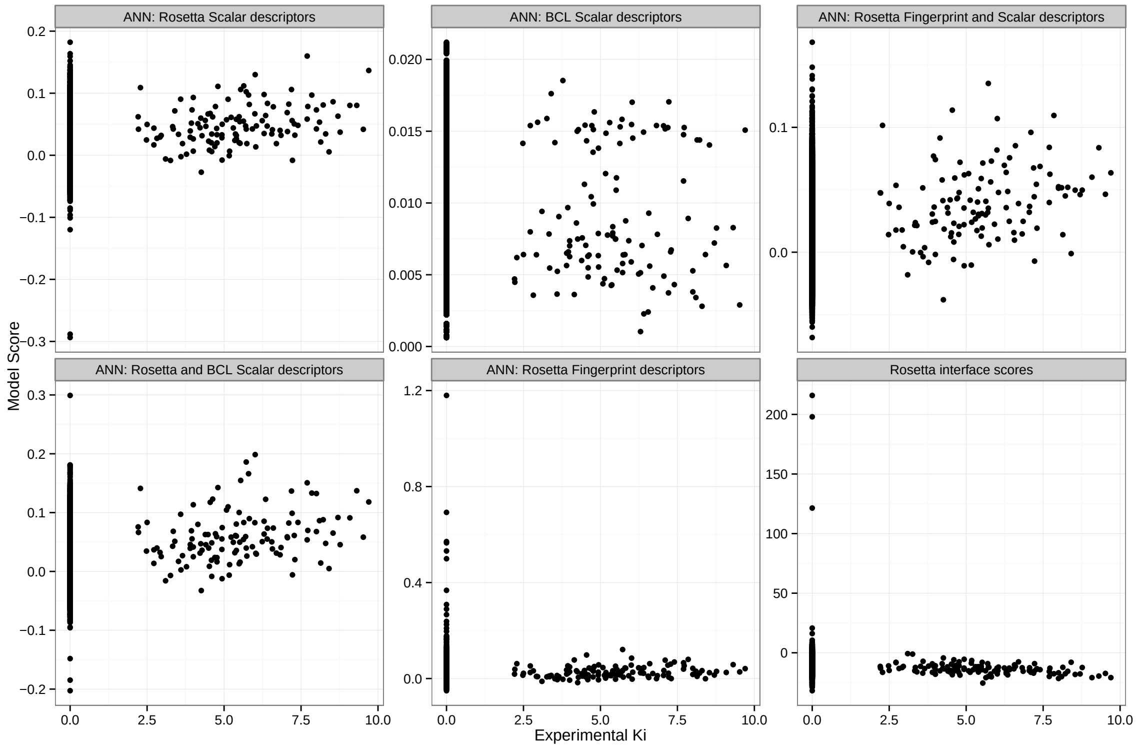


Figure IV.5: A plot of the correlations between ANN and Rosetta model score and experimental K$_i$.

future experiment to address this concern would be obtain a larger set of known active ligands with a wide range of experimental K$_i$ values. This set of ligands would be used to train a regression model over active ligands only and would be used in conjunction with the classification model. The classification model would then be used in an initial filtering stage to identify likely active compounds, and the K$_i$ prediction model would be used as refinement stage to rank the likely active compounds by binding affinity for further investigation. Successfully training this model would likely require a far larger set of ligands than the 120 protein-ligand pairs used in this study.

IV.2.4.7 Benchmarking of trained networks using DEKOIS 2.0

The performance metrics described in Section IV.2.4.5 clearly indicate that the ANN-based classifiers have some value at classifying ligand activity beyond the RosettaLigand interface score and have not been over-trained. However, the ability of the networks to gen-

Model Name	**Pearson Correlation**
ANN: BCL Scalar descriptors	-0.0005
ANN: Rosetta and BCL Scalar descriptors	0.0873
ANN: Rosetta Fingerprint descriptors	0.0385
ANN: Rosetta Fingerprint and Scalar descriptors	0.0594
Rosetta interface scores	-0.0654

Table IV.6: The Pearson correlations between experimentally measured K_i and model score.

eralize beyond the training dataset needs to be assessed. In order to answer the question of whether the ANN models demonstrated here are capable of making general predictions, the DEKOIS 2.0 (Bauer et al., 2013) dataset was used for benchmarking. All proteins and ligands were prepared in the same manner as the training dataset, and the best scoring model for each protein-ligand complex was selected. The DEKOIS 2.0 benchmarking set consists of 81 sets of known active and predicted inactive ligands across 80 proteins. As DEKOIS 2.0 is intended for the benchmarking of structure based screening methods, all known active ligands were confirmed to bind the same binding site on the protein. Unlike the training dataset in which putative inactives were generated by cross-docking, the inactive compounds in the DEKOIS benchmarking set are generated through a parameter matching approach, which is described in detail by Bauer *et al.* (Bauer et al., 2013).

As ANN models are known to be relatively ineffective at making predictions on input data outside of the range of their training set, the original DEKOIS 2.0 dataset was filtered using Lipinski's rule to contain only small drug like molecules with a range of properties similar to that in the initial set. This filtering process resulted in a total of 16,080 active and inactive ligands across 74 of the original 80 protein systems. The DEKOIS 2.0 dataset has a low overlap with the training dataset. Of the 120 ligands in the training set described in Section IV.2.1.2, 3 were identical to any ligand in the DEKOIS 2.0 set. Despite this, there is broad similarity in chemical properties between the 120 ligands in the training dataset and the ligands in the DEKOIS benchmarking set, as can be seen by comparing figures IV.6 and IV.1. This is important, as a low rate of structural overlap between training and benchmarking datasets is necessary to avoid artificially inflating success rates by allowing

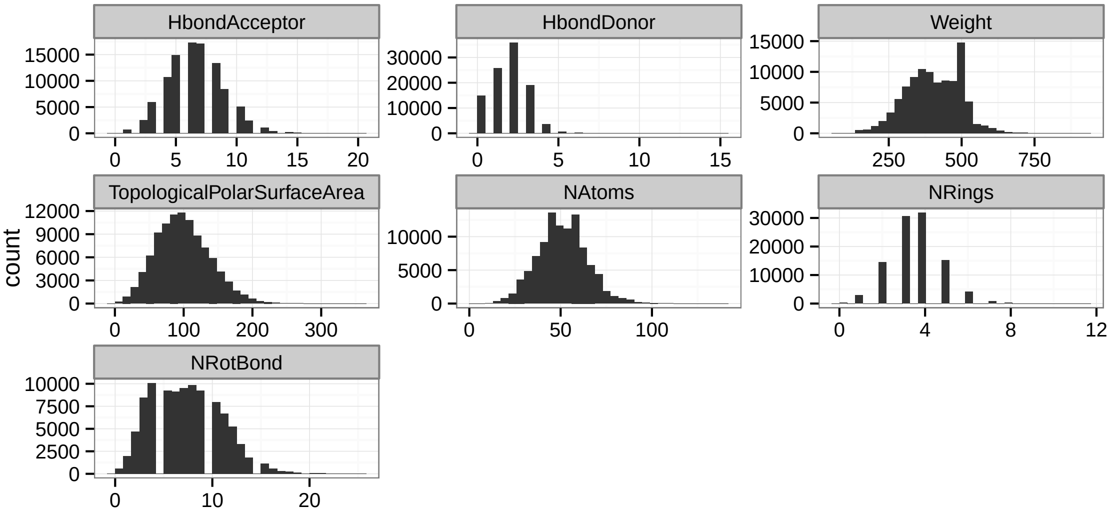


Figure IV.6: The basic property distribution of ligands in the DEKOIS 2.0 benchmarking dataset. Histograms are plotted of the hydrogen-bond donor and acceptor count, atom count, rotatable bond count, ring count, Topological Polar Surface Area, log(K_i) and molecular weight of the ligands.

the ANN to predict what it has already seen.

First, the classifiers were used to screen the entire DEKOIS 2.0 benchmarking set, and ROC curve plots were generated using each classifier. As the DEKOIS benchmarking set does not include experimentally determined binding information for each ligand, RosettaLigand was used to provide models as input to the classifiers. The protocol used to obtain models for this screen is described in Chapter F. Specifically, 200 models were created for each protein-ligand complex, and the lowest scoring model by RosettaLigand score was scored using as input into the classifiers. The protocol used here is identical to the protocol used to generate both the active and inactive models used for classifier training. The ROC curves are shown in Figure IV.8. Table IV.7 summarizes the average enrichment, ROC-AUC and log(AUC) for the benchmark. As expected, the control network (ANN: BCL Scalar Descriptors) has performance equivalent with a random classifier ROC-AUC and log(AUC) Specifically, we see a ROC-AUC of 0.48 and a log(AUC) of 0.34. The introduction of Rosetta scoring information into the ANN slightly increases ROC-AUC by

0.15-0.20, and log(AUC) by 0.28-0.40. There is no significant change in the Average Enrichment. These performance metrics were computed based on the combined set of all 16080 actives and inactive compounds described in Section IV.2.4.7. As shown in Chapter III, RosettaLigand is not capable of successfully docking ligands into all protein systems. As a result, the relatively low enrichments seen here are likely a combination of both limitations in the RosettaLigand docking algorithm, and the ANN classifiers themselves.

We can obtain further insight into the details of the ANN model performance by evaluating each of the 74 protein systems independently. It is possible that there are some sets of compounds that are easier or more difficult for the model to predict. Understanding any differences between the properties of successfully and unsuccessfully docked ligands could potentially lead to insights into how the models could be improved. Additionally, if there is a distinct subset of chemical space for which ligand activity can be effectively predicted, the models may have value in screening of those parts of chemical space. The distribution of ROC-AUC across each of the 74 protein-ligand systems is plotted in Figure IV.7A. As expected, the model created using only BCL ligand descriptors has no predictive power, and the three ANN-based classifiers have similar but positive predictive power. Figure IV.7B shows the Rosetta fingerprint and scalar model, the RosettaLigand interface score model, and the BCL-only descriptors. As expected, the non-control models have slightly better performance than the random model generated with only BCL ligand based descriptors. However, there appears to be no significant difference between the ANN based models and the use of Rosetta scores as a classifier. Further analysis and discussion of these results is presented in Section IV.3.2.

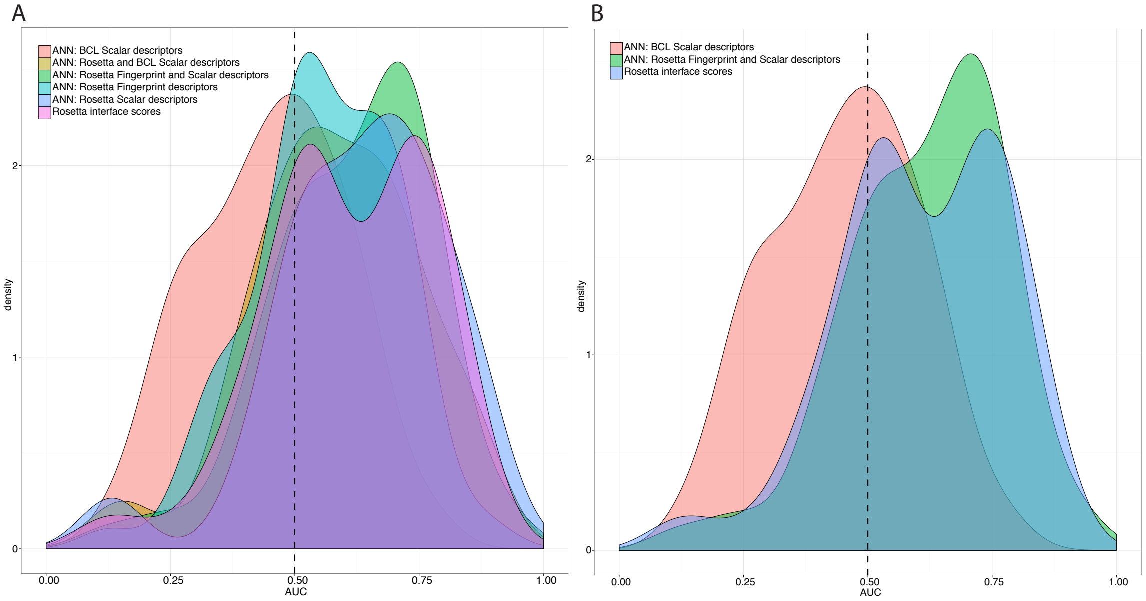


Figure IV.7: A) A plot of the distribution of ROC-AUC values for each of the 74 protein targets in the DEKOIS 2.0 benchmarking set when models were re-scored with each of the classifiers being evaluated. B) A plot of the distributions for 3 of the evaluated models. The dotted vertical line indicates the ROC-AUC associated with a random model. ROC-AUC values less than 0.5 are worse than random.

Classifier	**Average Enrichment**	**ROC-AUC**	**log(AUC)**
Rosetta Interface scores	3.81	0.62	0.62
ANN: BCL Scalar descriptors	5.79	0.48	0.34
ANN: Rosetta and BCL Scalar descriptors	5.51	0.63	0.62
ANN: Rosetta Fingerprint and Scalar descriptors	5.08	0.66	0.69
ANN: Rosetta Scalar descriptors	5.77	0.68	0.74

Table IV.7: ROC-AUC and average enrichment for the DEKOIS benchmarking study.. The Rosetta Interface Scores classifier uses only the sorted RosettaLigand interface scores for classification. All “ANN” classifiers are neural networks using the specified descriptors. ROC-AUC is the area under the ROC curve generated from each descriptor (Figure IV.8) Average enrichment is the average enrichment within the first 1% of the each dataset.

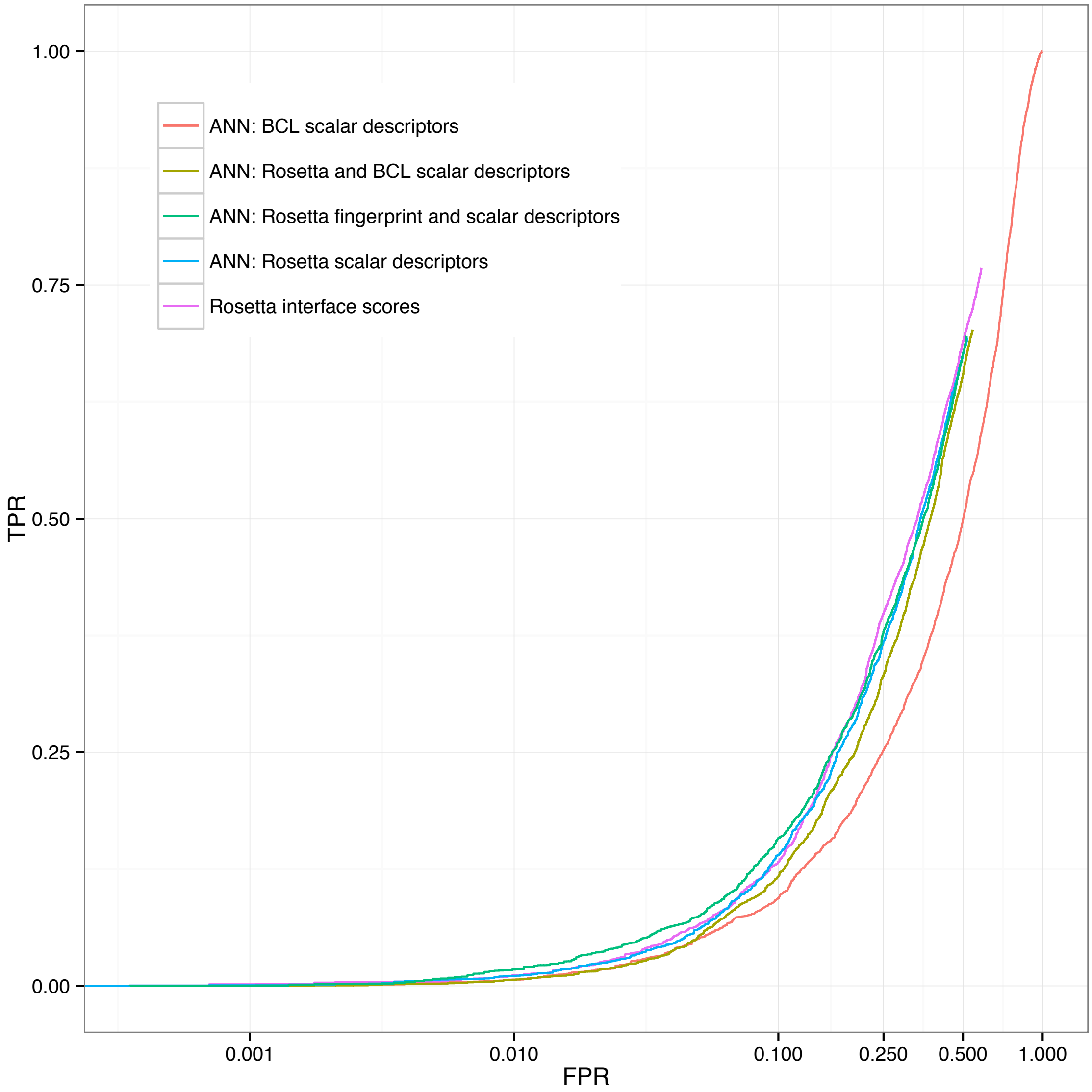


Figure IV.8: A plot showing the ROC curves of the various classification models trained across all protein targets in the DEKOIS 2.0 benchmark.

IV.3 Discussion

IV.3.1 We can construct neural network models that improve activity classification over a range of protein and ligand chemical space

The chemically balanced training dataset was successfully used to train ANNs to classify ligands based on their binding affinity. It is possible that, under certain circumstances, ligand descriptor information may be beneficial to this type of classification model and that some emergent property might exist between (for example) the flexibility of the ligand and the behavior of that ligand in the RosettaLigand docking simulation. However, at least in the training dataset used here, there is no evidence of such an emergent property, and if one does exist it, would almost certainly require a dataset comprising a very large range of ligand chemical space to become apparent. The loss of accuracy seen upon the addition of the Rosetta fingerprint descriptors suggests that, rather than conveying meaningful information about the protein-ligand interface, these descriptors are introducing noise into the system. While there is ample historical precedent (Mueller et al., 2010; Butkiewicz et al., 2013; Hristozov et al., 2007) for the value of RDF-based fingerprint descriptors as ANN input, it appears that in this case, the descriptors are not providing meaningful information. Regardless, the RosettaLigand-based scalar descriptors do provide sufficient information to create a well trained model that outperforms the RosettaLigand energy function alone.

IV.3.2 Development of a global classifier of protein-ligand binding affinity remains challenging

The cross-validation studies described in IV.2.4.5 suggest that the ANN models trained in this study are not overtrained and have some ability to distinguish between active and inactive compounds. However, application of these models to a fully independent benchmarking set suggests that, while the models are well trained, they are unable to act effectively as generally usable predictors. When comparing the ROC-AUC performance of the neural network models and the Rosetta-based model across the 74 proteins in the DEKOIS training set (Figure IV.7A), we see there are only minimal differences in ROC-AUC distribution across the range of protein system. To simplify the discussion of the data, we will therefore use the "ANN: Rosetta Fingerprint and Scalar descriptors" model as an example. Comparing this model to the control ("ANN: BCL Scalar descriptors"), we see a rightward shift in the performance distribution, with the mean ROC-AUC shifting from 0.45 to 0.61. Unfortunately, even with this rightward shift, the two curves (control and trained model) still mostly overlap, meaning that the ability of the trained model to make successful predictions is highly limited.

As mentioned in Section IV.2.4.7, it would be useful to be able to identify a subset of chemical space for which the ANN models are generally better able to perform. In order to attempt to identify such a subset, a range of chemical properties were computed for all the ligands in the filtered DEKOIS 2.0 benchmarking set. Binned distributions were then computed in order to compare the distribution of chemical properties for ligands in systems that could be successfully (ROC-AUC $>$ 0.7) and unsuccessfully (ROC-AUC $\leq$ 0.7) predicted. Figure IV.9 plots these binned distributions as a set of box plots. We see no meaningful difference in distribution across the range of evaluated ligand properties. This conclusion is consistent with previous studies (Mysinger et al., 2012) in which the inconsistent performance between various subsets of ligand chemical space could not be effectively eliminated.

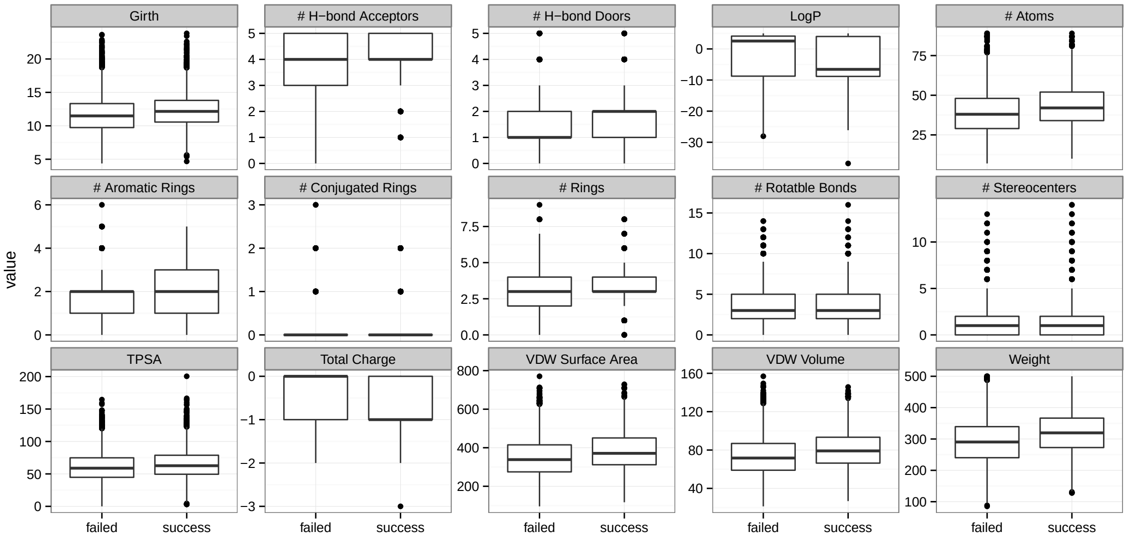


Figure IV.9: A plot showing the differences in distribution of various chemical properties between protein systems with ROC-AUC values for the classifier "ANN: Rosetta Fingerprint and Scalar descriptors" above 0.7 labelled "successful", and systems below that mark, labelled "failed". In these box-and-whisker plots, the midline corresponds to the mean, the hinges of the box correspond to the 25th and 75th percentile, the whiskers correspond to 1.5 times the Inter-Quartile Range, and points outside of this range are indicated as individually plotted dots.

CHAPTER V

Conclusions and Future Directions

V.1 Summary of Findings

V.1.1 Development of a novel energy function and benchmarking method for protein design

The development of tools for accurate structural biology predictions is an important area of current research. Tools that can accurately model the effect of mutations on protein stability will make it possible to both explore the basic physical principles that drive protein stability, and engineer new proteins and enzymes for pharmaceutical and chemical purposes. While supercharging (Lawrence et al., 2007; Kurnik et al., 2012; Simeonov et al., 2011) of protein surfaces has been an effective means of improving the solubility of designed proteins, this technique has an impact on the rate of folding, artificially recapitulating the balance between solubility and ability to fold seen in natural proteins. In the service of the goal of improving the quality of protein design, Chapter IIdescribes an innovative method for designing native-like protein surfaces, as well as a novel quality metric for assessing protein designs.

The new energy function implements a KBP previously developed by Durham *et al.*, (Durham et al., 2009), which was computed based on the propensity of amino acids existing at various degrees of burial in X-ray crystal structures of soluble proteins. This energy function is effectively an environment potential, which provides an energy bonus to amino acids frequently found in nature at a given degree of burial within the protein. In addition to the implementation of this knowledge based potential, the weights of the RosettaDesign energy function were re-optimized to maximize the PSSM score of the protein, based on a PSSM generated using BLAST (Altschul et al., 1997). To assess the quality of the proteins designed with the new energy function, two metrics were used. The first was sequence recovery (Kuhlman and Baker, 2000) the percentage of amino acids which were designed to be identical to the native amino acid. The second, PSSM recovery, was originally presented in Chapter II. PSSM recovery is the percentage of amino acids with a favorable PSSM score according to a PSSM created using BLAST. PSSM recovery is advantageous over sequence recovery as a metric of good protein design because it measures evolutionarily favorable mutations, rather than simply counting exact amino acid recovery. The combination of this optimization function and the environment based knowledge based potential resulted in protein designs more native-like relative to the previously published RosettaDesign function. Specifically, PSSM recovery improved from 72% with the standard energy function to 77.2% with the optimized energy function.

V.1.2 Improvement of the speed and sampling efficiency of RosettaLigand

While RosettaLigand has been previously successful at ligand docking (Lemmon et al., 2012; Combs et al., 2011; Allison et al., 2014), it is too slow for use in high throughput ligand docking applications. To address this, chapters III and C describe the development of a grid based Monte Carlo sampling algorithm for initially placing ligands in the protein binding site prior to refinement, as well as a modular scoring system for defining Cartesian grid-based scoring functions. The best performing results with the new method were obtained using the new Monte Carlo initial placement algorithm combined with the originally published grid based energy function. This method resulted in a 10-fold reduction in the number of protein models required to obtain a successful binding pose, as well as a 6-fold reduction in the time necessary to generate a single model. Additionally, the new RosettaLigand initial placement algorithm results in a significantly increased ability of RosettaLigand to successfully dock ligands into protein systems and an improved tolerance of backbone and side-chain misplacement. It is notable that these improvements resulted entirely from improved initial sampling. These results highlight the importance of high quality and efficient sampling in protein-ligand docking and demonstrate that substantial gains can be made in docking performance through more efficient utilization of existing scoring information. The results presented in Chapter III make it possible for the first time to efficiently use RosettaLigand for the structure based virtual screening of large compound libraries on a small academic computing cluster.

V.1.3 Development of RosettaHTS: A structure based virtual screening protocol

The methods developed in Chapter III lead naturally to Chapter IV, which describes the development of a protocol for docking large numbers of small molecules using RosettaLigand and a novel ANN-based classification model for predicting the activity of these compounds. To train the ANN model, a training dataset was constructed using cross-docking so as to be highly diverse in chemical and protein space, as well as balanced in chemical space between active and inactive compounds. In addition to previously developed Rosetta energy terms, interface quality metrics, and ligand descriptor information, a new set of protein-ligand interface descriptors were developed based on RDFs. Several networks were trained using various combinations of these descriptor sets. Analysis of the cross-validation performance of the trained networks indicated that the majority of the useful information to the networks was provided by the Rosetta energy term and interface quality metric data.

While the ANN models developed in this study were not capable of generalizing enough to make accurate predictions of ligand activity in the DEKOIS 2.0 benchmarking set, the ANN scoring approach warrants further investigation. The lack of generalizability seen here reflects the recent history in the field. The study described in Chapter IV represents only preliminary investigation into the feasibility of using RosettaLigand as a source of descriptor features for ANN-based scoring functions. It appears, based on the analysis presented above, and on the existing literature reviewed, that this may be a method worth pursuing. To continue this project, there are a wide range of additional method development, benchmarking, and analysis steps that could be undertaken. Section V.2.2 outlines some of these next steps.

V.2 Future Directions

V.2.1 RosettaLigand method development

V.2.1.1 Scoring function development

While the RosettaLigand algorithm improvements described in Chapter III are highly encouraging, there are further improvements that can be made in both the speed and scientific performance of the software. One obvious area of further development is in the grid-based energy function used by the initial placement algorithm. The current energy function serves primarily to identify regions of the protein binding site in which atoms placed would result in major clashes with the protein backbone. Additional information would potentially improve the efficiency with which the initial placement algorithm can identify high quality binding poses, further reducing the number of binding poses required for a high quality prediction. The shape complementarity and hydrogen bonding potentials described in Chapter C represent one attempt to address this problem, but these energy functions did not improve the performance of RosettaLigand. The likely culprit is the over-reliance on side-chain atom positions. Because the shape complementarity and hydrogen bonding potentials are constructed using all protein atoms, incorrect side-chain positions will result in poor binding position predictions. Because starting side-chain positions can be assumed to be incorrect if a ligand is being cross-docked, or docked into a comparative model or relaxed structure, any method which relies on side-chain information is likely to be unsuccessful. To address this problem, research into the development of KBPs using backbone, Cα and Cβ atoms is ongoing. These potentials are generated using well packed crystal structures with non-covalently bound ligands as input and encode the propensity of said ligand atoms relative to the protein backbone atoms. Several methods of accomplishing generating these potentials are being explored, including the use of purely distance-based potentials, 3-D potentials utilizing a distance and two angles, and 2-D potentials utilizing a distance and one angle.

V.2.1.2 Sampling method development

The Monte Carlo initial placement algorithm described in Chapter III has proven useful, but has room for significant improvement. The Metropolis Monte Carlo search is performed at a constant Boltzmann temperature. While a constant temperature search has proven sufficient, a Monte Carlo simulated annealing algorithm, in which the temperature is initially raised and then slowly lowered, could result in faster and more reliable convergence. Additionally, the temperature can be dynamically modulated to reach a target acceptance rate. In this method, the target acceptance rate is gradually lowered over the course of the simulation to cause convergence. To reach the target acceptance rate, the Boltzmann temperature is modulated continuously.

Pre-computed grid-based scoring functions lend themselves well to a variety of sampling methods, some of which may be far more efficient than a Monte Carlo search. In particular, if more informative scoring grids can be developed, geometric hashing methods, such as those previously used for CryoEM fitting (Woetzel et al., 2011), could be of great value. A geometric hashing algorithm would be capable of identifying potential initial ligand positions far more rapidly than a Monte Carlo search. Additionally, such an algorithm would make it possible to greatly expand the size of the scoring grid, potentially allowing RosettaLigand to be used to perform binding site detection, rather than requiring that the user provide the initial binding site.

V.2.2 RosettaHTS method development

V.2.2.1 Exploring additional methods for computing protein-ligand interaction descriptors

In addition to the RDFs described in Section IV.2.2.4, there are a wide array of descriptors that may be applicable to RosettaHTS. Some of these have been previously successful in various forms of vHTS, while others are new. It has been historically difficult to predict what type of descriptors will be most useful for a given application, so further research should include the implementation and evaluation of the descriptor methodologies described below.

3D Autocorrelation (3DA) functions are a class of vector based descriptors similar to RDFs. Autocorrelation is a general class of methods for comparing a structure with itself. While autocorrelation methods are most frequently used in signal analysis, they can be adapted to encode the composition and structure of a small molecule. ADRIANA (Molecular Networks GmbH Computerchemie, 2011) describes a 3DA function as

$$A(d_n) = \frac{1}{2L_n} \sum_{\substack{i,j \\ i \neq j}} p_i p_j \tag{V.1}$$

where L_n is the number of inter-atomic distances within a given interval, and p_i and p_j are properties of two atoms. As with the previously described RDFs, 3DAs can be used to generate fingerprint vectors. 3DA functions have been previously successful (Butkiewicz et al., 2013) in ligand based virtual screening studies, and their lower resolution relative to RDF descriptors may result in lower noise, more effective descriptors.

Atom pair counts can also be used as ANN descriptors. In the case of NNScore 1.0 and 2.0 (Durrant and Mccammon, 2010, 2011), the total number of protein-ligand atom pairs within 2.0 and 4.0 -Å shells was computed. This metric effectively provides a low resolution fingerprint of the protein-ligand environment in a way that is independent of both receptor structure and orientation. Because the shells are broad, these atom pair count descriptors will be less sensitive to small perturbations in predicted ligand binding predictions and therefore less noisy when used as the input to a classification model. As implementation of atom pair count descriptors is straightforward, their utility as descriptors for RosettaHTS should be evaluated in combination with, and as an alternative to, the 3DA and RDF descriptors.

When high quality protein structural information is available, binary interaction fingerprints can be a useful class of descriptor. binary interaction fingerprints describe the protein-ligand interface in terms of a binary bitstring. In this scheme, of which Structural Interaction Fingerprint (SiFT) (Deng et al., 2004) is one example, each bit in the string represents an interaction between the ligand and each amino acid in the protein receptor. Additional information can be encoded in the fingerprint by adding bitstrings representing specific types of bonding interaction, as opposed to than simple physical proximity. As demonstrated in the initial SiFT paper, there is a statistically significant correlation between bitstring similarity and ligand binding activity to a given protein target. There are a number of advantages to advantage of bitstring based fingerprints, including that they are conceptually simple and straightforward to implement and analyze. Additionally, as each field in the bitstring can contain only a 1 or 0, rather than a range of floating point numbers, there is less potential for noise to be introduced in the descriptor signal relative to 3DA or RDF models. The ease of calculation, diversity of information content, and low noise would seem to make SiFT style interface bitstrings an ideal choice of descriptor for RosettaHTS, barring one major limitation: their dependence on sequence alignment. As each bit represents a single residue-ligand interaction, interface bitstrings can only be accurately compared if the bits represent the same regions of protein structure. For this reason, most implementations of interface bitstring fingerprints are used to classify ligand activity against a single protein target or a group of highly conserved similar proteins (Chupakhin et al., 2013). The concept of a binary fingerprint is still useful, particularly in the context of ANN based models, and it may be possible to devise a binary interaction fingerprint that is useful for making comparisons between a diverse range of target proteins.

In the field of image recognition, bitmaps have been successfully used as descriptors for Deep Belief Network (DBN)s (Hinton et al., 2012; Dean et al., 2013). For training image recognition models, image bitmaps representing color and intensity are used directly as input to the network. The advantage of image descriptors is that they are fully independent of both rotation and scale. The concept of 2-D bitmaps employed in image recognition can potentially be adapted for use in 3-D pattern recognition. Protein ligand interactions could be represented as space-filling voxels, with voxel values representing chemical properties. A 3-D image descriptor-based method could potentially take advantage of the resiliency to noise, rotation, translation, and scale that has made DBNs successful in image recognition tasks.

The downside of such an image-based descriptor method is the number of descriptor bins required to represent the protein-ligand interface. An image with a 0.25 Å voxel width and an 8 Å box size would require 32^3, or 32,768 values for each chemical property explored. Such a large number of inputs is substantially larger than the ANNs and DBNs previously reported in the literature and is computationally infeasible at the present time. However, the generation and training of extremely large networks is an area of intensive research in both the public and private sector, and it is likely that networks of this size will become computationally tractable in the near future.

V.2.2.2 Further optimization of ANN Models

The ANN models described in this study represent preliminary work and thus have not been subjected to the rigorous optimization that is required to obtain maximal network performance. There are a range of neural network parameters that affect model performance. At a minimum, learning rate, number of hidden nodes, descriptor set, and number of hidden layers should be optimized. These parameters can be optimized through a brute force search of a reasonable range of the parameter space until an optimal set is located. However, there are methods (Attik et al., 2005) for searching the network topology and parameter space more efficiently.

In addition, simple feed-forward ANNs were the only machine learning method assessed in this study. A number of other supervised learning techniques, such as support vector machines and random forest models should also be evaluated. More exotic methods, such as DBNs and recurrent neural networks, may also be of value, though a larger benchmarking set and more sophisticated descriptor methods, such as those described in Section

V.2.2.1 may be required in order to take advantage of them. Evaluation of additional machine learning methods may also enable the use of a consensus model for predicting activity based on a jury of classifiers trained using different methods.

V.2.2.3 Development of a multi-stage model pipeline

In Section IV.2.4.6, we attempted to train a regression model to directly predict K_i, rather than simple binary classification. The models trained were unable to attain meaningful regression correlation during cross-validation. However, it is possible that the failure of these models was the result of the inclusion of inactive compounds. If a reliable, generalized classification model could be trained to classify active and inactive models, a regression model trained on only active models could be useful as a second stage in a multi-stage classification pipeline. In this pipeline, the classification model would be used first to identify likely active compounds, and the second stage model would attempt to predict K_i. This approach would allow researchers to prioritize compounds for further study in a more effective way than a simple classification model would provide. An additional advantage of this approach is that it would make it possible to remove the inactive compounds from the training dataset for the K_i model, as the model would only be applied to presumed active compounds. As inactive compounds rarely have measured binding affinities, it is necessary to fix their K_i values to 0.0, which likely adds noise to the regression model, as even inactive compounds do not have exactly identical binding affinities. Removing these compounds will almost certainly reduce noise in the training set and improve the quality of the regression model.

V.2.2.4 Improvement of benchmarking protocol

To eliminate the possibility of the ANN models learning a decoy selection algorithm rather than the actual methods, the active compounds were cross-docked to generate a set of inactive models during training. On the other hand, the set of predicted inactives provided by DEKOIS 2.0 were used during benchmarking. Because each prediction made by the neural network is made in isolation, a well generalized model should be capable of distinguishing between active and inactive compounds, regardless of the manner in which those active and inactive compounds were selected. As an additional control experiment, it may be useful to perform a benchmarking study in which the active compounds from the DEKOIS 2.0 set are cross-docked in the same manner as the training set.

V.2.2.5 Exploration of quantum chemical and molecular dynamics descriptor information

One major component missing from this study is the addition of any descriptor information reflecting quantum chemical effects, or the molecular dynamics of the system as a whole. The importance of this information cannot be understated, and it is likely that both categories of descriptor information are critical to fully describe the protein-ligand interaction. Recent work by Steven Combs (Combs, 2013) introduces electron orbital placement into the Rosetta atom typing scheme. The incorporation of this orbital information will enable a wide range of fingerprint and scalar descriptors derived from the relationship between protein and ligand orbital positions. The limitation of this system is that the orbitals are represented as points, rather than as 3-D structures. The advantage of the point representation is that it maintains computational efficiency. However, if more accurate electron orbitals could be computed in a computationally efficient manner, they could be used directly as a source of descriptor information.

Subtle changes in the mode of vibration can have a dramatic effect on protein-ligand binding affinity (Baugh et al., 2010). Molecular dynamics trajectories could be used to compute the normal modes of vibration for a complex, and these modes of vibration could then be used directly as a descriptor in an ANN model. The inclusion of this information could allow the network to identify changes in the normal modes of vibration between strong and weakly binding compounds. The major downside of including this information is an expected decrease in computational efficiency. In even a small vHTS screen, thousands of trajectories would need to be computed and analyzed prior to ANN classification.

While continuing developments in computational power may make this approach effective eventually, it is unlikely to be feasible given current available computing hardware and modeling tools.

Appendix A

Rosetta HTS ligand pre-processing

A.1 Background

vHTS in Rosetta poses a number of unique challenges. Large numbers of ligands must be prepared and managed. If multiple protein systems are being screened, the appropriate ligands must be matched with the appropriate proteins. Additionally, because Rosetta currently stores ligand information after it is required, the memory requirements of the protocol increase as more ligands are screened. The most straight forward way to address this is to limit the number of ligands screened in each process.

This protocol describes a series of scripts that can be used to rapidly prepare a set of proteins and ligand for virtual high throughput screening.

All scripts referenced can be found in `tools/hts_tools`

A.2 Prerequisites

This protocol requires the following:

- Python
- BioPython
- The most recent weekly release of Rosetta.
- A directory containing conformations of all the proteins in your screening study.
- An Structure Data File (SDF) containing all the conformers of all the ligands you want to dock. All conformers of the same ligand must have the same name.
  - All conformers of all ligands must have 3D coordinates and hydrogens. You will receive no errors or warnings if you provide ligands with 2D coordinates or without hydrogens, but you will get very poor ligand docking results.

A.3 Protocol

This protocol concerns the processing and parameterization of SDF files for use in RosettaLigand docking calculations. While the RosettaLigand is capable of handling a wide range of chemistry and protein systems, some additional preparation is often required to make the best use of the system. High quality, minimized protein structural models are required, and the location of the ligand binding site must be known to roughly 5.0 Å.

The Nature Protocols paper published by Combs et al (Combs et al., 2013) describes in detail the step by step process for preparing individual protein and ligand files for input, as well as best practices for the use of Rosetta in general as a tool for protein-ligand docking.

A.3.1 Split ligands

The first step of the protocol is to split the ligand file. `sdf_split_organize.py` will accomplish this task. It takes as input a single SDF file, and will split that file into multiple files, each file containing all the conformers for one ligand. Different ligands must have different names in the SDF records, and all conformers for one ligand must have the same name. Output filenames are based on the sha1 hash of the input filename, and are placed in a directory hashed structure. Thus, a ligand with the name "Written by BCL::WriteToMDL,CHEMBL29197" will be placed in the path
`/41/412d1d751ff3d83acf0734a2c870faaa77c28c6c.mol`.

The script will also output a list file in the following format:

```
ligand_id,filename
string,string
ligand_1,path/to/ligand1
ligand_2,path/to/ligand2
```

The list file is a mapping of protein names to SDF file paths.

Many filesystems perform poorly if large numbers of files are stored in the same directory. The hashed directory structure is a method for splitting the generated ligand files across 256 roughly evenly sized subdirectories, improving filesystem performance.

The script is run as follows:

```
sdf_split_organize.py input.sdf output_dir/ file_list.csv
```

Note that `output_dir/` must already exist prior to running `sdf_split_organize.py`

A.3.2 Create "project database"

The ligand preparation pipeline uses an sqlite3 database for organization during the pipeline. The database keeps track of ligand metadata and the locations of ligand files. The project database will be used in Sections A.3.3 andA.3.4, and does not need to be preserved after those steps are completed. The project database is created using the following command:

```
setup_screening_project.py file_list.csv output.db3
```

A.3.3 Append binding information to project database

The next step is to create a binding data file. The binding data file should be in the following format:

```
ligand_id,tag,value
string,string,float
ligand_1,foo,1.5
ligand_2,bar,-3.7
```

The columns are defined as follows:

1. **ligand_id** — ligand_id is the name of the ligand, which must match the ligand_id in the file_list.csv file created by `sdf_split_organize.py`.

2. **tag** — The name of the protein the ligand should be docked into. If a ligand should be docked into multiple proteins, it should have multiple entries in the binding data

file. Note that this protocol makes a distinction between protein name, and file name. If you have 4 files: foo_0001.pdb, foo_0002.pdb, bar_0001.pdb, and bar_0002.pdb, then you have two proteins with the names foo and bar. The scripts expect that the protein PDB files begin with the protein name.

3. **value** — The activity of the ligand. If you are doing a benchmarking study and know the activity of your ligand, you should enter it here. If you are not doing a benchmarking study, or if ligand activity is not relevant to your study, value can be set to 1.0 (or anything else). This field is currently only used in a few specific Rosetta protocols that are in the experimental stages, and is typically ignored, so it is safe to set arbitrarily in almost every case. Regardless, the scripts require that you provide some value.

Once you have created this file, you can insert it into the project database with the following command:

```
add_activity_tags_to_database.py output.db3 tag_file.csv
```

A.3.4 Generate params files

The next step is to generate params files. make_params.py is a script which wraps around

molfile_to_params.py and generates params files in an automated fashion. params files will be given random names that do not conflict with existing Rosetta residue names (no ligands will be named ALA, for example). This script routinely results in warnings from molfile_to_params.py, these warnings are not cause for concern. Occasionally, molfile_to_params.py is unable to properly process an SDF file, if this happens, the ligand will be skipped. In order to run make_params.py you need to specify the path to a copy of molfile_to_params.py, as well as the path to the Rosetta database.

molfile_to_params.py is part of the standard rosetta distribution and is located in main/source/src/python/apps/public/

make_params.py should be run like this:

```
make_params.py -j 4 \
--database path/to/Rosetta/main/database \
--path_to_params path/to/molfile_to_params.py \
output.db3 params/
```

In the command line above, the -j option indicates the number of CPU cores which should be used when generating params files. If you are using a multiple core machine, setting -j equal to the number of available cpu cores.

The script will create a directory params/ containing all params files, PDB files and conformer files.

A.3.5 Create job files

Because of the memory usage limitations of Rosetta, it is necessary to split the screen up into multiple jobs. The optimal size of each job will depend on the following factors:

- The amount of memory available per CPU
- The number of CPUs being used
- The number of atoms in each ligand
- The number of conformers of each ligand
- The number of protein residues involved in the binding site.

