

DEVELOPMENT OF METHODS FOR DOCKING AND DESIGNING SMALL  
MOLECULES WITHIN THE ROSETTA CODE FRAMEWORK

by

GORDON HOWARD LEMMON

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

December, 2012

Nashville, Tennessee

Approved by:

David L. Tabb

Jarrold A. Smith

Brian O. Bachmann

Richard T. D'Aquila

Copyright © 2012 by Gordon Howard Lemmon

All Rights Reserved

## ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Tabb, and my committee members, Dr. Bachmann, Dr. D'Aquila, Dr. Smith for their constructive criticism and for helping me graduate on time. Thanks also to my doctoral advisor, Professor Jens Meiler, for many hours reviewing my papers and offering his advice.

Thanks also go to my colleagues in the Meiler lab for their friendship and many helpful scientific discussions. Also thanks for helping me hunt down bugs in the code. The staff at the Advanced Computing Center for Research and Education (ACCRE) as well as the CSB administrators must also be acknowledged for providing and maintaining the resources that made this work possible.

Thanks to my mother and father for their encouragement and to my best friend for her patience and love. Thanks to the LDS church for providing countless opportunities for meaningful service, and for helping me remember that “he that loseth his life...shall find it” (Matt 10:39) isn't a scripture about time spent in lab. Finally thanks to my Creator for providing guidance, comfort, assurance, and purpose to life.

Funding for this research was provided through the Molecular Biophysics Training Grant (NIH GM 08320-18).

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	iii
LIST OF FIGURES .....	ix
LIST OF TABLES .....	xi
NOMENCLATURE .....	xii
STATEMENT OF DISSERTATION .....	xiii
References.....	xv
I. INTRODUCTION .....	1
Ligand Docking: Sampling.....	3
Ligand Docking: Scoring .....	5
Ligand Design.....	6
Rosetta .....	8
RosettaLigand.....	9
HIV/AIDS.....	9
HIV Protease.....	11
HIV Protease Inhibitors .....	11
References.....	12
II. PREDICTION OF HIV-1 PROTEASE/INHIBITOR AFFINITY USING ROSETTALIGAND .....	19
Introduction.....	19
Materials and Methods .....	23

176 experimental PR/PI binding energies have been collected.....	23
171 high resolution template PR structures have been collected. ....	24
Threading of sequence onto structure for comparative modeling. ....	24
High resolution refinement of comparative models. ....	24
Low resolution initial placement of ligand.....	24
Docking of PIs into comparative models. ....	25
Predicting $\Delta\Delta G$ s using the standard approach. ....	26
Predicting $\Delta\Delta G$ s using the constant-unbound approach. ....	26
Optimization of RosettaLigand score term weights. ....	28
Partitioning data by location of PR mutations.....	28
 Results/Discussion.....	 29
Assessment of uncertainty in experimental binding affinity data. ....	29
Building of comparative models from HIV-1 templates.....	30
RosettaLigand docking protocol allows local flexibility.....	31
Usage of experimental data for weight optimization. ....	31
Analysis of optimized score term weights.....	32
Predicting $\Delta\Delta G$ s using the standard approach. ....	33
Predicting $\Delta\Delta G$ s using the constant-unbound approach. ....	34
Optimized score term weights predict binding affinity in independent data set. ....	35
Analysis of data partitioned by location of PR mutations.....	35
 Conclusion.....	 37
 Future Directions.....	 38
 Acknowledgements.....	 39
 Supporting Information.....	 39
 References.....	 39
 III. ROSETTA LIGAND DOCKING WITH FLEXIBLE XML PROTOCOLS.....	 43
Summary.....	43
Introduction.....	43
Materials.....	45

Preparation of protein PDB input file.....	46
Preparation of ligand PDB and “params” input files.....	47
Relevant command line or flags file options.....	48
Methods .....	49
StartFrom.....	49
Translate .....	50
Translates.....	50
CompoundTranslate .....	51
Rotate.....	51
Rotates .....	52
SlideTogether .....	52
HighResDocker .....	53
FinalMinimizer.....	54
InterfaceScoreCalculator .....	55
Putting it all together .....	55
Notes .....	58
Score Term reweighting .....	58
How many models should I make? .....	58
LIGAND_AREAS.....	58
INTERFACE_BUILDERS.....	60
MOVEMAP_BUILDERS .....	60
How long will this take to run? .....	61
How do I analyze my results .....	62
Fragmentation for flexibility and design .....	62
References.....	63
IV. TOWARDS LIGAND DOCKING INCLUDING EXPLICIT INTERFACE WATER MOLECULES.....	65
Abstract.....	65
Introduction.....	66
Materials and Methods .....	73