Because of the number of factors that affect RosettaLigand memory usage, it is usually necessary to determine the optimal job size manually. Jobs should be small enough to fit into available memory.

To make this process easier, the make_evenly_grouped_jobs.py script will attempt to group your protein-ligand docking problem into a set of jobs that are sized as evenly possible. The script is run like this:

```
make_evenly_grouped_jobs.py --n_chunks=10 \
 --max_per_job=1000 param_dir/ structure_dir/ output_prefix
```

If the script was run as written above, it would use param files from the directory param_dir/, and structure files from the directory structure_dir/. It would attempt to split the available protein-ligand docking jobs into 10 evenly grouped job files (-n_chunks). The script will attempt to keep all the docking jobs involving one protein system in one job file. However, if the number of jobs in a group exceeds 1000, the jobs involving that protein system will be split across multiple files (-max_per_job). The script will output the 10 job files with the given prefix, so in the command above, you would get files with names like "output_prefix_01.js". The script will output to the screen the total number of jobs in each file. All the numbers should be relatively similar. If a job file at the beginning of the list is much larger than the others, it is a sign that you should reduce the value passed to -max_per_job. If the sizes of all jobs are larger than you want, increase -n_chunks.

A.3.5.1 Job file specification

RosettaLigand Job files are JavaScript Object Notation (JSON) files which contain the paths to protein and ligand PDB files, the names of the protein systems, and the params files necessary to load the ligand PDB files. An example file is below:

```
 "jobs": [
  {
   "proteins": [
    "set2_28_0001.pdb"
   ],
   "ligands": [
    "A9.pdb"
   ],
   "group_name": "set2_28"
  },
  {
   "proteins": [
    "set1_42_0001.pdb"
   ],
   "ligands": [
    "AJ.pdb"
   ],
   "group_name": "set1_42"
  }
 ],
 "params": [
  "A9.params",
  "AJ.params"
 ]
```

A.3.6 Submit jobs

Once the jobs have been created they can be input into rosetta using the option `-in:file:screening_job_file job_file.js`. If this option is being used, `-s`, `-l`, `-list` and `-in:file:extra_res_fa` should not be used. Other than this, any Rosetta application or XML protocol may be used without restriction. For example, the screening job inputter would be used in conjunction with RosettaScripts using the following command line:

```
   rosetta_scripts.default.linuxgccrelease @flags.txt -in:file:screenin
job_file.js -parser:protocol script.xml
```

Where `flags.txt` is a file containing any set of RosettaLigand flags, and `script.xml` is a RosettaScripts XML file.

A.3.7 Result processing

The results of the a ligand docking job can be handled in the same way as any other RosettaLigand docking calculation. For examples of the range of detailed analysis that mayb be performed to analyze RosettaLigand results, see Chapters E, IV and C. As the protocol above applies only to the processing of files for input, it has no impact on the on the output of docking results. For smaller RosettaLigand studies, results should be output as PDB files using the `-out:pdb` flag. However, if very large numbers of models will be generated, the Rosetta SQL output framework described in Chapter B may be of use.

Appendix B

Managing Rosetta datasets with SQL

B.1 Introduction

Screening small molecule libraries with RosettaHTS requires the generation of very large number of models. For example, a screen of 250,000 compounds will result in 50,000,000 generated Rosetta models. Because RosettaHTS only needs the top 1 model generated for each ligand, 99.995% of this data will not be needed. To avoid unnecessarily storing data, RosettaHTS stores generated models in a relational database. Here the technical details of the database storage framework used by RosettaHTS are described. This framework was developed collaboratively by myself, Chris Miles at the University of Washington, Matt O'Meara at the University of North Carolina, and Tim Jacobs at the University of North Carolina, all of whom contributed equally to the project.

B.2 Features reporter architecture

The data storage framework used by RosettaHTS is a subset of a statistical analysis framework within Rosetta called the Features Reporter. The Features Reporter framework consists of an Object-Relational Mapping (ORM) that allows SQL schema definitions, queries and insertion statements to be composed in C++, and a reporting system used for computing statistical data and storing it in the database. The ORM was developed using CppDB(http://cppcms.com/sql/cppdb/) SQL library, which allows the features reporter to work with SQLite, MySQL and Postgres database backends. As all 3 of these back ends have different caveats, behaviors and syntax idioms, the ORM is responsible for constructing the appropriate MySQL command and sending it to the server.

In addition to allowing for the transparent support of multiple Database Backends, the ORM system makes it possible for a C++ programmer to easily define new SQL objects without being familiar with SQL syntax. For example, the C++ code reproduced below defines an SQL table containing structure ID, residue number, 3 letter residue name, and residue type fields, then generate the appropriate SQL syntax to create this table, and execute the SQL command on the database server.

```
Column struct_id("struct_id", new DbBigInt(), false);
Column resNum("resNum", new DbInteger(), false);
Column name3("name3", new DbText(), false);
Column res_type("res_type", new DbText(), false);

utility::vector1<Column> residues_pkey_cols;
residues_pkey_cols.push_back(struct_id);
residues_pkey_cols.push_back(resNum);

Schema residues("residues", PrimaryKey(residues_pkey_cols));
residues.add_column(struct_id);
residues.add_column(resNum);
residues.add_column(name3);
residues.add_column(res_type);
residues.add_foreign_key(
  ForeignKey(struct_id, "structures", "struct_id", true));

residues.write(db_session);
```

This model of database interaction allows multiple database systems to be supported with a single code base, which is critical given the diversity of hardware and software operated by Rosetta users. Additionally, it allows support to be added for additional database systems without any change to the code defining feature tables.

The Features Reporter has been previously described in the literature (Leaver-Fay et al., 2013) as a method for collecting and analyzing statistics from large numbers of Rosetta models. In the sections below we will describe the specific application and usage of this framework for the storage and retrieval of protein models in a high throughput screening environment.

B.3 Pose storage schema

RosettaHTS uses a subset of the Features Reporter system to store and retrieve protein structure information from the database. In the most simple case, it would be possible to store only the atom and residue names, chain information, and cartesian coordinates for each atom, and effectively replicate the information present in a PDB file. However, storing this limited subset of information requires that Rosetta rebuild the internal data-structures representing the protein structure every time the pose is loaded, as is done when taking input from a PDB file. While this is normally acceptable, in rare cases, there are multiple correct configurations that these data structures can take on, and thus, recomputing them can result in a slightly different input model, and numeric inconsistencies in scoring across multiple loads of the structure. While these numeric inconsistencies do not typically lead to scientific inconsistencies, their presence can complicate data analysis.

In order to rectify this problem, the RosettaHTS pose storage schema stores not only the information reflected in the protein PDB file, but also all of the derived data created by Rosetta. The storage of this information makes it possible to precisely store and recover the protein structure as it is internally represented by Rosetta. The Pose storage schema is illustrated in Figure B.1. Each block in the figure represents an SQL table storing a subset of of the information needed to reconstruct the pose. Connections between the blocks indicate “Relationships” between the tables. Connecting the tables with relationships makes it possible to link data elements that related to each other, while simultaneously allowing for Pose data to be queried efficiently.

A critical component of the design of a relational database schema is a means of uniquely identifying models. In this case, uniqueness from the point of view of the database has a different meaning than biochemical uniqueness. If Rosetta is run twice with the same protocol and produces two absolutely identical output models, those models should be stored separately in the database and given unique identifiers, even though they contain the same information. Additionally, given a specific stored Pose, it should be possible to

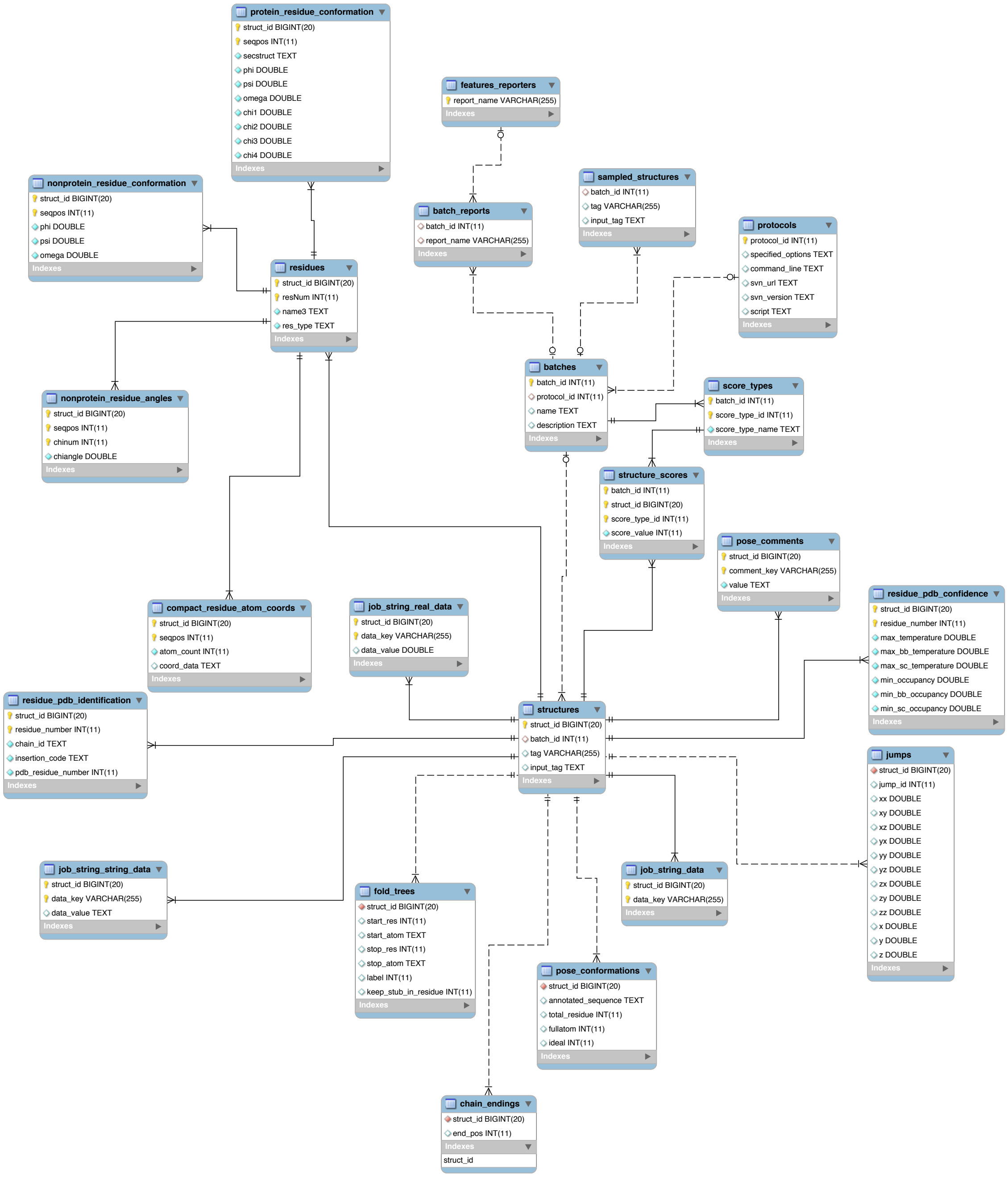


Figure B.1: A schematic representation of the information stored by the RosettaHTS pose schema

identify which execution of the Rosetta application resulted in that Pose, and at what point in the overall protocol the Pose (or other statistical data) was generated. To accomplish this, the identification of structures is handled with a three tiered structure. The top tier is the "protocol" level. A new protocol is generated every time a Rosetta process outputs to a database. Each protocol has a unique number identifying it which is assigned by the database server. The protocol stores the version of Rosetta that was used, as well as the command line options and XML script specified by the user. Each protocol can generate one or more "batch" which is the second level of the identification system. A batch is generated every time a protocol outputs structural data. If RosettaHTS is only outputting completed protein models, only one batch exists per protocol. It is possible to output protein models at intermediate steps within a long RosettaScripts protocol. If intermediate models are output, an additional batch will be created for each intermediate step. If the Features Reporter is also being used in the middle of the protocol to generate extra statistics, additional batches will be generated. Each batch has a unique number identifying it, and references the protocol that was responsible for creating it. The last tier in the identification system is the "structure". A new structure is generated every time Rosetta produces a model. Each structure has a unique number identifying it, and references the batch that resulted in the creation of that structure. Additionally, each structure record has a human readable "tag" describing its output, and references the input structure that was used as a basis for the model. In this way, the batch, protocol and structure tables can be joined by their related IDs such that, given a structure stored in the database, one can determine precisely what Rosetta command was used to generate that structure, and at what point in the Rosetta protocol the structure was generated.

The structure ID described above is used as the central unique identifier, and all the information associated with an individual model is related using this ID.

The MySQL database schema stores the following information:

- Protocol information

    - Text and numeric comment data that some Rosetta components insert in a pose for scoring and tagging purposes during the protocol.

- Rosetta-specific structural metadata
    - The "Jumps" in the structure. Jumps are data structures which represent the spatial relationship between protein chains as a Rotation and Translation matrix.
    - The "Fold Tree" defining the structure. The Fold tree is a data structure used by Rosetta to describe how different parts of the protein are related in space. Fold Trees can be configured to make it possible to rapidly perturb large parts of the protein as a rigid body. DiMaio et al. (DiMaio et al., 2011) provides an example of a fold tree manipulation in practice.

- Structural information
    - The range of B-factors for atoms in each residue.
    - The sequence associated with the structure, annotated with the identity of any non-canonical, modified or ligand residues.
    - The ending position of each chain.
    - The Chain ID, insertion code, and residue number present in the original PDB file.
    - The values of each energy term in the score function when the structure was last scored.
    - The cartesian coordinates of each atom.
    - The chi angles of each canonical amino acid.
    - The rotation angles of each non-canonical amino acid or non-protein molecule.

In addition to providing infrastructure for outputting Rosetta models, the RosettaHTS pose IO schema provides infrastructure for using this data as input into Rosetta. Both output to and input from the database is handled by the Rosetta job distribution system. This means that any Rosetta application can use a database as a data storage mechanism, without any additional programming on the part of the developer of the application. Furthermore, because the same database can be used for both output and input simultaneously, a Rosetta modeling protocol that relies on different applications being used in different stages can use the database for data management at each stage.

B.4 Database Filters

Historically, Rosetta applications have considered each model independently of any other model. This is primarily an engineering decision. If each model is considered independently, it means that only the current structure needs to be stored in memory, reducing system resource requirements. Additionally, if each model is considered independently, the design of Rosetta is simpler, as it each module of the code does not need to consider the accumulation of state between jobs. However, while this decision was critical to making Rosetta a maintainable piece of software, it has severely limited the ability to perform filtering and output of models. Specifically, filters could historically be created only based on single protein metrics. A filter could, for example, output models with a score lower than a certain fixed cutoff, or that had a degree of packing better than a cutoff. However, in the case of protein-ligand docking, we are unable to set fixed score cutoffs, as the range of scores seen will vary from ligand to ligand. The preferred means of filtering ligand models is to accept the lowest scoring model generated for each protein-ligand pair. This type of filter cannot be created with the traditional Rosetta filter system, requiring that all models be output, and filtered after the fact. The RosettaHTS docking protocol requires that 200 models be generated per protein-ligand pair. Given that each compressed model requires approximately 90 kilobytes of storage space, the total requirement per protein-ligand pair is roughly 18 megabytes. Thus, a 250,000 compound screen would require about 4.5 terabytes of compressed storage space. The infrastructure required to store and analyze a dataset of this size outstrips the abilities of most research groups, and the storage of the complete dataset is particularly senseless as nearly all of it will be deleted after the initial round of filtering.

The Database Filter system leverages the properties of the SQL database system to make it possible for the first time to create Rosetta filters which take into account the context of previously generated models. SQL is designed to conduct rapid queries of stored data, and as these queries are conducted by the database engine itself, rather than Rosetta, the operation of the filter does not require that state be kept between Rosetta jobs and does not result in additional memory requirements.

A Rosetta Database Filter can have three possible outcomes: The current model is not output, the current model is added, or the current model replaces an existing model. To accomplish this the Database Filter conducts a query of the existing structures in the database to identify if the current model is suitable to be output, and to identify the model that it will replace if necessary. Currently, filters exist to filter models by the top percent or top count by score. The creation of additional filters can be accomplished by the implementation of a new DatabaseFilter class with a single method. While the current usage for filtering by score percentile is relatively simple, the Database Filter framework could hypothetically be used to implement histogram matching filters or other filters requiring deep analysis of the existing dataset. This would be effectively impossible in the context of the classical Rosetta filtering system, but relatively trivial with a Database Filter.

A description of the currently implemented database filters and their use is described in section B.6.2

B.5 Performance Considerations and Selection of a Database backends

Currently, the Rosetta database system supports three different database back ends. The selection of the appropriate back end is an important consideration, and will depend largely on the way in which the database will be used.

B.5.1 SQLite3

SQLite3 (https://www.sqlite.org/) is the default option for Rosetta database support. SQLite3 stores the entire database as a single binary file, and the database engine itself is built into Rosetta. The primary advantage of using SQLite3 is that it requires no additional hardware or software infrastructure, which makes it idea for prototyping a protocol or handling small “one-off” analysis of data. SQLite3 was designed to be used explicitly with single threaded applications, this means that only a single Rosetta process at a time can write to an SQLite database. Additionally, SQLite3 produces a very large number of random access operations to the filesystem. The nature of these operations is such that heavy usage of SQLite3 can severely degenerate the performance of a network file system. For this reason, we recommend that SQLite3 only be used only with local disks, and preferably with Solid State Drives rather than traditional “Spinning” disks. Despite these limitations, Rosetta protocols requiring large amounts of memory and complex statistical analysis can be performed very efficiently using SQLite3. The majority of the data collection and analysis described in Leaver-Fay et al. (Leaver-Fay et al., 2013) was conducted using an SQLite3 database.

B.5.2 PostgreSQL and MySQL

PostgreSQL (http://www.postgresql.org/) and MySQL (http://www.mysql.com/) are freely available server based SQL solutions. In both cases, an independent server is used, and Rosetta communicates with this server over the network. The details of the installation and configuration of these systems are beyond the scope of this document, however either piece of software is acceptable for use with RosettaHTS. For large scale data operations, PostgreSQL and MySQL have substantial advantages over SQLite3. In both cases, many processes can simultaneously write to a single database, and the database server ensures that data is written correctly even for large numbers of concurrent reads and writes. Additionally, the use of a server based SQL solution offloads the computational and IO complexity involved with reading from and writing to the database to a separate machine than the one running Rosetta. This separation of work can have a substantial performance benefit. It is possible to create a Rosetta protocol which produces database requests that outstrip the processing ability of even a reasonably powerful database server. The precise nature of these limitations will depend on the Rosetta protocol, as well as the specific details not only of the database server software configuration but also the server and network infrastructure being used. For this reason, we advise empirically determining the performance limitations and requirements of your specific protocol through benchmarking tests prior to large scale screening.

B.6 Storing and retrieving poses from the Rosetta command line

Rosetta provides a set of command line options for storing and retrieving poses from a database. These command line options can be used with most Rosetta applications, the most notable exception being the "abinitio" application, and will function identically in any protocol.

B.6.1 Connecting to a database

Connecting to a Rosetta database depends on the type of database back end being used. Regardless of what back end is being used, back end mode and the database name must be specified:

```
-inout:dbms:mode <mode>
-inout:dbms:database_name <db_name>
```

where `<mode>` should be replaced by the database mode (either `mysql`, `sqlite` or `postgres`), and `<db_name>` should be replaced by the name of the database file if using sqlite, the schema name if using MySQL, and the database name if using PostgreSQL.

If you are using MySQL or PostgreSQL as a back end, you must specify several additional options connect to the server

```
-inout:dbms:host <host>
-inout:dbms:user <username>
-inout:dbms:password <password>
-inout:dbms:port <port>
```

Where `<host>` is the address of the database server, `<username>` is the username of a user with permission to read and write from the database, `<password>` is the password of that user, and `<port>` is the TCP port that the server is running on.

Additionally, if you will be running RosettaHTS or any other protocol that results in very large numbers of structures being output to a single database (more than 100,000), the database schema can be altered to reduce the number of `INSERT` statements necessary to write a complete model. This is accomplished by modifying the Rosetta database storage schema to produce a single record per residue rather than a single record per atom. This enormously reduces the storage and network bandwidth requirements at the cost of data analyzability, and should be considered a required setting when using RosettaHTS. Compact schema mode is enabled with the following option:

```
-inout:use_compact_residue_schema true
```

B.6.2 Writing to a database

Once the database connection options above are specified, Rosetta can be configured to write all output models to the specified database with a single output option:

```
-out:use_database
```

If the use of a Database Filter is desired, it should be specified with another option:

```
-out:database_filter <filter_name> <score_term> <value>
```

Where `<filter_name>` is the name of the database filter to use, `<score_term>` is the name of the scoring term to filter by, and `<value>` is the cutoff value to use. At the time of writing, the following filters exist:

- `TopPercentOfEachInput` – Output models in the top *n*% of models for each input structures by specified score. The specified cutoff value should be a decimal between 0 and 1.

- `TopPercentOfAllInputs` – Output models in the top *n*% of models over all input structures by the specified score. The specified cutoff value should be a decimal between 0 and 1.

- `TopCountOfEachInput` – Output the top *n* models for each input structures by the specified score. The specified cutoff value should be an integer.

- `TopCountOfAllInputs` – Output the top *n* models over all input structures by the specified score. The specified cutoff value should be an integer.

B.6.3 Reading from a database

Rosetta can be configured to read input models from the specified database with a single input option:

```
-in:use_database
```

Note that both `-in:use_database` and `-out:use_database` can be used simultaneously. This is valuable in an interactive protocol, as the results from the current round of iteration can be written to the same database as the results from the previous round.

It is frequently desirable to read only a subset of structures from a database. If the structure IDs of the desired structures are known they can be specified with the following command:

```
-in:dbms:struct_ids <struct_id_list>
```

Where `<struct_id_list>` is a space separated list of structure IDs.

If the structure IDs are not known, it is possible to select a subset of the database using an SQL select statement:

```
-in:select_structures_from_database <statement>
```

where `<statement>` is an SQL statement that performs a `SELECT` operation that returns either a struct_id or tag column from the database. As a simple example, the statement:

```
SELECT struct_id FROM job_string_real_data
        WHERE data_key = "total_score"
        ORDER BY data_value ASC LIMIT 500;
```

Would select the structure IDs of the 500 lowest scoring models by total score. This command line option is potentially dangerous as minimal syntax validation is performed by Rosetta and the specified statement is executed on the SQL server. Make sure only to use this command line option with trusted user input, and do not expose any protocol that uses it as a web server.

Appendix C

Development and testing of knowledge based scoring functions for RosettaLigand initial placement

C.1 Introduction

The improvements in the performance of the RosettaLigand docking algorithm described in Chapter III result entirely from changes to the sampling algorithm in the initial placement phase of docking. The ligand scoring grids used in the initial placement algorithm and originally described by (Davis and Baker, 2009) are relatively primitive, containing only broad information about the acceptable and unacceptable regions for ligand placement within the protein binding pocket. Adding more detailed information to these scoring grids has the potential to further improve the accuracy and speed of RosettaLigand.

Here we describe the development of two new sets of scoring grids based on on KBPs. The first is a set of two scoring grids encoding shape complementarity and hydrogen bonding information. The second is a set of 20 scoring grids encoding the probability of a ligand atom existing within a given distance and angle of a canonical amino acid. The aim of these KBPs is to provide additional information about energy favorable ligand binding positions. While the results of benchmarking these algorithms, described in Section C.3, indicate that neither of the sets of scoring grids described in this chapter result in any significant improvement in RosettaLigand performance, KBP-based scoring grids may still be of use as a broad category of scoring method.

C.2 Methods

C.2.1 Shape complementarity scoring grid

C.2.1.1 The need for and challenge of shape complementarity calculation

Shape complementarity is a useful means of determining whether a ligand is in a well packed, non-clashing conformation relative to the protein binding pocket. However, rigorous computation of shape complementarity using a metric such as S_c (Lawrence and Colman, 1993), is too time consuming for our purposes. Therefore, a rapid approximation of shape complementarity was developed.

C.2.1.2 Description of shape complementarity calculations

The shape complementarity scoring grid is computed using the distance between each square in the grid and the edge of the nearest protein atom. For this method, the standard atom radii included in Rosetta were used. The fully populated grid represents the maximum possible radius of an atom at any given point without clashing with any other atom. An energy term representing the complementarity of a given ligand atom is then computed by subtracting the radius of that atom from the value in the nearest grid square, resulting in length of the gap between the ligand atom and the nearest protein atom (Figure C.1A). This gap length was converted to an energy by using a KBP.

The KBP was derived by computing the $-log(propensity)$ of a pair of protein and ligand atoms having a gap of a given distance. The KBP was derived using the Top8000[1] set of high quality crystal structures curated by the Richardson Lab at UNC. The resulting potential is shown in Figure C.1C. The left hand (clashing) side of the potential is modeled as a linear slope (indicated in red) when it passes above zero on the X axis, while the left hand side is modeled as zero when it reaches the zero axis. The rationale for scoring atoms with large gaps as zero is to avoid unnecessarily penalizing correct poses when a large portion of the ligand is exposed to solvent. As RosettaLigand uses an implicit solvation model, solvent molecules are not present, and thus not represented in the scoring grid. The solvation score is applied during the refinement step when the full RosettaLigand score function is used.

[1] http://kinemage.biochem.duke.edu/databases/top8000.php

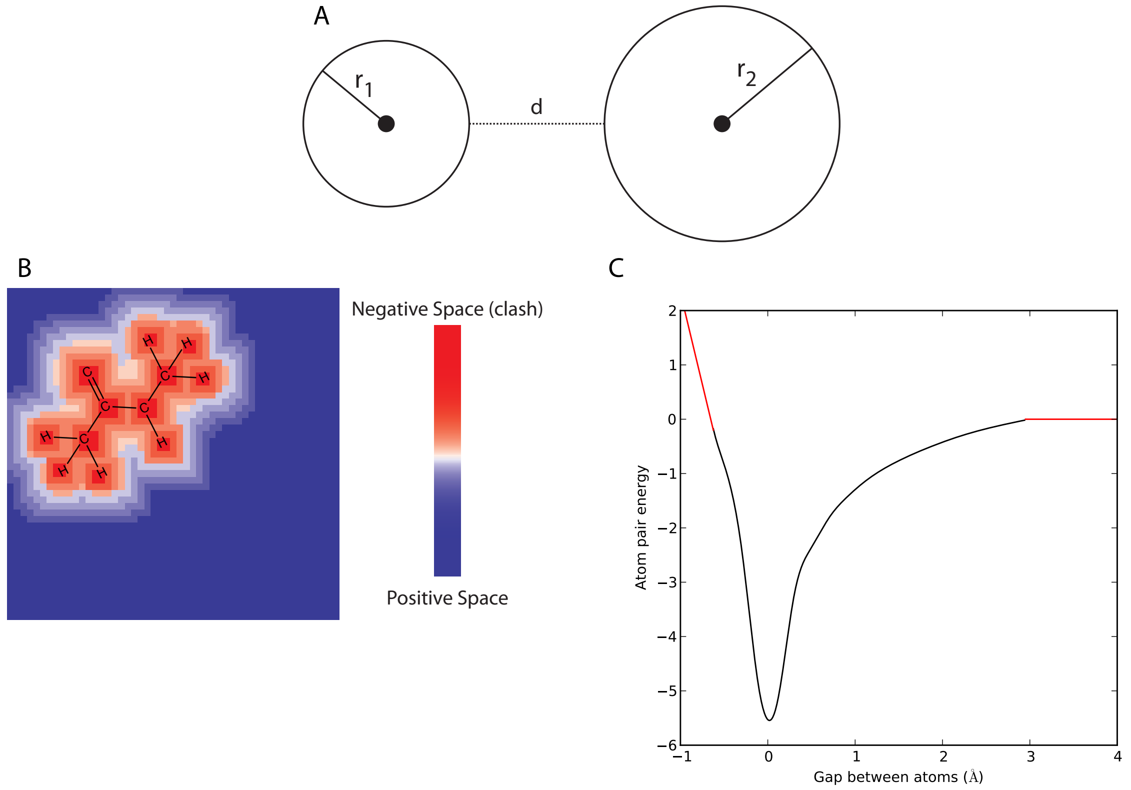


Figure C.1: A schematic of the shape complementarity grid. A) illustrates the computation of the gap distance between two atoms. B) represents the gap distances placed in the scoring grid. C) is a plot of the KBP. The region directly computed based on the KBP is in black. Linear functions applied to values beyond the bounds of the potential are in red.

C.2.2 Hydrogen bond scoring grid

C.2.2.1 Description of hydrogen bond scoring calculations

A pair of scoring grids are used to model hydrogen bonding. As the goal of the initial placement algorithm is to provide a rapid first approximation of ligand position, only distances between hydrogen bond donors and acceptors are taken into account, and angular information is ignored. A KBP was created based on the $-log(propensity)$ of a hydrogen bond donor atom being within some distance of a hydrogen bond acceptor atom, using the same Top8000 protein set described previously (Figure C.2A). Separate scoring grids are computed for hydrogen bond donors and acceptors. To compute the hydrogen bond donor scoring grid, the distance from each grid square to the nearest hydrogen bond donor is computed, and the score from the KBP is stored in the grid square (Figure C.2B). The same process is followed to compute the hydrogen bond acceptor scoring grid. Hydrogen bond donor atoms are scored based on the value of the nearest grid square in the appropriate scoring grid.

C.2.3 3D angle based scoring grid

The shape complementarity scoring grid described in Section C.2.1 is relatively simplistic. Many interactions between ligand atoms and protein side-chains can only be properly expressed using both the angle and distance between the atoms. A purely distance-based scoring grid will be unable to properly measure interactions such as hydrogen bonding, $\pi - \pi$ stacking, etc. To attempt to remedy that situation, another novel scoring grid was developed. A KBP was computed for each of the 20 canonical amino acids, based on a distance and two angles between a ligand atom and a protein amino acid. The distance is calculated between the ligand (query) atom and the Cβ atom of the amino acid. The Θ angle is computed as the cosine of the angle Query-Cβ-Cα, and the Φ angle is computed as the dihedral angle between Hα-Cα-Cγ-Query.

The Top8000 set of protein structures was used to generate the KBP. The distance and

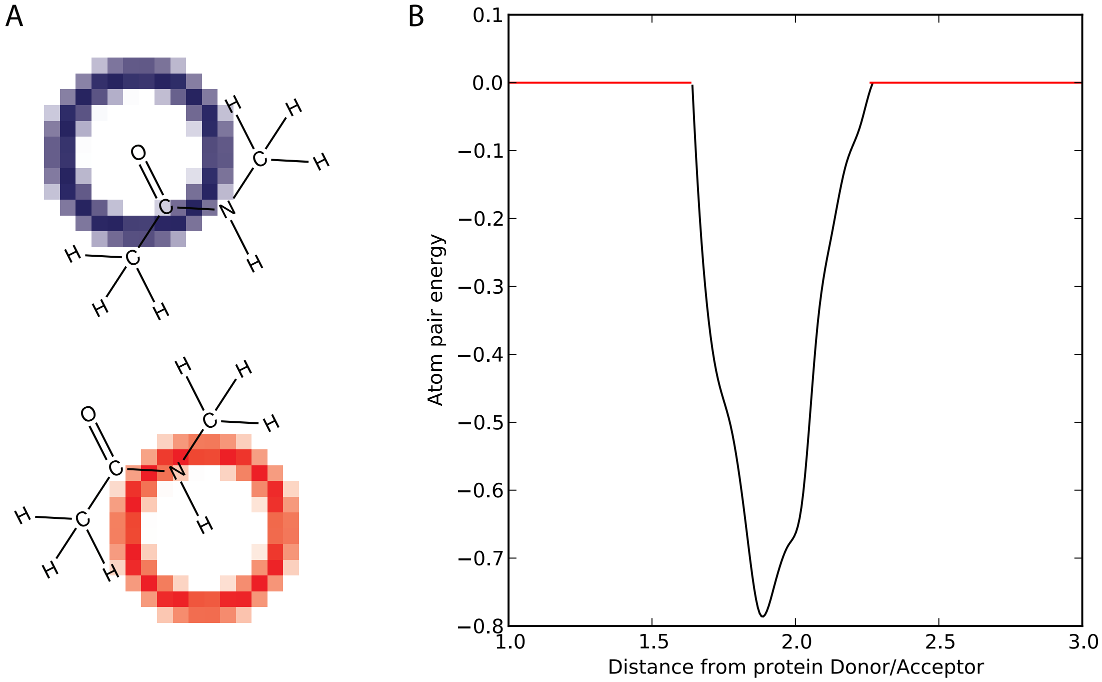


Figure C.2: A schematic of the hydrogen bonding grid. A) A schematic illustrating the placement of scoring information in the hydrogen bonding scoring grids. B) The KBP used to populate the scoring grids. The region directly computed based on the KBP is in black. Linear functions applied to values beyond the bounds of the potential are in red.

two angles described above were computed for each ligand atom and each protein amino acid in the set. The distance and angle data were then aggregated for each of the 20 amino acids, resulting in 20 KBPs representing the energy associated with a ligand atom located in the vicinity of a given canonical amino acid. The scores generated by these 20 KBPs were then summed to produce a single scoring grid representing the total energy of ligand atoms located in the protein binding site.

By computing the KBP using internal coordinates relative to the Cβ atom, the resulting energy term is independent of the orientation of the amino acid in the protein structure. The internal coordinate frame of the energy term can be converted to cartesian coordinates during scoring grid construction. When the scoring grid is constructed, the sum of the KBP energies associated with each amino acid surrounding a given scoring grid space is used as the total energy for that grid square.

As the KBP is based on the position of atoms in X-ray crystal structures, it does not take into account the presence of unordered solvent. The result is that areas of the ligand binding pocket normally occupied by water appear empty and will therefore have unfavorable scores. Additionally, clashes with the protein backbone should be avoided regardless of the KBP score. To address both of these issues, the angular KBP-based grid described above was combined with the previously implemented binary scoring grid, ensuring that available space in the ligand binding pocket has a nominally favorable score, while severely clashing positions are always avoided. The role of the KBP-based grid is to add nuance to the initial placement scoring function.

C.2.4 Description of benchmarking sets

The CSAR derived benchmarking set described in Chapter III is capable of docking the majority of ligands successfully using the TRANSFORM initial placement algorithm and the previously implemented binary scoring grids. In order to make it easier to distinguish the effects of the newly implemented scoring functions on docking success, a more difficult benchmark derived from the Q-Dock(Brylinski and Skolnick, 2008) homology modeling benchmark was used to test the new scoring grids. The new benchmarking set consists of 154 protein-ligand pairs. Each protein model in the benchmark was a comparative model made using an X-ray crystal structure template. The homology models in the Q-Dock benchmarking set are of a wide range of quality, ranging from 1.4 Å to 24.0 Å RMSD from the template structure. For each protein-ligand pair, an ensemble of 10 models was created by relaxing the homology model provided as part of the Q-Dock homology modeling benchmarking set.

C.3 Results and Discussion

C.3.1 Neither of the evaluated KBPs significantly improves docking performance

Figure C.3 compares the effect of the KBPs described in Section C.2 on docking performance and efficiency. In the left panel, the fraction of systems in the Q-Dock benchmark in which the lowest scoring model was under 2.0 Å RMSD to the crystal structure was plotted. In the right panel, the fraction of systems in which the best scoring model was under 4.0 Å RMSD to the crystal structure was plotted. In these figures, the designation "3-D" refers to the internal coordinate based 3-D KBP described in Section C.2.3, and "1-D" refers to the 1-D KBPs described in sections C.2.1 and C.2.2

In order to determine whether any apparent differences in model performance seen in Figure C.3 are statistically significant, a Welch's T-Test was used to compare the performance of pairs of protocols. The Welch's T-test is a method that tests the hypothesis that two independent samples with potentially unequal variances have identical means. This test was applied to pairs of methods for each set of samples plotted in Figure C.3, allowing the statistical significance of method changes to be plotted as a function of sample size. Figure C.4 shows that the only statistically meaningful improvement in RosettaLigand performance is obtained by the implementation of the TRANSFORM initial placement algorithm, while choice of scoring grid appears to have no impact.

Figure C.3 appears to indicate a slight change in the performance of the TRANSFORM/1-D/MCM protocol relative to the TRANSFORM/1-D/MIN protocol at samples sizes over 800. The T-test analysis in Figure C.4 suggests that the differences in performance observed here may be statistically significant, with T-test values dipping slightly below the 0.05 threshold. However, when a 2.0-Å cutoff is used as the metric for docking success, the TRANSFORM/1-D/MCM protocol exhibits improved performance over TRANSFORM/1-D/MIN, while the TRANSFORM/1-D/MIN protocol exhibits improved performance when a 4.0-Å cutoff is used. This discrepancy in performance when the cutoff threshold is changed warrants further investigation. Overall, we see that the TRANSFORMbased protocols result in a roughly doubled success rate over the TRANSROTbased protocols at both the 2.0 and 4.0-Å success cutoffs.

The requirement that the single lowest scoring model is below the RMSD cutoff is a stringent one. It is also worthwhile to assess whether any of the docking protocols studied here can improve the ability of RosettaLigand to arrive at a nearly correct solution, even if it is ultimately unsuccessful. In this case, a "nearly correctly docked" protein-ligand system is defined as one in which any model within 2.0 Rosetta Energy Units (REU) of the lowest scoring model were also within 2.0 or 4.0 Å RMSD of the crystal structure. Figure C.5 plots this property in a way analogous to Figure C.3. As expected, we see an increase in success rate with this less stringent criterion, with the new, less stringent criterion resulting in a roughly 35% success rate at the 2.0 Å cutoff, rather than the 20% success rate seen with the previously discussed criteria. As with Figure C.3, we see that the TRANSFORM initial placement algorithm generally performs better than the TRANSROT algorithm, but that the 1-D and 3-D KBPs do not result in notable changes compared to the original binary scoring grid.

Because final scoring is performed using the Rosetta energy function regardless of the choice of initial placement grid, the new grids described in this paper should primarily impact ligand sampling. In order to study the effect of sampling specifically, the percentage of

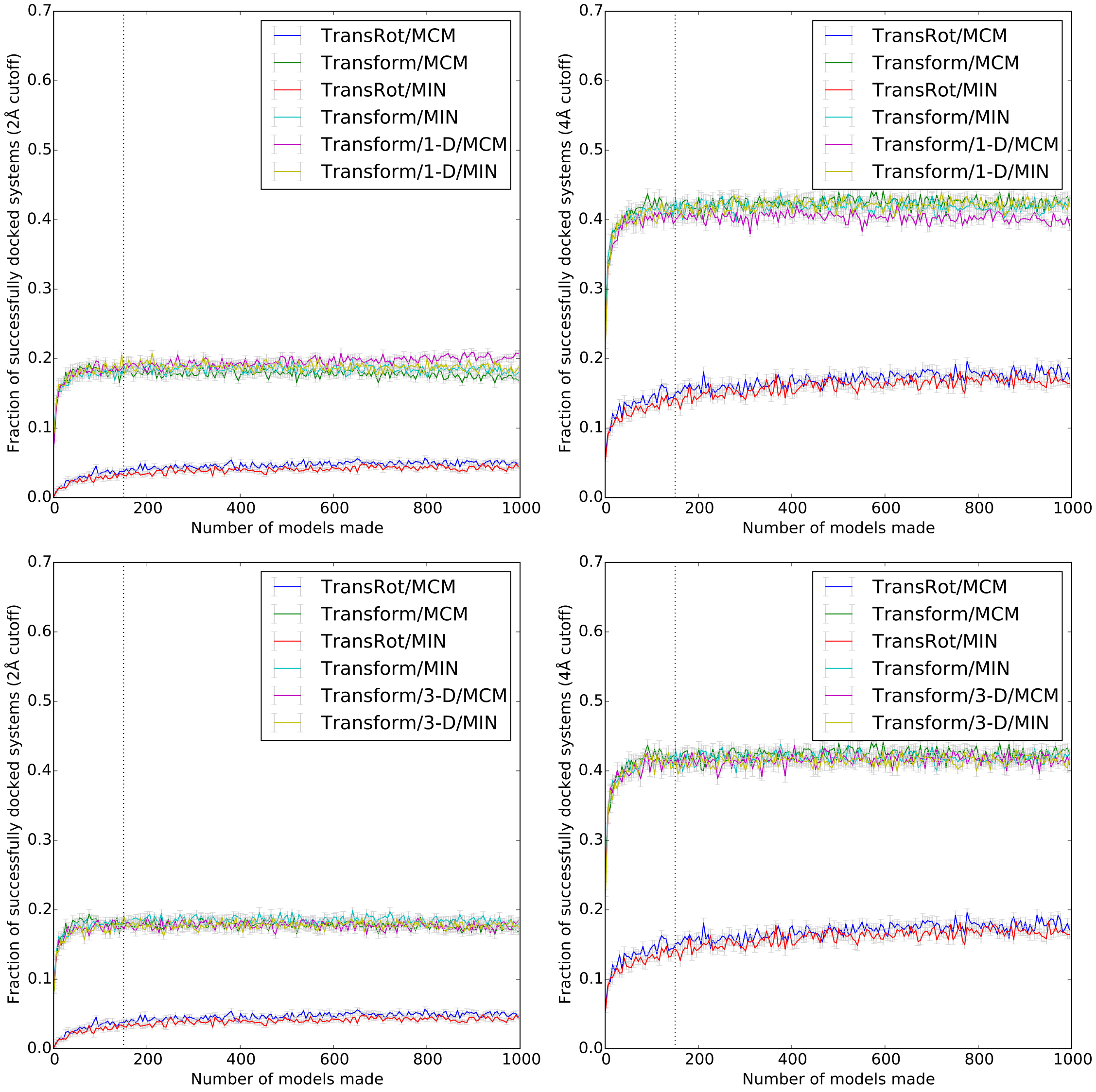
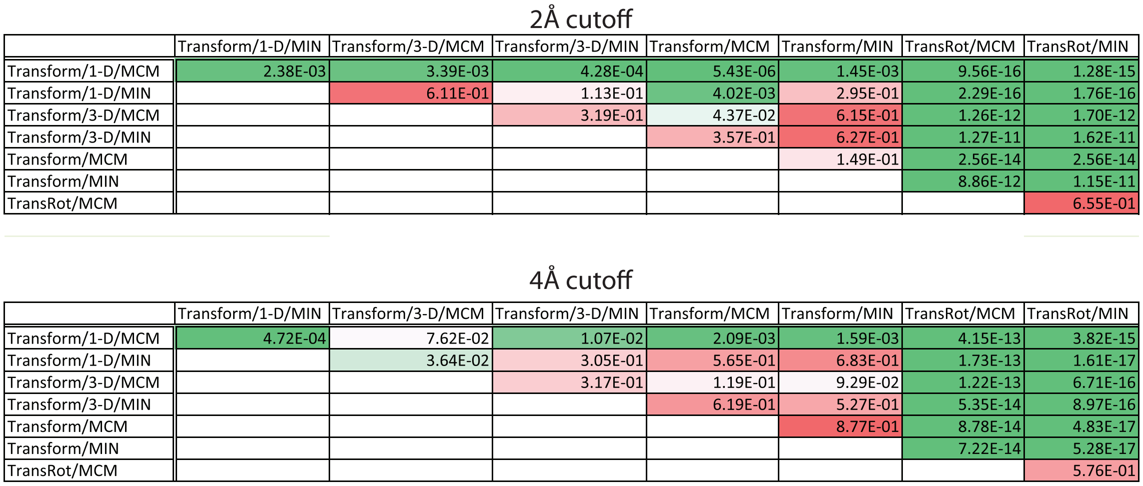


Figure C.3: The fraction of protein systems in which the lowest scoring model has an RMSD < 2.0 Å (left), and < 4.0 Å (right) to the native structure as a function of the total number of structures when docked into structures in the Q-Dock benchmarking set. A large pool of models were generated, and random subsamples were taken. Twenty random samples were taken for each point, and the means are plotted, with the error bars representing the standard deviation. Docking protocols that make use of the TRANSFORM algorithm reliably converge after approximately 150 models (dotted line).