Preparation of HIV-1 PR inputs for cross docking. ....	73
Preparation of CSAR dataset. ....	74
Standard docking – low resolution sampling. ....	75
Standard docking – high resolution refinement. ....	76
HIV-1 PR/PI cross-docking with a protein-centric water. ....	76
CSAR self-docking with ligand-centric waters. ....	77
Creation of ligand conformers. ....	78
Placement of decoy waters. ....	78
CSAR self-docking with ligand conformers and water decoys. ....	78
Docking Model Production and Analysis. ....	79
Ranking metrics. ....	79
RMSD calculations. ....	79
Results/Discussion. ....	80
Protein-centric water docking improves placement of HIV-1 PR inhibitors. ....	80
Protein-centric water docking improves ranking of HIV-1 PR inhibitors: ....	81
Analysis of CSAR dataset. ....	82
Ligand-centric water docking improves CSAR inhibitor placement. ....	85
Ligand-centric water docking improves CSAR inhibitor ranking. ....	86
Ligand-centric water docking improves CSAR docking “success rates”. ....	87
Analysis of successes and failures in CSAR water docking. ....	88
Docking of CSAR interface waters does not improve binding affinity predictions. ....	90
Docking with ligand conformers is improved by a water placement algorithm. ....	91
References. ....	93
V. CONCLUSION & FUTURE DIRECTIONS. ....	97
RosettaLigand binding affinity prediction. ....	97
Advances to RosettaLigand software. ....	98
Modeling of explicit waters within RosettaLigand. ....	99
References. ....	100
APPENDICES. ....	101
A. Experimental $\Delta\Delta G$ & $K_i$ values for HIV-1 PR/PI binding. ....	101

B.	Template structures used for comparative modeling.....	104
C.	Sequence Alignment of 171 HIV-1 PR backbone templates.....	109
D.	HIV-1 PR/PI data partitioned by location of mutations.....	113
E.	Distribution of Rosetta energy scores for a set of HIV-1 PR/PI models .....	114
F.	Comparison of top Rosetta models and experimental structures.....	115
G.	HIV-1 PR/PI docking protocol .....	116
	Preliminary Relax Step.....	116
	Ligand docking step .....	116
H.	Prediction of clinical outcomes.....	118
I.	Ligand docking through incremental construction .....	122
J.	Ligand design through incremental construction.....	127
	General Rosetta code modifications.....	127
	Creation of ligand fragments .....	129
	GrowLigand.....	129
	Filters for terminating ligand design growth .....	129
	AddHydrogens.....	130
	Ligand design XML .....	130
	Future directions.....	132
K.	XML used with standard docking (chapter 4) .....	134
L.	XML used with protein-centric docking.....	135
M.	XML used with ligand-centric docking .....	137
N.	Summary of my commits to the Rosetta SVN server .....	138



## LIST OF FIGURES

Figure I-1: Accuracy of RosettaLigand binding affinity predictions.....	9
Figure I-2: HIV-1 Protease bound to Ritonavir. ....	11
Figure I-3: Skeletal formula of several HIV-1 protease inhibitors.....	12
Figure II-1. Effect of binding site and non-binding site mutations on rigid and flexible proteins.....	20
Figure II-2. HIV-1 protease binding site flexibility.....	22
Figure II-3. Explanation of $\Delta\Delta\Delta G$ .....	27
Figure II-4. HIV-1 PR Multiple sequence alignment using ClustalX 2.1. ....	30
Figure II-5. Predicted vs.experimental HIV-1 PR/PI binding affinity.....	34
Figure III-1: Multiple ligand docking captures induced-fit. ....	45
Figure III-2. RosettaLigand design of HisF/DHT interface.....	53
Figure IV-1. Sampling of the HIV-1 protease binding pocket by Ritonavir and a conserved water. ....	75
Figure IV-2. Comparison of ranks and RMSDs from standard docking and protein-centric water docking predictions of HIV-1 PR/PI interfaces. ....	80
Figure IV-3. CSAR inhibitor properties by number of interface waters. ....	84
Figure IV-4. Effect of ligand-centric water docking of CSAR data by interface water count.....	86
Figure IV-5. Comparison of rank & RMSDs from standard docking and ligand-centric water docking of CSAR data. ....	87

Figure IV-6. Example CSAR docking result where ligand-centric waters improve inhibitor placement. ....	88
Figure IV-7. Example CSAR docking result where ligand-centric waters worsen inhibitor placement. ....	89
Figure IV-8. Relation between binding pocket crowdedness and ligand-centric water based improvements in CSAR model ranking. ....	90