2Å cutoff

	Transform/1-D/MIN	Transform/3-D/MCM	Transform/3-D/MIN	Transform/MCM	Transform/MIN	TransRot/MCM	TransRot/MIN
Transform/1-D/MCM	2.38E-03	3.39E-03	4.28E-04	5.43E-06	1.45E-03	9.56E-16	1.28E-15
Transform/1-D/MIN		6.11E-01	1.13E-01	4.02E-03	2.95E-01	2.29E-16	1.76E-16
Transform/3-D/MCM			3.19E-01	4.37E-02	6.15E-01	1.26E-12	1.70E-12
Transform/3-D/MIN				3.57E-01	6.27E-01	1.27E-11	1.62E-11
Transform/MCM					1.49E-01	2.56E-14	2.56E-14
Transform/MIN						8.86E-12	1.15E-11
TransRot/MCM							6.55E-01

4Å cutoff

	Transform/1-D/MIN	Transform/3-D/MCM	Transform/3-D/MIN	Transform/MCM	Transform/MIN	TransRot/MCM	TransRot/MIN
Transform/1-D/MCM	4.72E-04	7.62E-02	1.07E-02	2.09E-03	1.59E-03	4.15E-13	3.82E-15
Transform/1-D/MIN		3.64E-02	3.05E-01	5.65E-01	6.83E-01	1.73E-13	1.61E-17
Transform/3-D/MCM			3.17E-01	1.19E-01	9.29E-02	1.22E-13	6.71E-16
Transform/3-D/MIN				6.19E-01	5.27E-01	5.35E-14	8.97E-16
Transform/MCM					8.77E-01	8.78E-14	4.83E-17
Transform/MIN						7.22E-14	5.28E-17
TransRot/MCM							5.76E-01

Figure C.4: A Welch's T-Test was computed comparing the success rates between all pairs of protocols with a model sample size of 1000. At top, the cutoff for docking success is considered to be 2.0 Å RMSD. At bottom, the cutoff is 4.0 Å RMSD. Models below the statistical significance cutoff of 0.05 are colored in a gradient from white to green. Models above the statistical significance cutoff are colored in a gradient from white to red.

total models with an RMSD below 2.0 and 4.0 Å was plotted for each of the RosettaLigand protocols studied. As with previous figures, these percentages are plotted as a function of sample size in Figure C.6. Of note in this figure is that, when a 4.0-Å success cutoff is used, the rate of sampling success is slightly lower for the protocols using the 1-D KBPs (success rate of approximately 0.23) relative to the binary and 3-D KBPs (success rate of 0.25). Interestingly, when Figure C.6 is compared to C.3, we see that this slight decrease in overall sampling success rate does not correspond to a decreased ability of RosettaLigand to successfully select a correct binding pose using the Rosetta energy score. As the requirement for a system to be successfully docked in Figure C.3 is that the single lowest scoring model has an RMSD of < 4.0 Å, a 2% decrease in overall sampling success is unlikely to significantly impact the final docking success rate. Thich would explain the phenomenon seen here.

Figure C.7 compares the performance of RosettaLigand protocols at docking individual protein-ligand systems in the Q-Dock benchmarking set. Here, RMSD vs RMSD plots compare the average score of the lowest scoring models from 20 random samples of 1,000

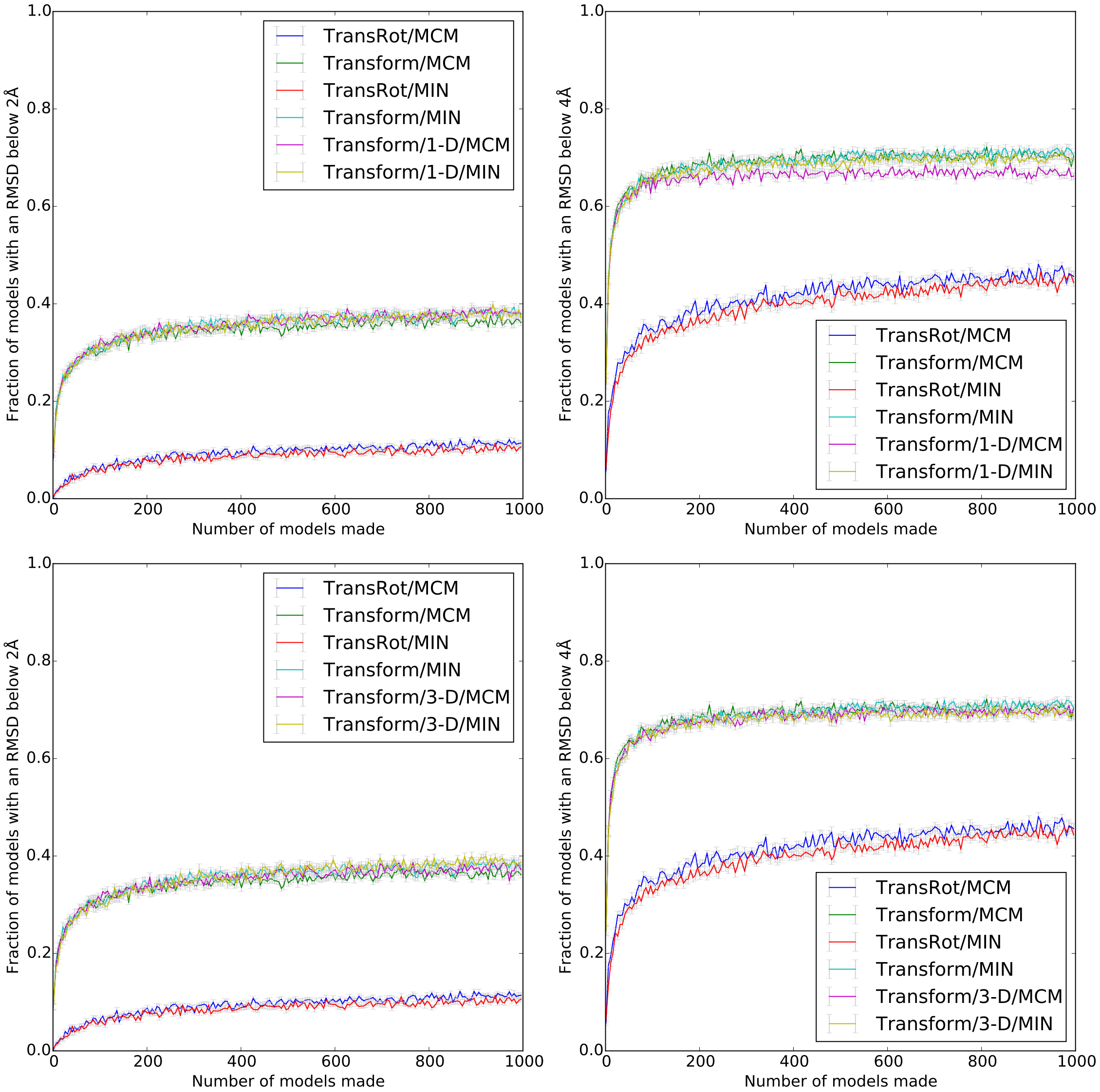


Figure C.5: The fraction of protein systems in which any model within 2.0 REU of the lowest scoring mode has an RMSD < 2.0 Å (left), and < 4.0 Å (right) to the native structure as function of the total number of structures when docked into structures in the Q-Dock benchmarking set. A large pool of models were generated, and random subsamples were taken. Twenty random samples were taken for each point, and the means are plotted, with the error bars representing the standard deviation. Docking protocols that make use of the TRANSFORM algorithm are reliably converged after approximately 150 models (dotted line).

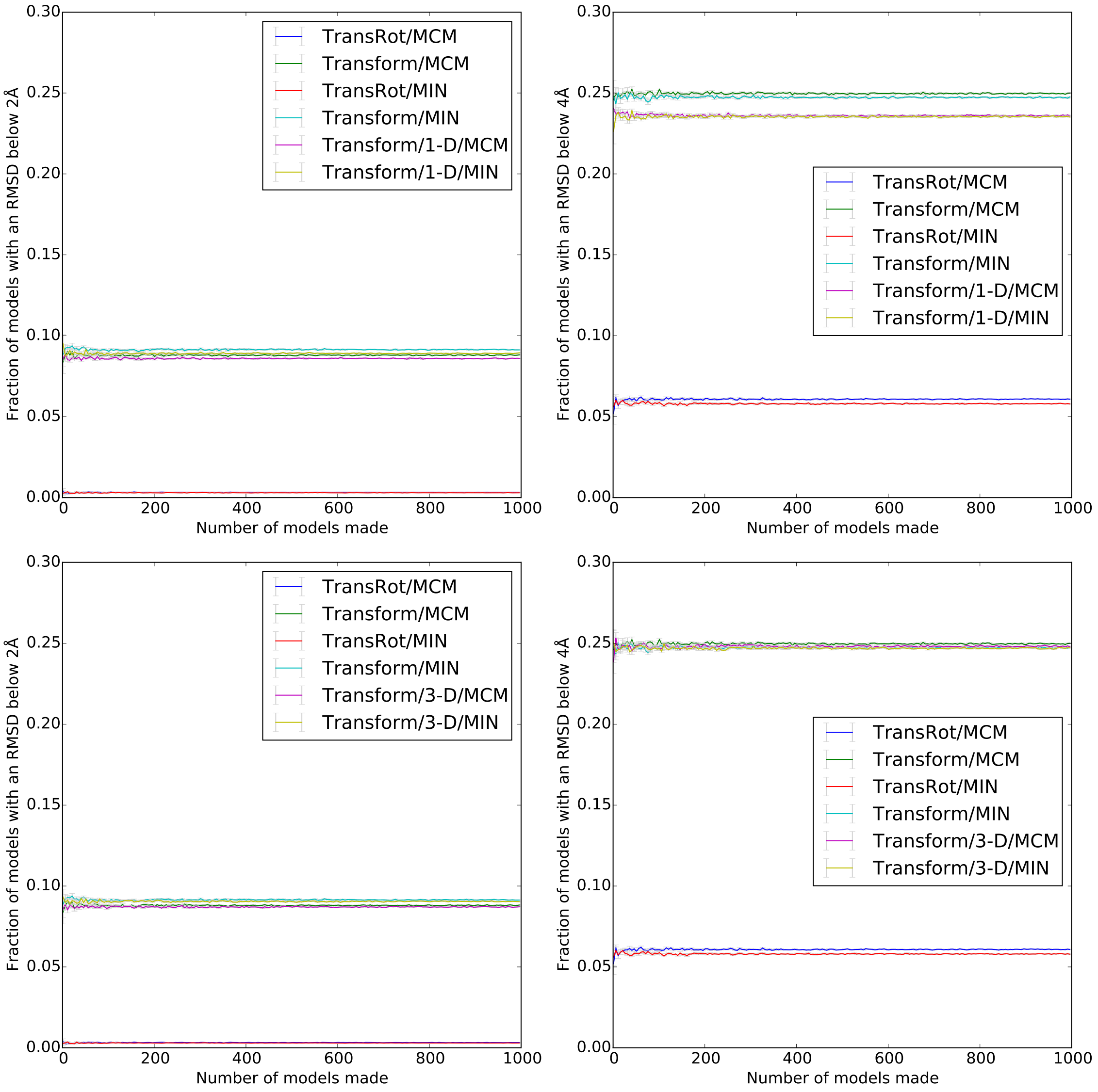


Figure C.6: The fraction of models with an RMSD below 2.0 Å (left) and 4.0 Å (right) regardless of score. A large pool of models was generated, and random subsamples were taken. Twenty random samples were taken for each point, and the means are plotted, with the error bars representing the standard deviation.

models each. The top left panel of Figure C.7 compares the performance of the TRANSROT/MCM and TRANSFORM/MCM protocols. The TRANSFORM initial placement algorithm resulted in the successful binding prediction of 16 protein-ligand systems that could not be docked successfully with the TRANSROT method, and one method could be successfuly docked with TRANSROT but not TRANSFORM. The TRANSFORM/3-D/MCM protocol (top right) resulted in three systems in which binding was successful over TRANSFORM/MCM, and four systems in which binding was successful in TRANSFORM/MCM over TRANSFORM/3-D/MCM. The TRANSFORM/1-D/MCM protocol (bottom left), resulted in four successfully docked system over TRANSFORM/MCM, while the TRANSFORM/MCM protocol resulted in three successfully docked systems over TRANSFORM/1-D/MCM. The TRANSFORM/1-D/MCM protocol (bottom right) resulted in eight models that were successfully docked over the TRANSFORM/3-D/MCM protocol, while three systems were successfully docked over TRANSFORM/1-D/MCM. From Figure C.7, we see that the TRANSFORM/1-D/MCM protocol affords only minimal improvement over the TRANSFORM/MCM protocol and that the TRANSFORM/3-D/MCM protocol results in a decrease in performance over TRANSFORM/1-D/MCM.

To further investigate the success and failure cases plotted in Figure C.7, we have rendered 3D representations for each system that was docked successfully in one system, but unsuccessfully in another. As a negative comparison, the five best scoring models from systems that could not be correctly docked by any of the studied docking protocols are rendered as well. Galleries of these models are shown in figures C.8, C.10, C.12, C.14, and C.16. Through a qualitative side-by-side comparison of these galleries, there are several broad conclusions we can draw regarding the properties of the RosettaLigand protocols studied in this chapter.

In Figure C.8, the models docked successfully by TRANSFORM/MCM or TRANSROT/MCM are plotted. The only difference between these two protocols is the method of sampling during initial placement, and the set of ligands which could be successfully docked by TRANS-

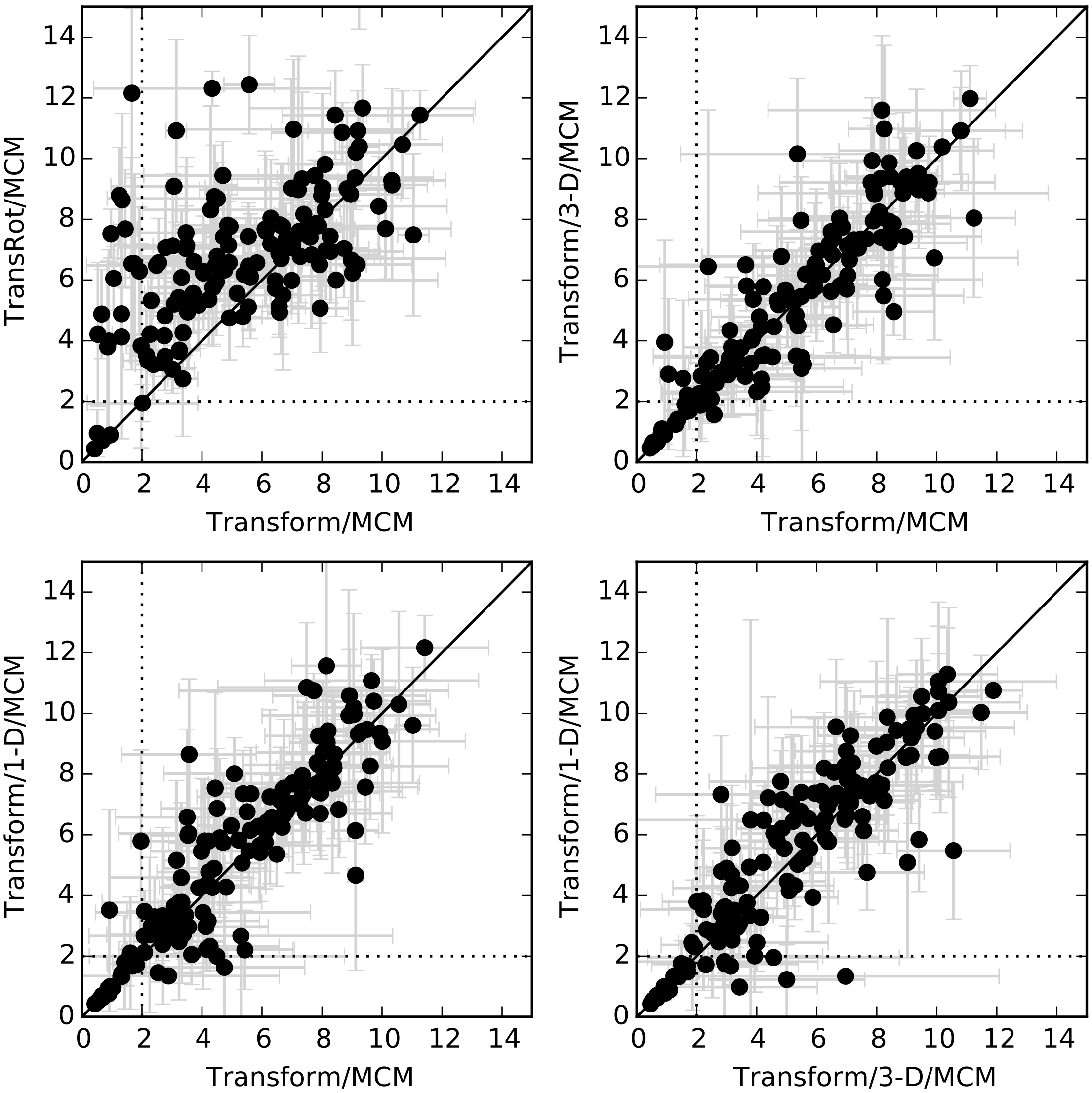


Figure C.7: RMSD vs RMSD plots comparing the performance of several pairs of protocols examined in this study. The average RMSD of the lowest scoring model for 20 samples of 1,000 models is plotted for each of the ligands in the Q-Dock benchmarking set.

FORM/MCM but not TRANSROT/MCM provide excellent examples of the value of improved sampling. This set of ligands is dominated by flat, relatively symmetric molecules, such as heme. The roughly symmetric nature of these ligands results in several similar (but not identical) poses that the ligand can assume, all with similar scores. Panels C.8D and C.8O are examples of this issue, in which the low scoring, high RMSD model obtained by the TRANSROT/MCM is rotated about the flat axis of the ligand.

On the other hand, Panels C.8A and C.8B are examples of highly flexible ligands in relatively large binding pockets. In the case of a highly flexible ligand, the ligand can assume a wide range of positions, many of which likely have similar scores but high RMSDs relative to the native structure. In both cases (large almost symmetric ligands and long highly flexible ligands), the large number of reasonably scoring local minima makes sampling difficult, and the additional sampling of the TRANSFORM initial placement algorithm is necessary to successfully dock these models.

Panel C.8Q plots the only system for which the TRANSROT/MCM protocol, rather than TRANSFORM/MCM, is successful. Here, the ligand is nearly symmetric, and while both protocols placed the ligand in the generally correct pose, the TRANSFORM/MCM model is inverted. This failure demonstrates that even the additional sampling of the TRANSFORM initial placement cannot guarantee successful sampling.

The overall impact on sampling of these systems is seen in Figure C.9, in which the Score vs. RMSD is plotted for all models produced by each RosettaLigand protocol for each system. In most of the 16 cases, TRANSROT/MCM is unable to sample any models with an RMSD below the 2.0 Å cutoff. Additionally, we see that in most cases, the symmetric ligand have multiple relatively low energy minima. The additional sampling afforded by the TRANSFORM initial placement algorithm increases the probability of sampling in the lowest energy, low RMSD binding position.

In figures C.10, C.12, and C.14, the sampling and refinement methods were kept constant, and the impact of the initial placement scoring grids were compared. There are a few interesting conclusions to be drawn from these figures. The TRANSFORM/1-D/MCM and TRANSFORM/MCM protocols are both more successful in docking large, flat, heme-like ligands relative to TRANSFORM/3-D/MCM, with TRANSFORM/3-D/MCM being unable to dock a heme-like ligand over either of the other scoring methods. While the TRANSFORM/1-D/MCM and TRANSFORM/MCM protocols seem better able to dock large ligands relative to TRANSFORM/3-D/MCM, the three smallest ligands which showed improvement (panels C.14A, C.14B, and C.10F), were successfully docked by TRANSFORM/3-D/MCM over the two other evaluated protocols. While three success cases out of a 148-system benchmark is insufficient to draw statistically meaningful conclusions, further investigation of the ability of the 3-D KBP to aid in the docking of small ligands may be warranted.

In Figure C.9 we see a drastic difference in Score vs. RMSD distribution as a result of the change in initial placement sampling method. In Figures C.11, C.13, and C.15, however, the initial placement sampling method is held constant, and only the initial placement scoring method is changed. In general, we see that the score vs. RMSD plots are nearly identical, indicate that the new scoring grids have minimal impact on the overall ability of RosettaLigand to sample low energy poses. In many of these cases, such as Figure C.13C, RosettaLigand docks the ligand in two distinct energy minima with nearly identical scores, and the differences between success and failure are the result of only tiny differences in the score of the lowest scoring model.

The examination of cases in which none of the RosettaLigand protocols were capable of successful docking is also informative. While the vast majority of systems in the Q-Dock benchmarking set could not be successfully docked, we selected “the best of the worst” models for rendering. Specifically, only systems in which the protein comparative model RMSD was < 2.0 Å to the template were considered for selection. Of these systems, the five lowest scoring models from systems which could not be successfully docked by any of the RosettaLigand protocols were selected and rendered in Figure C.16.

These failure cases provide good examples of a few broad categories in which RosettaLigand will, in general, have difficulty succeeding. While all the models are below 2.0 Å RMSD to the template structure, Panels C.16A and B are cases in which the ligand is surrounded by loops, and in which the loops initially occlude the ligand binding site. The occlusion of the binding site is a result of incorrect loop modeling during homology model generation. In both cases, RosettaLigand has moved the loops during the refinement stage to avoid clashing, but it is unable to correctly dock the ligands. In general, if loop modeling is required in the generation of a protein model, great care must be taken to accurately model loops which are in direct contact with the ligand prior to attempting docking, so as to avoid the behaviors seen in these two panels.

In Panel C.16C, we attempt to dock a highly flexible ligand with several binding site loops into a shallow pocket. This case is difficult for a number of reasons. In addition to the previously discussed difficulty of docking into binding sites involving loops, the large number of rotatable bonds increases the difficulty of sampling. Finally, the shallow binding pocket reduces the number of direct atom-atom interactions, which in turn reduces the amount of information available to the RosettaLigand energy function.

Panel C.16D involves a long, flexible ligand docked into a wide barrel structure. While this is not a surface binding site, as in Panel C.16C, the wide binding pocket similarly reduces the number of interactions. Panel C.16E involves a flexible, surface binding ligand, which will require substantial sampling to correctly dock. While RosettaLigand is potentially capable of successfully handling protein-ligand systems with these properties, the cases represented here are inherently challenging, and the current system will be unlikely to succeed relative to other cases.

Figure C.17 plots the Score vs. RMSD plots associated with Figure C.16. Of the five systems plotted, only the system plotted in Panel C.17A was able to successfully sample substantial numbers of models below 2.0 Å. In the case of Panel C.17A, we see that while some models were sampled below the 2.0 Å cutoff, the system has several local minima, some of which have a lower score than the correct binding position. Based on the Score vs. RMSD plots presented here, and on qualitative inspection of individual success and failure cases, it appears that while the new TRANSFORM initial placement algorithm greatly improves the effectiveness of sampling, further improvements in both sampling and in the Rosetta energy function are required.

C.3.2 There is minimal correlation between ligand docking success and comparative model accuracy

While the qualitative analysis above provides a number of valuable insights into the performance characteristics of RosettaLigand, a more quantitative analysis is also valuable. The range of accuracy of the comparative models in the Q-Dock benchmarking set gives us the opportunity to investigate the relationship between model accuracy and docking success. If such a relationship did exist, it would be useful for making decisions about the feasibility of computational docking experiments. However, we see from Figure C.18 that no such relationship exists. In this figure, we plot the relationship between the RMSD of the lowest scoring ligand to the crystal structure, and the RMSD of the protein comparative model to the protein crystal structure. At left in Figure C.18 is the relationship between the ligand RMSD and the all-atom RMSD of the comparative model. At right in Figure C.18 is the relationship between the ligand RMSD and the binding pocket RMSD of the comparative model. Here, the binding pocket is defined as the set of residues within 10.0 Å of the center of the ligand. We see no correlation between the ligand RMSD and the all atom comparative model RMSD ($R^2 = 0.002$), or the binding pocket comparative model RMSD ($R^2 = 0.007$).

While there is no meaningful correlation, there are some rough guidelines that may be obtained from this analysis. We see that the successful ligand docking predictions involve protein models with an all-atom RMSD of < 10 Å to the template, and nearly all have RMSDs of < 5 Å. All successful predictions have a binding pocket RMSD of < 5 Å, indicating that, as expected, the accuracy of the ligand binding pocket is somewhat more

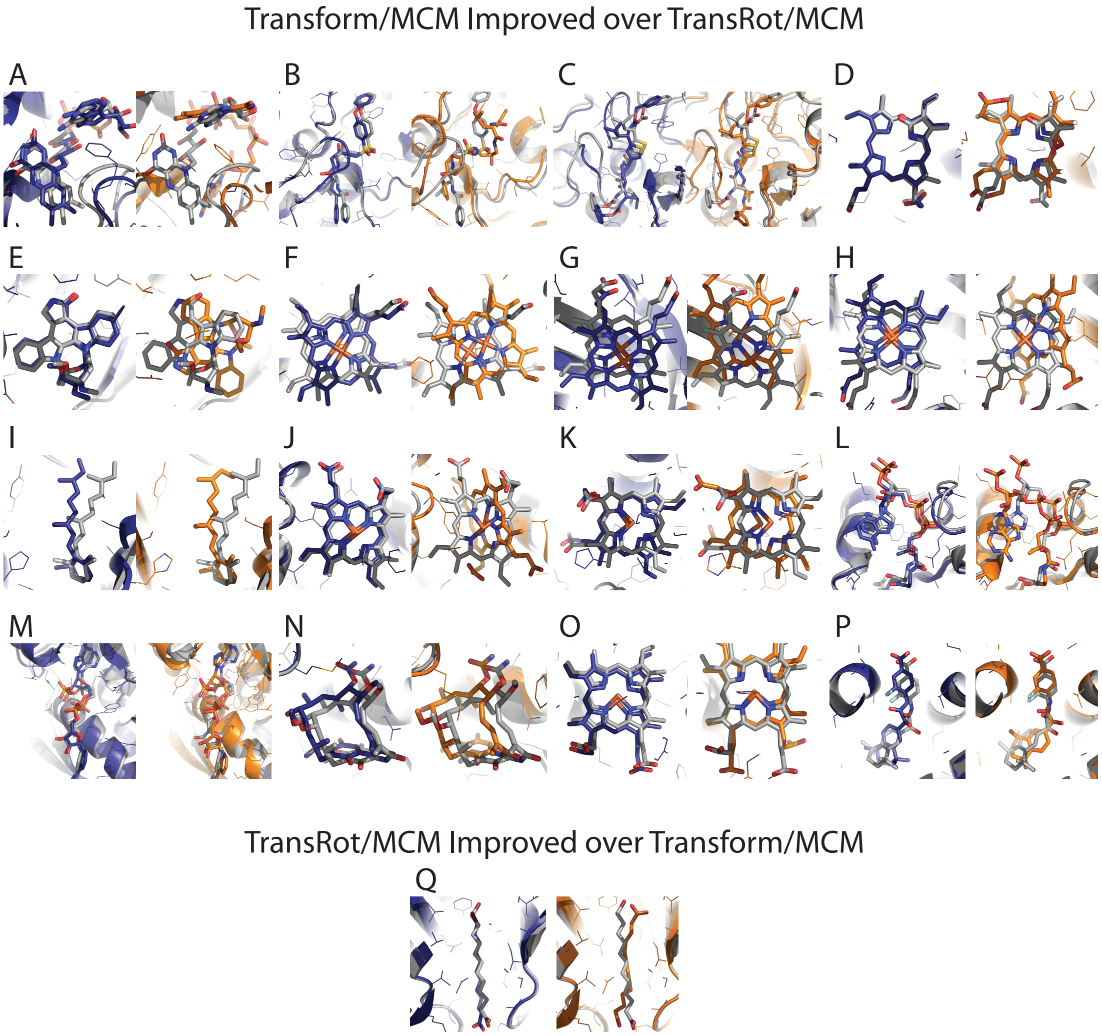

Transform/MCM Improved over TransRot/MCM
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
TransRot/MCM Improved over Transform/MCM
Q

Figure C.8: The lowest scoring model for protein-ligand systems that were docked correctly by TRANSFORM/MCM and incorrectly by TRANSROT/MCM (and vice versa). Correctly docked ligands are shown in blue, and incorrectly docked ligands are shown in orange. This Figure corresponds to the top left panel in Figure C.7. The following proteins are plotted (PDB IDs): A) 1a8p, B) 1au2, C) 1ayv, D) 1bvd, E) 1byg, F) 1cxc, G) 1d06, H) 1drm, I) 1fen, J) 1flp, K) 1myt, L) 1qsr, M) 1tcs, N) 1yet, O) 2hbg, P) 4lbd, Q) 1bso.

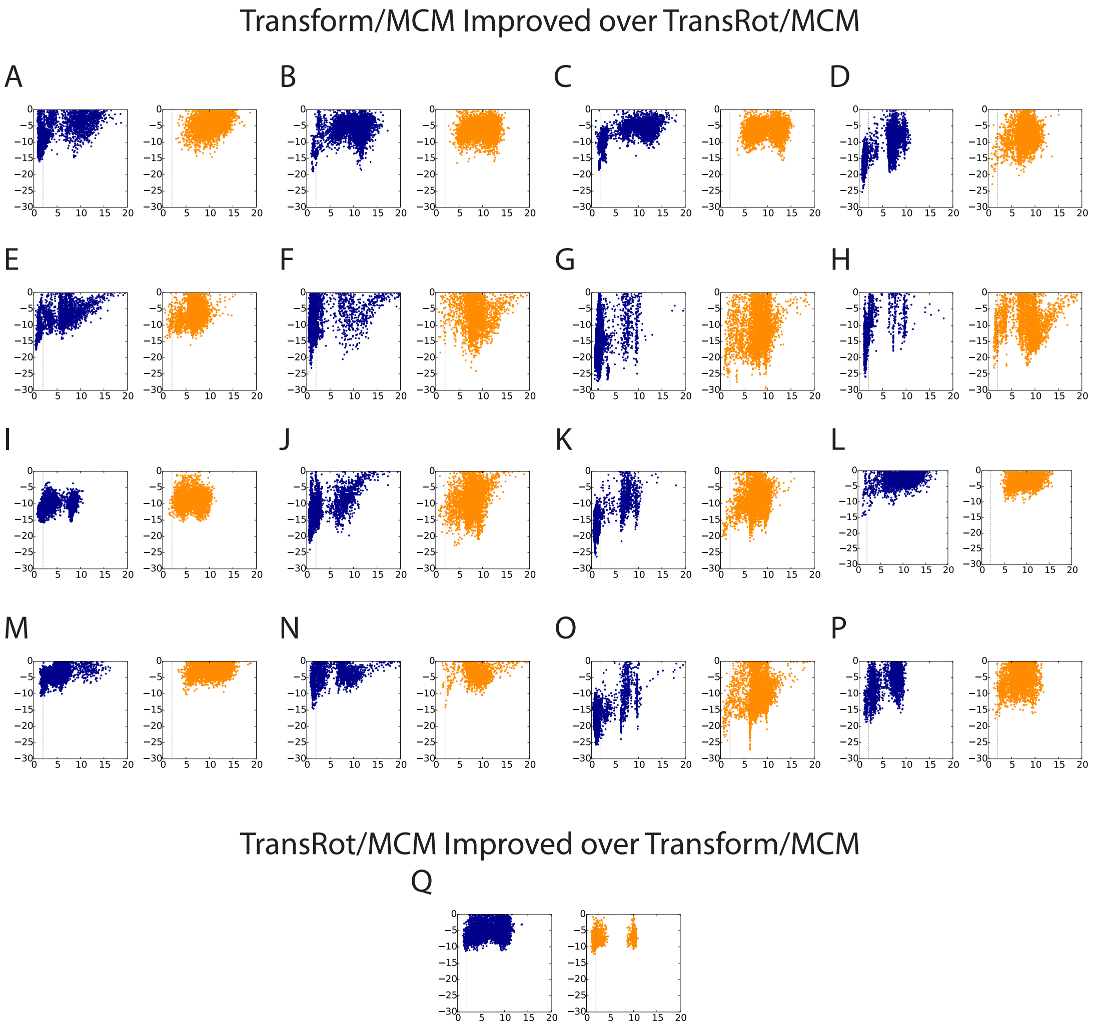
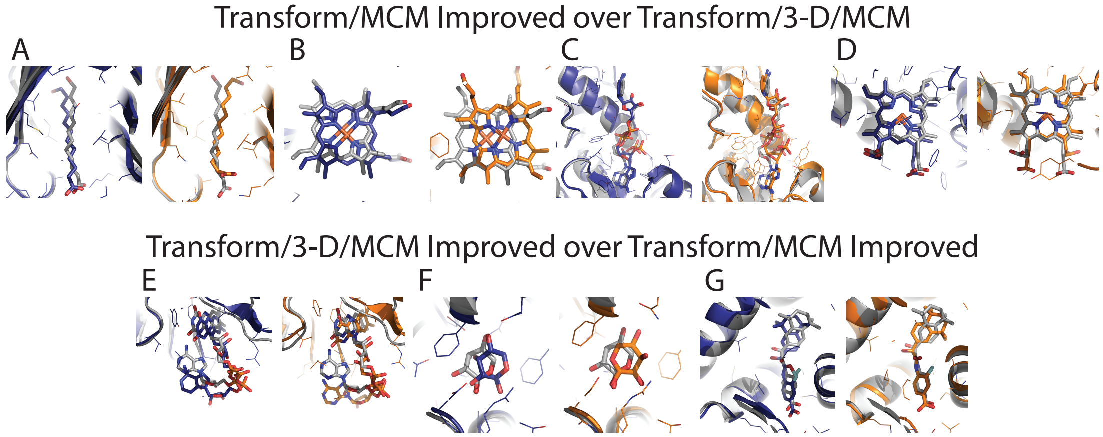


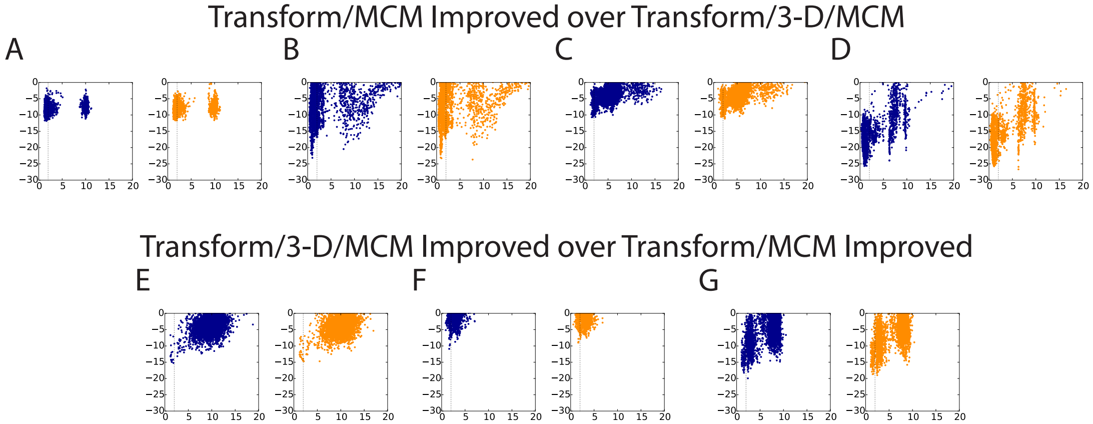
Figure C.9: Score vs. RMSD plots for protein-ligand systems that were docked correctly by TRANSFORM/MCM and incorrectly by TRANSROT/MCM (and vice versa). The Score vs. RMSD plot for the protocol that correctly docked the ligand is shown in blue, and and the plot for the protocol that incorrectly docked the ligand is shown in orange. The x-axis plots the RMSD in Å. The y-axis plots the score in Rosetta energy units. This Figure corresponds to the top left panel in Figure C.7. The following systems are plotted (PDB IDs): A) 1a8p, B) 1au2, C) 1ayv, D) 1bvd, E) 1byg, F) 1cxc, G) 1d06, H) 1drm, I) 1fen, J) 1flp, K) 1myt, L) 1qsr, M) 1tcs, N) 1yet, O) 2hbg, P) 4lbd, Q) 1bso.



Transform/MCM Improved over Transform/3-D/MCM

Transform/3-D/MCM Improved over Transform/MCM Improved

Figure C.10: The lowest scoring model for protein-ligand systems that were docked correctly by Transform/MCM and incorrectly by Transform/3-D/MCM (and vice versa). Correctly docked ligands are shown in blue, and incorrectly docked ligands are shown in orange. This Figure corresponds to the top right panel in Figure C.7. The following proteins are plotted (PDB IDs): A) 1bso, B) 1cxc, C) 1tcs, D) 2hbg, E) 1fdr, F) 2dri, G) 4lbd.



Transform/MCM Improved over Transform/3-D/MCM

Transform/3-D/MCM Improved over Transform/MCM Improved

Figure C.11: Score vs. RMSD plots for protein-ligand systems that were docked correctly by Transform/MCM and incorrectly by Transform/3-D/MCM (and vice versa). The Score vs. RMSD plot for the protocol that correctly docked the ligand is shown in blue, and and the plot for the protocol that incorrectly docked the ligand is shown in orange. The x-axis plots the RMSD in Å. The y-axis plots the score in Rosetta energy units. This Figure corresponds to the top right panel in Figure C.7. The following systems are plotted (PDB IDs): A) 1bso, B) 1cxc, C) 1tcs, D) 2hbg, E) 1fdr, F) 2dri, G) 4lbd.