## LIST OF TABLES

Table II-1. Score term weights optimized for HIV-1 PR binding affinity prediction. ....	32
Table II-2. Correlations between RosettaLigand predictions and experimental data. ....	35
Table III-1: Relative speed for various components of RosettaLigand XML. ....	61
Table IV-1. Comparison of protein-centric and ligand-centric water docking. ....	77
Table IV-2. Change in RMSD of top Rosetta model of HIV-1 PR/Pis when water is docked. ....	81
Table IV-3. Change in whether top scoring model of HIV-1 PR/PI is <2.0 Å RMSD when water is docked. ....	82
Table IV-4: Summary statistics describing the CSAR dataset. ....	85
Table IV-5. Average CSAR docking results. ....	85
Table IV-6.: Ranking metrics for CSAR docking studies. ....	86

## NOMENCLATURE

PR	HIV-1 Protease
PI	HIV-1 Protease Inhibitor
$\Delta G$	Gibbs Free Energy
$\Delta\Delta G$	Binding affinity
$\Delta\Delta\Delta G$	Change in binding affinity due to mutation
$\Delta G_b$	Free energy of a protein/ligand complex (“bound”)
$\Delta G_u$	Free energy of a protein without ligand (“unbound”)

## STATEMENT OF DISSERTATION

Structure-based drug design is a key challenge for pharmaceutical chemists. By studying the structure of proteins bound to natural substrates, researchers can design small molecules which they predict will bind in a similar fashion. Docking software such as RosettaLigand (Meiler and Baker 2006) assists researchers in predicting how a small molecule and a protein will interact. These docking algorithms play a crucial role in in-silico drug screening and drug design (Zoete, Grosdidier et al. 2009).

The original aims of my doctoral research as presented in my qualifying proposal were to (1) make improvements to the RosettaLigand docking algorithm, and (2) develop a protocol for designing small molecules within Rosetta. My work began by investigating ways to improve predictions of HIV-1 protease/protease inhibitor (PR/PI) binding affinity ( $\Delta\Delta G$ ). It was clear from the literature that attempts to predict PR/PI  $\Delta\Delta G$ s had been unsuccessful (Kim and Skolnick 2008; Cheng, Li et al. 2009). Yet accurate  $\Delta\Delta G$  predictions have the potential to streamline the drug discovery process. Accurate  $\Delta\Delta G$  predictions mean in silico screening can correctly identify high affinity lead compounds. We also suggest these predictions could be coupled with HIV-1 genotype assays to determine which inhibitors to prescribe (structure based). Based on work presented in chapter 2, accurate prediction (correlation coefficient of  $R=0.71$ ) of HIV-1 PR/PI  $\Delta\Delta G$ s is now a reality (Lemmon, Kaufmann et al. 2012).

While we were thrilled to see such drastic improvements in HIV-1 PR/PI  $\Delta\Delta G$  prediction we recognized that our methods included two significant limitations. First, PIs largely retained their crystallographic conformations during docking simulations. Only small adjustments in PI torsion angles were allowed. Second, a key water molecule known mediate the PR/PI interaction

was not considered. The first limitation was part of a more general problem of efficient sampling of flexibility. The second limitation was a result of the *one protein/one ligand* approach of most ligand docking software. To overcome these limitations significant refactoring of Rosetta code was necessary.

Rather than modify the one-file, hard-coded RosettaLigand protocol, I completely rewrote RosettaLigand as collection of independent classes. The user defines a custom ligand docking protocol via an XML script, where each of the XML tags correspond to a Rosetta class of the same name, and attributes define options for that class. Chapter 3 is a guide to the XML specification and describes *multiple ligand docking* (Lemmon and Meiler 2012). The new code also allows large-scale ligand flexibility through fragmentation (Appendix I) and implements the rudiments of ligand design (Appendix J).

Having developed code that allows for multiple ligand docking, I was positioned to add a key interface water to my HIV-1 PR/PI  $\Delta\Delta G$  docking studies. After showing that this water improves prediction accuracy, I studied the effect of water on docking accuracy within a dataset of diverse protein/ligand complexes (Dunbar, Smith et al. 2011). The results from water docking are found in chapter 4.

I am responsible for the data preparation, data analysis, interpretation of results, and text found in this dissertation. With the exception of occasional help from Sam DeLuca in debugging a C++ syntax error, I am solely responsible for all modifications to RosettaCode summarized in Appendix N. Sarel Fleishman put in place the XML framework that enabled my refactoring of RosettaLigand code (Fleishman, Leaver-Fay et al. 2011). Kristian Kaufmann collected the initial binding affinity data and HIV-1 template structures used in HIV-1 PR/PI  $\Delta\Delta G$  predictions. The

CSAR dataset used in chapter 4 is a community resource prepared for the uniform evaluation of docking and scoring methods (Dunbar, Smith et al. 2011). Jens Meiler helped review and revise of the text of chapters 2-4. Figure I-1 is taken from the original RosettaLigand publication (Meiler and Baker 2006). Brittany Allison contributed Figure III-2. The fragment-based docking algorithm presented in Appendix I was inspired by similar approaches for incremental construction (Rarey, Kramer et al. 1996).