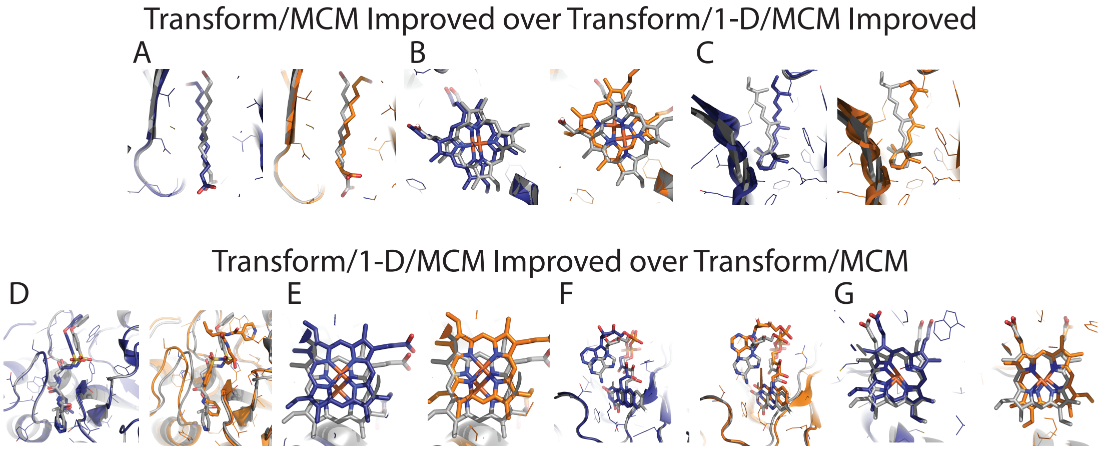

Transform/MCM Improved over Transform/1-D/MCM Improved
A
B
C
Transform/1-D/MCM Improved over Transform/MCM
D
E
F
G

Figure C.12: The lowest scoring model for protein-ligand systems that were docked correctly by TRANSFORM/MCM and incorrectly by TRANSFORM/1-D/MCM (and vice versa). Correctly docked ligands are shown in blue, and incorrectly docked ligands are shown in orange. This Figure corresponds to the bottom left panel in Figure C.7. The following proteins are plotted (PDB IDs): A) 1bso, B) 1cxc, C) 1fen, D) 1au2, E) 1cyo, F) 1fdr, G) 3c2c

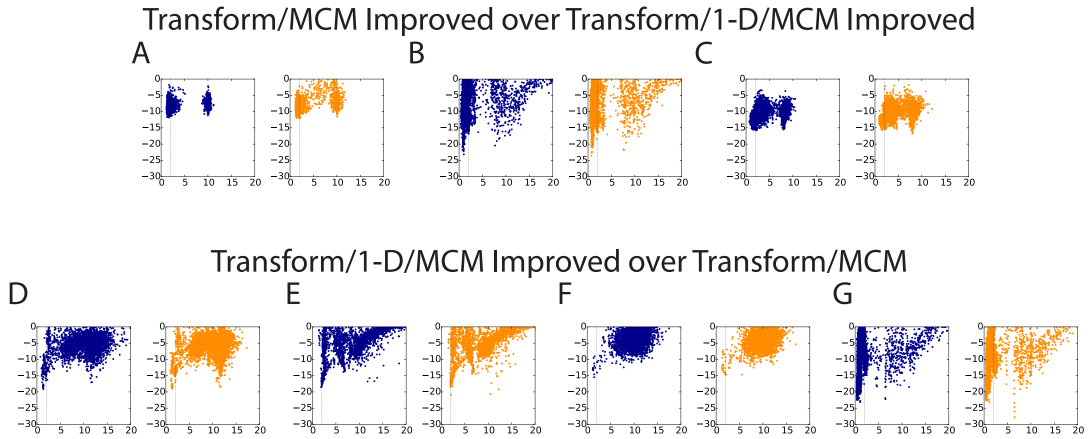


Figure C.13: Score vs. RMSD plots for protein-ligand systems that were docked correctly by TRANSFORM/MCM and incorrectly by TRANSFORM/1-D/MCM (and vice versa). The Score vs. RMSD plot for the protocol that correctly docked the ligand is shown in blue, and and the plot for the protocol that incorrectly docked the ligand is shown in orange. The x-axis plots the RMSD in Å. The y-axis plots the score in Rosetta energy units. This Figure corresponds to the bottom left panel in Figure C.7. The following systems are plotted (PDB IDs): A) 1bso, B) 1cxc, C) 1fen, D) 1au2, E) 1cyo, F) 1fdr, G) 3c2c

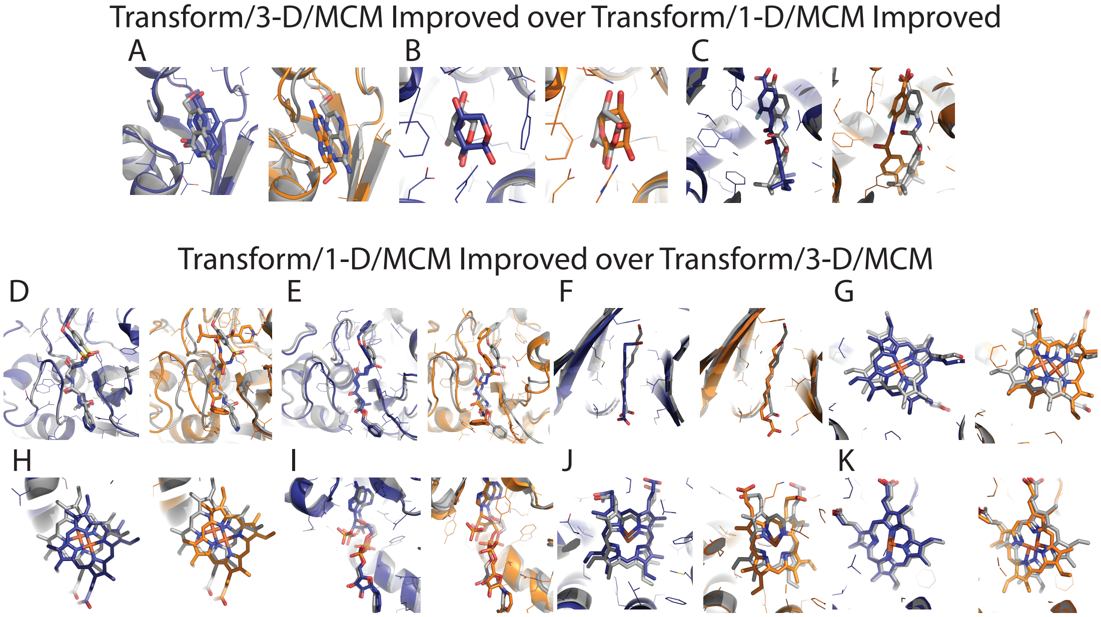

Transform/3-D/MCM Improved over Transform/1-D/MCM Improved
A
B
C
Transform/1-D/MCM Improved over Transform/3-D/MCM
D
E
F
G
H
I
J
K

Figure C.14: The lowest scoring model for protein-ligand systems that were docked correctly by TRANSFORM/3-D/MCM and incorrectly by TRANSFORM/1-D/MCM (and vice versa). Correctly docked ligands are shown in blue, and incorrectly docked ligands are shown in orange. This Figure corresponds to the bottom right panel in Figure C.7. The following proteins are plotted (PDB IDs): A) 2dhn, B) 2dri, C) 4lbd, D) 1au2, E) 1ayw, F) 1bso, G) 1cxc, H) 1cyo, I) 1tcs, J) 2hbg, K) 3c2c

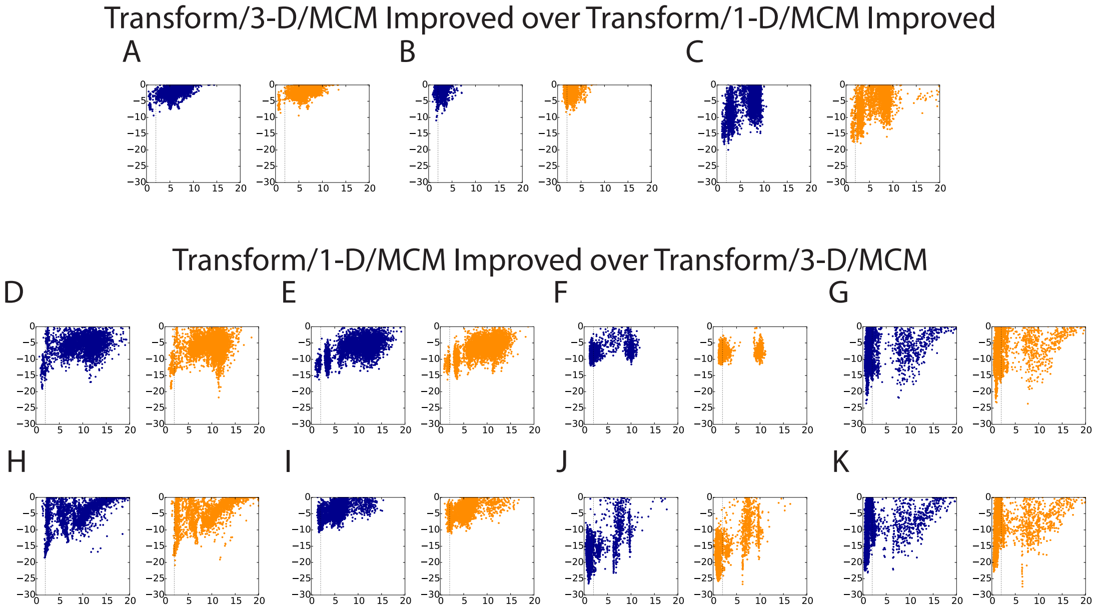


Figure C.15: Score vs. RMSD plots for protein-ligand systems that were docked correctly by Transform/3-D/MCM and incorrectly by Transform/1-D/MCM (and vice versa). The Score vs. RMSD plot for the protocol that correctly docked the ligand is shown in blue, and and the plot for the protocol that incorrectly docked the ligand is shown in orange. The x-axis plots the RMSD in Å. The y-axis plots the score in Rosetta energy units. This Figure corresponds to the bottom right panel in Figure C.7. The following systems are plotted (PDB IDs): A) 2dhn, B) 2dri, C) 4lbd, D) 1au2, E) 1ayw, F) 1bso, G) 1cxc, H) 1cyo, I) 1tcs, J) 2hbg, K) 3c2c

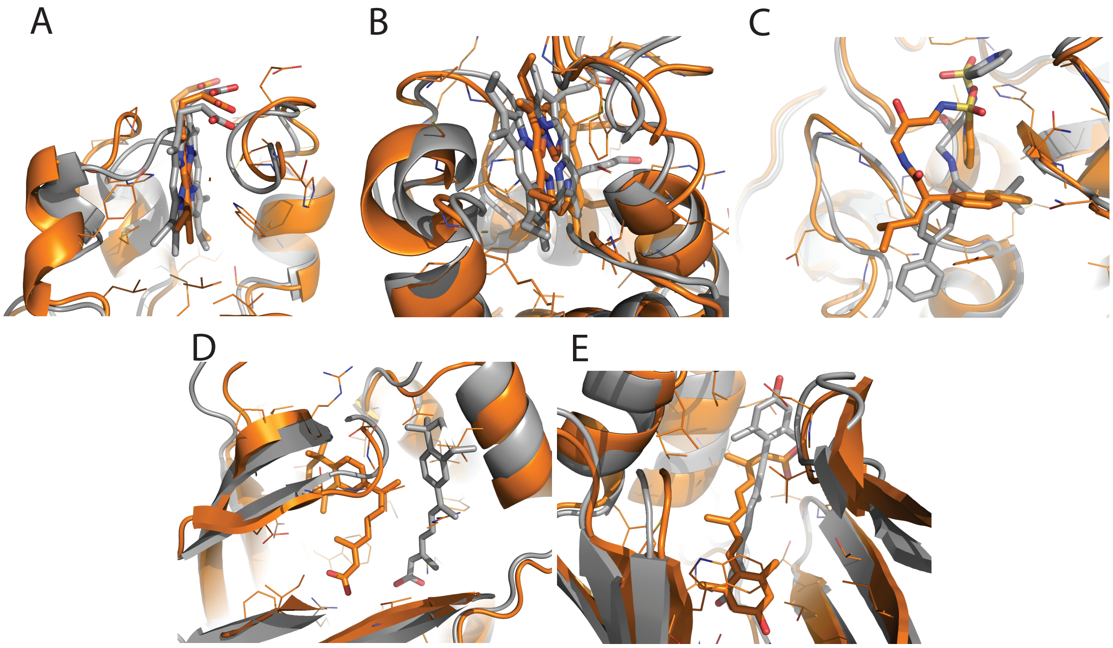


Figure C.16: The lowest scoring model for the five lowest scoring protein-ligand systems that were incorrectly docked by all evaluated RosettaLigand protocols. The following proteins are plotted (PDB IDs): a) 1cxy, b) 451c, c) 1bgo, d) 2cbs, e) 3cbs

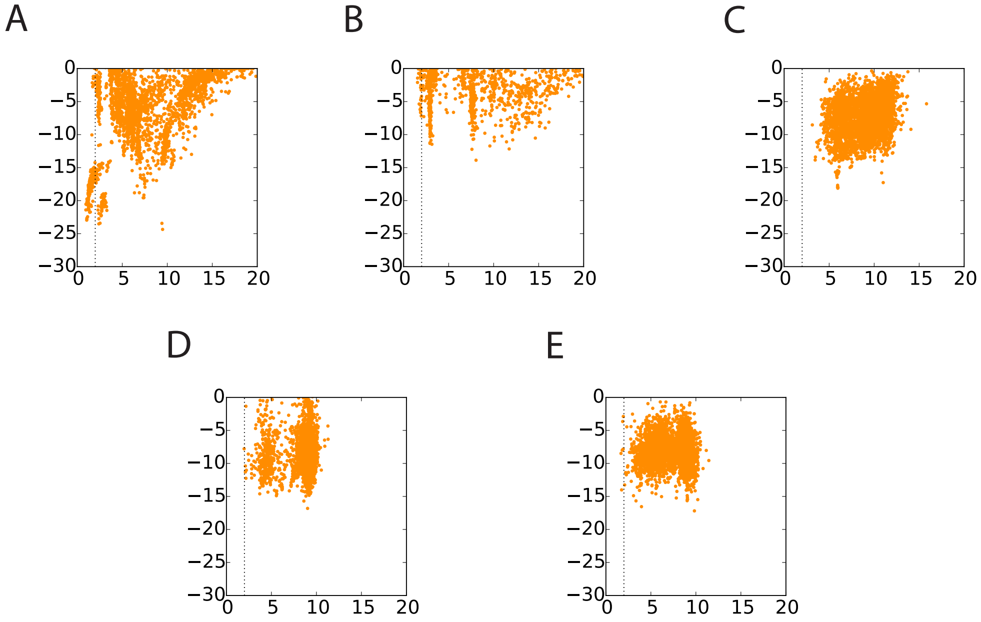


Figure C.17: Score vs. RMSD plots from the TRANSFORM/MCM protocol for protein-ligand systems that were incorrectly docked by all evaluated RosettaLigand protocols. The following systems are plotted (PDB IDs): a) 1cxy, b) 451c, c) 1bgo, d) 2cbs, e) 3cbs

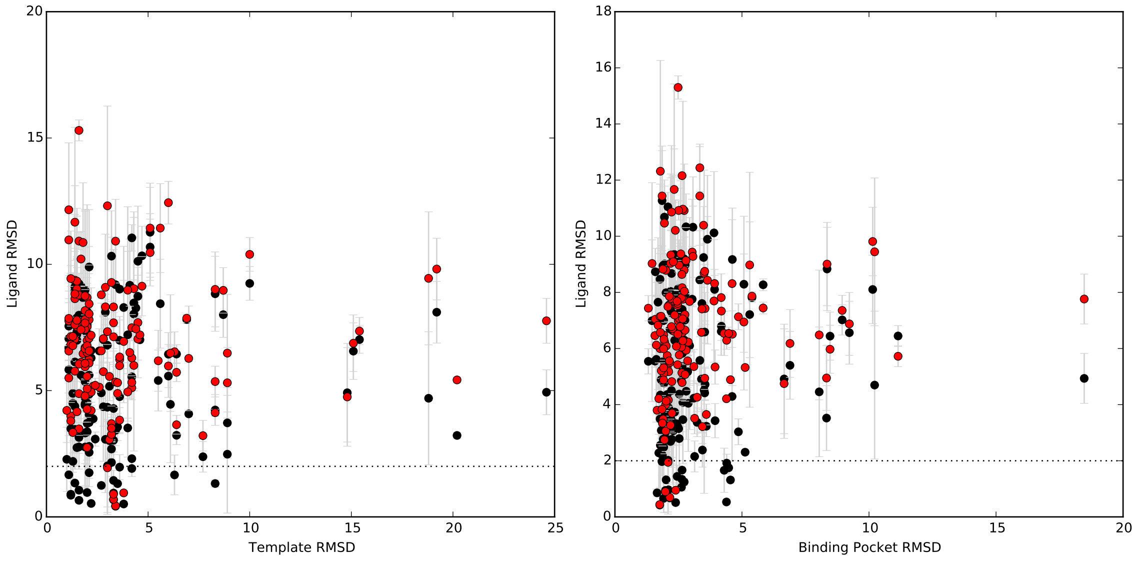


Figure C.18: At left, the correlation between the RMSD of the lowest scoring ligand docking prediction made by RosettaLigand and the all atom RMSD of the protein homology models to the template. At right, the correlation between the RMSD of the lowest scoring ligand docking prediction, and the binding pocket RMSD of the protein homology models to the template. Here, the binding pocket is defined any residue with at least one atom within 10.0 Å of the geometric center of the ligand. Models generated using the TRANSFORM/MCM algorithm are indicated in black, and models generated using the TRANSROT/MCM algorithm are in red. Error bars represent standard deviation.

important than the overall accuracy of the protein model.

It is notable that even among models with an RMSD to the template below < 5 Å, only 21 (17.3%) of protein-ligand pairs could be correctly docked by the TRANSFORM/MCM algorithm, which is the best performing of those tested in this study. This suggests that, while an accurate protein model is critical to successful docking, the success or failure of RosettaLigand is driven by properties other than protein model accuracy.

C.3.3 Ligand chemical properties do not appear to be well correlated with docking success

For pairs of RosettaLigand docking protocols tested in this experiment, the results were analyzed to look for correlations between docking success and ligand chemical properties. Specifically, the distribution of chemical properties among ligands that were successfully docked by both protocols in a pair, unsuccessfully docked by both protocols, or successfully docked by one protocol and unsuccessfully docked by another were studied. The aim of this analysis was to identify ligand properties that may lead to increased or decreased docking success among the tested protocols. The BCL was used to compute the girth, count of hydrogen-bond acceptors and donors, predicted $log(p)$, atom count, aromatic ring count, conjugated ring count, ring count, rotatable bond count, stereo center count, Total Polarizable Surface Area (TPSA), total charge, VDW surface area, VDW volume, molecular weight and radius of gyration for each of the 154 ligands in the benchmarking set.

Figure C.19 plots the distribution of ligand properties for successful and unsuccessful protein-ligand systems in the Q-Dock benchmarking set docked with the TRANSFORM/MCM and TRANSROT/MCM protocols. We see that ligands that are successfully docked by the TRANSFORM-based protocol but not the TRANSROT-based protocol are generally slightly larger and more flexible than ligands that can be docked successfully by both protocols. However, the distributions of the chemical properties generally overlap, so we are unable to extract hard predictors of docking success based on ligand property.

Figures C.20 and C.21 compare the performance of two pairs of protocols as a function of the chemical properties of the ligands in the Q-Dock benchmarking set. As these two pairs of protocols do not have statistically significant performance differences (Figure C.4), it is not expected that we will see significant differences in the chemical property distributions of successfully and unsuccessfully docked systems. In general, the measured chemical properties do not seem to have a significant impact on success rate between the 1-D KBP, 3-D KBP, and binary scoring grids. However, there are a few aspects of the analysis that many warrant further investigation. In Figure C.20, we see that the number of rings in systems that could be successfully docked when using the 1-D KBP scoring grid, rather than the binary scoring grid, is increased. Specifically, systems that were docked correctly with the 1-D grid and incorrectly with the binary grid had between 6-8 rings, while systems that were docked successfully with the binary grid and unsuccessfully with

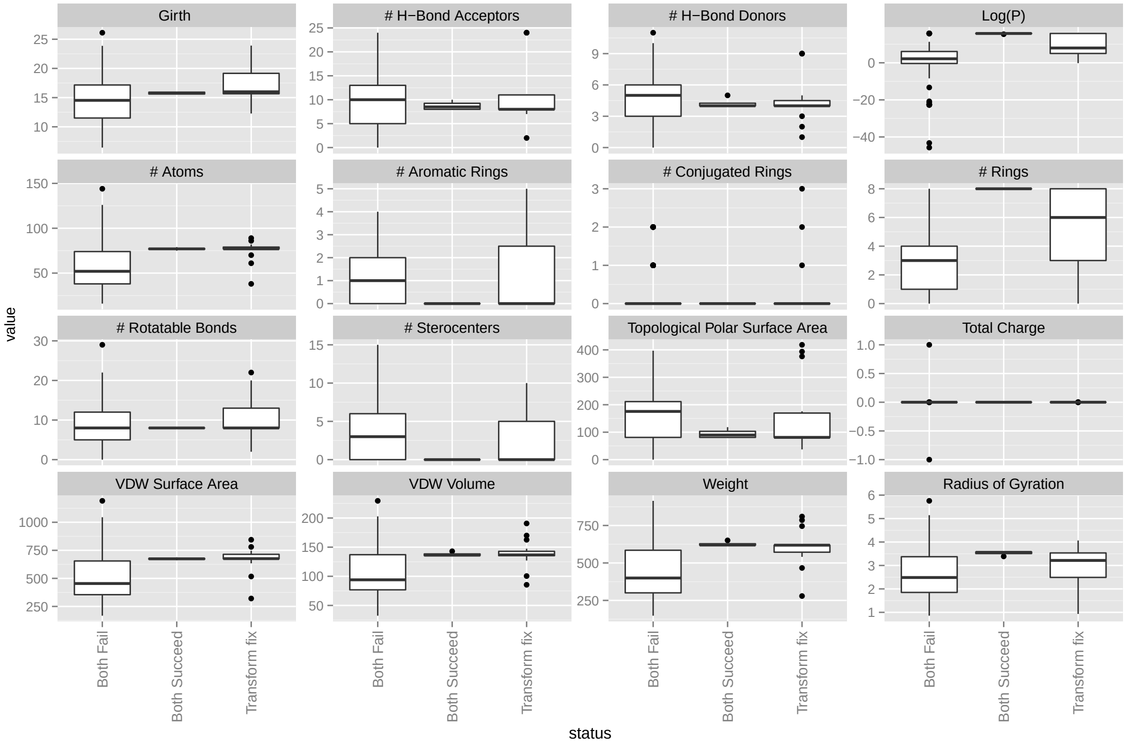


Figure C.19: Comparison of ligand property distributions for ligands docked with the Transform/MCM and TransRot/MCM protocols. "Transform Fix" is the group of ligands that could not be docked with the TransRot/MCM protocol but were successfully docked by the Transform/MCM protocol.

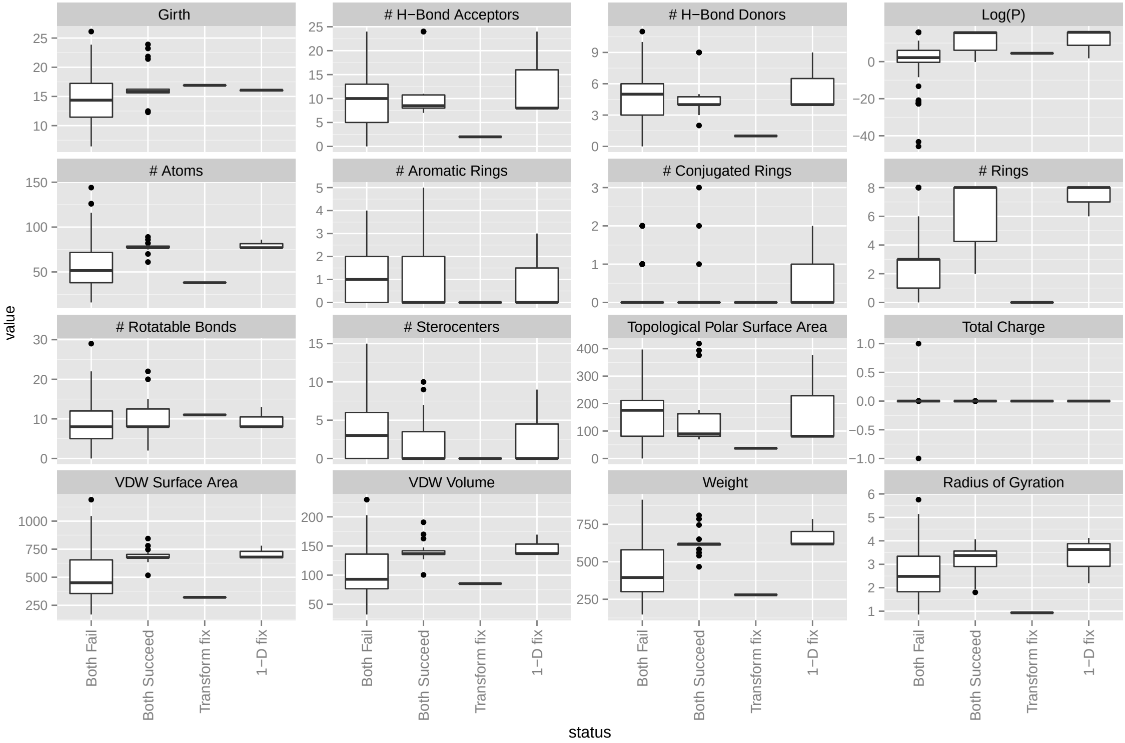


Figure C.20: Comparison of ligand property distributions for ligands docked with the TRANSFORM/MCM and TRANSFORM/1-D/MCM protocols.

the transform grid had no rings. While this distribution overlaps the distribution of ligands that could be successfully docked with both protocols (2-8 rings), the impact of the 1-D KBP scoring grid on the ability of RosettaLigand to dock large or complex ring systems is a potential area of further investigation.

C.3.4 Protein structural and chemical properties do not appear to be well correlated with docking success

The relationship between protein properties and ligand docking success can be analyzed in a similar manner to the analysis of ligand properties described in Section C.3.3. For each protein-ligand pair in the Q-Dock benchmarking set, the ligand was placed at the experimentally determined crystallographic position within the comparative model. Rosetta was then used to compute a set of 21 metrics describing the protein-ligand binding site. The following metrics were computed: Number of unsaturated hydrogen-bonds, total Rosetta

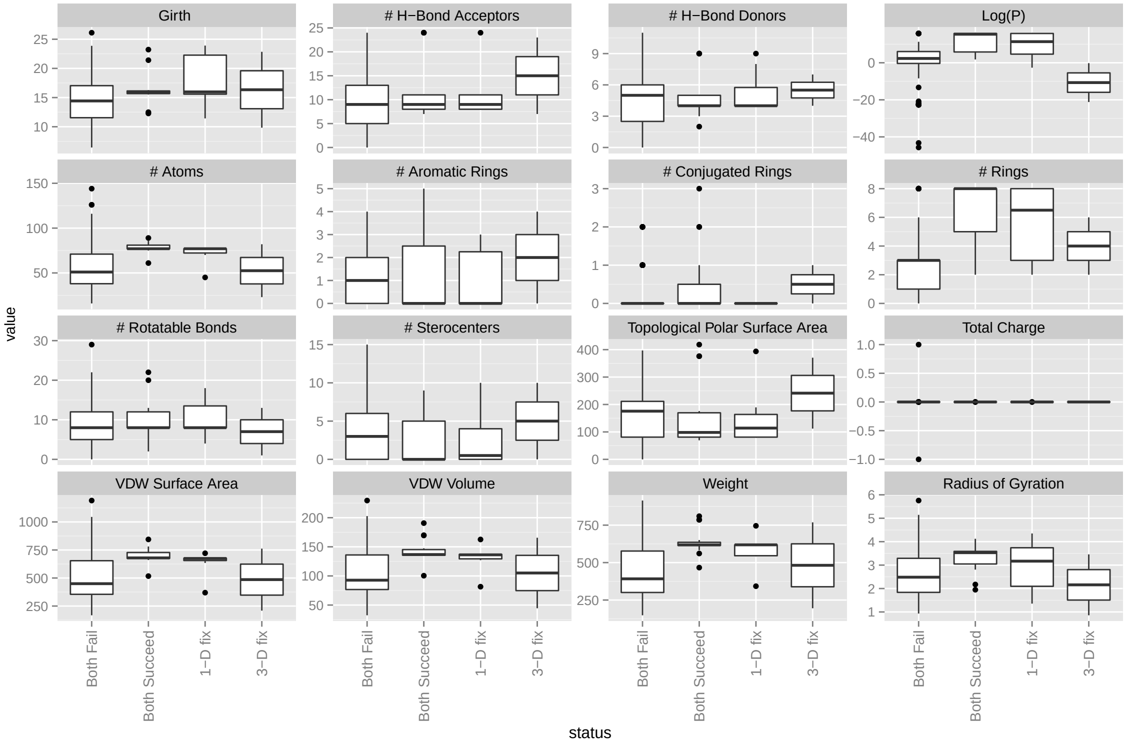


Figure C.21: Comparison of ligand property distributions for ligands docked with the TRANSFORM/1-D/MCM and TRANSFORM/3-D/MCM protocols.

energy of the bound complex, total Rosetta energy for the bound complex normalized by SASA, unbound Rosetta energy, unbound Rosetta energy normalized by SASA, hydrophobic SASA, total SASA, polar SASA, percentage of total energy accounted for by H-bonding, Rosetta attractive energy, Rosetta electrostatic energy, Rosetta pair energy, Rosetta repulsive energy, Rosetta solvation energy, Rosetta backbone-side-chain H-bond energy, Rosetta side-chain-side-chain H-bond energy, total protein residue count,interface residue count, Rosetta packing statistic, total pre-residue Rosetta energy, and S_c.

Figure C.22 compares the protein properties of proteins that succeeded and failed with the TRANSFORM/MCM and TRANSROT/MCM protocols. As the TRANSFORM/MCM and TRANSROT/MCM protocols had a large, statistically significant difference in terms of overall performance, this is the most likely case to see the impact of protein chemical property on success. While the distributions do overlap substantially, it appears that the proteins in which both protocols are able to dock successfully have slightly lower attractive energy scores and slightly lower backbone-sidechain hydrogen-bond energy scores.

Figure C.23 plots the differences in protein property distributions for success and failure cases of the TRANSFORM/MCM and TRANSFORM/1-D/MCM. Inspection of the distributions here indicate that in general, the binary scoring grid is more capable than the 1-D KBP grid of docking ligands into proteins with higher protein-ligand interface energies and lower attractive energies. However, as with the previous analysis, the distributions of the protein properties are generally overlapping, limiting the conclusions which we can draw.

Figure C.24 plots the differences in protein property distributions between the TRANSFORM/1-D/MCM and TRANSFORM/3-D/MCM methods. Here, more than in the previous two figures, we see overlap between nearly all the plotted distributions. However, we see that the proteins for which TRANSFORM/3-D/MCM had improved performance above TRANSFORM/1-D/MCM generally have a lower backbone-side-chain hydrogen-bond Energy and an increased number of unsaturated H-bonds relative to the overall distributions.

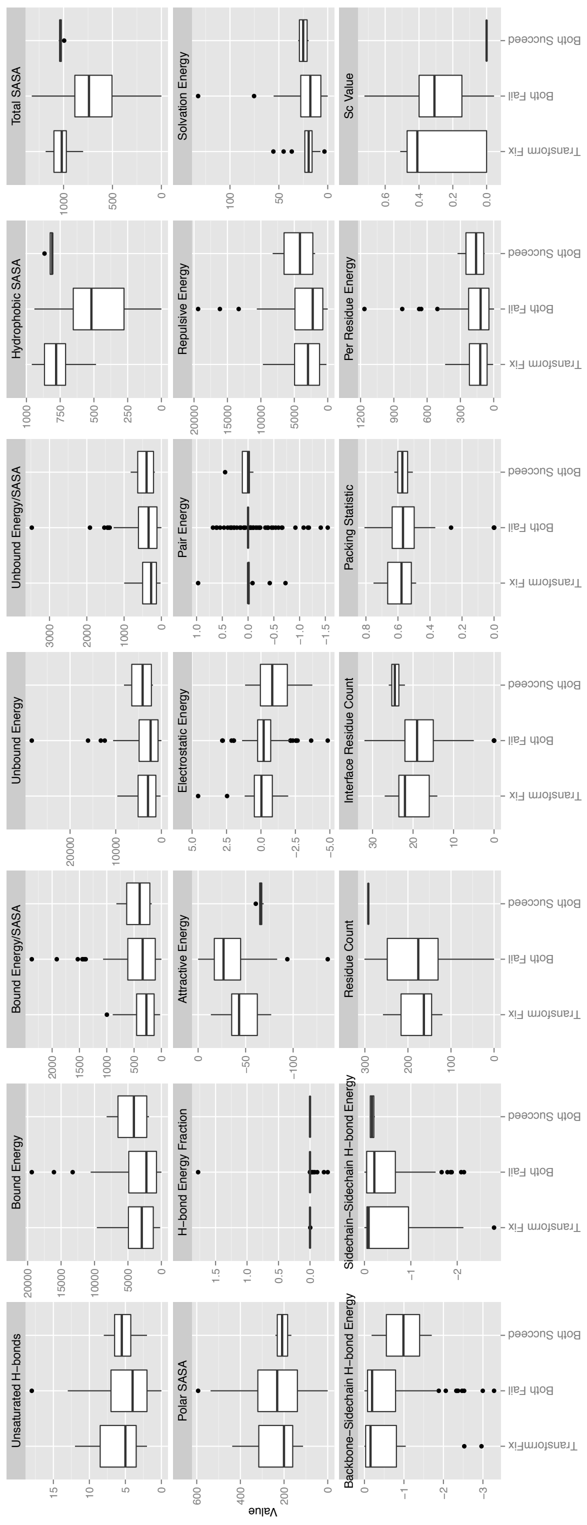


Figure C.22: Comparison of protein property distributions for protein-ligand pairs docked with the TRANSFORM/MCM and TRANSROT/MCM protocols. “Transform Fix” is the group of ligands that could not be docked with the TRANSROT/MCM protocol but were successfully docked by the TRANSFORM/MCM protocol.

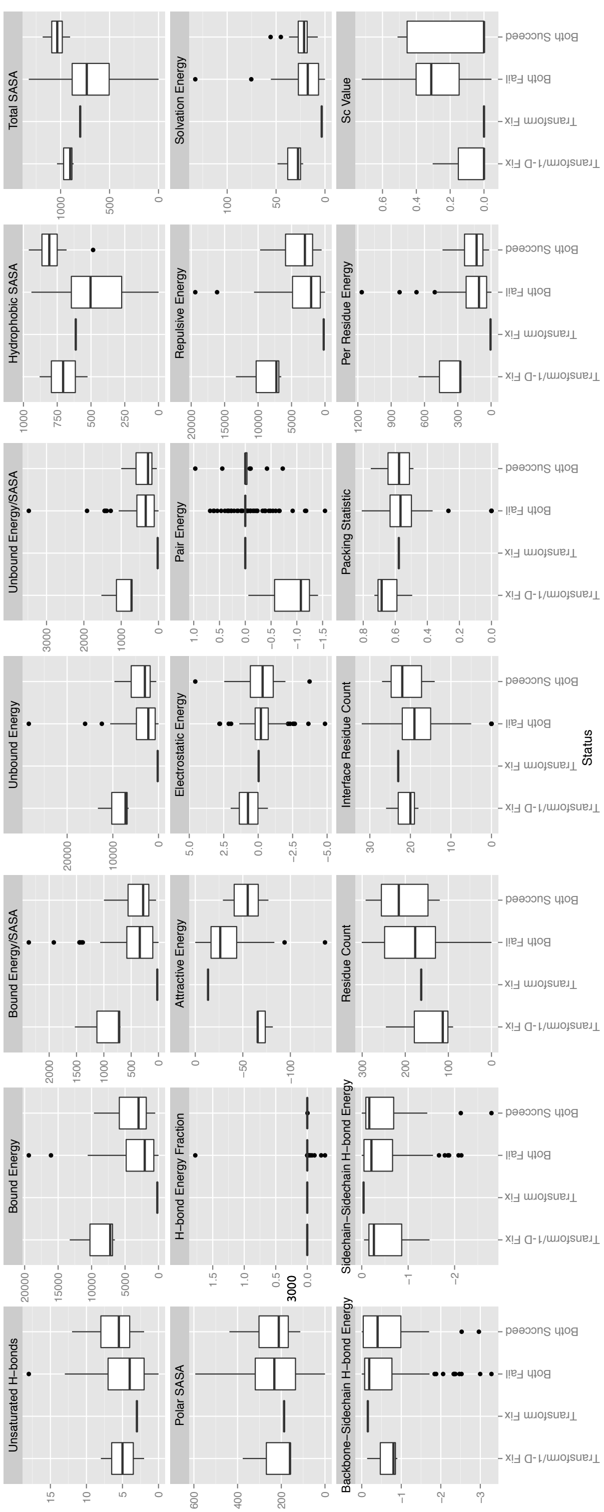


Figure C.23: Comparison of protein property distributions for protein-ligand pairs docked with the TRANSFORM/MCM and TRANSFORM/1-D/MCM protocols. “Transform Fix” is the group of ligands that could not be docked with the TRANSFORM/1-D/MCM protocol but were successfully docked by the TRANSFORM/MCM protocol. “Transform/1-D Fix” is the group of ligands that could not be docked with the TRANSFORM/MCM protocol but were successfully docked by the TRANSFORM/1-D/MCM protocol.

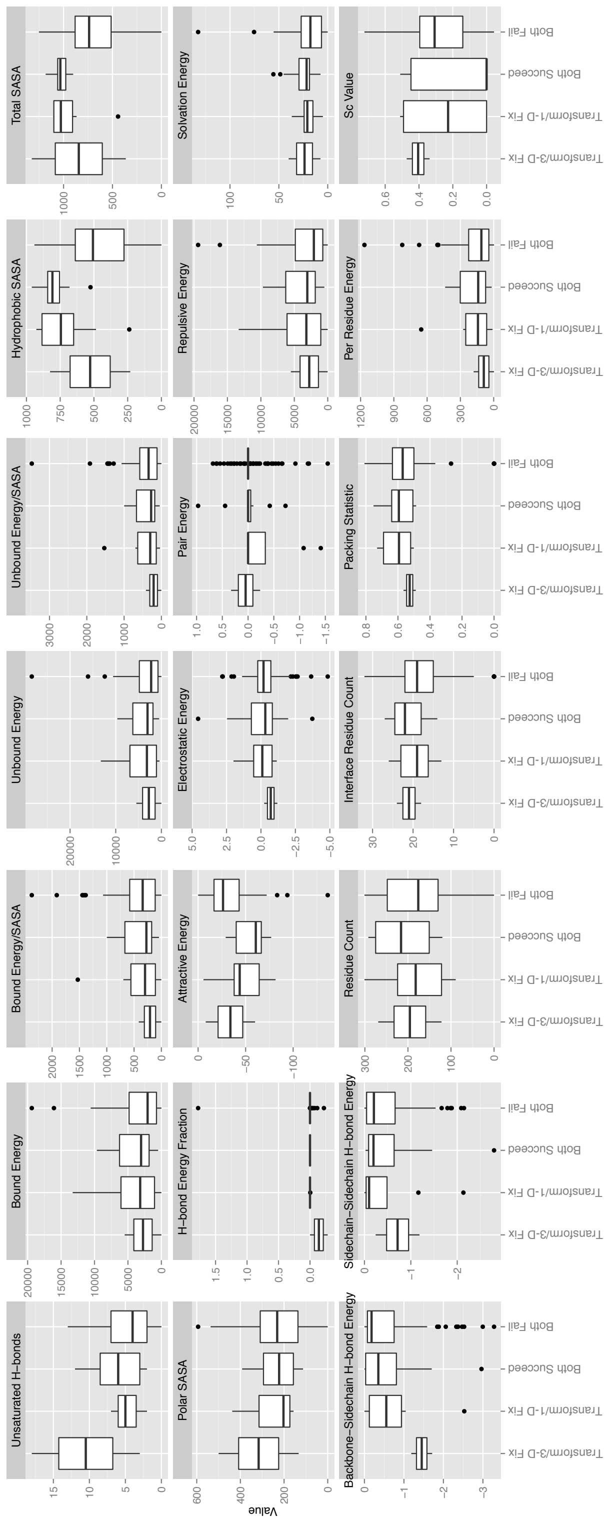


Figure C.24: Comparison of protein property distributions for protein-ligand pairs docked with the TRANSFORM/1-D/MCM and TRANSFORM/3-D/MCM protocols. “Transform/3-D Fix” is the group of ligands that could not be docked with the TRANSFORM/1-D/MCM protocol but were successfully docked by the TRANSFORM/3-D/MCM protocol. “Transform/1-D Fix” is the group of ligands that could not be docked with the TRANSFORM/3-D/MCM protocol but were successfully docked by the TRANSFORM/1-D/MCM protocol.

C.3.5 Future Directions

As described above, the newly implemented KBP-based scoring grids do not afford any significant performance improvement relative to the previously discussed binary scoring function. There are several possible explanations for this issue and a number of paths of further research and development which may lead to remedies.

The lack of correlation between protein structure accuracy and ligand docking success (Figure C.18) suggests that high accuracy protein models are necessary but not sufficient to correctly model protein-ligand interactions. Based on prior research, it is likely that protein-ligand dynamics (Baugh et al., 2010) and quantum chemical description of interactions (Weber et al., 2014) are critical to accurate and reliable modeling.

It is also likely that continued research efforts should be focused on the high resolution sampling and scoring components of RosettaLigand, rather than low resolution docking. Figure C.18 indicates that most of the protein-ligand pairs in the Q-Dock binding with comparative model RMSDs of < 5 Å could not be docked successfully by the TRANSFORM/MCM protocol. For this analysis, protein systems with comparative model RMSDs above 5.0 Å were excluded, to focus on RosettaLigand's performance in docking ligands into reasonably high quality models. Analysis of the entire pool of models generated by RosettaLigand indicates that a small percentage (3.3%) of the models had ligand RMSDs of < 2.0 Å relative to the crystal structure. Figure C.25 plots the distribution of the RMSDs of models produced for incorrectly docked protein-ligand systems by several of the studied docking protocols. As in previous analyses, a system is defined as incorrectly docked if the RMSD of the ligand is > 2.0 Å relative to the crystal structure. While each of the docking protocols was able to sample correct poses over the entire Q-Dock benchmarking set, none of the protocols were capable of successfully sampling each of the 148 proteins in the benchmark set. Specifically: the TRANSROT/MCM protocol sampled at least one pose at < 2.0 Å for 5/148 proteins, the TRANSFORM/MCM and TRANSFORM/1-D/MCM protocols successfully sampled at least one model for 20/148 proteins, and the TRANS-

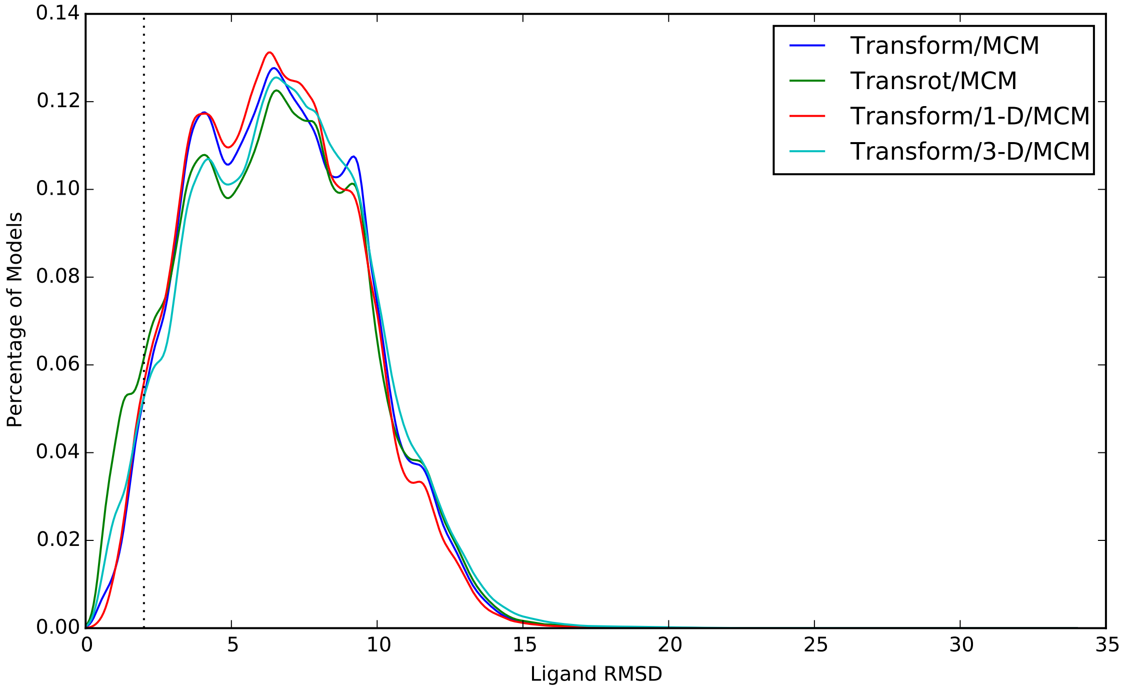


Figure C.25: The distribution of ligand RMSDs across all models generated by a selection of protocols for incorrectly docked protein-ligand pairs with a protein template RMSD of < 5 Å.

FORM/3-D/MCM protocol correctly sampled at least one model for 18/140 proteins. This analysis supports the conclusion that the RosettaLigand performance is currently limited by both sampling and scoring.

The lack of trends in protein and ligand property distributions discussed in sections C.3.3 and C.3.4 suggest that there is likely no single root cause for the inability to properly distinguish between ligand poses. Rather, it is likely that an overall improvement in high resolution energy function accuracy and information content is necessary. The work on development of an electron orbital-based KBP by Combs et al. (Combs, 2013), as well as improvements in the conventional RosettaLigand energy function by Leaver-Fay et al. (unpublished) may improve the performance of RosettaLigand and should be investigated. Additionally, re-scoring of RosettaLigand predictions with QM/MM based force fields may also yield beneficial results.

There is likely still room for improvement within the scope of KBP based modeling methods. The lack of improvement over the originally developed binary scoring function seen with the two new methods described here indicates that these methods are not providing significant new information to the initial placement algorithm. To build upon the current methods, a reasonable next step might be to collapse the 3-D KBP described in Section C.2.3 into 2 dimensions. KBPs are prone to artifacts in the data tables resulting from low counts in any particular bin. As the number of bins increases, the likelihood of some bins having unusably low counts, and therefore skewed energy values, increases. Typically, when developing a 1-D or 2-D KBP, we manually inspect the KBP table to look for unwanted artifacts. These artifacts typically manifest themselves as sharp discontinuities in the KBP, and can be corrected via smoothing. The difficulty with the 3-D KBP described here is two-fold. First, a 3-D KBP is extremely difficult to effectively visualize, second, it is not immediately apparent what the KBP should look like. These two factors make the manual inspection component of the KBP development process extraordinarily difficult. Additionally, the number of bins present in the KBP increases with the power of the number of dimensions. Thus, reducing the KBP from 3-D to 2-D will drastically reduce the total number of bins, increasing the number of samples per bin and hopefully reducing the number of artifacts. When developing the 2-D KBP, a decision must be made as to how to compute the angular component of the potential. Potentials computed using both the Query-Cβ-Cα angle and the Hα-Cα-Cγ-Query dihedral could be computed and benchmarked, and the best performing potential would be selected. The Query-Cβ distance is still likely the best choice for the distance potential.

The methods described and analyzed in this chapter provides a promising starting point for a wide range of further study. Qualitative and quantitative study of the RosettaLigand protocols evaluated suggest that the failure cases of RosettaLigand are a combination of limitations in both sampling and scoring. To address the sampling limitations, more sophisticated methods of sampling can be employed in both initial placement and refinement.

While the Monte Carlo methods used in the initial placement and refinement stages of RosettaLigand are relatively effective, the analysis performed in this chapter indicates that the currently implemented sampling methods are not able to adequately sample the binding site in all cases. The RosettaLigand initial placement system described in this dissertation has been designed to fully separate the implementation of the sampling algorithm and scoring function. As a result, it should be feasible to implement and evaluate a wide range of sampling functions, including genetic algorithms, Fast Fourier Transform, and Particle Swarm Optimization.

In addition to the evaluation of more sophisticated sampling methods, the bounds of the currently implemented sampling algorithms should be more thoroughly investigated. As the goal in this study was to assess the ability of RosettaLigand to rapidly dock ligands for screening applications, sampling was limited to a relatively small number of models (less than 1,000). In order to fully assess the capabilities of the existing sampling algorithm, the experiments should be repeated with tens or hundreds of thousands of models. While time consuming, these experiments may provide additional insight into the full abilities and limitations of the existing system, which can be used to guide future experiments.

In the analysis of individual success and failure techniques described in this chapter, there were numerous cases in which substantially different binding poses had relatively similar RosettaLigand scores after refinement. This suggests a complex energy landscape with numerous minima. Because the RosettaLigand energy function is largely knowledge based, it is difficult to directly assign the cause of any single scoring decision. However, with sufficient sampling, it should be possible to “map” the energy surface. Given a set of high resolution protein-ligand structures, a large amount of small perturbation sampling and scoring could be performed, and the impact of these small conformational changes statistically analyzed as a function of the protein and ligand conformations at the binding pocket. By performing this analysis, it may be possible to identify systematic failures in various components of the scoring function. If so, this analysis could be valuable in identifying specific areas of improvement in the RosettaLigand refinement energy function. A similar study was performed in the context of Rosetta protein design in 2013 (Leaver-Fay et al., 2013) and was valuable in providing an analytical basis for improvement of the energy function.

Appendix D

Protocol capture for Chapter II

D.1 Introduction

This chapter describes the weight optimization, benchmarking and analysis performed in the work detailed in Chapter II. Note that the protocol described here was originally performed using Rosetta SVN revision 39040. In the time since the work described in Chapter II was performed, the OptE application used here has been drastically rewritten. As a result, this procedure should not be expected to function correctly (or at all!) when using Rosetta revisions after 39040.

D.2 Protocol

D.2.1 Weight Optimization

D.2.1.1 Overview

This protocol performs a five way cross validation optimization of the neighbor vector score function using the rosetta optE_parallel application. In this setup, 20 rounds of optimization are performed, and the reference energies, fa_sol and neigh_vect scoring function are allowed to freely optimize. The weights are optimized both to maximize PSSM score and to maintain the overall native sequence composition.

D.2.1.2 Preparing input structures

Prior to running OptE, all input crystal structures should be cleaned and relaxed. To clean the input structures, remove all PDB lines other than ATOM records. Relaxation is performed using the Rosetta relax application, and the sequence relax protocol. This protocol can be executed as follows:

```
relax.default.linuxgccrelease \
-database minirosetta_database/ \
-l input.list -relax:sequence -ex1 -ex2 -ex1aro
```

Where input.list contains a list of paths to the input PDB files. Relaxed files using this method have been provided in `Optimization/input_files/input_pdbs/`

D.2.1.3 Generating PSSM files

PSSM files must additionally be generated for each PDB file prepared in Section D.2.1.2. To generate PSSM files, first use the provided script, `getFastaFromCoords.pl` to create a fasta file based on the relaxed crystal structure. `getFastaFromCoords.py` is run as follows:

```
getFastaFromCoords.pl -pdbfile input.pdb > input.pdb.fasta
```

The resulting fasta file will then be used as input to BLAST to create a PSSM file. As the BLAST webserver does not provide PSSM files as output in the proper format, the BLAST application will be used, and is executed as follows:

```
runblast input.pdb.fasta
```

The resulting file will produce, among other things, a PSSM file with the file extension .ascii. This .ascii file will be converted to the format required by Rosetta using the provided script `convertpssm.py`. The Rosetta expects that the PSSM information be provided as a text file, in which each line of the text file contains the one letter code of a native amino acid, followed by the percentage of observed mutations seen by blast, ordered in alphabetical order by one letter code, and separated by spaces. This file can be produced using the PSSM generated by blast by running the `convertpssm.py` script:

```
convertpssm.py -i input.pdb.ascii -o input.fasta.probs
```

Note that OptE requires that the file produced by convertpssm.py begin with with the name of the original pdb file, and end with the suffix .ascii.probs. Additionally, this file must be present in the same directory as the input PDB file. Thus, a PDB file called "input.pdb" should have a PSSM file of the same name titled input.fasta.probs. These files are provided in the `Optimization/input_files/input_pdbs` directory.

D.2.1.4 Running OptE

Three sets of optimization were performed using OptE: the optimization of the NV environment KBP, optimization of the reference energies only, and optimization of the reference energies of the final averaged NV environment KBP energy function. The command files for each optimization are designated by their suffix, and are located in the `Optimization/input_files` directory. "kbp" for files relating to the NV environment KBP optimization, "ref" for files relating to reference energy optimization, and "avg" for files relating to optimization of the averaged energy function.

In this case, a template Portable Batch System (PBS) file was used, and variables were passed in to this PBS file to start each section of the five way cross validation. The template file is located in `input_files/optimization_x.pbs`, and the submission commands are located in `input_files/submit_commands_x.txt`.

See `input_files/flags_x.txt` for the options and comments describing what these options do for each of the 3 optimization experiments performed.

D.2.2 Weight validation and analysis

D.2.2.1 Benchmarking of optimized weights

The weight sets optimized in Section D.2.1 were benchmarked using the Rosetta fixbb application. Fixbb conducts fixed backbone design over the entire protein, using the specified weight set. An example command line and flags files can be found in the `Benchmark/input_files` directory. Fixed backbone design was performed on all proteins in both the 100 and 42 protein benchmark sets described in Chapter D.2.1.2.

D.2.2.2 Analysis of benchmarking data

After the benchmarking designs were performed, the computation of sequence recovery and PSSM recovery was carried out using the script `design_benchmark_protocol.py`, which is provided in the `Benchmark/input_files` directory. This script takes as input a list of paths to native protein structures, and a list of paths to designed protein structures, and outputs a set of Comma Separated Value (CSV) files containing the statistics reported in Chapter II. The script should be run as:

```
design_benchmark_protocol.py --prefix prefix_file \
native_list.txt designed_list.txt
```

Appendix E

Protocol capture for Chapter III

E.1 Introduction

This chapter describes the details of the protocol which was described in Chapter III.

E.2 Protocol

E.2.1 Conformer Generation

Conformers can be generated with a number of tools, including MOE and OMEGA. In this case, the Conformer Generation tool included as part of the BCL suite was used. The following command was used to generate conformers:

```
bcl.exe molecule:ConformerGeneration -conformers \
pdb_refinedsupplemented_lib.sdf.bz2 -ensemble \
rosetta_inputs/ligands/all_ligands.sdf \
-conformation_comparer DihedralBins \
-temperature 1.0 -max_iterations 1000 \
-top_models 100 -bin_size 30.0
-scheduler PThread 8 \
-add_h -conformers_single_file conformers
```

You can use any conformer generation tool you have available to you for this step. Your generated conformers should be output to a single SDF file. Every conformer must have 3D coordinates and hydrogens added. Conformers of the same ligand should have the same name in the SDF file. For convenience, an example conformer file is provided at `rosetta_inputs/ligands/all_ligands.sdf`.

E.2.2 Params file generation

Params files contain the parameterization information for a ligand. Every ligand or Residue in a protein structure input into Rosetta must have a corresponding params file. Rosetta is distributed with a script called `molfile_to_params.py` which generates these files. However, this script is generally cumbersome for the generation of more than a small handful of ligands. The protocol below is designed for the preparation of large numbers of ligands.

All the scripts needed for this process are in the tools directory in the Rosetta distribution. Each of the scripts below would normally be preceded by Rosetta/tools/hts_tools, but this directory prefix has been omitted for brevity.

1. *Split ligand files*

   The conformers for all ligands are initially stored in a single SDF file, but `molfile_to_params.py` expects one SDF file per ligand. `sdf_split_organize.py` accomplishes this task. It takes as input a single SDF file, and will split that file into multiple files, each file containing all the conformers for one ligand. Different ligands must have different names in the SDF records, and all conformers for one ligand must have the same name. Output filenames are based on the SHA1 hash of the input filename, and are placed in a directory hashed structure. Thus, a ligand with the name "Written by BCL::WriteToMDL,CHEMBL29197" will be placed in the path
   /41/412d1d751ff3d83acf0734a2c870faaa77c28c6c.mol.

   The script will also output a list file in the following format:

   ```
   ligand_id,filename
   string,string
   ligand_1,path/to/ligand1
   ligand_2,path/to/ligand2
   ```

   The list file is a mapping of protein names to SDF file paths.

   Many filesystems perform poorly if large numbers of files are stored in the same directory. The hashed directory structure is a method for splitting the generated ligand files across 256 roughly evenly sized subdirectories, improving filesystem performance.

The script is run as follows:

```
sdf_split_organize.py \
rosetta_inputs/ligands/conformers.sdf \
split_conformers/ ligand_names.csv
```

Be sure the split_conformers/ directory exists before running the script. Examples of the output of this script are in `example_outputs/ligand_prep/`

2. *Create Projet Database*

   The ligand preparation pipeline uses an SQLite3 database for organization during the pipeline. The database keeps track of ligand metadata and the locations of ligand files. The project database is created using the following command:

   ```
   setup_screening_project.py ligand_names.csv ligand_db.db3
   ```

   An example of the project database is in example_outputs/ligand_prep

3. *Append binding information to project database*

   The next step is to create a binding data file. The binding data file should be in the following format:

   ```
   ligand_id,tag,value
   string,string,float
   ligand_1,foo,1.5
   ligand_2,bar,-3.7
   ```

   The columns are defined as follows:

- **ligand_id** — ligand_id is the name of the ligand, which must match the ligand_id in the file_list.csv file created by sdf_split_organize.py.
- **tag** — The name of the protein the ligand should be docked into. If a ligand should be docked into multiple proteins, it should have multiple entries in the binding data file. Note that this protocol makes a distinction between protein name, and file name. If you have 4 protein files: foo_0001.pdb, foo_0002.pdb, bar_0001.pdb, and bar_0002.pdb, then you have two proteins with the names foo and bar. The scripts expect that the protein PDB files begin with the protein name.
- **value** — The activity of the ligand. If you are doing a benchmarking study and know the activity of your ligand, you should enter it here. If you are not doing a benchmarking study, or if ligand activity is not relevant to your study, value can be set to 1.0 (or anything else). This field is currently only used in a few specific Rosetta protocols that are in the experimental stages, and is typically ignored, so it is safe to set arbitrarily in almost every case.

An example input file is provided. you can insert it into the project database with the following command:

```
add_activity_tags_to_database.py ligand_db.db3 \
rosetta_inputs/ligand_activities.csv
```

4. *Generate Params Files*

The next step is to generate params files. make_params.py is a script which wraps around molfile_to_params.py and generates params files in an automated fashion. Params files will be given random names that do not conflict with existing Rosetta residue names (no ligands will be named ALA, for example). This script routinely results in warnings from molfile_to_params.py, these warnings are not cause for concern. Occasionally, molfile_to_params.py is unable to properly process an SDF file, if this happens, the ligand will be skipped. In order to run make_params.py you need to specify the path to a copy of molfile_to_params.py, as well as the path to the Rosetta database.

make_params.py should be run like this:

```
make_params.py -j 2 --database Rosetta/main/database \
--path_to_params \
Rosetta/main/source/src/python\
/apps/public/molfile_to_params.py \
ligand_db.db3 params/
```

In the command line above, the -j option indicates the number of CPU cores which should be used when generating params files. If you are using a multiple core machine, setting -j equal to the number of available CPU cores. Be sure that the params/ directory exists before running the script.

The script will create a directory params/ containing all params files, PDB files and conformer files.

An example of the output params/ directory is found in
example_outputs/ligand_prep

5. *Create job files*

Because of the memory usage limitations of Rosetta, it is necessary to split the screen up into multiple jobs. The optimal size of each job will depend on the following factors:

- The amount of memory available per CPU
- The number of CPUs being used

- The number of atoms in each ligand
- The number of conformers of each ligand
- The number of protein residues involved in the binding site.

Because of the number of factors that affect RosettaLigand memory usage, it is usually necessary to determine the optimal job size manually. Jobs should be small enough to fit into available memory.

To make this process easier, the `make_evenly_grouped_jobs.py` script will attempt to group your protein-ligand docking problem into a set of jobs that are sized as evenly possible. The script is run like this:

```
make_evenly_grouped_jobs.py --create_native_commands \
rosetta_inputs/proteins --n_chunks 1 \
--max_per_job 1000 \
params rosetta_inputs/proteins job
```

If the script was run as written above, it would use param files from the directory `param_dir/`, and structure files from the directory `structure_dir/`. It would attempt to split the available protein-ligand docking jobs into 10 evenly grouped job files (`-n_chunks`). The script will attempt to keep all the docking jobs involving one protein system in one job file. However, if the number of jobs in a group exceeds 1000, the jobs involving that protein system will be split across multiple files (`-max_per_job`). The script will output the 10 job files with the given prefix, so in the command above, you would get files with names like "output_prefix_01.js". The script will output to the screen the total number of jobs in each file. All the numbers should be relatively similar. If a job file at the beginning of the list is much larger than the others, it is a sign that you should reduce the value passed to `-max_per_job`. If the sizes of all jobs are larger than you want, increase `-n_chunks`.

Additionally, the script will take the default ligand positions from the ligand PDB files, and the protein files from the `rosetta_inputs/proteins` directory, and designate these as the "native" pose of the protein-ligand complex. This feature will allow Rosetta to compute ligand RMSDs automatically, and was used in the benchmarking studies described in the manuscript.

An example job file produced using this script is found in `example_outputs/ligand_prep`

E.2.3 Docking

After following the procedure above to prepare your ligands, you are ready to dock the ligands. The screening job file produced in the previous step contains the paths to the input proteins and ligands and the paths to the necessary params files. In this example, the ligand pdbs are already positioned in the ligand binding site.

RosettaLigand protocols are built in the RosettaScripts framework, a modular architecture for creating RosettaLigand protocols. The `rosetta_inputs/xml` directory contains all of the rosetta protocols were tested in the manuscript, and any of these XML files can be used with the docking commands described below. See the comments in the XML files for details on the usage and operation of the scripts.

The Rosetta ligand docking command should be run as follows:

```
rosetta_scripts.default.linuxgccrelease \
@rosetta_inputs/flags.txt \
-in:file:screening_job_file rosetta_inputs/job_01.js \
-parser:protocol rosetta_inputs/tr_repack.xml \
-out:file:silent results.out
```

`rosetta_inputs/flags.txt` contains flags that are always the same regardless of the input file.

This command will dock every protein-ligand binding pair and place the output in the specified silent file. In the benchmarking case described in Chapter III, 2000 models were made for each protein-ligand binding pair. However, in a practical application 200 models would be appropriate.

E.3 Analysis

E.3.1 Practical analysis

If this protocol is being used for an application project in which the correct ligand binding position is not known, the lowest scoring model for each protein-ligand binding pair should be selected. From that point, we recommend filtering by protein-ligand interface score (interface_delta_X), as well as the packstat score (Sheffler and Baker, 2009) which can be computed through the InterfaceAnalyzer mover. The cutoffs for these filtering steps should depend on the range of scores present, and the number of compounds it is possible to test.

After filtering, the selected compounds should be visually inspected. If a crystal structure exists with a known binding pose, the predicted binding poses of the unknown compounds should be compared. Additionally, the overall binding poses of the filtered compounds should be inspected to assess whether or not they make chemical sense. While this is a qualitative process, human intuition has proven a valuable aid in the drug design process. (Voet et al., 2014)

E.3.2 Benchmarking analysis

Statistical analysis of the benchmarking study provided in this paper was performed using Python. analysis.ipynb is an iPython Notebook (http://ipython.org/notebook.html) containing the code necessary to reproduce these figures, as well as comments and description of that code. See the iPython documentation for installation and usage instructions.

Appendix F

Protocol capture for Chapter IV

F.1 Introduction

This chapter describes the details of the protocol which was described in Chapter IV.

F.2 Protocol

F.2.1 Training data preparation

The PDBBind refined dataset was obtained from http://www.pdbbind.org.cn/. for each protein in the refined dataset, 3 protein crystal structure files are provided. The “complex” file contains the protein in complex with the ligand, the “pocket” file contains only protein atoms within 10 Å of the binding pocket, and the “protein” file contains the entire protein, but no ligand. For the purposes of docking in Rosetta, the “protein” files will be used for our protein input.

F.2.1.1 protein structure preparation

For the purposes of this protocol, the only protein structure preparation required is the addition of hydrogen atoms, which can be performed using the Rosetta score_jd2 application. A list of the protein structure PDBs is generated with the following command:

```
ls -1 v2013-refined/*/*protein.pdb > input_files.txt
```

Hydrogens can then be added with:

```
score_jd2.default.linuxgccrelease -s input_files.txt -out:pdb
```

F.2.1.2 ligand structure preparation

PDBBind provides SDF files for each input ligand. The provided SDF files are in the crystallographic conformation, and already have 3D coordinates and hydrogens present. Params files are produced by concatenating the SDF files using the command:

```
cat v2013-refined/*/*ligand.sdf > crystal_ligands.sdf
```

The resulting crystal_ligands.sdf file is then used to prepare conformers and param files. Conformers are generated using the unpublished BCL::ConformerGeneration command:

```
bcl.exe molecule:ConformerGeneration -conformers \
pdb_refinedsupplemented_lib.sdf.bz2 -ensemble \
crystal_ligands.sdf -conformation_comparer \
DihedralBins -temperature 1.0 -max_iterations 1000 \
-top_models 100 -bin_size 30.0 -scheduler PThread 8 \
-add_h -conformers_single_file conformers
```

After conformers are generated, params files and rosetta screening job files are produced using the protocol described in Chapter A. This protocol will generate screening job files for the active ligands. To generate files used for cross-docking, the `make_evenly_grouped_jobs.py` script should be re-run with the addition of the flag `-inactive_cross_dock`, which will result in the creation of a second set of screening job files which will dock every ligand into every protein except the native protein.

F.2.2 Training data docking

F.2.2.1 Docking script

The RosettaLigand docking protocol used in this protocol is reported in detail in Chapter III. As RosettaLigand is implemented through the RosettaScripts system, the following XML script implements the protocol:

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <ligand_soft_rep weights="ligand_soft_rep">
      <Reweight scoretype="fa_elec" weight="0.42"/>
      <Reweight scoretype="hbond_bb_sc" weight="1.3"/>
      <Reweight scoretype="hbond_sc" weight="1.3"/>
      <Reweight scoretype="rama" weight="0.2"/>
    </ligand_soft_rep>

    <hard_rep weights=ligand>
      <Reweight scoretype="fa_intra_rep" weight="0.004"/>
      <Reweight scoretype="fa_elec" weight="0.42"/>
      <Reweight scoretype="hbond_bb_sc" weight="1.3"/>
      <Reweight scoretype="hbond_sc" weight="1.3"/>
      <Reweight scoretype="rama" weight="0.2"/>
    </hard_rep>
  </SCOREFXNS>
  <LIGAND_AREAS>
    <docking_sidechain chain="X" cutoff="6.0"
      add_nbr_radius="true" all_atom_mode="true"
      minimize_ligand="10"/>
    <final_sidechain chain="X" cutoff="6.0"
      add_nbr_radius="true" all_atom_mode="true"/>
    <final_backbone chain="X" cutoff="7.0"
      add_nbr_radius="false" all_atom_mode="true"
      Calpha_restraints="0.3"/>
  </LIGAND_AREAS>

  <INTERFACE_BUILDERS>
    <side_chain_for_docking
      ligand_areas="docking_sidechain"/>
    <side_chain_for_final
      ligand_areas="final_sidechain"/>
    <backbone ligand_areas="final_backbone"
      extension_window="3"/>
  </INTERFACE_BUILDERS>

  <MOVEMAP_BUILDERS>
    <docking sc_interface="side_chain_for_docking"
      minimize_water="true"/>
    <final sc_interface="side_chain_for_final"
      bb_interface="backbone" minimize_water="true"/>
  </MOVEMAP_BUILDERS>

  <SCORINGGRIDS ligand_chain="X" width="15">
    <vdw grid_type="ClassicGrid" weight="1.0"/>
  </SCORINGGRIDS>

  <MOVERS>
    <Transform name="transform" chain="X"
      box_size="5.0" move_distance="0.1"
      angle="5" cycles="500" repeats="1"
      temperature="5" initial_perturb="5.0"/>
    <HighResDocker name="high_res_docker"
      cycles="1" repack_every_Nth="1"
      scorefxn="ligand_soft_rep"
      movemap_builder="docking"/>
    <FinalMinimizer name="final"
      scorefxn="hard_rep"
      movemap_builder="final"/>
    <InterfaceScoreCalculator name="add_scores"
      chains="X" scorefxn="hard_rep"
      compute_grid_scores="0"/>
    <AddJobPairData name="system_name"
      key="system_name"
      value_type="string"
      value_from_ligand_chain="X" />
    <AddJobPairData name="log_ki"
      key="log_ki" value_type="real"
      value_from_ligand_chain="X" />

    <ParsedProtocol name="low_res_dock">
      <Add mover_name="transform"/>
    </ParsedProtocol>

    <ParsedProtocol name="high_res_dock">
      <Add mover_name="high_res_docker"/>
      <Add mover_name="final"/>
    </ParsedProtocol>

    <ParsedProtocol name="reporting">
      <Add mover_name="add_scores"/>
      <Add mover_name="system_name"/>
      <Add mover_name="log_ki"/>
    </ParsedProtocol>
  </MOVERS>

  <PROTOCOLS>
    <Add mover_name="low_res_dock"/>
    <Add mover_name="high_res_dock"/>
```

```
    <Add mover_name="reporting"/>
  </PROTOCOLS>

</ROSETTASCRIPTS>
```

The XML script above is used for docking the native ligands into the associated proteins. The script for cross-docking is nearly identical, with this mover:

```
        <AddJobPairData name="log_ki"
            key="log_ki" value_type="real"
            value_from_ligand_chain="X" />
```

Replaced by this mover:

```
        <AddJobPairData name="log_ki"
            key="log_ki" value_type="real"
            value="0.0" />
```

This change will cause the stored log(K_i) value for each ligand to be 0.0 rather than the experimental value stored in the params files for the native ligands.

F.2.2.2 Docking command

The full cross-docked training dataset requires the generation of an extremely large number of models. For each protein-ligand complex, 200 models will be generated. Since the training dataset contains 507 proteins and 507 ligands, a total of 50,000,000 models must be calculated, the storage of which would require an unreasonable amount of disk space. Because only the lowest scoring model for each protein-ligand complex is required, the structures will be stored in a MySQL database, and a database filter will be used to ensure that only the lowest scoring models are stored.

In addition to the normal command line options used in ligand docking, an additional set of MySQL related flags are required:

```
-inout
 -dbms
  -mode mysql
  -host <host>
  -port <port>
  -user <username>
  -password <password>
```

Here, `<host>`, `<port>`, `<username>`, and `<password>` should be replaced with the address of MySQL server, the port it runs on, and a valid mysql username and password. The following flags control the ligand docking process itself:

```
-packing:ignore_ligand_chi true
-ex1
-ex2
-qsar
 -max_grid_cache_size 1
-restore_pre_talaris_2013_behavior true
-nstruct 200
-out
 -use_database
 -database_filter TopCountOfEachInput interface_delta_X 1
-inout
 -dbms
 -use_compact_residue_schema
 -database_name <db_name>
```

Here, `<db_name>` should be replaced with the name of an existing schema in the MySQL server. The `-database_filter` option instructs Rosetta to only output models that are better than any existing model for that protein-ligand pair to the database server.

The RosettaLigand processes can be executed as follows:

```
rosetta_scripts.mpimysql.linuxgccrelease  @flags.txt \
-in:file:screening_job_file <job_file> -parser:protocol <xml>
```

Where `<job_file>` is the path to a screening job file, and `<xml>` is the XML script used for docking.

F.2.3 Training data descriptor generation

Once the ligand docking process is complete, descriptors and SDF files for each of the generated ligand poses need to be produced.

F.2.3.1 Rosetta descriptors

The Rosetta descriptors are generated using RosettaScripts. Specifically, the RDF fingerprint functions are generated using the ComputeLigandRDF mover, the interface score descriptors are generated with the InterfaceScoreCalculator mover, and the interface quality descriptors are generated with the InterfaceAnalyzerMover mover. After the features are computed, they are output along with an SDF file containing the ligand poses using the WriteLigandMolFile mover. WriteLigandMolFile will produce once file per CPU core that RosettaScripts is run on. The following XML file is used for descriptor computation:

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <ligand_soft_rep weights="ligand_soft_rep">
      <Reweight scoretype="fa_elec" weight="0.42"/>
      <Reweight scoretype="hbond_bb_sc" weight="1.3"/>
      <Reweight scoretype="hbond_sc" weight="1.3"/>
      <Reweight scoretype="rama" weight="0.2"/>
    </ligand_soft_rep>

    <hard_rep weights="ligand">
      <Reweight scoretype="fa_intra_rep" weight="0.004"/>
      <Reweight scoretype="fa_elec" weight="0.42"/>
      <Reweight scoretype="hbond_bb_sc" weight="1.3"/>
      <Reweight scoretype="hbond_sc" weight="1.3"/>
      <Reweight scoretype="rama" weight="0.2"/>
    </hard_rep>

  </SCOREFXNS>
  <LIGAND_AREAS>
    <docking_sidechain chain="X" cutoff="6.0"
      add_nbr_radius="true" all_atom_mode="true"
      minimize_ligand="10"/>
    <final_sidechain chain="X" cutoff="6.0"
      add_nbr_radius="true" all_atom_mode="true"/>
    <final_backbone chain="X" cutoff="7.0"
      add_nbr_radius="false" all_atom_mode="true"
      Calpha_restraints="0.3"/>
  </LIGAND_AREAS>

  <INTERFACE_BUILDERS>
    <side_chain_for_docking
      ligand_areas="docking_sidechain"/>
    <side_chain_for_final
      ligand_areas="final_sidechain"/>
    <backbone ligand_areas="final_backbone"
      extension_window="3"/>
  </INTERFACE_BUILDERS>

  <MOVEMAP_BUILDERS>
    <docking sc_interface="side_chain_for_docking"
      minimize_water="true"/>
    <final sc_interface="side_chain_for_final"
      bb_interface="backbone" minimize_water="true"/>
  </MOVEMAP_BUILDERS>

  <MOVERS>
    <ComputeLigandRDF name="rdf_compute_pocket"
      ligand_chain="X" mode="pocket" range="6"
      bin_count="24">
      <RDF name="RDFEtableFunction"
        scorefxn="hard_rep"/>
      <RDF name="RDFElecFunction"
        scorefxn="hard_rep"/>
      <RDF name="RDFChargeFunction"
        sign_mode="ligand_plus" />
      <RDF name="RDFChargeFunction"
        sign_mode="ligand_minus" />
      <RDF name="RDFChargeFunction"
        sign_mode="same_sign" />
      <RDF name="RDFHbondFunction"
        sign_mode="ligand_acceptor"/>
      <RDF name="RDFHbondFunction"
        sign_mode="ligand_donor"/>
      <RDF name="RDFBinaryHbondFunction"
        sign_mode="ligand_acceptor"/>
      <RDF name="RDFBinaryHbondFunction"
        sign_mode="ligand_donor"/>
      <RDF name="RDFBinaryHbondFunction"
        sign_mode="matching_pair"/>

    </ComputeLigandRDF>
    <ComputeLigandRDF name="rdf_compute_interface"
      ligand_chain="X" mode="interface" range="6"
      bin_count="24">
      <RDF name="RDFEtableFunction"
        scorefxn="hard_rep"/>
      <RDF name="RDFElecFunction"
        scorefxn="hard_rep"/>
      <RDF name="RDFChargeFunction"
        sign_mode="ligand_plus" />
      <RDF name="RDFChargeFunction"
        sign_mode="ligand_minus" />
      <RDF name="RDFChargeFunction"
        sign_mode="same_sign" />
      <RDF name="RDFHbondFunction"
        sign_mode="ligand_acceptor"/>
      <RDF name="RDFHbondFunction"
        sign_mode="ligand_donor"/>
      <RDF name="RDFBinaryHbondFunction"
        sign_mode="ligand_acceptor"/>
      <RDF name="RDFBinaryHbondFunction"
        sign_mode="ligand_donor"/>
      <RDF name="RDFBinaryHbondFunction"
        sign_mode="matching_pair"/>
    </ComputeLigandRDF>

    <InterfaceAnalyzerMover name="interface_analyzer"
      scorefxn="hard_rep" packstat="true"
      pack_separated="true" ligandchain="X"/>
    <InterfaceScoreCalculator name="add_scores"
      chains="X" scorefxn="hard_rep"/>

    <AddJobPairData name="system_name" key="system_name"
      value_type="string" value_from_ligand_chain="X" />
    <AddJobPairData name="log_ki" key="log_ki"
      value_type="real" value_from_ligand_chain="X" />

    <WriteLigandMolFile name="write_ligand" chain="X"
      directory="output_ligands" prefix="%%PREFIX%%"/>

    <ParsedProtocol name="reporting">
      <Add mover_name="rdf_compute_pocket"/>
      <Add mover_name="rdf_compute_interface"/>
```

```
      <Add mover_name="interface_analyzer"/>
      <Add mover_name="add_scores"/>
      <Add mover_name="system_name"/>
      <Add mover_name="log_ki"/>
    </ParsedProtocol>
  </MOVERS>

  <PROTOCOLS>
    <Add mover_name="reporting"/>
    <Add mover_name="write_ligand"/>
  </PROTOCOLS>

</ROSETTASCRIPTS>
```

This script will be applied to all of the structures produced by the docking step described in Section F.2.2.2. The command line used to run the script is as follows:

```
rosetta_scripts.mpimysql.linuxgccrelease @flags.txt \
-out:path:pdb <pdb_dir> -inout:dbms:database_name <db_name> \
-in:use_database -parser:protocol <xml> \
-script_vars PREFIX=<prefix> \
-in:file:extra_res_batch_path <params> -out:pdb_gz \
-restore_pre_talaris_2013_behavior true \
-packing:ignore_ligand_chi true \
-inout:dbms:use_compact_residue_schema \
-inout:dbms:retry_failed_reads true
```

Where flags.txt contains the database authentication flags described in Section F.2.2.2. `<prefix>` should be replaced with the desired prefix for the output SDF files.

F.2.3.2 BCL descriptors

Once the descriptors and SDF files have been generated using Rosetta, a BCL binary dataset can be constructed. This dataset files contain all the descriptor information for each ligand pose used in training. Rosetta derived descriptors are read out of the miscellaneous properties of the SDF files output by Rosetta, while the ligand descriptors are produced directly by the BCL. The features which will be used for the input and output of the network are described using object files. The output object file is very simple, as it contains only the log(K_i) value stored in the SDF files output by Rosetta:

```
Combine(
  MiscProperty(log_ki,values per molecule=1)
)
```

The input object contains all the features that may be used as input to the neural networks:

```
Combine(
  MiscProperty(solv_interface_rdf,values per molecule=24),
  MiscProperty(solv_pocket_rdf,values per molecule=24),
  MiscProperty(rep_interface_rdf,values per molecule=24),
  MiscProperty(rep_pocket_rdf,values per molecule=24),
  MiscProperty(hbond_acceptor_interface_rdf,
    values per molecule=24),
  MiscProperty(hbond_acceptor_pocket_rdf,
    values per molecule=24),
  MiscProperty(hbond_binary_acceptor_interface_rdf,
    values per molecule=24),
  MiscProperty(hbond_binary_acceptor_pocket_rdf,
    values per molecule=24),
  MiscProperty(hbond_binary_donor_interface_rdf,
    values per molecule=24),
  MiscProperty(hbond_binary_donor_pocket_rdf,
    values per molecule=24),
  MiscProperty(hbond_donor_interface_rdf,
    values per molecule=24),
  MiscProperty(hbond_donor_pocket_rdf,
    values per molecule=24),
  MiscProperty(hbond_matching_pair_interface_rdf,
    values per molecule=24),
  MiscProperty(hbond_matching_pair_pocket_rdf,
    values per molecule=24),
  MiscProperty(elec_interface_rdf,
    values per molecule=24),
  MiscProperty(elec_pocket_rdf,
    values per molecule=24),
  MiscProperty(charge_minus_interface_rdf,
    values per molecule=24),
  MiscProperty(charge_minus_pocket_rdf,
    values per molecule=24),
  MiscProperty(charge_plus_interface_rdf,
    values per molecule=24),
  MiscProperty(charge_plus_pocket_rdf,
    values per molecule=24),
  MiscProperty(charge_unsigned_interface_rdf,
    values per molecule=24),
  MiscProperty(charge_unsigned_pocket_rdf,
    values per molecule=24),
  MiscProperty(dSASA_hphobic,values per molecule=1),
  MiscProperty(dSASA_int,values per molecule=1),
  MiscProperty(dSASA_polar,values per molecule=1),
  MiscProperty(delta_unsatHbonds,values per molecule=1),
  MiscProperty(hbond_E_fraction,values per molecule=1),
  MiscProperty(hbond_lr_bb,values per molecule=1),
  MiscProperty(hbond_sc,values per molecule=1),
  MiscProperty(hbond_sr_bb,values per molecule=1),
  MiscProperty(if_X_fa_atr,values per molecule=1),
  MiscProperty(if_X_fa_elec,values per molecule=1),
  MiscProperty(if_X_fa_intra_rep,values per molecule=1),
  MiscProperty(if_X_fa_pair,values per molecule=1),
  MiscProperty(if_X_fa_rep,values per molecule=1),
  MiscProperty(if_X_fa_sol,values per molecule=1),
  MiscProperty(if_X_hbond_bb_sc,values per molecule=1),
  MiscProperty(if_X_hbond_lr_bb,values per molecule=1),
  MiscProperty(if_X_hbond_sc,values per molecule=1),
  MiscProperty(if_X_hbond_sr_bb,values per molecule=1),
  MiscProperty(interface_delta_X,values per molecule=1),
  MiscProperty(nres_int,values per molecule=1),
  MiscProperty(packstat,values per molecule=1),
  Divide(
    lhs=MiscProperty(total_score,values per molecule=1),
    rhs=MiscProperty(nres_all,values per molecule=1)
  ),
  Weight,
  HbondDonor,
  HbondAcceptor,
  LogP,
  TotalCharge,
  NRotBond,
  NAromaticRings,
  NRings,
  TopologicalPolarSurfaceArea,
  Girth
)
```

The input and output object files and the SDF files produced by Rosetta are provided to the BCL descriptor::GenerateDataset application, which produces the binary dataset file needed for neural network training:

```
bcl.exe descriptor:GenerateDataset -source \
'Randomize(SdfFile(filename="output_ligands.sdf"))' \
-feature_labels input.obj \
-result_labels output.obj \
-output dataset.bin
```

This command will produce the file dataset.bin containing all necessary descriptor data.

F.2.4 Neural network training

The BCL is also used for neural network training. The DVD attached to the thesis contains the specific configuration file used for training (config.ini). The following network and training architecture was used:

```
NeuralNetwork( transfer function = Sigmoid,
weight update = Simple(eta=0.1,alpha=0.5),
objective function = EnrichmentAverage(
  cutoff=0.5,
  enrichment max=0.01,
  step size=0.00001,
  parity=1),
steps per update=1,
hidden architecture(100,100),
iteration weight update=MaxNorm(in=10,out=1),
shuffle=True,data selector=Tolerant(tolerance=0.1),
dropout(0.125,0.5))
```

Here, the average enrichment over the first 1% of the dataset is used as an objective function. 2 layers of hidden neurons are used, with each layer containing 100 neurons. The network dropout method (Hinton et al., 2012) is used to regularize the network and prevent over-fitting. A 90 fold cross-validation is used, so the `config.ini` file will result in the creation of 90 networks.

The submit.py script is run from the directory containing the config.ini file as follows:

```
submit.py -t cross_validation
```

This script will set up the cross-validation, and dispatch each of the 90 network training processes to a cluster.

F.2.5 Neural network analysis

The submit.py script also performs a basic analysis of the trained networks. The average enrichment across the entire cross-validation is computed, as are the TPR, FPR and PPV plots which are used to assess the performance of the network. The results of this analysis are output in the results/ directory produced by the submit.py script.

F.2.6 Benchmark data preparation

The DEKOIS 2.0 (Bauer et al., 2013) dataset was used for benchmarking purposes. The ligand SDF files used for the dataset are obtained from http://www.dekois.com/, and the associated protein files were obtained directly from the PDB. Cleaned and prepared data are in the attached DVD, and were prepared and docked identically to the process described in Section F.2.1

F.2.7 Benchmark data analysis

F.2.7.1 descriptor computation

Rather than using the SDF files of the DEKOIS 2.0 docked ligand poses to train a network, we will score our dataset using an existing network. This process is performed using the BCL molecule:Properties command. The cross-validation process used for network training produces an ensemble of 90 trained models, so the predicted activity is computed as the average of the output of all 90 models. molecule:Properties is run with the following command:

```
bcl.exe molecule:Properties -input_filenames \
input_ligands.sdf \
-add 'Mean(
  PredictedActivity(
    storage=File(directory=<model_dir>,prefix=model)))' \
-rename 'Mean(
  PredictedActivity(
    storage=File(directory=<model_dir>,prefix=model)))' \
     predicted_activity \
-tabulate 'Cached(Name)' 'Cached(system_name)' \
'Cached(log_ki)' 'Cached(predicted_activity)' \
-output_table output.csv
```

Where `<model_dir>` is the path to a directory of trained neural network models, and `output.csv` is the path to an output file.

F.2.7.2 ROC curve generation

The CSV files produced in the previous step contain all predicted and experimental activities for all ligands in the DEKOIS 2.0 set. For the purposes of this study, ROC curves will be created individually for each system. The BCL application model:ComputeStatistics is used to compute ROC plot and ROC-AUC values for a set of network predictions. This application requires an input file in the following format:

```
bcl::linal::Matrix<float>
1207 2
0.000000 0.123812
1.000000 0.123218
...
```

The first line is a header indicating that the data is a BCL matrix, the second line indicates that the matrix has 1207 rows and 2 columns, and the remaining rows are the data values. In this case, the first column should be the experimental values, and the second column should be the predicted values. The second column should be sorted such that the best predicted scores are first. To accomplish this task, the `make_bcl_inputs_for_plotting.py` script is used. The script is run as follows:

```
make_bcl_inputs_for_plotting.py \
--exp_label 'Cached(log_ki)' \
--pred_label 'Cached(predicted_activity)' \
input_data.csv output_dir/
```

This script will create BCL matrix files for each protein system found in the dataset and output these files to the specified directory `output_dir/`. The ROC-AUC values are then computed using these output table files and the script `collect_enrichment.py` This script is run as:

```
collect_enrichment.py output_dir/ tag
```

Where tag is a user specified tag. The script will output the protein system name, ROC-AUC and the tag for each protein system to standard output. Additionally, it will produce data files for each protein system in the `output_dir` directory, which can be used to graph ROC curves.

Appendix G

Predicting ligand binding poses in KRAS and RPA70

G.1 Introduction

KRAS and RPA70 are two proteins with potential value as targets in cancer research. This appendix describes a molecular modeling project to identify predicted binding modes in a set of small drug-like molecules previously identified through a ligand-based vHTS protocol.

G.1.1 KRAS Overview

KRAS is well studied oncogene, known to be activated in 17-25% of human tumors (Kranenburg, 2005). KRAS is a self-inactivating GTPase which is active in its GTP bound from, and inactive in the GDP bound form. Activation of KRAS drives a wide range of downstream cell signaling processes involved in cell proliferation and metabolism (Eser et al., 2014).

Since the 1970s, the RAS family of genes to which KRAS belongs has been extensively studied and validated as a target for cancer treatment (Pylayeva-Gupta et al., 2011). Despite a clear role as an oncogene in many cancers, it has historically been a difficult target for small molecule drugs (Sun et al., 2014). GDP and GTP binding molecules are typically difficult to selectively target at their primary binding sites due to the large number of signaling proteins which natively bind GDP/GTP. In an attempt to address this issue, an Structure Activity Relationship (SAR) by Nuclear Magnetic Resonance (NMR) study was performed, which resulted in an allosteric binding site which was selectively targetable by small drug-like molecules(Sun et al., 2012).

The experimental chemical data obtained from the 2012 SAR by NMR study was used to create a ligand based QSAR model. This model was then used to provide a small set of potential vHTS leads which are investigated using structure based methods in this study.

G.1.2 RPA70 Overview

RPA70 is the 70S subunit of RPA, and is an ssDNA binding protein involved in cell cycle regulation. Under normal circumstances, RPA activity is regulated by phosphorylation, typically as a result of cellular DNA damage (Wold, 1997). Activating mutation of RPA70 is implicated in a number of cancers (Hass et al., 2010), and the inhibition of the RPA complex is known to suppress tumor growth (Glanzer et al., 2014).

Although RPA70 is a highly validated target for cancer therapeutics, it has been historically difficult to identify tightly binding compounds. Recently, engineering of the protein surface made it possible to obtain high resolution X-Ray crystal structures with co-crystallized inhibitors (Feldkamp et al., 2013). These crystal structures were used as the source of native binding pose data for the study described in this chapter.

G.2 Methods

At the time when this study was conducted, the RosettaHTS protocol described in previous chapters was still under development. As a result, the docking study was performed using the RosettaLigand protocol originally described in (Meiler and Baker, 2006). Identical methods were used for preparing the small molecules and protein structures used in the KRAS and RPA70 studies.

G.2.1 Small molecule preparation

Small molecules were initially obtained as 2D structures. A library of 3D conformations was created using the LowModeMD (Labute, 2010) method implemented by the Molecular Operating Environment (MOE) software package. The following settings were used during LowModeMD conformer generation

- **Rejection Limit** — 100
- **RMS Gradient** — 0.005
- **Iteration Limit** — 10000

- **MM Iteration Limit** — 500
- **Enforce chair conformations** — Yes
- **RMSD Limit** — 0.25
- **Energy Window** — 7
- **Conformation Limit** — 100
- **Separate strain energy by stereo class** — Yes
- **Exclude fixed atoms from shape descriptors** No

The actual number of conformers generated depends on the flexibility of the individual ligands. The resulting conformer libraries were exported as concatenated PDB files suitable for use as Rosetta single residue rotamer libraries.

G.2.2 Protein structure preparation

A set of X-Ray crystal structures of both KRAS and RPA70 was used as the source of protein models for the docking study. In each case, the protein was co-crystallized with a ligand having known binding activity. To produce an ensemble of low energy models, each protein crystal structure was minimized in the absence of the ligand using the Rosetta relax protocol. Twenty independent minimization trajectories were computed to produce an ensemble of twenty protein structures.

G.2.3 Cross-docking to validate RosettaLigand performance

It is known from previous research that RosettaLigand is not capable of effectively predicting the binding affinity or binding pose of all protein systems (Davis et al., 2009). For this reason, prior to carrying out a docking study, it is critical to determine whether Rosetta is capable of docking ligands into each protein system being studied. Because the available set of X-Ray crystal data consists entirely of co-crystals with relatively diverse docked ligands, we can perform a cross-docking study to validate RosettaLigand's performance.

Two cross-docking studies were performed independently with RPA70 and KRAS. In both cases, each ligand was docked into each of the relaxed protein models described in Section G.2.2. 1000 ligand docking trajectories were computed for each protein model. The following parameters were used for ligand docking:

- **Translation distribution** — Uniform
- **Maximum Translation distance** — 5.0 Å
- **Maximum Translation cycles** — 50
- **Rotation distribution** — Uniform
- **Maximum Rotation angle** — 360°
- **Maximum Rotation cycles** — 500
- **High resolution docking cycles** – 6
- **High resolution Repack every** – 3rd cycle

This resulted in a set of 196,000 ligand binding predictions for the KRAS study and 95,000 binding predictions for the RPA70 study. By plotting the Score versus the RMSD of each ligand, we can verify that RosettaLigand can reliably position each ligand in the active conformation with a low score.

G.2.4 Predicting binding modes of vHTS identified leads

After validation of RosettaLigands ability to handle the KRAS and RPA70 protein systems, the same protocols used for the cross-docking studies described above were repeated to dock the vHTS leads into the ensemble of relaxed protein models.

A ligand-based vHTS study (unpublished) was performed by Will Lowe in the Meiler lab. This study used a machine learning approach based on the approach described by Mueller et al. (Mueller et al., 2012). The result of this study was a small set of compounds with high levels of predicted activity for both KRAS and RPA70.

For both studies, three letter Ligand IDs were arbitrarily assigned to each ligand. These three letter IDs will be used to reference the ligands through this chapter.

G.3 Results and Discussion

G.3.1 KRAS

G.3.1.1 Results of self and Cross-Docking validation

When each ligand from the set of protein-ligand co-crystal structures was docked into its own respective protein model and score vs RMSD plots are generated, we see that in the majority of cases, there are low scoring, low RMSD predicted binding poses with scores in a similar range to the minimized structure (Figure G.1).

When the ligands are cross-docked, we also see the overall presence of low scoring, low RMSD models (Figure G.2). However, we also see, in all cases, the presence of a second distribution of binding poses, with an RMSD of roughly of 6.0 Å to the native binding position. The narrow band of the 2nd distribution suggests that it represents a distinct binding pose.