## References

- Cheng, T., X. Li, et al. (2009). "Comparative assessment of scoring functions on a diverse test set." Journal of Chemical Information and Modeling **49**(4): 1079-1093.
- Dunbar, J. B., Jr., R. D. Smith, et al. (2011). "CSAR benchmark exercise of 2010: selection of the protein-ligand complexes." Journal of Chemical Information and Modeling **51**(9): 2036-2046.
- Fleishman, S. J., A. Leaver-Fay, et al. (2011). "RosettaScripts: a scripting language interface to the Rosetta macromolecular modeling suite." PLoS One **6**(6): e20161.
- Kim, R. and J. Skolnick (2008). "Assessment of programs for ligand binding affinity prediction." Journal of Computational Chemistry **29**(8): 1316-1331.
- Lemmon, G., K. Kaufmann, et al. (2012). "Prediction of HIV-1 Protease/Inhibitor Affinity using RosettaLigand." Chemical Biology and Drug Design **79**(6): 888-896.
- Lemmon, G. and J. Meiler (2012). "Rosetta Ligand docking with flexible XML protocols." Methods in Molecular Biology **819**: 143-155.
- Meiler, J. and D. Baker (2006). "ROSETTALIGAND: protein-small molecule docking with full side-chain flexibility." Proteins **65**(3): 538-548.
- Rarey, M., B. Kramer, et al. (1996). "A fast flexible docking method using an incremental construction algorithm." Journal of Molecular Biology **261**(3): 470-489.
- Zoete, V., A. Grosdidier, et al. (2009). "Docking, virtual high throughput screening and in silico fragment-based drug design." Journal of Cellular and Molecular Medicine **13**(2): 238-248.

# CHAPTER 1

## INTRODUCTION

The development of advanced, accurate structure-based drug design (SBDD) has been seen as a promising research direction for at least 4 decades (Fastier 1964). Even as techniques for structural elucidation began to develop (Kendrew, Bodo et al. 1958), chemists realized the potential for designing drugs based on these structures (Beddell, Goodford et al. 1976; Cushman, Cheung et al. 1977; Matthews, Alden et al. 1977). Yet as promising as rational design appears, random methods such as high throughput screening (HTS) (Gane and Dean 2000), remain a large component of drug discovery efforts (Bibette 2012). This is largely due to an inability to accurately predict the structure and thermodynamics of the protein-ligand complex (Hubbard 1997). Nevertheless SBDD has been coupled with HTS as a way to reduce the number of molecules to be screened. Sadly, using both random and rational methods, the pharmaceutical industry has seen a continued decline in productivity over several decades (Brown and Superti-Furga 2003; Macarron, Banks et al. 2011).

Computational methods were championed in the early 1980s as the necessary tools that would bring success to the field of structure-based drug design (Goodford 1984). Early exponents of computer-aided drug design (CADD) understood that accurate predictions would involve modeling of flexibility, solvation and entropic effects (Marshall 1987). Yet decades later these challenges persist (Waszkowycz, Clark et al. 2011). With continued technological and methodological advances, however, recent years have seen steady advancement in CADD prediction quality (Durrant and McCammon 2010; Xiang, Cao et al. 2012).



Ligand docking is an element of CADD that attempts to predict how a small molecule will interact with a given protein target. This is an essential step towards CADD, since designed small molecules are evaluated based on their ability to bind to their target. Ligand docking methods have proven successful in cases where the protein and/or ligand structures are well known and fairly rigid. However, flexible proteins and large flexible ligands continue to present a challenge to ligand docking applications (Huang and Zou 2010).

HIV-1 protease (PR) is an example of highly flexible protein and HIV-1 protease inhibitors (PIs) are large flexible ligands (Louis, Ishima et al. 2007). Functional PR mutants abound, leading to additional conformational diversity and the ability to develop PI resistance (Bennett, Camacho et al. 2009). These 3 types of flexibility we believe are the culprit behind poor PR/PI binding affinity predictions (Cheng, Li et al. 2009). In chapter 2 we discuss ways to improve predictions of protease inhibitor binding affinity using RosettaLigand software (Lemmon, Kaufmann et al. 2012).

In chapter 3 we showcase a new version of RosettaLigand software which allows efficient modeling of ligands with many rotatable bonds, such as certain HIV-1 protease inhibitors. The new software also allows for docking of multiple ligands, small molecules, ions and cofactors simultaneously. Additionally, we implement the rudiments of ligand design – specifically the building of novel small molecules from fragments (Lemmon and Meiler 2012).

In chapter 4 the multiple ligand docking introduced in chapter 3 is used to explore the role of water molecules in the accuracy of ligand docking studies. HIV-1 protease predictions improve when a key water is considered. Additionally, the CSAR dataset of diverse protein/ligand complexes is used to examine the effect of water on docking predictions.

## Ligand Docking: Sampling

Ligand docking seeks to predict the interaction between a protein and a small molecule. A number of applications exist for predicting protein/small molecule interactions. These include FlexX (Rarey, Kramer et al. 1996), DOCK (Ewing, Makino et al. 2001), ICM (Abagyan, Totrov et al. 1994), QXP (McMartin and Bohacek 1997), Prodock (Trosset and Scheraga 1999), Pro\_LEADS (Baxter, Murray et al. 1998), Hammerhead (Welch, Ruppert et al. 1996), FLOG (Martin 2007), GOLD (Gohlke, Kuhn et al. 2004), AutoDock (Langer, Li et al. 2008), FlipDock (Zhao and Sanner 2007), GREEN (Johnson, Evanoff et al. 2008), and Glide (Martinez-Ramirez, Jeurond et al. 2008). While these programs perform well with small rigid proteins, most struggle to correctly predict interactions that involve conformational selection or induced-fit effects (Taylor, Jewsbury et al. 2002).

Accurate docking must take into account the flexibility of both the protein and the ligand. The many degrees of freedom involved in protein flexibility makes this problem appear computationally intractable; therefore many docking applications maintain rigid proteins (Taylor, Jewsbury et al. 2002). A few programs, including DOCK (Ewing, Makino et al. 2001), GOLD (Gohlke, Kuhn et al. 2004), and AutoDock (Langer, Li et al. 2008) allow limited receptor flexibility through side chain torsion angle sampling within the active site. Others such as FlexE (Simons, Ruczinski et al. 1999) represent receptor flexibility through user provided ensembles of protein conformations. While some docking software now allows *limited* protein flexibility, an efficient representation of the protein flexibility seen in nature is still needed (Durrant and McCammon 2010).

Since docking applications have generally incorporated options for flexibility as an afterthought, they fail to fully capture flexibility in a scalable manner. Rosetta, on the other

hand, was developed initially as an application for protein structure prediction. As such, efficient and scalable protein flexibility is an integral part of the code base.

Techniques for modeling ligand flexibility can be grouped into 3 categories: random/stochastic methods, simulation methods, and systematic methods (Sousa, Fernandes et al. 2006). Random/Stochastic methods include Monte Carlo approaches, genetic algorithms, and Tabu searches. Simulation methods include molecular dynamics and energy minimization. Systematic methods include conformational search (sampling of all torsion angle combinations), fragmentation, and database methods using conformational ensembles (Sousa, Fernandes et al. 2006).

Systematic conformational searches can become unreasonably time-consuming. For example, using a strategy that samples only small perturbations in ligand torsion angles, RosettaLigand can handle ligands with no more than 7 rotatable bonds (Kaufmann, Glab et al. 2008). Sampling from experimentally derived structural ensembles is limited by the number and diversity of conformations available. Yet for large flexible ligands, in silico conformer generation can lead to ensembles too large for efficient sampling.

Therefore, we suggest a ligand fragment rotamer selection strategy similar to those used for protein side-chain sampling. This strategy splits the ligand into several fragments. A conformer library derived from experimental structures is generated for each fragment. RosettaLigand samples ligand conformations one fragment at a time, thereby reducing the problem of combinatorial explosion, while retaining the efficiency of knowledge-based conformer sampling.

The ability to dock multiple ligands simultaneously may be necessary to find the correct binding mode. Sequential docking of two ligands is not sufficient, because induced-fit

effects may require the presence of several ligands, cofactors, and metal ions (Sousa, Fernandes et al. 2006). Simultaneous docking of multiple small molecules is a missing feature in extant docking software.

## **Ligand Docking: Scoring**

As ligand docking applications sample ligand position and orientation, and protein and ligand flexibility, assessments must be made as to the quality of these models. Numerous score functions are available for assessing protein/ligand conformations. Because of the scale of the sampling problem, score functions often sacrifice accuracy for speed. Accuracy and speed must be delicately balanced when choosing a score function for ligand docking.