Detailed inspection of low scoring members of the two binding poses confirms that there is a second binding pose predicted by RosettaLigand. As the second pose was observed in all cross-docked ligands, we use a single ligand (ID: 000), as an example in this analysis. In the left hand panel of Figure G.3, we see the score vs. RMSD plot for ligand 000. We see here that the lowest scoring models in the 6 Å RMSD distribution are of a slightly lower score. As a result, we can compare the two distributions of poses by comparing the lowest scoring models to the lowest RMSD models (right hand panel).

This comparison is striking, we see that there are indeed two distinct populations of lig-

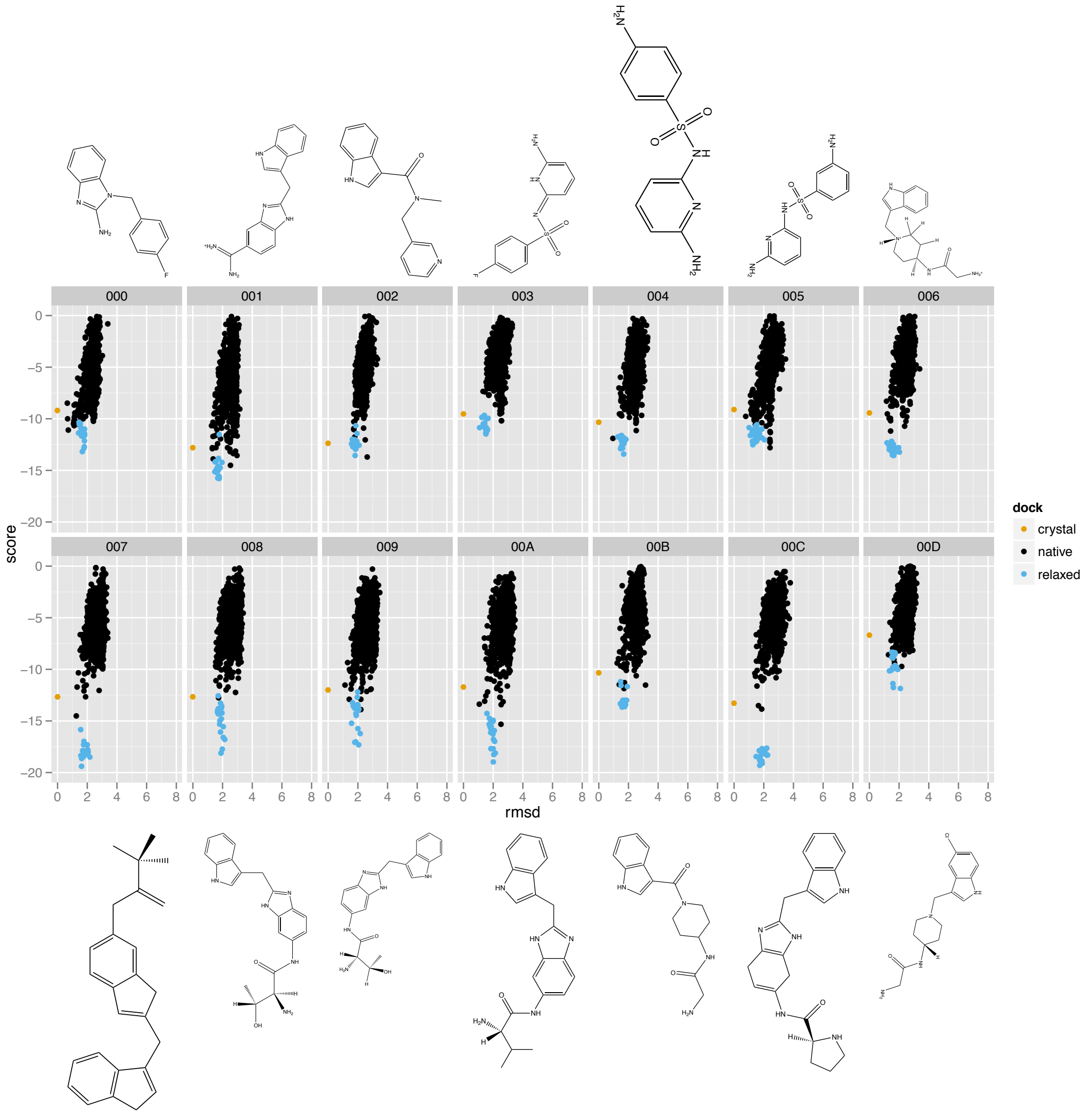


Figure G.1: Score vs RMSD plots for KRAS ligands self-docked into the relaxed models of the proteins they were crystalized with. The 2D structures of the ligands are pictured above and below each plot.

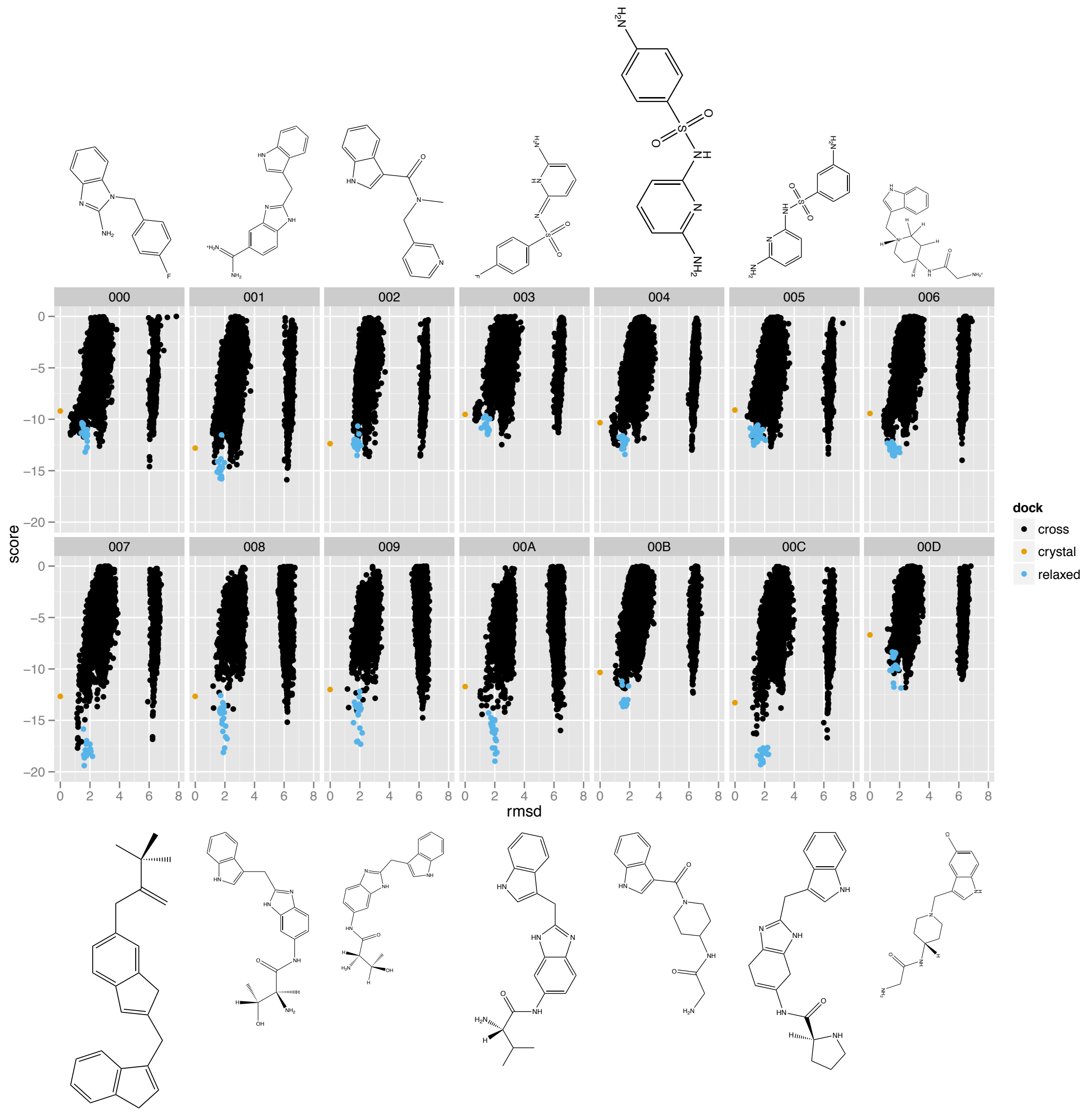


Figure G.2: Score vs RMSD plots for KRAS ligands cross-docked into the relaxed models of the entire KRAS protein set. The 2D structures of the ligands are pictured above and below each plot.

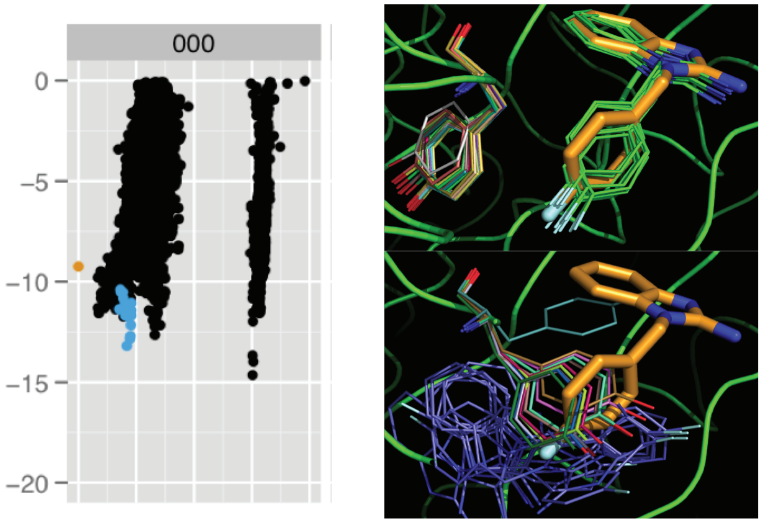


Figure G.3: A comparison of the two binding modes seen in the KRAS cross-docking study. At left, the score vs. RMSD plot for the 000 ligand. At right, the lowest RMSD models (in green), Best scoring models (blue), and crystallographic position, (orange).

ands, and furthermore that these two populations are mutually exclusive due to a rotamer shift. Inspection of the two distributions indicates that the Y71 side-chain (rendered as sticks in Figure G.3 changes positions to accommodate the ligand binding position represented by the 6 Å RMSD distribution. This position change is seen consistently across the range of RosettaLigand models, and results in the Y71 side-chain intersecting the native binding position.

G.3.1.2 Analysis of vHTS lead predictions

The KRAS virtual screening leads described in Section were docked using the same protocol that was used for the validation studies described previously. Because the Rosetta energy function varies with atom count, the scores of the models were normalized by the number of atoms in each ligand so as to allow for a fair comparison between ligands. As with the cross-docking study, each of the vHTS leads was observed in low energy poses at both the crystallographic binding position and the secondary binding position described above. Figure G.4 compares the normalized score of the lowest scoring model of each

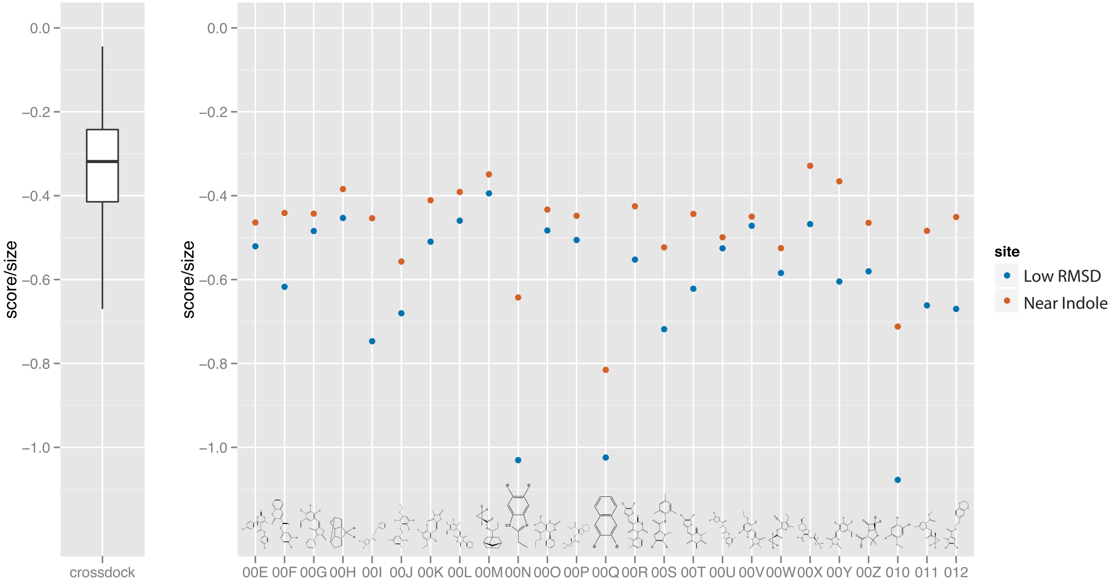


Figure G.4: A plot of the size normalized scores of the lowest scoring KRAS vHTS lead compounds, relative to the distribution of scores for low RMSD ligands in the cross-ndocking validation study. For each vHTS lead compound, The score of lowest scoring ligand near the native binding site (“low_rmsd”, in blue), and at the secondary predicted binding position described in Section G.3.1.1 (“near_indole”, in red)

vHTS lead at both binding positions. These models are compared to the overall distribution of model scores in the cross-docking study.

We see from Figure G.4 that in general, all ligands have relatively low scoring poses at both ligand positions. Three ligands stand out as having particularly low scores relative to both the models in the cross-docking study and the overall set of vHTS leads. Specifically, ligands with IDs 00N, 00Q, and 010. These three ligands have poses with normalized interface scores of < 1.0, while all other models in both the vHTS screen and the cross-docking study have scores of > 0.8.

Two-Dimensional structures of these three lowest scoring ligands are pictured in Figure G.5. Inspection of the 3D structures of the low scoring models for 00N, 00Q, and 010 indicates that the predicted binding poses are structurally similar to the observed binding poses known active compounds (Figure G.6).

While superficially, the predicted binding poses of these ligands appear plausible, the

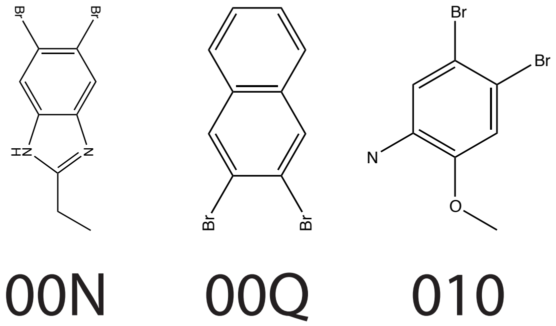


Figure G.5: Two dimensional structures of the three KRAS vHTS leads with normalized scores below -1.0

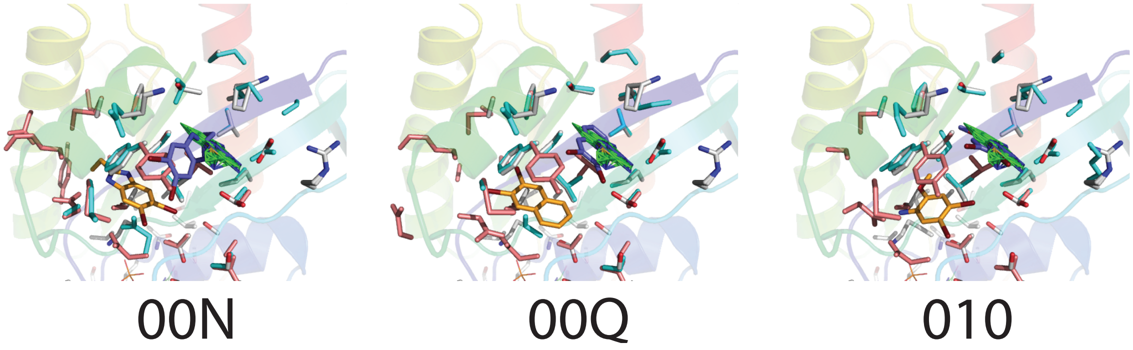


Figure G.6: Representations of 00N, 00Q, and 010, the lowest scoring RPA70 vHTS leads. In each case, the low scoring "low RMSD" and "near Indole" binding positions are plotted. The natural crystallographic ligand positions are plotted in bright green.

chemical properties of the ligands cast doubt on their usefulness as drug discovery leads, and on the accuracy of the unusually low Rosetta interface scores. Specifically, all three of the ligands have two bromine atoms. Since bromine is a large and relatively hydrophobic atom, and since Van der Waals packing is a highly weighted component of the Rosetta energy function, it is possible that the low normalized scores for these ligands is due to the ease with which a large, spherical hydrophobic mass can be packed into the binding pocket.

Even if the energy function is producing accurately low scores for these three compounds, the presence of multiple bromines presents a serious challenge to their usefulness as lead compounds for drug development. The high molecular weight and greasy properties of bromine atoms makes optimization of these compounds challenging, and they would probably be excluded from continuing drug development.

G.3.1.3 Comparison of Y71 side-chain shift predictions with new experimental data

At the time that the ligand binding study described above was performed in 2012, there was no experimental evidence of the shift in the Y71 residue seen in Figure G.3 existing in nature. More recently, in 2014, the Fesik lab published a series of crystal structures investigating a second binding site distal from the original binding site described in this study (Sun et al., 2014). As part of this study, five crystal structures[1] were prepared, with optical resolutions ranging from 1.20-1.88 Å. In four of the five published structures, an indole-derived ligand was covalently bound to the cystine mutated protein at S39C. The purpose of this covalently bound ligand is to occupy the first binding site so that binding of ligands into the second binding site could be effectively studied. Figure G.7 illustrates the relative position of the two binding positions.

Low scoring models for ligand ID 000 were compared to the new crystal structures produced by the 2014 Fesik lab study. A rendering of the superimposition of these structures is shown in Figure G.8. In one of the five new crystal structures (PDB ID 4q01), the Y71 rotamer is found in the shifted position observed in the 6.0 Å RMSD distribution described

[1] PDB IDs: 4q01, 4q02, 4q03, 4pzy, and 4pzz

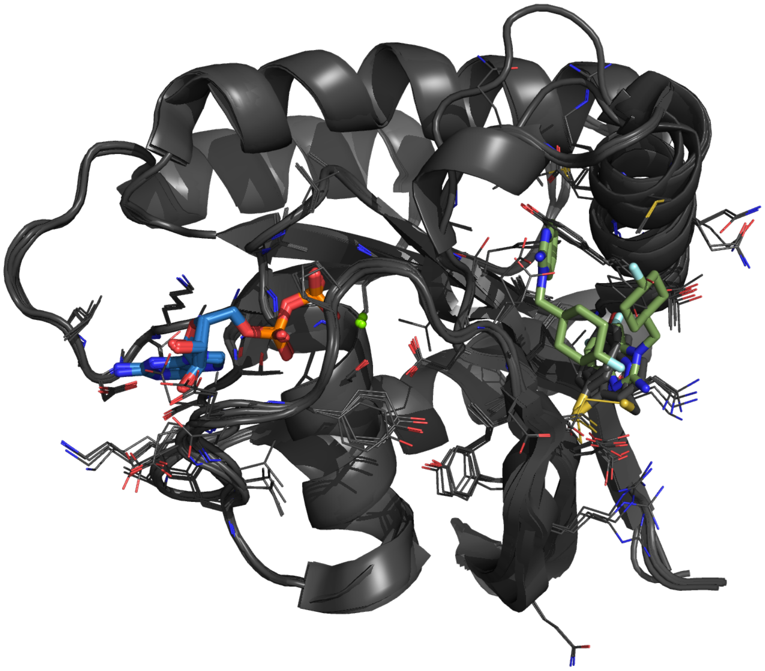


Figure G.7: Superimposition of PDBs 4q01, 4q02, 4q03, 4pzy, and 4pzz with two low scoring RosettaLigand models. The GDP ligand co-crystallized at the second binding site is highlighted in blue sticks. Two low scoring RosettaLigand models (Ligand ID 000), one in each of the two predicted binding binding modes are highlighted in green sticks.

previously and illustrated in Figure G.3.

The experimental observation of the Y71 rotamer shift predicted in nature suggests that the second binding mode described in Section G.3.1.1 may be physically possible. That being said, there are several critical differences between the RosettaLigand predictions made in this study and the binding modes observed in the 2014 Fesik lab study.

First, we see in Figure G.3 that the position taken by the ligand in the primary binding mode is directly blocked by the position of the Y71 rotamer in the second binding mode. This occurs because the tail of the docked ligand in the primary binding mode occupies the space of the Y71 rotamer in the second binding mode. In contrast, the covalently bound ligands used in the 2014 Fesik lab study consist only of the head of the ligand, and thus cannot occupy the space of the Y71 side-chain.

The second difference between the two studies is the presence of a ligand at the second

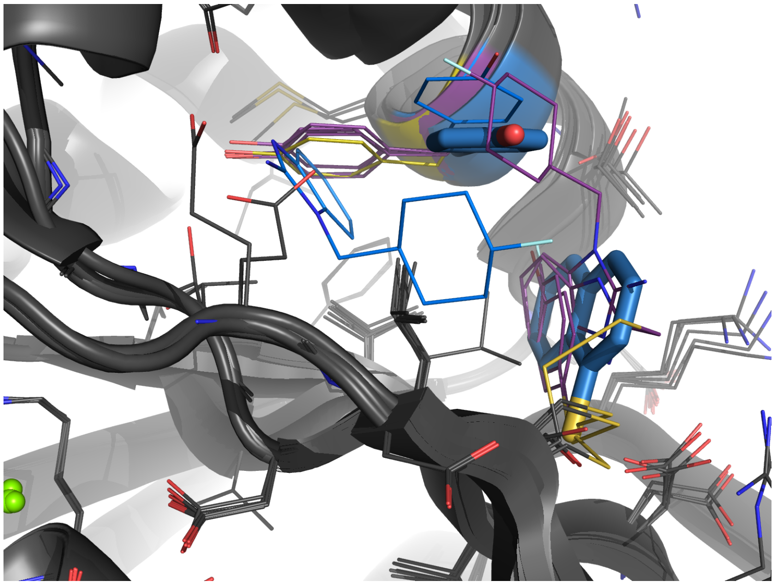


Figure G.8: Superimposition of PDBs 4q01, 4q02, 4q03, 4pzy, and 4pzz with two low scoring RosettaLigand models. In purple, crystal structures and predicted ligand binding poses associated with the primary binding mode at the first binding site. In blue, crystal structures and predicted ligand binding posts associated with the secondary binding mode at the first binding site. In blue sticks, the covalently bound ligand and Y71 of crystal structure 4q01 are highlighted.

binding site at in the 2014 Fesik lab study. In the RosettaLigand models described in this study (and the crystal structures which formed the basis of that study), the second binding site was empty. This difference is a critical one, as the presence of GDP at the second binding site may result in significant changes to the energy landscape of the first binding site.

These two differences between the study performed in this chapter and the 2014 Fesik lab study limit the conclusions we can draw from the comparison between the two. We can conclude that the Y71 side-chain is physically capable of performing the rotamer shift observed in the RosettaLigand predictions in this study. However, further studies will be required to determine if the predicted second binding mode associated with this side-chain shift is realistic. It is entirely possible that the presence of a ligand at the second binding site is required in order for the Y71 rotamer to change position.

G.3.2 RPA70

G.3.2.1 Results of self and Cross-Docking validation

As with KRAS in Section G.3.1.1, a cross-docking study was performed to assess the ability of RosettaLigand to predict binding modes against RPA70. Figure G.9 plots score vs RMSD for each of the ligands cross-docked into the RPA70 binding site. In most cases, low scoring predicted binding poses are found within 2.0 Å of the crystallographic binding position. However, unlike the results of the KRAS cross-docking study in Figure G.2, the low scoring models are distributed across a wide range of RMSD.

Upon inspection of the RPA70 binding site, a possible explanation emerges. Figure G.10 illustrates that the overall structure of the RPA70 binding site is a long, deep trough across the surface of the protein, bounded by two ligands, T55 and S60, which are critical for ligand binding (Feldkamp et al., 2013). Figure G.11 shows the crystallographic positions of the ligands involved in the cross-docking study. From this we see that although all the ligands in the binding site are in close proximity to T55 and/or S60, the overall range of binding position is substantial. This wide range of acceptable binding position is the likely explanation for the wide range in RMSD of low scoring models.

The results of the cross-docking study illustrated in Figure G.9 suggest that RosettaLigand is generally capable of identifying high quality binding poses within the RPA70 active site. However, it does not appear that it is generally capable of determining whether a ligand will bind preferentially in the vicinity of T55 or S60.

G.3.2.2 Analysis of vHTS lead predictions

As in the KRAS study described in Section G.3.1.2, each of the vHTS leads was docked into RPA70, and the normalized scores of the best scoring models for each ligand were plotted against the range of normalized scores seen in the cross-docking study. Figure G.12 plots the normalized scores of the lowest scoring models in the vicinity of both the T55 and T60 ligands relative to the overall distribution of models produced in the cross-docking

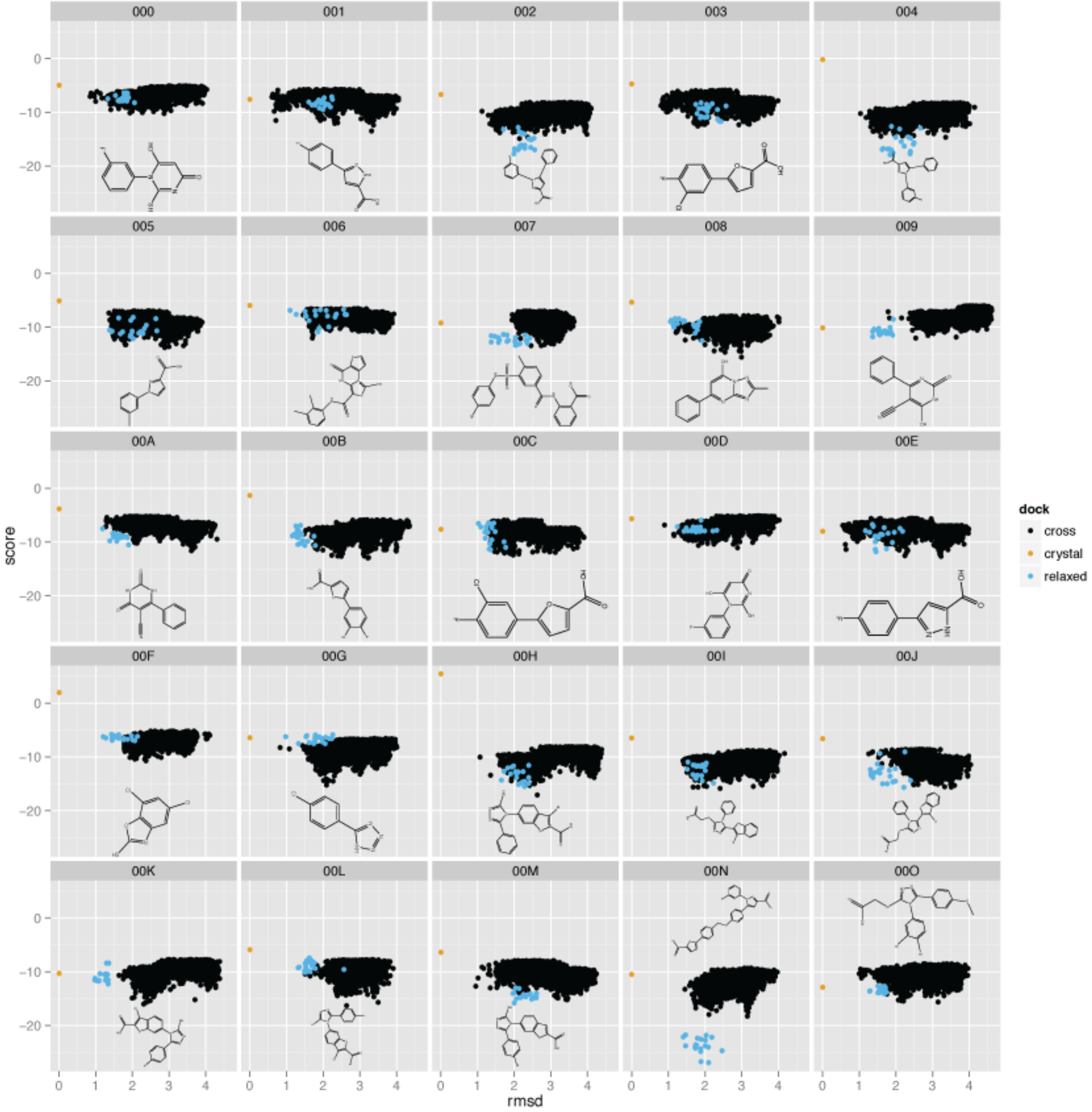


Figure G.9: Score vs RMSD plots for RPA70 ligands cross-docked into the relaxed models of the proteins they were crystalized with. The 2D structures of the ligands are pictured above and below each plot.

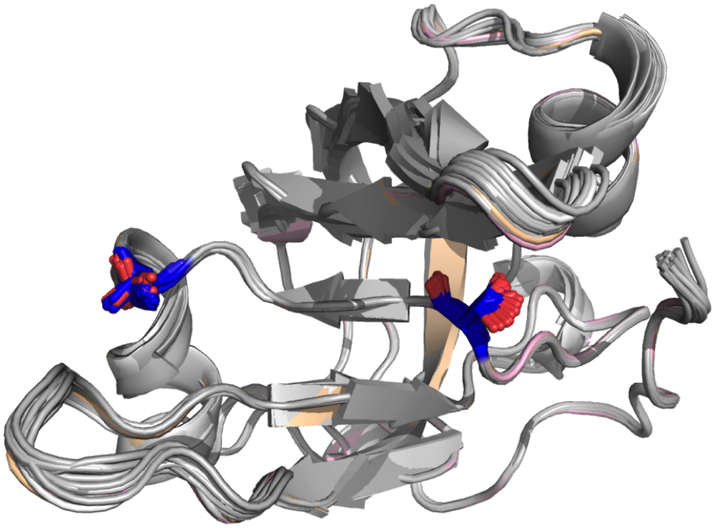


Figure G.10: Ribbon diagram of RPA70. Residues T55 and S60, highlighted in blue sticks, are both critical for ligand binding and represent the edges of a long narrow ligand binding pocket.

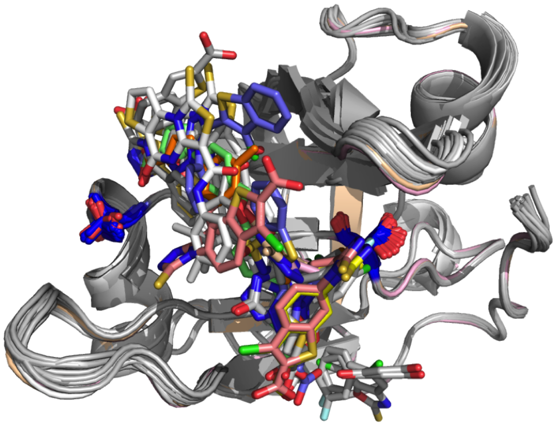


Figure G.11: Ribbon diagram of RPA70 with the crystallized position of the co-crystallized ligands shown as sticks.

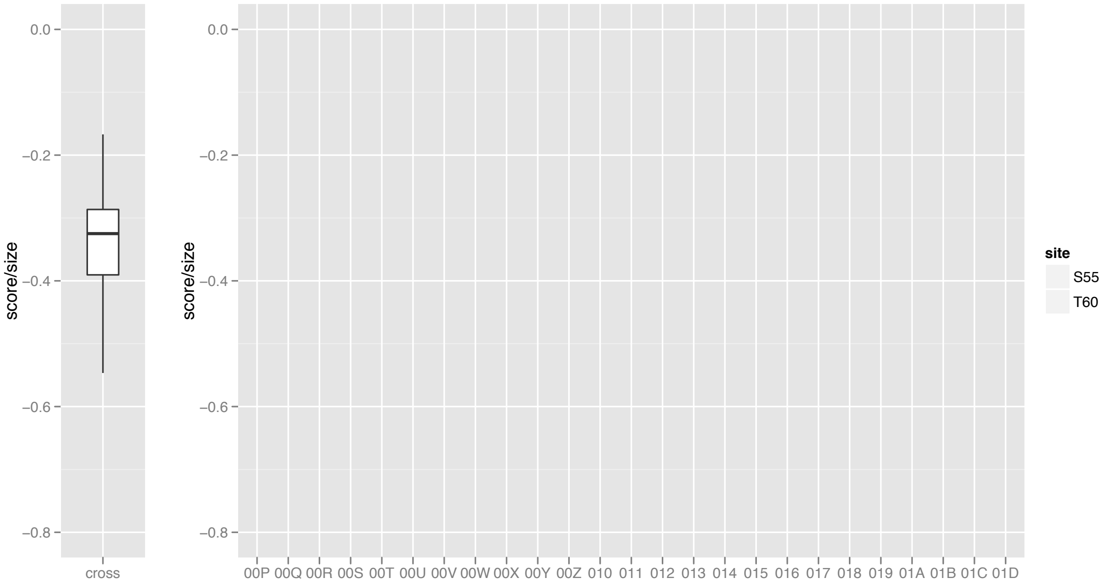


Figure G.12: A plot of the size normalized scores of the lowest scoring RPA70 vHTS lead compounds, relative to the distribution of scores for low RMSD ligands in the cross-docking validation study. For each vHTS lead compound, The score of lowest scoring ligand model near the T55 location ('T55', in blue), and the T60 location('T60', in red) is plotted.

study.

Of the ligands considered in this study, all have low scores within the average range of predictions produced in the cross-docking study. However, two ligands stand out with scores substantially below the median, specifically ligand IDs 00Y and 011. The binding sites of the low scoring 00Y and 011 models plotted in Figure G.12 are rendered in Figure G.13. A comparison of these predicted binding sites to the crystallographic binding sites of known active ligands (Figure G.11) indicates that the predicted binding positions of these low scoring ligands are similar to those observed in nature, suggesting that the low scores of these ligands may be plausible.

G.3.3 Future Directions

The study described in this chapter provides a starting point for a broader study into the pharmacology of both KRAS and RPA70. There are a number of potential avenues for

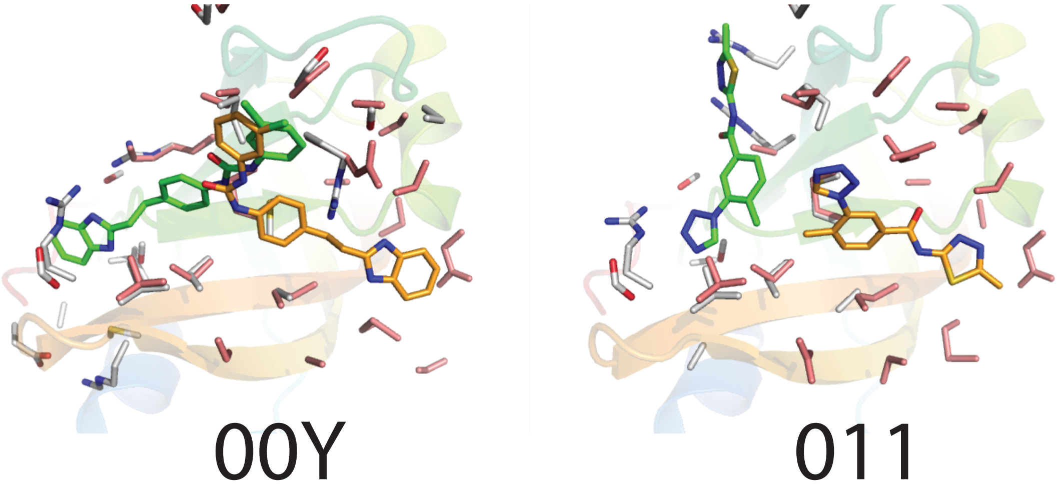


Figure G.13: Representations of 00Y and 011, the lowest scoring RPA70 vHTS leads. In each case, the lowest scoring ligands in the vicinities of the T55 and S60 binding sites are plotted.

continuation of this study, both computational and experimental.

In the case of the KRAS vHTS study, more stringent filtering of the vHTS leads is required prior to further computational studies. Specifically, compounds with multiple halogens should be excluded.

Further study is required to understand the secondary binding mode predicted by RosettaLigand and partially observed by (Sun et al., 2014). While it is now known that the Y71 amino acid is capable of existing in both orientations predicted by RosettaLigand in this study, further study is required to determine the conditions under which this secondary orientation can occur. One potential means of investigating the behavior of the Y71 amino acid is through Molecular Dynamics (MD) studies. In the 2014 Fesik Lab study in which the secondary orientation of the Y71 amino acid was observed, both KRAS binding sites were fully occupied. In this case, the goal of the MD study would be to determine whether it is possible for the secondary Y71 orientation to exist without the second binding site being occupied.

In addition to computational studies, further experimental work could be used to investigate the secondary binding mode predicted at the first KRAS site. In the 2014 Fesik lab study, the primary binding site was occupied by a covalently bound ligand. A similar study could be performed to investigate ligand binding in the primary binding site by covalently attaching a ligand to the secondary site.

Appendix H

The Limitations of Rosetta Atom typing and Orbital assignment

H.1 The origins of the Rosetta Atom types

The system of atom parameterization used by Rosetta has evolved dramatically as the software suite has expanded in functionality. In order to gain a full understanding of the capabilities and limitations of this system, some discussion of the history its development is required. Rosetta was originally conceived as a method for *ab initio* protein folding. The original fragment based protein folding algorithm used a coarse-grained model in which each ligand was modeled as a backbone with a single centroid atom representing the side chain. For the centroid based model, the atom-typing system consisted of one type for each backbone atom, and one type for each canonical amino acid (Simons et al., 1997).

Rosetta was later extended to perform full-atom protein modeling with an atomic detail energy function (Rohl et al., 2004) and a new set of atom types (Kortemme et al., 2003). The parameters for these atom types were derived from a range of sources. Partial charges were based on those used in the CHARMM Molecular Mechanics (MM) function (Brooks et al., 1983) and solvation parameters were provided by the Lazaridis-Karplus solvation method (Lazaridis and Karplus, 1999). 34 distinct atom types were created which map unambiguously to the atoms in the 20 canonical amino acids. Additionally, atom types representing Halogens (F, Cl, Br, I), Metals (Zn, Fe, Mg, C, Na, K), Phosphorus and Silicon were added, brining the total count to 49 distinct atom types.

H.2 Rosetta is limited in its ability to parametrize small molecules

While the atom types described in Section H.1 can effectively parameterize the canonical amino acids, as well as common co-factors and metal ions, the parameterization of arbitrary small molecules is more limited. When a molecule that is not a standard co-factor, amino acid or metal is used in Rosetta, it is parameterized using the decision tree illustrated in

Figure H.1.

Close inspection of the decision tree illustrates the problems with it as a parameterization algorithm. The decisions itself was designed to reliably parameterize the canonical amino acids. The chemical space represented by canonical amino acids is extremely limited, and these limitations are directly reflected by the decision tree.

The atom typing decision tree has a default “fall through” case for every element type it supports. For this reason, it will assign atom types to every atom with a supported element, but it will frequently assign these atom types incorrectly.

As an example of this problem, take the case of the COO atom type. The COO atom type is meant to parameterize the carbon in a COOH group, such as that found in Aspartic acid. However, as shown in Figure H.1, the COO atom type is the default typing for unsaturated carbons. The effect of this choice of default is seen in uracil. Uracil is illustrated in Figure H.2. Here, we draw the molecule with the assigned Rosetta atom-types labelled in place of the element names. Note the atom-types of the atoms involved in the two $\mathrm{C{=}O}$ groups. In both cases, the carbon is assigned as COO, and the oxygen is assigned as ONH2. The parameters of both these atom-types were derived from vey different chemical contexts than they contexts in which they are used in uracil.

This is only one of many cases in which the Rosetta atom-typing decision tree will incorrectly type atoms. The parameterization of nitrile ($\mathrm{R{-}C{\equiv}N}$) groups is a case which the standard Rosetta atom typing method is consistently unable to handle. In this case, the difficulty stems from the lack of an atom-type that reflects either the C or N in the nitrile. Rosetta will use the decision tree to parameterize $\mathrm{R{-}C{\equiv}N}$ as $\mathrm{R{-}COO{\equiv}Npro}$, where COO (as described previously) represents the C in a COOH group, and Npro represents the N in a proline ring. The properties of both atom types are sufficiently different than those of a nitrile group that it is unlikely that a nitrile parameterized as $\mathrm{R{-}COO{\equiv}Npro}$ will result in realistic Rosetta models.

Overall, the issues with the Rosetta atom-typing scheme can be viewed as a classic

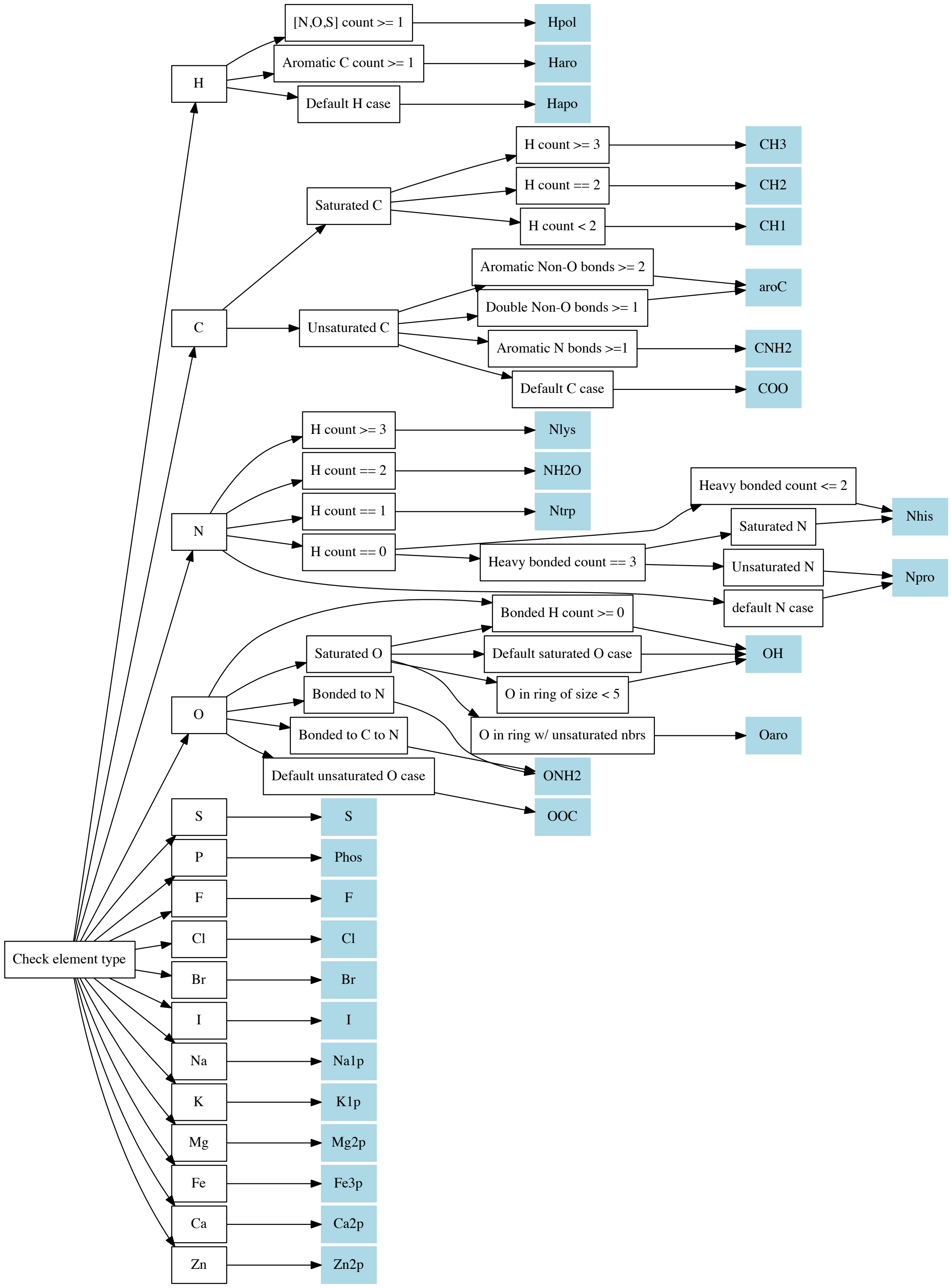


Figure H.1: Rosetta atom types are assigned to small molecule atoms using a decision tree, which is illustrated here. Decision tree conditions are marked in black boxes, Rosetta atom type assignments are marked in light blue boxes.