Score functions can be categorized as force-field based, empirical, and knowledge-based (Sousa, Fernandes et al. 2006). Molecular mechanics force-fields such as AMBER (Weiner and Kollman 1981) and CHARMM (Brooks, Bruccoleri et al. 1983) calculate internal ligand energies as well as protein/ligand interaction energies, and favor accuracy over speed. Empirical score functions such as ChemScore (Eldridge, Murray et al. 1997) and SCORE (Wang, Liu et al. 1998) and Bohm's score function (Bohm 1994) are trained to predict binding affinities using experimental data. These methods are fast, but limited by the datasets used in training. Knowledge based score functions such as DrugScore (Velec, Gohlke et al. 2005) are derived from statistics of physical parameters found in experimentally determined protein/ligand complexes. These methods are very efficient, but only indirectly predict binding affinity. Instead scores relate to how closely a model's physical parameters recapitulate those seen in structural databases (Sousa, Fernandes et al. 2006).

The accuracy of score functions can be assessed by comparing their scores with measured binding affinities. Often  $K_i$  data is more readily available than binding affinity measurements. In these cases the equation  $\Delta G^0 = -RT \ln K_i$  can be used to convert inhibition constants to binding affinities. This conversion assumes that  $K_i$  values were obtained under the quasi-equilibrium conditions of Michaelis-Menton kinetics (Jenwitheesuk and Samudrala 2003). Additionally it assumes inhibition measurements are a suitable proxy for equilibrium constants. This should hold in the case of competitive inhibition where the rate of enzyme/inhibitor complex dissociation is much lower than the rate of enzyme/substrate dissociation. Finally, measurements must be made under conditions of free diffusion.

Such conversions may also be hampered by the fact that  $K_i$ s are measured using a variety of experimental techniques. We considered these limitations, however in chapter 2 we find that where HIV-1 protease/inhibitor binding affinity and  $K_i$  data are available, the conversion of  $K_i$  to binding affinity is appropriate. The standard deviation between ITC measurements and converted  $K_i$  values is 1.07 kJ/mol.

## **Ligand Design**

Rational drug design has been described as 3 problems: (1) Construction: how can we assemble synthetically feasible novel structures? (2) Docking: how does a virtual ligand interact with the receptor? (3) Scoring: how can the quality of a designed structure be estimated (Schneider and Bohm 2002)? Although the field of structure-based drug design acknowledges the importance of receptor and ligand flexibility in accurately accomplishing step 2 (docking), static structures are still generally used for this purpose (Cozzini, Kellogg et al. 2008). Our proposed method allows receptor flexibility.

Many ligand design strategies employ a fragment search approach. For instance, Dakshanamurthy et al. designed inhibitors of VEGFR2 kinase. The design process consisted of (1) collecting a database of organic fragments, (2) using the UNITY module of Sybyl (Ghose, Jaeger et al. 1993) to find fragments with key characteristics, (3) searching for these fragments in a database of 82 million organic compounds, and (4) docking filtered results using FlexX (Dakshanamurthy, Kim et al. 2007). Of 613 compounds with a FlexX predicted binding affinity below a cutoff of -20 kcal/mol, 17 were acquired for additional testing. Of these 17 compounds, 7 significantly inhibited angiogenesis. Our method differs in that it presents an integrated and more fully automated approach within Rosetta. RosettaLigandDesign will be capable of database search, filtering, and docking, as well as assembling novel ligands from fragments.

Structure generating applications such as BUILDER (Lewis, Roe et al. 1992), CLIX (Lawrence and Davis 1992), GROUPBUILD (Rotstein and Murcko 1993), HOOK (Eisen, Wiley et al. 1994), LEGEND (Nishibata and Itai 1991), LUDI (Bohm 1992), and SPROUT (Gillet, Johnson et al. 1993) construct small molecules, often starting with known key fragments. Some first construct a skeleton that spans the binding pocket and then find fragments to fill the skeleton. Others first place key fragments and then build an interconnecting skeleton. LEGEND (Nishibata and Itai 1991), GEMINI (Singh, Saldanha et al. 1991), GROW (Moon and Howe 1991), GROUPBUILD (Rotstein and Murcko 1993), and GenStar (Rotstein and Murcko 1993) all use a fragment-extension approach. Fragments are added sequentially around a key fragment, without the use of a skeleton (Burt, Hutchins et al. 1997).

These methods often lack automation and require prior knowledge to be effective. This knowledge includes correct selection of a key starting fragment, and placement of the fragment in the correct starting position. Many require the user to dock large collections of generated structures using another application. A more intelligent algorithm that integrated sampling and scoring could inform its sampling approach based on the results from scoring the generated structures. RosettaLigandDesign integrates design with docking. All of the listed methods represent the receptor as a rigid body, whereas Rosetta has the power to efficiently model protein flexibility during design.