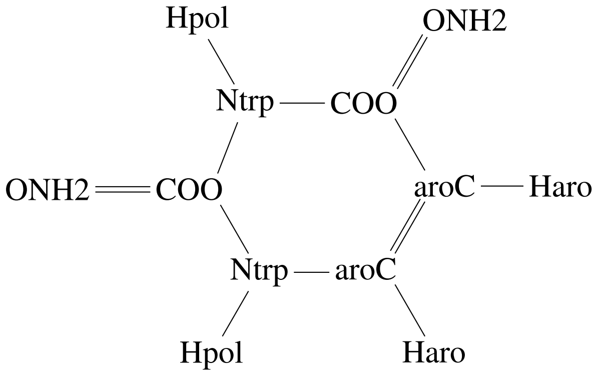


Figure H.2: A uracil molecule labeled with the atom-types assigned by the standard Rosetta decision tree algorithm.

case of model over-fitting. Both the atom-types themselves and the decision tree model for assigning them were based primarily on a narrow range of chemical space, and the resulting model therefore applies to that specific set of chemical space. RosettaLigand appears to be successful as a ligand docking system in spite of the parameterization system, rather than as a result of it. Regardless of ligand parameterization, Rosetta is effectively able to parameterize proteins, meaning that a large component of the ligand docking model is correct.

H.3 The limitations of Rosetta atom-typing impact the applicability of scoring functions to small molecule studies

In addition to being a source of chemical parameters (Size, Charge, etc) Rosetta atom-types are used for atom identification. This can cause problems when new energy terms relying on the Rosetta atom-types are implemented. In an attempt to improve the quality of Rosetta protein models, Steven Combs developed a set of KBP energy terms using electron orbital positions placed using Valence Shell Electron Pair Repulsion (VSEPR) theory (Combs, 2013). These energy functions were largely successful at improving the ability of Rosetta to recover the fine details of side-chain rotamer positioning.

Given this improvement, an obvious future direction was to extend the use of these orbital based potentials to the modeling of protein-small molecule interactions. The addition of molecular orbital information to small molecules has the potential to greatly improve the accuracy of Rosetta protein ligand interaction models.

However, since the orbital based KBP was developed based on the standard Rosetta energy types, there was concern that the errors in parameterization described in Section H.2 would limit the effectiveness of the method when applied to small molecules.

To investigate this, a small, preliminary, qualitative study was performed comparing the positions of orbitals as assigned by the Combs algorithm with molecular orbital densities computed using an DFT method. Combs method orbital assignments were computed with Rosetta. DFT computations were computed using Gaussian 09. The X-Ray crystal structures of the ligands were used as a starting point, and geometry optimization was performed using B3LYP with the 6-31G basis set and the SCRF implicit solvent model. After geometry optimization, HOMO molecular density models were computed using Gaussian CubeGen. H.3 illustrates the results of the two orbital assignments. Through a qualitative comparison of the two orbital assignment methods, we can see significant differences between the molecular orbitals assigned by the Combs method and the DFT computations. In some cases, the Combs method has performed well. For example, in Figure H.3B and H.3D, we see that the pi-orbitals associated with the benzene rings have been correctly assigned, with both peak density and orbital assignments running perpendicular to the ring plane. In other cases, however the Combs method is less successful. In H.3C and H.3D, There are several cases in which the Combs method has placed orbitals on oxygen atoms with a 90 degree orientation relative to the DFT predicted density. In other cases, such as the ring nitrogen in H.3B and H.3C, the Combs method has assigned orbitals where the DFT predicts no orbital density.

While this was only a preliminary investigation, it strongly suggests that the incorrect atom-typing provided by the default typing method is insufficient. In order to effectively assign orbitals to ligands using the Combs method, it will be necessary to use an atom-typing method that can unambiguously type any atom in an arbitrary small molecule.

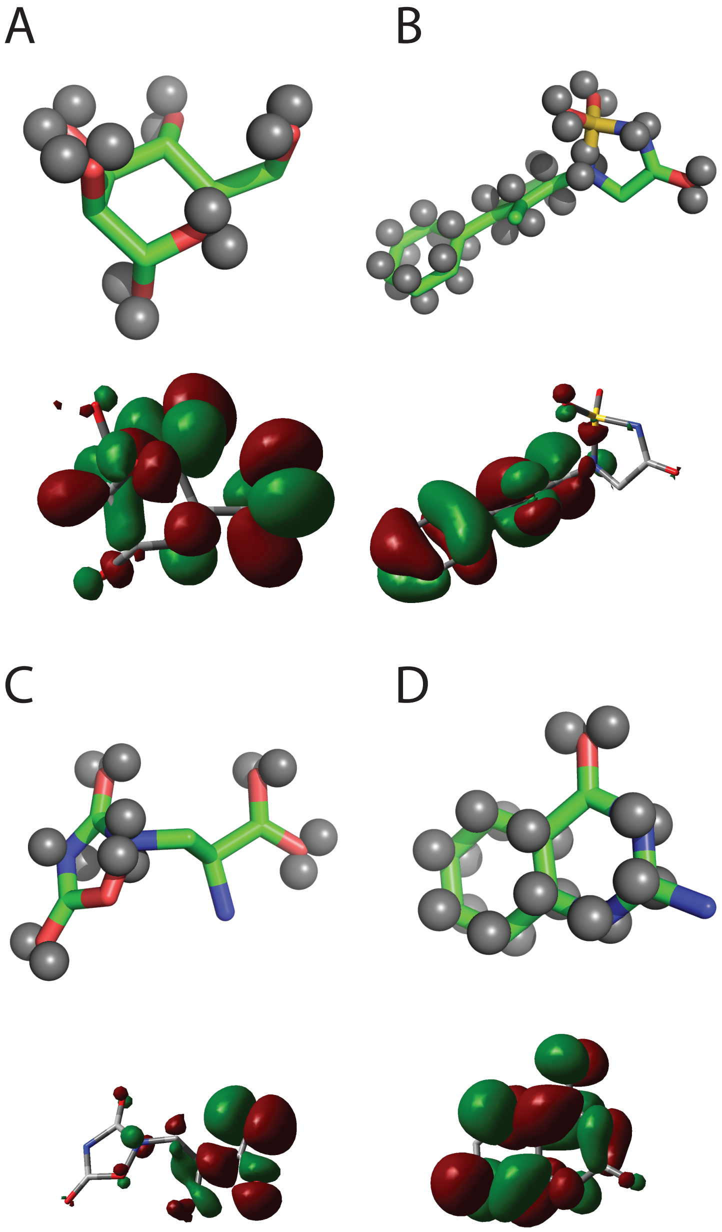


Figure H.3: Comparison of orbital positions assigned by the Combs algorithm with HOMO molecular orbital surfaces computed using a DFT method. Four molecules are plotted, with the Combs method assignments at top and the DFT surface at bottom. A) Alpha-D-Mannose crystallized with lectin (PDB ID: 2ARE). B) Inhibitor of Tyrosine-protein phosphatase non-receptor type 1 (PDB ID: 2NTA). C) Inhibitor of GluR6 (PDB ID: 1S9T). D) Inhibitor of tRNA-Guanine Transglycosylase (PDB ID: 1S39)

H.4 Potential improvements to the atom typing system

The existing Rosetta atom-typing system, while well suited to canonical amino acids, is not sufficient to correctly handle small molecules. However, the Rosetta software architecture allows for the simultaneous use of multiple atom-typing systems in a simulation, with different terms in the energy function making use of different atom-type assignments. As a result, we have the ability to easily implement new atom-typing schemes without major revision of the existing modeling algorithms and scoring functions. At the time of this writing, research in the Meiler lab is underway to develop a new atom-typing system with the ability to unambiguously assign atom-types to all small molecules. The atom-typing system being developed is derived from the research by Gasteiger et al on the effect of atomic substituents on atomic properties (Hutchings and Gasteiger, 1983; Gasteiger and Hutchings, 1983). The general concept of this new atom-type set is to type atoms based on the element of the atom, and the number and geometry of its substituents. For example, a carbon with four single bonded substituents (as in CH_4) would have the atom-type C_TeTeTe, while a phosphorous with two double bonded to 2 substituents (as in PO_2) would have the atom-type P_DiDiPiPi.

This method for atom-typing has several potential advantages, particularly in the context of the Rosetta energy function. Gasteiger atom-types can be rapidly and unambiguously for any atom in any molecule. Because the atom-type assignment relies only on the query atom and its immediate neighbors, loop and aromaticity detection are not required. Because the atom-type assignments are unambiguous, the previously described problem of mis-typing atoms can be avoided entirely. Additionally, because the Gasteiger atom-types are explicitly based on the geometry of the substituent atoms, VSEPR orbital positions can be computed using only the atom-type name.

By implementing Gasteiger atom-types, it will be possible to extend the orbital based energy function developed by Combs et al. to correctly handle arbitrary small-molecules. Since Rosetta easily supports the use of multiple simultaneous atom-typing schemes, this new Gasteiger based energy function could be used in conjunction with the existing Rosetta energy terms. While augmenting the existing energy function with a new set of terms based on Gasteiger atom-types would be beneficial to the modeling of protein-ligand interactions, the noise introduced by the incorrect assignment of Rosetta types will still limit the performance of the models. To address this problem in full, the KBPs which comprise the majority of the Rosetta energy functions could be recomputed using the Gasteiger atom-type assignments. This re-computation of the KBPs would represent a massive undertaking, but could potentially result in a substantial improvement in ligand docking performance.

BIBLIOGRAPHY

Abad, M. C., Askari, H., O'Neill, J., Klinger, A. L., Milligan, C., Lewandowski, F., Springer, B., Spurlino, J., and Rentzeperis, D. (2008). Structural determination of estrogen-related receptor gamma in the presence of phenol derivative compounds. *The Journal of steroid biochemistry and molecular biology*, 108(1-2):44–54.

Ahuja, E. G., Janning, P., Mentel, M., Graebsch, A., Breinbauer, R., Hiller, W., Costisella, B., Thomashow, L. S., Mavrodi, D. V., and Blankenfeldt, W. (2008). PhzA/B catalyzes the formation of the tricycle in phenazine biosynthesis. *Journal of the American Chemical Society*, 130(50):17053–17061.

Alguel, Y., Meng, C., Terán, W., Krell, T., Ramos, J. L., Gallegos, M.-T., and Zhang, X. (2007). Crystal structures of multidrug binding protein TtgR in complex with antibiotics and plant antimicrobials. *Journal of molecular biology*, 369(3):829–840.

Alicea, I., Marvin, J. S., Miklos, A. E., Ellington, A. D., Looger, L. L., and Schreiter, E. R. (2011). Structure of the Escherichia coli phosphonate binding protein PhnD and rationally optimized phosphonate biosensors. *Journal of molecular biology*, 414(3):356–369.

Allison, B., Combs, S., DeLuca, S., Lemmon, G., Mizoue, L., and Meiler, J. (2014). Computational design of protein-small molecule interfaces. *Journal of structural biology*, 185(2):193–202.

Almrud, J. J., Kern, A. D., Wang, S. C., and Czerwinski, R. M. (2002). The crystal structure of YdcE, a 4-oxalocrotonate tautomerase homologue from Escherichia coli, confirms the structural basis for oligomer diversity. *Biochemistry*.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.

Ambroggio, X. I. and Kuhlman, B. (2006). Computational Design of a Single Amino Acid Sequence that Can Switch between Two Distinct Protein Folds. *Journal of the American Chemical Society*, 128(4):1154–1161.

Arockia Jeyaprakash, A., Jayashree, G., Mahanta, S. K., Swaminathan, C. P., Sekar, K., Surolia, A., and Vijayan, M. (2005). Structural basis for the energetics of jacalin-sugar interactions: promiscuity versus specificity. *Journal of molecular biology*, 347(1):181–188.

Aronov, A. M., Baker, C., Bemis, G. W., Cao, J., Chen, G., Ford, P. J., Germann, U. A., Green, J., Hale, M. R., Jacobs, M., Janetka, J. W., Maltais, F., Martinez-Botella, G., Namchuk, M. N., Straub, J., Tang, Q., and Xie, X. (2007). Flipped out: structure-guided design of selective pyrazolylpyrrole ERK inhibitors. *Journal of medicinal chemistry*, 50(6):1280–1287.

Attik, M., Bougrain, L., and Alexandre, F. (2005). Neural network topology optimization. *Artificial Neural Networks: Formal Models and Their Applications*, 3697:53–58.

Austin, T. M., Nannemann, D. P., Deluca, S. L., Meiler, J., and Delpire, E. (2014). In silico analysis and experimental verification of OSR1 kinase - Peptide interaction. *Journal of structural biology*, 187(1):58–65.

Avvakumov, G. V., Grishkovskaya, I., Muller, Y. A., and Hammond, G. L. (2002). Crystal structure of human sex hormone-binding globulin in complex with 2-methoxyestradiol reveals the molecular basis for high affinity interactions with C-2 derivatives of estradiol. *The Journal of biological chemistry*, 277(47):45219–45225.

Axarli, I., Dhavala, P., Papageorgiou, A. C., and Labrou, N. E. (2009). Crystallographic and functional characterization of the fluorodifen-inducible glutathione transferase from Glycine max reveals an active site topography suited for diphenylether herbicides and a novel L-site. *Journal of molecular biology*, 385(3):984–1002.

Bandarage, U. K., Wang, T., Come, J. H., Perola, E., Wei, Y., and Rao, B. G. (2008). Novel thiol-based TACE inhibitors. Part 2: Rational design, synthesis, and SAR of thiol-containing aryl sulfones. *Bioorganic & Medicinal Chemistry Letters*, 18(1):44–48.

Barth, P., Schonbrun, J., and Baker, D. (2007). Toward high-resolution prediction and design of transmembrane helical protein structures. *Proceedings of the National Academy of Sciences of the United States of America*, 104(40):15682–15687.

Bauer, M. R., Ibrahim, T. M., Vogel, S. M., and Boeckler, F. M. (2013). Evaluation and Optimization of Virtual Screening Workflows with DEKOIS 2.0 - A Public Library of Challenging Docking Benchmark Sets. *Journal Of Chemical Information And Modeling*, 53(6):1447–1462.

Baugh, L., Le Trong, I., Cerutti, D. S., Gülich, S., Stayton, P. S., Stenkamp, R. E., and Lybrand, T. P. (2010). A distal point mutation in the streptavidin-biotin complex preserves structure but diminishes binding affinity: experimental evidence of electronic polarization effects? *Biochemistry*, 49(22):4568–4570.

Becker, J. W., Rotonda, J., Cryan, J. G., Martin, M., Parsons, W. H., Sinclair, P. J., Wiederrecht, G., and Wong, F. (1999). 32-Indolyl ether derivatives of ascomycin: three-dimensional structures of complexes with FK506-binding protein. *Journal of medicinal chemistry*, 42(15):2798–2804.

Beddell, C. R., Goodford, P. J., Norrington, F. E., Wilkinson, S., and Wootton, R. (1976). Compounds designed to fit a site of known structure in human haemoglobin. *British Journal of Pharmacology*, 57(2):201–209.

Bera, A. K., Atanasova, V., Robinson, H., Eisenstein, E., Coleman, J. P., Pesci, E. C., and Parsons, J. F. (2009). Structure of PqsD, a Pseudomonas quinolone signal biosynthetic enzyme, in complex with anthranilate. *Biochemistry*, 48(36):8644–8655.

Bertini, I., Calderone, V., Cosenza, M., Fragai, M., Lee, Y. M., Luchinat, C., Mangani, S., Terni, B., and Turano, P. (2005). Conformational variability of matrix metalloproteinases: beyond a single 3D structure. *Proceedings of the National Academy of Sciences of the United States of America*, 102(15):5334–5339.

Bingham, R. J., Findlay, J. B. C., Hsieh, S.-Y., Kalverda, A. P., Kjellberg, A., Perazzolo, C., Phillips, S. E. V., Seshadri, K., Trinh, C. H., Turnbull, W. B., Bodenhausen, G., and Homans, S. W. (2004). Thermodynamics of binding of 2-methoxy-3-isopropylpyrazine and 2-methoxy-3-isobutylpyrazine to the major urinary protein. *Journal of the American Chemical Society*, 126(6):1675–1681.

Björkman, A. J., Binnie, R. A., Zhang, H., Cole, L. B., Hermodson, M. A., and Mowbray, S. L. (1994). Probing protein-protein interactions. The ribose-binding protein in bacterial transport and chemotaxis. *The Journal of biological chemistry*, 269(48):30206–30211.

Blum, M.-M., Löhr, F., Richardt, A., Rüterjans, H., and Chen, J. C.-H. (2006). Binding of a designed substrate analogue to diisopropyl fluorophosphatase: implications for the phosphotriesterase mechanism. *Journal of the American Chemical Society*, 128(39):12750–12757.

Boeckler, F. M., Joerger, A. C., Jaggi, G., Rutherford, T. J., Veprintsev, D. B., and Fersht, A. R. (2008). Targeted rescue of a destabilized mutant of p53 by an in silico screened drug. *Proceedings of the National Academy of Sciences*, 105(30):10360–10365.

Böhm, H.-J., Flohr, A., and Stahl, M. (2004). Scaffold hopping. *Drug discovery today. Technologies*, 1(3):217–224.

Bolla, J. R., Do, S. V., Long, F., Dai, L., Su, C.-C., Lei, H.-T., Chen, X., Gerkey, J. E., Murphy, D. C., Rajashankar, K. R., Zhang, Q., and Yu, E. W. (2012). Structural and functional analysis of the transcriptional regulator Rv3066 of Mycobacterium tuberculosis. *Nucleic acids research*, 40(18):9340–9355.

Borsi, V., Calderone, V., and Fragai, M. (2010). Entropic contribution to the linking coefficient in fragment based drug design: a case study. *Journal of medicinal ...*, 53(10):4285–4289.

Brenk, R., Meyer, E. A., Reuter, K., Stubbs, M. T., Garcia, G. A., Diederich, F., and Klebe, G. (2004). Crystallographic study of inhibitors of tRNA-guanine transglycosylase suggests a new structure-based pharmacophore for virtual screening. *Journal of molecular biology*, 338(1):55–75.

Brick, P. and Blow, D. M. (1987). Crystal structure of a deletion mutant of a tyrosyl-tRNA synthetase complexed with tyrosine. *Journal of molecular biology*, 194(2):287–297.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., and Karplus, M. (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4(2):187–217.

Brown, C. K., Vetting, M. W., Earhart, C. A., and Ohlendorf, D. H. (2004). Biophysical analyses of designed and selected mutants of protocatechuate 3,4-dioxygenase1. *Annual review of microbiology*, 58:555–585.

Brumshtein, B., Greenblatt, H. M., Butters, T. D., Shaaltiel, Y., Aviezer, D., Silman, I., Futerman, A. H., and Sussman, J. L. (2007). Crystal structures of complexes of N-butyl- and N-nonyl-deoxynojirimycin bound to acid beta-glucosidase: insights into the mechanism of chemical chaperone action in Gaucher disease. *The Journal of biological chemistry*, 282(39):29052–29058.

Brylinski, M. and Skolnick, J. (2008). Q-Dock: Low-resolution flexible ligand docking with pocket-specific threading restraints. *Journal of Computational Chemistry*, 29(10):1574–1588.

Burkhard, P., Taylor, P., and Walkinshaw, M. D. (2000). X-ray structures of small ligand-FKBP complexes provide an estimate for hydrophobic interaction energies. *Journal of molecular biology*, 295(4):953–962.

Butkiewicz, M., Lowe, E., Mueller, R., Mendenhall, J., Teixeira, P., Weaver, C., and Meiler, J. (2013). Benchmarking Ligand-Based Virtual High-Throughput Screening with the PubChem Database. *Molecules*, 18(1):735–756.

Buts, L., Garcia-Pino, A., Imberty, A., Amiot, N., Boons, G.-J., Beeckmans, S., Versées, W., Wyns, L., and Loris, R. (2006). Structural basis for the recognition of complex-type biantennary oligosaccharides by Pterocarpus angolensis lectin. *The FEBS journal*, 273(11):2407–2420.

Bystroff, C. and Kraut, J. (1991). Crystal structure of unliganded Escherichia coli dihydrofolate reductase. Ligand-induced conformational changes and cooperativity in binding. *Biochemistry*, 30(8):2227–2239.

Bystroff, C., Simons, K. T., Han, K. F., and Baker, D. (1996). Local sequence-structure correlations in proteins. *Current opinion in biotechnology*, 7(4):417–421.

Campanacci, V., Lartigue, A., Hällberg, B. M., Jones, T. A., Giudici-Orticoni, M.-T., Tegoni, M., and Cambillau, C. (2003). Moth chemosensory protein exhibits drastic conformational changes and cooperativity on ligand binding. *Proceedings of the National Academy of Sciences of the United States of America*, 100(9):5069–5074.

Castell, A., Short, F. L., Evans, G. L., Cookson, T. V. M., Bulloch, E. M. M., Joseph, D. D. A., Lee, C. E., Parker, E. J., Baker, E. N., and Lott, J. S. (2013). The substrate capture mechanism of Mycobacterium tuberculosis anthranilate phosphoribosyltransferase provides a mode for inhibition. *Biochemistry*, 52(10):1776–1787.

Chandler, D. (2005). Interfaces and the driving force of hydrophobic assembly. *Nature Cell Biology*, 437(7059):640–647.

Chen, C. C. H., Han, Y., Niu, W., Kulakova, A. N., Howard, A., Quinn, J. P., Dunaway-Mariano, D., and Herzberg, O. (2006). Structure and kinetics of phosphonopyruvate hydrolase from Variovorax sp. Pal2: new insight into the divergence of catalysis within the PEP mutase/isocitrate lyase superfamily. *Biochemistry*, 45(38):11491–11504.

Chen, H., Liu, B., Huang, H., Hwang, S., and Ho, S. (2007). SODOCK: Swarm optimization for highly flexible protein-ligand docking. *Journal of Computational Chemistry*, 28:612–623.

Cho, Y., Vermeire, J. J., Merkel, J. S., Leng, L., Du, X., Bucala, R., Cappello, M., and Lolis, E. (2011). Drug repositioning and pharmacophore identification in the discovery of hookworm MIF inhibitors. *Chemistry & Biology*, 18(9):1089–1101.

Chothia, C. and Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. *The EMBO journal*, 5(4):823–826.

Chu, M., Lang, Z., Chavas, L., and Neres, J. (2010). Biophysical and X-ray crystallographic analysis of Mps1 kinase inhibitor complexes. *Biochemistry*.

Chupakhin, V., Marcou, G., Baskin, I., Varnek, A., and Rognan, D. (2013). Predicting ligand binding modes from neural networks trained on protein-ligand interaction fingerprints. *Journal Of Chemical Information And Modeling*, 53(4):763–772.

Clark, R. D. and Webster-Clark, D. J. (2008). Managing bias in ROC curves. *Journal of computer-aided molecular design*, 22(3-4):141–146.

Combs, S., Kaufmann, K., Field, J. R., Blakely, R. D., and Meiler, J. (2011). Y95 and E444 Interaction Required for High-Affinity S-Citalopram Binding in the Human Serotonin Transporter. *ACS chemical neuroscience*, 2(2):75–81.

Combs, S. A. (2013). *Identification and Scoring of Partial Covalent Interactions in Proteins and Protein-Ligand Complexes*. PhD thesis, Vanderbilt University.

Combs, S. A., Deluca, S. L., DeLuca, S. H., Lemmon, G. H., Nannemann, D. P., Nguyen, E. D., Willis, J. R., Sheehan, J. H., and Meiler, J. (2013). Small-molecule ligand docking into comparative models with Rosetta. *Nature Protocols*, 8(7):1277–1298.

Crepin, T., Schmitt, E., Mechulam, Y., Sampson, P. B., Vaughan, M. D., Honek, J. F., and Blanquet, S. (2003). Use of analogues of methionine and methionyl adenylate to sample conformational changes during catalysis in Escherichia coli methionyl-tRNA synthetase. *Journal of molecular biology*, 332(1):59–72.

Cuneo, M. J., Changela, A., Warren, J. J., Beese, L. S., and Hellinga, H. W. (2006). The crystal structure of a thermophilic glucose binding protein reveals adaptations that interconvert mono and di-saccharide binding sites. *Journal of molecular biology*, 362(2):259–270.

Dantas, G., Corrent, C., Reichow, S. L., Havranek, J. J., Eletr, Z. M., Isern, N. G., Kuhlman, B., Varani, G., Merritt, E. A., and Baker, D. (2007). High-resolution Structural and Thermodynamic Analysis of Extreme Stabilization of Human Procarboxypeptidase by Computational Protein Design. *Journal of molecular biology*, 366(4):1209–1221.

Dantas, G., Kuhlman, B., Callender, D., Wong, M., and Baker, D. (2003). A large scale test of computational protein design: folding and stability of nine completely redesigned globular proteins. *Journal of molecular biology*, 332(2):449–460.

Das, R. and Baker, D. (2008). Macromolecular Modeling with Rosetta. *Annual Review of Biochemistry*, 77(1):363–382.

Davis, I. W. and Baker, D. (2009). RosettaLigand Docking with Full Ligand and Receptor Flexibility. *Journal of molecular biology*, 385(2):381–392.

Davis, I. W., Raha, K., Head, M. S., and Baker, D. (2009). Blind docking of pharmaceutically relevant compounds using RosettaLigand. *Protein Science*, 18(9):1998–2002.

Dean, T., Ruzon, M. A., Segal, M., Shlens, J., Vijayanarasimhan, S., and Yagnik, J. (2013). Fast, Accurate Detection of 100,000 Object Classes on a Single Machine.

Deka, R. K., Brautigam, C. A., Yang, X. F., Blevins, J. S., Machius, M., Tomchick, D. R., and Norgard, M. V. (2006). The PnrA (Tp0319; TmpC) lipoprotein represents a new family of bacterial purine nucleoside receptor encoded within an ATP-binding cassette (ABC)-like operon in Treponema pallidum.

DeLuca, S., Dorr, B., and Meiler, J. (2011). Design of native-like proteins through an exposure-dependent environment potential. *Biochemistry*, 50(40):8521–8528.

Deng, Z., Chuaqui, C., and Singh, J. (2004). Structural interaction fingerprint (SIFt): a novel method for analyzing three-dimensional protein-ligand binding interactions. *Journal of medicinal chemistry*, 47(2):337–344.

Devesse, L., Smirnova, I., Lönneborg, R., Kapp, U., Brzezinski, P., Leonard, G. A., and Dian, C. (2011). Crystal structures of DntR inducer binding domains in complex with salicylate offer insights into the activation of LysR-type transcriptional regulators. *Molecular microbiology*, 81(2):354–367.

DiMaio, F., Leaver-Fay, A., Bradley, P., Baker, D., and André, I. (2011). Modeling symmetric macromolecular structures in Rosetta3. *PloS one*, 6(6):e20450.

Done, S. H., Brannigan, J. A., Moody, P. C., and Hubbard, R. E. (1998). Ligand-induced conformational change in penicillin acylase. *Journal of molecular biology*, 284(2):463–475.

Dranchak, P., MacArthur, R., Guha, R., Zuercher, W. J., Drewry, D. H., Auld, D. S., and Inglese, J. (2013). Profile of the GSK published protein kinase inhibitor set across ATP-dependent and-independent luciferases: implications for reporter-gene assays. *PloS one*, 8(3):e57888.

Dunbar Jr., J. B., Smith, R. D., Yang, C.-Y., Ung, P. M.-U., Lexa, K. W., Khazanov, N. A., Stuckey, J. A., Wang, S., and Carlson, H. A. (2011). CSAR Benchmark Exercise of 2010: Selection of the Protein–Ligand Complexes. *Journal Of Chemical Information And Modeling*, 51(9):2036–2046.

Dunbrack Jr, R. L. and Karplus, M. (1993). Backbone-dependent Rotamer Library for Proteins Application to Side-chain Prediction. *Journal of molecular biology*, 230(2):543–574.

Durham, E., Dorr, B., Woetzel, N., Staritzbichler, R., and Meiler, J. (2009). Solvent accessible surface area approximations for rapid and accurate protein structure prediction. *Journal of molecular modeling*, 15(9):1093–1108.

Durrant, J. D., Friedman, A. J., Rogers, K. E., and Mccammon, J. A. (2013). Comparing Neural-Network Scoring Functions and the State of the Art: Applications to Common Library Screening. *Journal Of Chemical Information And Modeling*, 53(7):1726–1735.

Durrant, J. D. and Mccammon, J. A. (2010). NNScore: A Neural-Network-Based Scoring Function for the Characterization of Protein–Ligand Complexes. *Journal Of Chemical Information And Modeling*, 50(10):1865–1871.

Durrant, J. D. and Mccammon, J. A. (2011). NNScore 2.0: a neural-network receptor-ligand scoring function. *Journal Of Chemical Information And Modeling*, 51(11):2897–2903.

Dvir, H., Jiang, H. L., Wong, D. M., Harel, M., Chetrit, M., He, X. C., Jin, G. Y., Yu, G. L., Tang, X. C., Silman, I., Bai, D. L., and Sussman, J. L. (2002). X-ray structures of Torpedo californica acetylcholinesterase complexed with (+)-huperzine A and (-)-huperzine B: structural evidence for an active site rearrangement. *Biochemistry*, 41(35):10810–10818.

Eldridge, M. D., Murray, C. W., Auton, T. R., Paolini, G. V., and Mee, R. P. (1997). Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of computer-aided molecular design*, 11(5):425–445.

Eser, S., Schnieke, A., Schneider, G., and Saur, D. (2014). Oncogenic KRAS signalling in pancreatic cancer. *British Journal of Cancer*, 111(5):817–822.

Ewing, T. J., Makino, S., Skillman, A. G., and Kuntz, I. D. (2001). DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. *Journal of computer-aided molecular design*, 15(5):411–428.

Feder, D., Hussein, W. M., Clayton, D. J., Kan, M. W., Schenk, G., McGeary, R. P., and Guddat, L. W. (2012). Identification of purple acid phosphatase inhibitors by fragment-based screening: promising new leads for osteoporosis therapeutics. *Chemical Biology & Drug Design*, 80(5):665–674.

Feldkamp, M. D., Frank, A. O., Kennedy, J. P., Patrone, J. D., Vangamudi, B., Waterson, A. G., Fesik, S. W., and Chazin, W. J. (2013). Surface reengineering of RPA70N enables cocrystallization with an inhibitor of the replication protein A interaction motif of ATR interacting protein. *Biochemistry*, 52(37):6515–6524.

Ferrara, P., Gohlke, H., Price, D. J., Klebe, G., and Brooks, C. L. (2004). Assessing scoring functions for protein-ligand interactions. *Journal of medicinal chemistry*, 47(12):3032–3047.

Fleishman, S. J., Leaver-Fay, A., Corn, J. E., Strauch, E.-M., Khare, S. D., Koga, N., Ashworth, J., Murphy, P., Richter, F., Lemmon, G., Meiler, J., and Baker, D. (2011). RosettaScripts: a scripting language interface to the Rosetta macromolecular modeling suite. *PloS one*, 6(6):e20161.

Fortenberry, C., Bowman, E. A., Proffitt, W., Dorr, B., Combs, S., Harp, J., Mizoue, L., and Meiler, J. (2011). Exploring symmetry as an avenue to the computational design of large protein domains. *Journal of the American Chemical Society*, 133(45):18026–18029.

Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P., and Shenkin, P. S. (2004). Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *Journal of medicinal chemistry*, 47(7):1739–1749.

Fushinobu, S., Jun, S.-Y., Hidaka, M., Nojiri, H., Yamane, H., Shoun, H., Omori, T., and Wakagi, T. (2005). A series of crystal structures of a meta-cleavage product hydrolase from Pseudomonas fluorescens IP01 (CumD) complexed with various cleavage products. *Bioscience, biotechnology, and biochemistry*, 69(3):491–498.

Gajiwala, K. S., Wu, J. C., Christensen, J., Deshmukh, G. D., Diehl, W., DiNitto, J. P., English, J. M., Greig, M. J., He, Y.-A., Jacques, S. L., Lunney, E. A., McTigue, M., Molina, D., Quenzer, T., Wells, P. A., Yu, X., Zhang, Y., Zou, A., Emmett, M. R., Marshall, A. G., Zhang, H.-M., and Demetri, G. D. (2009). KIT kinase mutants show unique mechanisms of drug resistance to imatinib and sunitinib in gastrointestinal stromal tumor patients. *Proceedings of the National Academy of Sciences*, 106(5):1542–1547.

Gallant, S. I. (1990). Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks*, 1(2):179–191.

García-Sáez, I., Reverter, D., Vendrell, J., Avilés, F. X., and Coll, M. (1997). The three-dimensional structure of human procarboxypeptidase A2. Deciphering the basis of the inhibition, activation and intrinsic activity of the zymogen. *The EMBO journal*, 16(23):6906–6913.

Gardiner, E. J., Holliday, J. D., O'Dowd, C., and Willett, P. (2011). Effectiveness of 2D fingerprints for scaffold hopping. *dx.doi.org.proxy.library.vanderbilt.edu*, 3(4):405–414.

Gasteiger, J. and Hutchings, M. G. (1983). New empirical models of substituent polarisability and their application to stabilisation effects in positively charged species. *Tetrahedron Letters*.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., and Overington, J. P. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(Database issue):D1100–7.

Ghosh, M., Meerts, I. A., Cook, A., Bergman, A., Brouwer, A., and Johnson, L. N. (2000). Structure of human transthyretin complexed with bromophenols: a new mode of binding. *Acta crystallographica Section D, Biological crystallography*, 56(Pt 9):1085–1095.

Glanzer, J. G., Liu, S., Wang, L., Mosel, A., Peng, A., and Oakley, G. G. (2014). RPA inhibition increases replication stress and suppresses tumor growth. *Cancer Research*, 74(18):5165–5172.

Gloster, T. M., Meloncelli, P., Stick, R. V., Zechel, D., Vasella, A., and Davies, G. J. (2007). Glycosidase inhibition: an assessment of the binding of 18 putative transition-state mimics. *Journal of the American Chemical Society*, 129(8):2345–2354.

Gloster, T. M., Williams, S. J., Roberts, S., Tarling, C. A., Wicki, J., Withers, S. G., and Davies, G. J. (2004). Atomic resolution analyses of the binding of xylobiose-derived deoxynojirimycin and isofagomine to xylanase Xyn10A. *Chemical communications (Cambridge, England)*, (16):1794–1795.

Gohlke, H. and Klebe, G. (2002). Approaches to the description and prediction of the binding affinity of small-molecule ligands to macromolecular receptors. *Angewandte Chemie (International ed. in English)*, 41(15):2644–2676.

Gonin, S., Arnoux, P., Pierru, B., Lavergne, J., Alonso, B., Sabaty, M., and Pignol, D. (2007). Crystal structures of an Extracytoplasmic Solute Receptor from a TRAP transporter in its open and closed forms reveal a helix-swapped dimer requiring a cation for alpha-keto acid binding. *BMC Structural Biology*, 7:11.

Goodford, P. J. (1985). A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of medicinal chemistry*, 28(7):849–857.

Grädler, U., Gerber, H. D., Goodenough-Lashua, D. M., Garcia, G. A., Ficner, R., Reuter, K., Stubbs, M. T., and Klebe, G. (2001). A new target for shigellosis: rational design and crystallographic studies of inhibitors of tRNA-guanine transglycosylase. *Journal of molecular biology*, 306(3):455–467.

Gray, J. J., Moughon, S., Wang, C., Schueler-Furman, O., Kuhlman, B., Rohl, C. A., and Baker, D. (2003). Protein–Protein Docking with Simultaneous Optimization of Rigid-body Displacement and Side-chain Conformations. *Journal of molecular biology*, 331(1):281–299.

Haapalainen, A. M., van Aalten, D. M., Meriläinen, G., Jalonen, J. E., Pirilä, P., Wierenga, R. K., Hiltunen, J. K., and Glumoff, T. (2001). Crystal structure of the liganded SCP-2-like domain of human peroxisomal multifunctional enzyme type 2 at 1.75 A resolution. *Journal of molecular biology*, 313(5):1127–1138.

Halperin, I., Glazer, D. S., Wu, S., and Altman, R. B. (2008). The FEATURE framework for protein function annotation: modeling new functions, improving performance, and extending to novel applications. *BMC genomics*, 9(Suppl 2):S2.

Hamelryck, T. (2005). An amino acid has two sides: a new 2D measure provides a different view of solvent exposure. *Proteins: Structure, Function, and Bioinformatics*, 59(1):38–48.

Hanekop, N., Höing, M., Sohn-Bösser, L., Jebbar, M., Schmitt, L., and Bremer, E. (2007). Crystal structure of the ligand-binding protein EhuB from Sinorhizobium meliloti reveals substrate recognition of the compatible solutes ectoine and hydroxyectoine. *Journal of molecular biology*, 374(5):1237–1250.

Hass, C. S., Gakhar, L., and Wold, M. S. (2010). Functional characterization of a cancer causing mutation in human replication protein A. *Molecular cancer research : MCR*, 8(7):1017–1026.

Hindle, S. A., Rarey, M., Buning, C., and Lengauer, T. (2002). Flexible docking under pharmacophore type constraints. *Journal of computer-aided molecular design*, 16(2):129–149.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv.org*.

Horsfall, L. E., Garau, G., Liénard, B. M. R., Dideberg, O., Schofield, C. J., Frère, J. M., and Galleni, M. (2007). Competitive inhibitors of the CphA metallo-beta-lactamase from Aeromonas hydrophila. *Antimicrobial agents and chemotherapy*, 51(6):2136–2142.

Hoseki, J., Okamoto, A., Masui, R., Shibata, T., Inoue, Y., Yokoyama, S., and Kuramitsu, S. (2003). Crystal structure of a family 4 uracil-DNA glycosylase from Thermus thermophilus HB8. *Journal of molecular biology*, 333(3):515–526.

Hothi, P., Roujeinikova, A., Khadra, K. A., Lee, M., Cullis, P., Leys, D., and Scrutton, N. S. (2007). Isotope effects reveal that para-substituted benzylamines are poor reactivity probes of the quinoprotein mechanism for aromatic amine dehydrogenase. *Biochemistry*, 46(32):9250–9259.

Hristozov, D. P., Oprea, T. I., and Gasteiger, J. (2007). Virtual screening applications: a study of ligand-based methods and different structure representations in four different scenarios. *Journal of computer-aided molecular design*, 21(10-11):617–640.

Hsieh, R. W., Rajan, S. S., Sharma, S. K., Guo, Y., DeSombre, E. R., Mrksich, M., and Greene, G. L. (2006). Identification of ligands with bicyclic scaffolds provides insights into mechanisms of estrogen receptor subtype selectivity. *The Journal of biological chemistry*, 281(26):17909–17919.

Hu, G., Gershon, P. D., Hodel, A. E., and Quiocho, F. A. (1999). mRNA cap recognition: dominant role of enhanced stacking interactions between methylated bases and protein aromatic side chains. *Proceedings of the National Academy of Sciences of the United States of America*, 96(13):7149–7154.

Huang, K.-F., Liu, Y.-L., Cheng, W.-J., Ko, T.-P., and Wang, A. H. J. (2005). Crystal structures of human glutaminyl cyclase, an enzyme responsible for protein N-terminal pyroglutamate formation. *Proceedings of the National Academy of Sciences of the United States of America*, 102(37):13117–13122.

Huang, N., Shoichet, B. K., and Irwin, J. J. (2006). Benchmarking sets for molecular docking. *Journal of medicinal chemistry*, 49(23):6789–6801.

Hung, A. W., Silvestre, H. L., Wen, S., Ciulli, A., Blundell, T. L., and Abell, C. (2009). Application of fragment growing and fragment linking to the discovery of inhibitors of Mycobacterium tuberculosis pantothenate synthetase. *Angewandte Chemie International Edition*, 48(45):8452–8456.

Hutchings, M. G. and Gasteiger, J. (1983). Residual electronegativity-an empirical quantification of polar influences and its application to the proton affinity of amines. *Tetrahedron Letters*, 24(25):2541–2544.

Hyre, D. E., Le Trong, I., Freitag, S., Stenkamp, R. E., and Stayton, P. S. (2000). Ser45 plays an important role in managing both the equilibrium and transition state energetics of the streptavidin-biotin system. *Protein Science*, 9(5):878–885.

Inanobe, A., Furukawa, H., and Gouaux, E. (2005). Mechanism of partial agonist action at the NR1 subunit of NMDA receptors. *Neuron*, 47(1):71–84.

Inoue, T., Ito, K., Tozaka, T., Hatakeyama, S., Tanaka, N., Nakamura, K. T., and Yoshimoto, T. (2003). Novel inhibitor for prolyl aminopeptidase from Serratia marcescens and studies on the mechanism of substrate recognition of the enzyme using the inhibitor. *Archives of biochemistry and biophysics*, 416(2):147–154.

Irwin, J. J. (2008). Community benchmarks for virtual screening. *Journal of computer-aided molecular design*, 22(3-4):193–199.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. (2012). ZINC: A Free Tool to Discover Chemistry for Biology. *Journal Of Chemical Information And Modeling*, 52(7):1757–1768.

Jiang, L., Althoff, E. A., Clemente, F. R., Doyle, L., Rothlisberger, D., Zanghellini, A., Gallaher, J. L., Betker, J. L., Tanaka, F., Barbas, C. F., Hilvert, D., Houk, K. N., Stoddard,

B. L., and Baker, D. (2008). De Novo Computational Design of Retro-Aldol Enzymes. *Science*, 319(5868):1387–1391.

Jones, G., Willett, P., and Glen, R. (1995). Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. *Journal of molecular biology*, 245:43–53.

Jones, G., Willett, P., Glen, R. C., Leach, A. R., and Taylor, R. (1997). Development and validation of a genetic algorithm for flexible docking. *Journal of molecular biology*, 267(3):727–748.

Karlberg, T., Hammarström, M., Schütz, P., Svensson, L., and Schüler, H. (2010a). Crystal structure of the catalytic domain of human PARP2 in complex with PARP inhibitor ABT-888. *Biochemistry*, 49(6):1056–1058.

Karlberg, T., Markova, N., Johansson, I., Hammarström, M., Schütz, P., Weigelt, J., and Schüler, H. (2010b). Structural basis for the interaction between tankyrase-2 and a potent Wnt-signaling inhibitor. *Journal of medicinal chemistry*, 53(14):5352–5355.

Kaufmann, K. W., Lemmon, G. H., Deluca, S. L., Sheehan, J. H., and Meiler, J. (2010). Practically Useful: What the R osettaProtein Modeling Suite Can Do for You. *Biochemistry*, 49(14):2987–2998.

Kaufmann, K. W. and Meiler, J. (2012). Using RosettaLigand for Small Molecule Docking into Comparative Models. *PloS one*, 7(12):e50769.

Khersonsky, O., Kiss, G., Röthlisberger, D., Dym, O., Albeck, S., Houk, K. N., Baker, D., and Tawfik, D. S. (2012). Bridging the gaps in design methodologies by evolutionary optimization of the stability and proficiency of designed Kemp eliminase KE59. *Proceedings of the National Academy of Sciences*, 109(26):10358–10363.

Klaholz, B. P., Mitschler, A., and Moras, D. (2000). Structural basis for isotype selectivity of the human retinoic acid nuclear receptor. *Journal of molecular biology*, 302(1):155–170.

Klieber, M. A., Underhill, C., Hammond, G. L., and Muller, Y. A. (2007). Corticosteroid-binding globulin, a structural basis for steroid transport and proteinase-triggered release. *The Journal of biological chemistry*, 282(40):29594–29603.