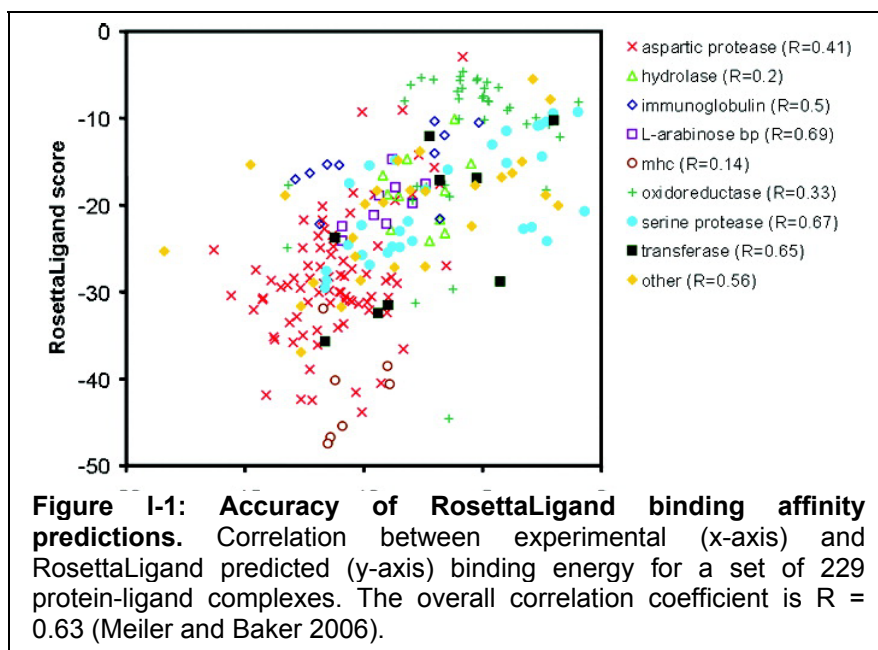
## **Rosetta**

Rosetta (Kaufmann, Lemmon et al. 2010) is an open source coding framework for a suite of applications for ab initio protein structure prediction, homology modeling, protein design, loop building, protein-protein interactions, ligand docking and more. Rosetta is freely available to the academic community and developed collaboratively at multiple universities and institutes. It has proven successful in multiple CASP experiments, where blind protein structure predictions from experts in computational structural biology are compared with experimental structures. In general, Rosetta predictions rely on a large number of independent folding simulations and selection of the lowest energy structures. Rosetta allows for both low-resolution centroid based scoring as well as high resolution atomic level scoring. Score terms include the 6-12 Lennard-Jones potential (Lennard-Jones 1924), the Lazaridis-Karplus implicit solvation model (Lazaridis and Karplus 1999), orientation-dependent hydrogen bonds (Morozov, Kortemme et al. 2004), a side-chain torsional potential derived from the Dunbrack backbone-dependent rotamer library set (Dunbrack and Karplus 1993), a backbone torsional potential dependent on secondary structure

(Kuhlman, Dantas et al. 2003), and a pair potential which accounts for long-range electrostatic interactions between polar atoms and  $\pi$ - $\pi$  and cation- $\pi$  interactions (Simons, Ruczinski et al. 1999). Current work in the Meiler Lab is directed toward representing the correct geometries of atomic orbitals and developing a score term that explicitly captures  $\pi$ - $\pi$  and cation- $\pi$  interactions.

## RosettaLigand

RosettaLigand was released in 2006 as an application for small molecule docking (Meiler and Baker 2006). It uses a Monte Carlo minimization approach, minimizing the protein side chain torsion angles. It also provides for



ligand flexibility and introduces limited backbone flexibility in the proximity of the ligand (Davis and Baker 2009). RosettaLigand prediction quality is estimated in Figure I-1. Note that RosettaLigand predictions within the aspartic proteases class, which includes HIV-1 protease, were poor. This obvious need for improvement was a motivation for this research.

## HIV/AIDS

Human Immunodeficiency Virus is a serious global health problem. As of 2005 over 40 million people were living with HIV worldwide (2005). The United States 2009 Federal Budget



includes an estimated \$24.1 billion (2008) for AIDS research and treatment. Much of these costs support the development of new anti-retroviral drugs aimed at various HIV targets.