Kong, Y.-h., Zhang, L., Yang, Z.-y., Han, C., Hu, L.-h., Jiang, H.-l., and Shen, X. (2008). Natural product juglone targets three key enzymes from Helicobacter pylori: inhibition assay with crystal structure characterization. *Acta pharmacologica Sinica*, 29(7):870–876.

Korkegian, A. (2005). Computational Thermostabilization of an Enzyme. *Science*, 308(5723):857–860.

Korndörfer, I. P., Schlehuber, S., and Skerra, A. (2003). Structural mechanism of specific ligand recognition by a lipocalin tailored for the complexation of digoxigenin. *Journal of molecular biology*, 330(2):385–396.

Kortemme, T., Joachimiak, L. A., Bullock, A. N., Schuler, A. D., Stoddard, B. L., and Baker, D. (2004). Computational redesign of protein-protein interaction specificity. *Nature Structural &#38; Molecular Biology*, 11(4):371–379.

Kortemme, T., Morozov, A. V., and Baker, D. (2003). An Orientation-dependent Hydrogen Bonding Potential Improves Prediction of Specificity and Structure for Proteins and Protein–Protein Complexes. *Journal of molecular biology*, 326(4):1239–1259.

Koshland, D. E. (1958). Application of a Theory of Enzyme Specificity to Protein Synthesis. *Proceedings of the National Academy of Sciences of the United States of America*, 44(2):98–104.

Kranenburg, O. (2005). The KRAS oncogene: Past, present, and future. *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*, 1756(2):81–82.

Kratz, J. M., Schuster, D., Edtbauer, M., Saxena, P., Mair, C. E., Kirchebner, J., Matuszczak, B., Baburin, I., Hering, S., and Rollinger, J. M. (2014). Experimentally Validated hERG Pharmacophore Models as Cardiotoxicity Prediction Tools. *Journal Of Chemical Information And Modeling*.

Kraut, D. A., Sigala, P. A., Pybus, B., Liu, C. W., Ringe, D., Petsko, G. A., and Herschlag, D. (2006). Testing electrostatic complementarity in enzyme catalysis: hydrogen bonding in the ketosteroid isomerase oxyanion hole. *PLoS biology*, 4(4):e99.

Kuhlman, B. (2003). Design of a Novel Globular Protein Fold with Atomic-Level Accuracy. *Science*, 302(5649):1364–1368.

Kuhlman, B. and Baker, D. (2000). Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences of the United States of America*, 97(19):10383–10388.

Kuntz, I. D., Blaney, J. M., Oatley, S. J., Langridge, R., and Ferrin, T. E. (1982). A geometric approach to macromolecule-ligand interactions. *Journal of molecular biology*, 161(2):269–288.

Kurnik, M., Hedberg, L., Danielsson, J., and Oliveberg, M. (2012). Folding without charges. *Proceedings of the National Academy of Sciences*, 109(15):5705–5710.

Labute, P. (2010). LowModeMD–implicit low-mode velocity filtering applied to conformational search of macrocycles and protein loops. *Journal Of Chemical Information And Modeling*, 50(5):792–800.

LaLonde, J. M., Kwon, Y. D., Jones, D. M., Sun, A. W., Courter, J. R., Soeta, T., Kobayashi, T., Princiotto, A. M., Wu, X., Schön, A., Freire, E., Kwong, P. D., Mascola, J. R., Sodroski, J., Madani, N., and Smith, A. B. (2012). Structure-based design, synthesis, and characterization of dual hotspot small-molecule HIV-1 entry inhibitors. *Journal of medicinal chemistry*, 55(9):4382–4396.

Larsen, N. A., Zhou, B., Heine, A., Wirsching, P., Janda, K. D., and Wilson, I. A. (2001). Crystal structure of a cocaine-binding antibody. *Journal of molecular biology*, 311(1):9–15.

Lartigue, A. (2003). The crystal structure of a cockroach pheromone-binding protein suggests a new ligand binding and release mechanism. *The Journal of biological chemistry*, 278(32):30213–30218.

Lassaux, P., Hamel, M., Gulea, M., Delbrück, H., Mercuri, P. S., Horsfall, L., Dehareng, D., Kupper, M., Frère, J.-M., Hoffmann, K., Galleni, M., and Bebrone, C. (2010). Mercaptophosphonate compounds as broad-spectrum inhibitors of the metallo-beta-lactamases. *Journal of medicinal chemistry*, 53(13):4862–4876.

Lawrence, M. C. and Colman, P. M. (1993). Shape complementarity at protein/protein interfaces. *Journal of molecular biology*, 234(4):946–950.

Lawrence, M. S., Philips, K. J., and Liu, D. R. (2007). *Supercharging Proteins Can Impart Unusual Resilience*, volume 129. American Chemical Society.

Lazaridis, T. and Karplus, M. (1999). Effective energy function for proteins in solution. *Proteins: Structure, Function, and Bioinformatics*, 35(2):133–152.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE.

Leaver-Fay, A., Kuhlman, B., and Snoeyink, J. (2005). Rotamer-pair energy calculations using a trie data structure. *Lecture Notes in Computer Science*.

Leaver-Fay, A., O'Meara, M. J., Tyka, M., Jacak, R., Song, Y., Kellogg, E. H., Thompson, J., Davis, I. W., Pache, R. A., Lyskov, S., Gray, J. J., Kortemme, T., Richardson, J. S., Havranek, J. J., Snoeyink, J., Baker, D., and Kuhlman, B. (2013). Scientific benchmarks for guiding macromolecular energy function improvement. *Methods in enzymology*, 523:109–143.

Lee, J. E., Cornell, K. A., Riscoe, M. K., and Howell, P. L. (2001). Structure of E. coli 5'-methylthioadenosine/S-adenosylhomocysteine nucleosidase reveals similarity to the purine nucleoside phosphorylases. *Structure (London, England : 1993)*, 9(10):941–953.

Lee, S. G., Kim, Y., Alpert, T. D., Nagata, A., and Jez, J. M. (2012). Structure and reaction mechanism of phosphoethanolamine methyltransferase from the malaria parasite Plasmodium falciparum: an antiparasitic drug target. *The Journal of biological chemistry*, 287(2):1426–1434.

Lemmon, G., Kaufmann, K., and Meiler, J. (2012). Prediction of HIV-1 protease/inhibitor affinity using RosettaLigand. *Chemical Biology & Drug Design*, 79(6):888–896.

Lemmon, G. and Meiler, J. (2013). Towards ligand docking including explicit interface water molecules. *PloS one*, 8(6):e67536.

Levinthal, C. (1968). Are there pathways for protein folding. *Journal de Chimie Physique*, 65:44.

Lewis, K. (2013). Platforms for antibiotic discovery. *Nature reviews Drug discovery*, 12(5):371–387.

Liang, G., Aldous, S., Merriman, G., Levell, J., Pribish, J., Cairns, J., Chen, X., Maignan, S., Mathieu, M., Tsay, J., Sides, K., Rebello, S., Whitely, B., Morize, I., and Pauls, H. W. (2012a). Structure-based library design and the discovery of a potent and selective mast cell β-tryptase inhibitor as an oral therapeutic agent. *Bioorganic & Medicinal Chemistry Letters*, 22(2):1049–1054.

Liang, G., Choi-Sledeski, Y. M., Shum, P., Chen, X., Poli, G. B., Kumar, V., Minnich, A., Wang, Q., Tsay, J., Sides, K., Kang, J., and Zhang, Y. (2012b). A β-tryptase inhibitor with a tropanylamide scaffold to improve in vitro stability and to lower hERG channel binding affinity. *Bioorganic & Medicinal Chemistry Letters*, 22(4):1606–1610.

Lietzan, A. D., Nagar, M., Pellmann, E. A., and Bourque, J. R. (2012). Structure of mandelate racemase with bound intermediate analogues benzohydroxamate and cupferron. *Biochemistry*, 51(6):1160–1170.

Lim, K., Owens, S. M., Arnold, L., Sacchettini, J. C., and Linthicum, D. S. (1998). Crystal structure of monoclonal 6B5 Fab complexed with phencyclidine. *The Journal of biological chemistry*, 273(44):28576–28582.

Lin, F.-Y., Liu, C.-I., Liu, Y.-L., Zhang, Y., Wang, K., Jeng, W.-Y., Ko, T.-P., Cao, R., Wang, A. H. J., and Oldfield, E. (2010). Mechanism of action and inhibition of dehydrosqualene synthase. *Proceedings of the National Academy of Sciences*, 107(50):21337–21342.

Lindsey, J. K. (1997). *Applying Generalized Linear Models*. Springer.

Lipinski, C. A., Lombardo, F., Dominy, B. W., and Feeney, P. J. (2001). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 46(1-3):3–26.

Liu, S., Chang, J. S., Herberg, J. T., Horng, M.-M., Tomich, P. K., Lin, A. H., and Marotti, K. R. (2006). Allosteric inhibition of Staphylococcus aureus D-alanine:D-alanine ligase revealed by crystallographic studies. *Proceedings of the National Academy of Sciences of the United States of America*, 103(41):15178–15183.

Lluis, M., Wang, Y., Monzingo, A. F., Fast, W., and Robertus, J. D. (2011). Characterization of C-alkyl amidines as bioavailable covalent reversible inhibitors of human DDAH-1. *ChemMedChem*, 6(1):81–88.

Loch, J., Polit, A., Górecki, A., Bonarek, P., Kurpiewska, K., Dziedzicka Wasylewska, M., and Lewiński, K. (2011). Two modes of fatty acid binding to bovine β-lactoglobulin–crystallographic and spectroscopic studies. *Journal of Molecular Recognition*, 24(2):341–349.

Lu, D., Bernstein, D. A., Satyshur, K. A., and Keck, J. L. (2010). Small-molecule tools for dissecting the roles of SSB/protein interactions in genome maintenance. *Proceedings of the National Academy of Sciences*, 107(2):633–638.

Magalhães, M. L. B., Czekster, C. M., Guan, R., Malashkevich, V. N., Almo, S. C., and Levy, M. (2011). Evolved streptavidin mutants reveal key role of loop residue in high-affinity binding. *Protein Science*, 20(7):1145–1154.

Mangani, S., Carloni, P., and Orioli, P. (1992). Crystal structure of the complex between carboxypeptidase A and the biproduct analog inhibitor L-benzylsuccinate at 2.0 A resolution. *Journal of molecular biology*, 223(2):573–578.

Mark, B. L., Vocadlo, D. J., Knapp, S., Triggs-Raine, B. L., Withers, S. G., and James, M. N. (2001). Crystallographic evidence for substrate-assisted catalysis in a bacterial beta-hexosaminidase. *The Journal of biological chemistry*, 276(13):10330–10337.

Martin, J. L., Begun, J., McLeish, M. J., Caine, J. M., and Grunewald, G. L. (2001). Getting the adrenaline going: crystal structure of the adrenaline-synthesizing enzyme PNMT. *Structure (London, England : 1993)*, 9(10):977–985.

Matera, I., Ferraroni, M., Kolomytseva, M., Golovleva, L., Scozzafava, A., and Briganti, F. (2010). Catechol 1,2-dioxygenase from the Gram-positive Rhodococcus opacus 1CP: quantitative structure/activity relationship and the crystal structures of native enzyme and catechols adducts. *Journal of structural biology*, 170(3):548–564.

Mathews, I. I., Vanderhoff-Hanaver, P., Castellino, F. J., and Tulinsky, A. (1996). Crystal structures of the recombinant kringle 1 domain of human plasminogen in complexes with the ligands epsilon-aminocaproic acid and trans-4-(aminomethyl)cyclohexane-1-carboxylic Acid. *Biochemistry*, 35(8):2567–2576.

Mayer, M. L. (2005). Crystal structures of the GluR5 and GluR6 ligand binding cores: molecular mechanisms underlying kainate receptor selectivity. *Neuron*, 45(4):539–552.

McGowan, S., Porter, C. J., Lowther, J., Stack, C. M., Golding, S. J., Skinner-Adams, T. S., Trenholme, K. R., Teuscher, F., Donnelly, S. M., Grembecka, J., Mucha, A., Kafarski, P., DeGori, R., Buckle, A. M., Gardiner, D. L., Whisstock, J. C., and Dalton, J. P. (2009). Structural basis for the inhibition of the essential Plasmodium falciparum M1 neutral aminopeptidase. *Proceedings of the National Academy of Sciences*, 106(8):2537–2542.

McGrath, M. E., Sprengeler, P. A., Hirschbein, B., Somoza, J. R., Lehoux, I., Janc, J. W., Gjerstad, E., Graupe, M., Estiarte, A., Venkataramani, C., Liu, Y., Yee, R., Ho, J. D., Green, M. J., Lee, C.-S., Liu, L., Tai, V., Spencer, J., Sperandio, D., and Katz, B. A. (2006). Structure-guided design of peptide-based tryptase inhibitors. *Biochemistry*, 45(19):5964–5973.

Meiler, J. and Baker, D. (2006). ROSETTALIGAND: Protein-small molecule docking with full side-chain flexibility. *Proteins: Structure, Function, and Bioinformatics*, 65(3):538–548.

Meyer, E. A., Furler, M., Diederich, F., Brenk, R., and Klebe, G. (2004). Synthesis and In Vitro Evaluation of 2-Aminoquinazolin-4(3H)-one-Based Inhibitors for tRNA-Guanine Transglycosylase (TGT). *Helvetica Chimica Acta*, 87(6):1333–1356.

Miller, J. R., Dunham, S., Mochalkin, I., Banotai, C., Bowman, M., Buist, S., Dunkle, B., Hanna, D., Harwood, H. J., Huband, M. D., Karnovsky, A., Kuhn, M., Limberakis, C., Liu, J. Y., Mehrens, S., Mueller, W. T., Narasimhan, L., Ogden, A., Ohren, J., Prasad, J. V. N. V., Shelly, J. A., Skerlos, L., Sulavik, M., Thomas, V. H., VanderRoest, S., Wang, L., Wang, Z., Whitton, A., Zhu, T., and Stover, C. K. (2009). A class of selective antibacterials derived from a protein kinase inhibitor pharmacophore. *Proceedings of the National Academy of Sciences*, 106(6):1737–1742.

Miyazono, K.-i., Miyakawa, T., Sawano, Y., Kubota, K., Kang, H.-J., Asano, A., Miyauchi, Y., Takahashi, M., Zhi, Y., Fujita, Y., Yoshida, T., Kodaira, K.-S., Yamaguchi-Shinozaki, K., and Tanokura, M. (2009). Structural basis of abscisic acid signalling. *Nature*, 462(7273):609–614.

Mobley, D. L., Graves, A. P., Chodera, J. D., McReynolds, A. C., Shoichet, B. K., and Dill, K. A. (2007). Predicting absolute ligand binding free energies to a simple model site. *Journal of molecular biology*, 371(4):1118–1134.

Mochalkin, I., Cheng, B., Klezovitch, O., and Scanu, A. M. (1999). Recombinant kringle IV-10 modules of human apolipoprotein (a): Structure, ligand binding modes, and biological relevance. *Biochemistry*, 38(7):1990–1998.

Molecular Networks GmbH Computerchemie (2011). *Descriptors of ADRIANA. Code*. ADRIANA.Code.

Morgunova, E., Illarionov, B., Sambaiah, T., Haase, I., Bacher, A., Cushman, M., Fischer, M., and Ladenstein, R. (2006). Structural and thermodynamic insights into the binding mode of five novel inhibitors of lumazine synthase from Mycobacterium tuberculosis. *The FEBS journal*, 273(20):4790–4804.

Morris, G., Goodsell, D., Halliday, R., Huey, R., Hart, W. E., Belew, R. K., and Olson, A. J. (1998). Automated docking using a Lamarckian genetic algorithm and an empirical binding free Energy Function. *Journal of Computational Chemistry*, 19(14):1639–1662.

Moustakas, D. T., Lang, P. T., Pegg, S., Pettersen, E., Kuntz, I. D., Brooijmans, N., and Rizzo, R. C. (2006). Development and validation of a modular, extensible docking program: DOCK 5. *Journal of computer-aided molecular design*, 20(10-11):601–619.

Moynié, L., Leckie, S. M., McMahon, S. A., Duthie, F. G., Koehnke, A., Taylor, J. W., Alphey, M. S., Brenk, R., Smith, A. D., and Naismith, J. H. (2013). Structural insights into the mechanism and inhibition of the β-hydroxydecanoyl-acyl carrier protein dehydratase from Pseudomonas aeruginosa. *Journal of molecular biology*, 425(2):365–377.

Mueller, R., Dawson, E. S., Niswender, C. M., Butkiewicz, M., Hopkins, C. R., Weaver, C. D., Lindsley, C. W., Conn, P. J., and Meiler, J. (2012). Iterative experimental and virtual high-throughput screening identifies metabotropic glutamate receptor subtype 4 positive allosteric modulators. *Journal of molecular modeling*, 18(9):4437–4446.

Mueller, R., Rodriguez, A. L., Dawson, E. S., Butkiewicz, M., Nguyen, T. T., Oleszkiewicz, S., Bleckmann, A., Weaver, C. D., Lindsley, C. W., Conn, P. J., and Meiler, J. (2010). Identification of Metabotropic Glutamate Receptor Subtype 5 Potentiators Using Virtual High-Throughput Screening. *ACS chemical neuroscience*, 1(4):288–305.

Müller, A., León-Kempis, M. d. R., Dodson, E., Wilson, K. S., Wilkinson, A. J., and Kelly, D. J. (2007). A bacterial virulence factor with a dual role as an adhesin and a solute-binding protein: the crystal structure at 1.5 A resolution of the PEB1a protein from the food-borne human pathogen Campylobacter jejuni. *Journal of molecular biology*, 372(1):160–171.

Myint, K. Z., Wang, L., Tong, Q., and Xie, X. Q. (2012). Molecular fingerprint-based artificial neural networks QSAR for ligand biological activity predictions. *Molecular pharmaceutics*.

Mysinger, M. M., Carchia, M., Irwin, J. J., and Shoichet, B. K. (2012). Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594.

Nicklaus, M. C., Wang, S., Driscoll, J. S., and Milne, G. W. (1995). Conformational changes of small molecules binding to proteins. *Bioorganic & Medicinal Chemistry*, 3(4):411–428.

Nirschl, A. A., Zou, Y., Krystek, S. R., Sutton, J. C., Simpkins, L. M., Lupisella, J. A., Kuhns, J. E., Seethala, R., Golla, R., Sleph, P. G., Beehler, B. C., Grover, G. J., Egan, D., Fura, A., Vyas, V. P., Li, Y.-X., Sack, J. S., Kish, K. F., An, Y., Bryson, J. A., Gougoutas, J. Z., DiMarco, J., Zahler, R., Ostrowski, J., and Hamann, L. G. (2009). N-aryl-oxazolidin-2-imine muscle selective androgen receptor modulators enhance potency through pharmacophore reorientation. *Journal of medicinal chemistry*, 52(9):2794–2798.

Notenboom, V., Williams, S. J., Hoos, R., Withers, S. G., and Rose, D. R. (2000). Detailed structural analysis of glycosidase/inhibitor interactions: complexes of Cex from Cellulomonas fimi with xylobiose-derived aza-sugars. *Biochemistry*, 39(38):11553–11563.

Oh, B. H., Pandit, J., Kang, C. H., Nikaido, K., Gokcen, S., Ames, G. F., and Kim, S. H. (1993). Three-dimensional structures of the periplasmic lysine/arginine/ornithine-binding protein with and without a ligand. *The Journal of biological chemistry*, 268(15):11348–11355.

Okrasa, K., Levy, C., Wilding, M., Goodall, M., Baudendistel, N., Hauer, B., Leys, D., and Micklefield, J. (2009). Structure-guided directed evolution of alkenyl and arylmalonate decarboxylases. *Angewandte Chemie International Edition*, 48(41):7691–7694.

Orita, M., Yamamoto, S., Katayama, N., Aoki, M., Takayama, K., Yamagiwa, Y., Seki, N., Suzuki, H., Kurihara, H., Sakashita, H., Takeuchi, M., Fujita, S., Yamada, T., and Tanaka, A. (2001). Coumarin and chromen-4-one analogues as tautomerase inhibitors of macrophage migration inhibitory factor: discovery and X-ray crystallography. *Journal of medicinal chemistry*, 44(4):540–547.

Oswald, C., Smits, S. H. J., Höing, M., Sohn-Bösser, L., Dupont, L., Le Rudulier, D., Schmitt, L., and Bremer, E. (2008). Crystal structures of the choline/acetylcholine substrate-binding protein ChoX from Sinorhizobium meliloti in the liganded and unliganded-closed states. *The Journal of biological chemistry*, 283(47):32848–32859.

Paesen, G. C., Adams, P. L., Harlos, K., Nuttall, P. A., and Stuart, D. I. (1999). Tick histamine-binding proteins: isolation, cloning, and three-dimensional structure. *Molecular cell*, 3(5):661–671.

Pappenberger, G., Schulz-Gasch, T., Kusznir, E., Müller, F., and Hennig, M. (2007). Structure-assisted discovery of an aminothiazole derivative as a lead molecule for inhibition of bacterial fatty-acid synthesis. *Acta crystallographica Section D, Biological crystallography*, 63(Pt 12):1208–1216.

Park, S., Yang, X., and Saven, J. G. (2004). Advances in computational protein design. *Current Opinion in Structural Biology*, 14(4):487–494.

Pavlovsky, A. G., Liu, X., Faehnle, C. R., Potente, N., and Viola, R. E. (2012). Structural characterization of inhibitors with selectivity against members of a homologous enzyme family. *Chemical Biology & Drug Design*, 79(1):128–136.

Pesenti, M. E., Spinelli, S., Bezirard, V., Briand, L., Pernollet, J.-C., Campanacci, V., Tegoni, M., and Cambillau, C. (2009). Queen bee pheromone binding protein pH-induced domain swapping favors pheromone release. *Journal of molecular biology*, 390(5):981–990.

Pierce, A. C., Jacobs, M., and Stuver-Moody, C. (2008). Docking study yields four novel inhibitors of the protooncogene Pim-1 kinase. *Journal of medicinal chemistry*, 51(6):1972–1975.

Prehna, G., Li, Y., Stoynov, N., Okon, M., Vuckovic, M., McIntosh, L. P., Foster, L. J., Finlay, B. B., and Strynadka, N. C. J. (2012). The zinc regulated antivirulence pathway of Salmonella is a multiprotein immunoglobulin adhesion system. *The Journal of biological chemistry*, 287(39):32324–32337.

Presnell, S. R., Patil, G. S., Mura, C., Jude, K. M., Conley, J. M., Bertrand, J. A., Kam, C. M., Powers, J. C., and Williams, L. D. (1998). Oxyanion-mediated inhibition of serine proteases. *Biochemistry*, 37(48):17068–17081.

Proisy, N., Sharp, S. Y., Boxall, K., Connelly, S., Roe, S. M., Prodromou, C., Slawin, A. M. Z., Pearl, L. H., Workman, P., and Moody, C. J. (2006). Inhibition of Hsp90 with synthetic macrolactones: synthesis and structural and biological evaluation of ring and conformational analogs of radicicol. *Chemistry & Biology*, 13(11):1203–1215.

Pylayeva-Gupta, Y., Grabocka, E., and Bar-Sagi, D. (2011). RAS oncogenes: weaving a tumorigenic web. *Nature reviews. Cancer*, 11(11):761–774.

Qi, X., Loiseau, F., Chan, W. L., Yan, Y., Wei, Z., Milroy, L.-G., Myers, R. M., Ley, S. V., Read, R. J., Carrell, R. W., and Zhou, A. (2011). Allosteric modulation of hormone release from thyroxine and corticosteroid-binding globulins. *The Journal of biological chemistry*, 286(18):16163–16173.

Rarey, M., Kramer, B., Lengauer, T., and Klebe, G. (1996). A Fast Flexible Docking Method using an Incremental Construction Algorithm. *Journal of molecular biology*, 261(3):470–489.

Ravaud, S., Robert, X., Watzlawick, H., Haser, R., Mattes, R., and Aghajari, N. (2007). Trehalulose synthase native and carbohydrate complexed structures provide insights into sucrose isomerization. *The Journal of biological chemistry*, 282(38):28126–28136.

Raves, M. L., Harel, M., Pang, Y. P., Silman, I., Kozikowski, A. P., and Sussman, J. L. (1997). Structure of acetylcholinesterase complexed with the nootropic alkaloid, (-)-huperzine A. *Nature structural biology*, 4(1):57–63.

Reymond, J. L., Ruddigkeit, L., and Blum, L. (2012). The enumeration of chemical space. *Wiley Interdisciplinary Reviews: Computer molecular science*, 2(5):717–733.

Richardson, T. I., Dodge, J. A., Durst, G. L., Pfeifer, L. A., Shah, J., Wang, Y., Durbin, J. D., Krishnan, V., and Norman, B. H. (2007). Benzopyrans as selective estrogen receptor beta agonists (SERBAs). Part 3: synthesis of cyclopentanone and cyclohexanone intermediates for C-ring modification. *Bioorganic & Medicinal Chemistry Letters*, 17(17):4824–4828.

Rigsby, R. E., Rife, C. L., Fillgrove, K. L., Newcomer, M. E., and Armstrong, R. N. (2004). Phosphonoformate: a minimal transition state analogue inhibitor of the fosfomycin resistance protein, FosA. *Biochemistry*, 43(43):13666–13673.

Rohl, C. A., Strauss, C. E. M., Misura, K. M. S., and Baker, D. (2004). Protein structure prediction using Rosetta. *Methods in enzymology*, 383:66–93.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408.

Ryu, K.-S., Kim, J.-I., Cho, S.-J., Park, D., Park, C., Cheong, H.-K., Lee, J.-O., and Choi, B.-S. (2005). Structural insights into the monosaccharide specificity of Escherichia coli rhamnose mutarotase. *Journal of molecular biology*, 349(1):153–162.

Sahigara, F., Mansouri, K., Ballabio, D., Mauri, A., Consonni, V., and Todeschini, R. (2012). Comparison of Different Approaches to Define the Applicability Domain of QSAR Models. *Molecules*, 17(12):4791–4810.

Schaeffer, R. D., Jonsson, A. L., Simms, A. M., and Daggett, V. (2010). Generation of a consensus protein domain dictionary. *Bioinformatics*, 27(1):46–54.

Schiefner, A., Breed, J., Bösser, L., Kneip, S., Gade, J., Holtmann, G., Diederichs, K., Welte, W., and Bremer, E. (2004a). Cation-pi interactions as determinants for binding of the compatible solutes glycine betaine and proline betaine by the periplasmic ligand-binding protein ProX from Escherichia coli. *The Journal of biological chemistry*, 279(7):5588–5596.

Schiefner, A., Holtmann, G., Diederichs, K., Welte, W., and Bremer, E. (2004b). Structural basis for the binding of compatible solutes by ProX from the hyperthermophilic archaeon Archaeoglobus fulgidus. *The Journal of biological chemistry*, 279(46):48270–48281.

Schneider, M., Fu, X., and Keating, A. E. (2009). X-ray vs. NMR structures as templates for computational protein design. *Proteins: Structure, Function, and Bioinformatics*, 77(1):97–110.

Schumacher, M. A., Carter, D., Scott, D. M., Roos, D. S., Ullman, B., and Brennan, R. G. (1998). Crystal structures of Toxoplasma gondii uracil phosphoribosyltransferase reveal the atomic basis of pyrimidine discrimination and prodrug binding. *The EMBO journal*, 17(12):3219–3232.

Shapovalov, M. V. and Dunbrack, R. L. (2011). A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure (London, England : 1993)*, 19(6):844–858.

Sharabi, O., Dekel, A., and Shifman, J. M. (2011). Triathlon for energy functions: Who is the winner for design of protein-protein interactions? *Proteins: Structure, Function, and Bioinformatics*, 79(5):1487–1498.

Shaw, P.-C., Wong, K.-B., Chan, D. S.-B., and Williams, R. L. (2003). Structural basis for the interaction of [E160A-E189A]-trichosanthin with adenine. *Toxicon : official journal of the International Society on Toxinology*, 41(5):575–581.

Sheffler, W. and Baker, D. (2009). RosettaHoles: rapid assessment of protein core packing for structure prediction, refinement, design, and validation. *Protein Science*, 18(1):229–239.

Shimada, A., Ueguchi-Tanaka, M., Nakatsu, T., Nakajima, M., Naoe, Y., Ohmiya, H., Kato, H., and Matsuoka, M. (2008). Structural basis for gibberellin recognition by its receptor GID1. *Nature*, 456(7221):520–523.

Shin, H., Cama, E., and Christianson, D. W. (2004). Design of amino acid aldehydes as transition-state analogue inhibitors of arginase. *Journal of the American Chemical Society*, 126(33):10278–10284.

Simeonov, P., Berger-Hoffmann, R., Hoffmann, R., Sträter, N., and Zuchner, T. (2011). *Protein Engineering Design and Selection*, (3).

Simons, K., Ruczinski, I., Kooperberg, C., and Fox, B. (1999). Improved Recognition of Native-Like Protein Structures Using a Combination of Sequence-Dependent and

Sequence-Independent Features of Proteins . *Proteins Structure Function and Genetics*, 34:82–95.

Simons, K. T., Kooperberg, C., Huang, E., and Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of molecular biology*, 268(1):209–225.

Sliwoski, G., Kothiwale, S., Meiler, J., and Lowe, E. W. (2014). Computational methods in drug discovery. *Pharmacological reviews*, 66(1):334–395.

Soisson, S. M., MacDougall-Shackleton, B., Schleif, R., and Wolberger, C. (1997). The 1.6 A crystal structure of the AraC sugar-binding and dimerization domain complexed with D-fucose. *Journal of molecular biology*, 273(1):226–237.

Stamper, C., Bennett, B., Edwards, T., Holz, R. C., Ringe, D., and Petsko, G. (2001). Inhibition of the aminopeptidase from Aeromonas proteolytica by L-leucinephosphonic acid. Spectroscopic and crystallographic characterization of the transition state of Peptide Hydrolysis. *Biochemistry*, 40:7035–7046.

Stengl, B., Meyer, E. A., Heine, A., Brenk, R., Diederich, F., and Klebe, G. (2007). Crystal structures of tRNA-guanine transglycosylase (TGT) in complex with novel and potent inhibitors unravel pronounced induced-fit adaptations and suggest dimer formation upon substrate binding. *Journal of molecular biology*, 370(3):492–511.

Stiffin, R. M., Sullivan, S. M., Carlson, G. M., and Holyoak, T. (2008). Differential inhibition of cytosolic PEPCK by substrate analogues. Kinetic and structural characterization of inhibitor recognition. *Biochemistry*, 47(7):2099–2109.

Sugiyama, S., Matsuo, Y., Maenaka, K., Vassylyev, D. G., Matsushima, M., Kashiwagi, K., Igarashi, K., and Morikawa, K. (1996). The 1.8-A X-ray structure of the Escherichia coli PotD protein complexed with spermidine and the mechanism of polyamine binding. *Protein Science*, 5(10):1984–1990.

Sumida, T., Stubbs, K. A., Ito, M., and Yokoyama, S. (2012). Gaining insight into the inhibition of glycoside hydrolase family 20 exo-β-N-acetylhexosaminidases using a structural approach. *Organic & biomolecular chemistry*, 10(13):2607–2612.

Sun, Q., Burke, J. P., Phan, J., Burns, M. C., Olejniczak, E. T., Waterson, A. G., Lee, T., Rossanese, O. W., and Fesik, S. W. (2012). Discovery of small molecules that bind to K-Ras and inhibit Sos-mediated activation. *Angewandte Chemie International Edition*, 51(25):6140–6143.

Sun, Q., Phan, J., Friberg, A. R., Camper, D. V., Olejniczak, E. T., and Fesik, S. W. (2014). A method for the second-site screening of K-Ras in the presence of a covalently attached first-site ligand. *Journal of biomolecular NMR*, 60(1):11–14.

Tang, G. W. and Altman, R. B. (2014). Knowledge-based fragment binding prediction. *PLoS computational biology*, 10(4):e1003589.

Tetko, I. V., Livingstone, D. J., and Luik, A. I. (1995). Neural network studies. 1. Comparison of overfitting and overtraining. *Journal Of Chemical Information And Modeling*, 35(5):826–833.

Thompson, M. J., Sievers, S. A., Karanicolas, J., Ivanova, M. I., Baker, D., and Eisenberg, D. (2006). The 3D profile method for identifying fibril-forming segments of proteins. *Proceedings of the National Academy of Sciences*, 103(11):4074–4078.

Trivella, D. B. B., Bleicher, L., Palmieri, L. d. C., Wiggers, H. J., Montanari, C. A., Kelly, J. W., Lima, L. M. T. R., Foguel, D., and Polikarpov, I. (2010). Conformational differences between the wild type and V30M mutant transthyretin modulate its binding to genistein: implications to tetramer stability and ligand-binding. *Journal of structural biology*, 170(3):522–531.

Trott, O. and Olson, A. J. (2009). AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, pages NA–NA.

Tsitsanou, K. E., Thireou, T., Drakou, C. E., Koussis, K., Keramioti, M. V., Leonidas, D. D., Eliopoulos, E., Iatrou, K., and Zographos, S. E. (2012). Anopheles gambiae odorant binding protein crystal complex with the synthetic repellent DEET: implications for structure-based design of novel mosquito repellents. *Cellular and molecular life sciences : CMLS*, 69(2):283–297.

Turlington, M., Noetzel, M. J., Chun, A., Zhou, Y., Gogliotti, R. D., Nguyen, E. D., Gregory, K. J., Vinson, P. N., Rook, J. M., Gogi, K. K., Xiang, Z., Bridges, T. M., Daniels, J. S., Jones, C., Niswender, C. M., Meiler, J., Conn, P. J., Lindsley, C. W., and Stauffer, S. R. (2013). Exploration of allosteric agonism structure-activity relationships within an acetylene series of metabotropic glutamate receptor 5 (mGlu5) positive allosteric modulators (PAMs): discovery of 5-((3-fluorophenyl)ethynyl)-N-(3-methyloxetan-3-yl)picolinamide (ML254). *Journal of medicinal chemistry*, 56(20):7976–7996.

Vassylyev, D. G., Tomitori, H., Kashiwagi, K., Morikawa, K., and Igarashi, K. (1998). Crystal structure and mutational analysis of the Escherichia coli putrescine receptor. Structural basis for substrate specificity. *The Journal of biological chemistry*, 273(28):17604–17609.

Vetting, M. W., D'Argenio, D. A., and Ornston, L. N. (2000). Structure of Acinetobacter strain ADP1 protocatechuate 3, 4-dioxygenase at 2.2 Å resolution: implications for the mechanism of an intradiol dioxygenase. *Biochemistry*.

Villa, F., Goebel, J., Rafiqi, F. H., Deak, M., Thastrup, J., Alessi, D. R., and van Aalten, D. M. F. (2007). Structural insights into the recognition of substrates and activators by the OSR1 kinase. *EMBO reports*, 8(9):839–845.

Vincent, F., Gloster, T. M., Macdonald, J., Morland, C., Stick, R. V., Dias, F. M. V., Prates, J. A. M., Fontes, C. M. G. A., Gilbert, H. J., and Davies, G. J. (2004). Common inhibition of both beta-glucosidases and beta-mannosidases by isofagomine lactam reflects different conformational itineraries for pyranoside hydrolysis. *Chembiochem : a European journal of chemical biology*, 5(11):1596–1599.

Voet, A. R. D., Kumar, A., Berenger, F., and Zhang, K. Y. J. (2014). Combining in silico and in cerebro approaches for virtual screening and pose prediction in SAMPL4. *Journal of computer-aided molecular design*, pages 1–11.

Vogel, S. M., Bauer, M. R., and Boeckler, F. M. (2011). DEKOIS: Demanding Evaluation Kits for Objective in Silico Screening — A Versatile Tool for Benchmarking Docking Programs and Scoring Functions. *Journal Of Chemical Information And Modeling*, 51(10):2650–2665.

Wan, Z.-K., Follows, B., Kirincich, S., Wilson, D., Binnun, E., Xu, W., Joseph-McCarthy, D., Wu, J., Smith, M., Zhang, Y.-L., Tam, M., Erbe, D., Tam, S., Saiah, E., and Lee, J. (2007). Probing acid replacements of thiophene PTP1B inhibitors. *Bioorganic & Medicinal Chemistry Letters*, 17(10):2913–2920.

Wang, C., Bradley, P., and Baker, D. (2007). Protein–Protein Docking with Backbone Flexibility. *Journal of molecular biology*, 373(2):503–519.

Wang, R., Fang, X., Lu, Y., and Wang, S. (2004). The PDBbind Database: Collection of Binding Affinities for Protein−Ligand Complexes with Known Three-Dimensional Structures. *Journal of medicinal chemistry*, 47(12):2977–2980.

Wang, S., Beck, R., Blench, T., Burd, A., Buxton, S., Malic, M., Ayele, T., Shaikh, S., Chahwala, S., Chander, C., Holland, R., Merette, S., Zhao, L., Blackney, M., and Watts, A. (2010). Studies of benzothiophene template as potent factor IXa (FIXa) inhibitors in thrombosis. *Journal of medicinal chemistry*, 53(4):1465–1472.

Watson, R. P., Demmer, J., Baker, E. N., and Arcus, V. L. (2007). Three-dimensional structure and ligand binding properties of trichosurin, a metatherian lipocalin from the milk whey of the common brushtail possum Trichosurus vulpecula. *Biochemical Journal*, 408(1):29–38.

Weber, C., Cole, D. J., O'Regan, D. D., and Payne, M. C. (2014). Renormalization of myoglobin-ligand binding energetics by quantum many-body effects. *Proceedings of the National Academy of Sciences*, 111(16):5790–5795.

Wei, B. Q., Baase, W. A., Weaver, L. H., Matthews, B. W., and Shoichet, B. K. (2002). A model binding site for testing scoring functions in molecular docking. *Journal of molecular biology*, 322(2):339–355.

White, A. W., Almassy, R., Calvert, A. H., Curtin, N. J., Griffin, R. J., Hostomsky, Z., Maegley, K., Newell, D. R., Srinivasan, S., and Golding, B. T. (2000). Resistance-modifying agents. 9. Synthesis and biological properties of benzimidazole inhibitors of the DNA repair enzyme poly(ADP-ribose) polymerase. *Journal of medicinal chemistry*, 43(22):4084–4097.

Whitworth, G. E., Macauley, M. S., Stubbs, K. A., Dennis, R. J., Taylor, E. J., Davies, G. J., Greig, I. R., and Vocadlo, D. J. (2007). Analysis of PUGNAc and NAG-thiazoline as transition state analogues for human O-GlcNAcase: mechanistic and structural insights into inhibitor selectivity and transition state poise. *Journal of the American Chemical Society*, 129(3):635–644.

Wilcken, R., Liu, X., Zimmermann, M. O., Rutherford, T. J., Fersht, A. R., Joerger, A. C., and Boeckler, F. M. (2012). Halogen-enriched fragment libraries as leads for drug rescue of mutant p53. *Journal of the American Chemical Society*, 134(15):6810–6818.

Wilder, P. T., Charpentier, T. H., Liriano, M. A., Gianni, K., Varney, K. M., Pozharski, E., Coop, A., Toth, E. A., Mackerell, A. D., and Weber, D. J. (2010). In vitro screening and structural characterization of inhibitors of the S100B-p53 interaction. *International journal of high throughput screening*, 2010(1):109–126.

Woetzel, N., Lindert, S., Stewart, P. L., and Meiler, J. (2011). BCL::EM-Fit: rigid body fitting of atomic structures into density maps using geometric hashing and real space refinement. *Journal of structural biology*, 175(3):264–276.

Wold, M. S. (1997). Replication protein A: a heterotrimeric, single-stranded DNA-binding protein required for eukaryotic DNA metabolism. *Annual Review of Biochemistry*, 66(1):61–92.

Woo, E. J., Marshall, J., Bauly, J., Chen, J. G., Venis, M., Napier, R. M., and Pickersgill, R. W. (2002). Crystal structure of auxin-binding protein 1 in complex with auxin. *The EMBO journal*, 21(12):2877–2885.

Wu, B., Chien, E. Y. T., Mol, C. D., Fenalti, G., Liu, W., Katritch, V., Abagyan, R., Brooun, A., Wells, P., Bi, F. C., Hamel, D. J., Kuhn, P., Handel, T. M., Cherezov, V., and Stevens, R. C. (2010). Structures of the CXCR4 chemokine GPCR with small-molecule and cyclic peptide antagonists. *Science*, 330(6007):1066–1071.

Xiao, S., Patsalo, V., Shan, B., Bi, Y., Green, D. F., and Raleigh, D. P. (2013). Rational modification of protein stability by targeting surface sites leads to complicated results. *Proceedings of the National Academy of Sciences of the United States of America*, 110(28):11337–11342.

Yang, L.-Q., Sang, P., Tao, Y., Fu, Y.-X., Zhang, K.-Q., Xie, Y.-H., and Liu, S.-Q. (2014). Protein dynamics and motions in relation to their functions: several case studies and the underlying mechanisms. *Journal of biomolecular structure & dynamics*, 32(3):372–393.

Yao, N., Trakhanov, S., and Quiocho, F. A. (1994). Refined 1.89-A structure of the histidine-binding protein complexed with histidine and its relationship with many other active transport/chemosensory proteins. *Biochemistry*, 33(16):4769–4779.

Yao, Y., Harrison, C. B., Freddolino, P. L., Schulten, K., and Mayer, M. L. (2008). Molecular mechanism of ligand recognition by NR3 subtype glutamate receptors. *The EMBO journal*, 27(15):2158–2170.

Yates, S. P., Taylor, P. L., Jørgensen, R., Ferraris, D., Zhang, J., Andersen, G. R., and Merrill, A. R. (2005). Structure-function analysis of water-soluble inhibitors of the catalytic domain of exotoxin A from Pseudomonas aeruginosa. *Biochemical Journal*, 385(Pt 3):667–675.

Yilmaz, O. G., Olmez, E. O., and Ulgen, K. O. (2013). Targeting the Akt1 allosteric site to identify novel scaffolds through virtual screening. *Computational biology and chemistry*, 48C:1–13.

Zechel, D. L., Boraston, A. B., Gloster, T., Boraston, C. M., Macdonald, J. M., Tilbrook, D. M. G., Stick, R. V., and Davies, G. J. (2003). Iminosugar glycosidase inhibitors: structural and thermodynamic dissection of the binding of isofagomine and 1-deoxynojirimycin to beta-glucosidases. *Journal of the American Chemical Society*, 125(47):14313–14323.

Zhang, H., Astrof, N. S., Liu, J.-H., Wang, J.-h., and Shimaoka, M. (2009a). Crystal structure of isoflurane bound to integrin LFA-1 supports a unified mechanism of volatile anesthetic action in the immune and central nervous systems. *FASEB journal : official publication of the Federation of American Societies for Experimental Biology*, 23(8):2735–2740.

Zhang, P., Ma, J., Zhang, Z., Zha, M., Xu, H., Zhao, G., and Ding, J. (2009b). Molecular basis of the inhibitor selectivity and insights into the feedback inhibition mechanism of citramalate synthase from Leptospira interrogans. *Biochemical Journal*, 421(1):133–143.

Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4):399–458.

Zhou, Z.-H., Jiang, Y., Yang, Y.-B., and Chen, S.-F. (2002). Lung cancer cell identification based on artificial neural network ensembles. *Artificial Intelligence in Medicine*, 24(1):25–36.

Zhu, X., Yan, X., Carter, L. G., Liu, H., Graham, S., Coote, P. J., and Naismith, J. (2012). A model for 3-methyladenine recognition by 3-methyladenine DNA glycosylase I (TAG) from Staphylococcus aureus. *Acta crystallographica. Section F, Structural biology and crystallization communications*, 68(Pt 6):610–615.

Zuegg, J., Gruber, K., Gugganig, M., Wagner, U. G., and Kratky, C. (1999). Three-dimensional structures of enzyme-substrate complexes of the hydroxynitrile lyase from Hevea brasiliensis. *Protein Science*, 8(10):1990–2000.