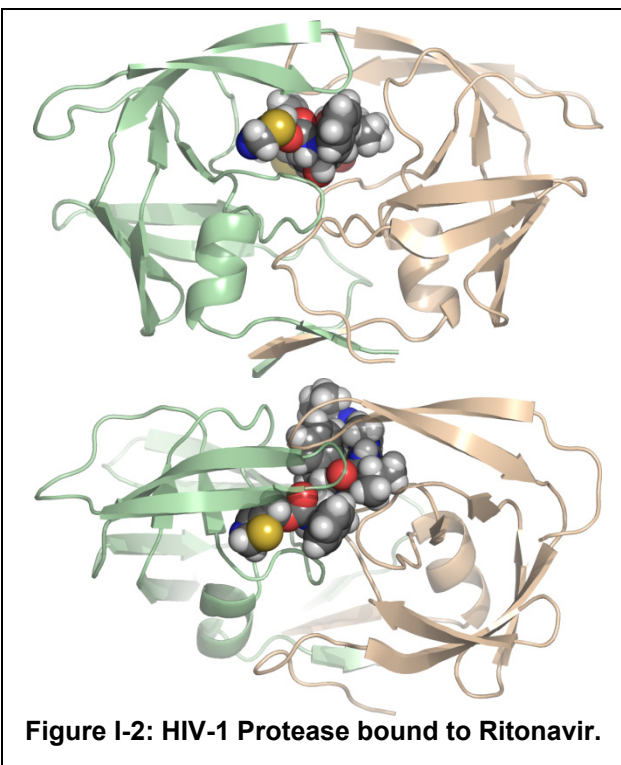
Because of its very high mutation rate, HIV has been successful at quickly becoming drug resistant upon treatment. Thus HIV inhibitors must be designed to bind broadly enough to neutralize the diversity of conformations their target can assume. Unfortunately, lack of strain specificity can lead to the need for high doses (because of low binding affinity) and high host toxicity (Shaikh, Jain et al. 2007). An alternative strategy would be to develop strain specific inhibitors. As HIV-1 continues to evolve drug resistance, the ability to quickly design and assess novel strain-specific drugs will be of paramount importance as we continue to combat HIV.

The HIV life cycle can be considered in 8 stages. First, HIV surface glycoproteins bind receptors on  $CD4^+$  T cells and macrophages, which leads to fusion of the adjacent membranes and delivery of capsid into a cell (Chan D 1998). Second, viral single stranded RNA is released from the viral proteins and a DNA complement is constructed using reverse transcriptase. Third, the DNA complement enters the nucleus and the viral enzyme called integrase assists its integration into the host genome. Fourth, T-cell activation leads to transcription of viral pro-RNA. Fifth, pro-RNA is cut into smaller pieces and translated into Tat, Rev, Gag and Env- the essential proteins for virion construction. Sixth, the Env polyprotein is transported through the ER and golgi where it is cleaved by HIV protease into 2 structural proteins. These proteins anchor to the cell membrane, are necessary for budding, and will become essential structural components of the developing virions. Seventh, other proteins assemble at the surface, and virus budding occurs. Eighth, HIVprotease continues to cleave polyproteins in the immature virion into functional proteins (Frankel and Young 1998) (Adamson and Freed 2007).

## HIV Protease

HIV protease's first role is in cleaving the Env polyprotein prior to virion budding. Before HIV virions can become infectious, HIV protease must cleave Gag polyprotein into 4 functional proteins. Additionally it must cleave Pol polyprotein into reverse transcriptase and more HIV protease (Nicholson, Yamazaki et al. 1995).

The structure of HIV protease is a dimer of two identical 99 amino acid subunits (Figure I-2). Each subunit contains 2 anti-



parallel  $\beta$ -sheets and 1  $\alpha$ -helix. The active site is part of a 4 stranded  $\beta$ -turn. It lies between the two chains, and generally contains the Asp-Thr-Gly signature of aspartic proteases (Piana and Carloni 2000). During enzyme/ligand binding, two flap regions are displaced by up to 7 Å (Louis, Ishima et al. 2007). At least 206 high resolution (2 Å or better) HIV-1 protease crystal structures exist in the protein databank. Most of these bind to a small molecule inhibitor.

## HIV Protease Inhibitors

Ten protease inhibitors are currently FDA approved for HIV treatment. A few of these are represented in Figure I-3. Each functions as a competitive inhibitor, binding in the active site through a network of hydrogen bonds, precluding substrate binding (Mastrolorenzo, Rusconi et al. 2007). Because of the high mutation rate of retroviruses, HIV protease mutations routinely

outmaneuver inhibitor treatment strategies (Rhee, Fessel et al. 2005). Drug cocktails containing several protease inhibitors resist such adaptations by requiring multiple mutations to overcome inhibitor binding. Nevertheless, even these treatments often fail.

HIV protease inhibitors have been an important and successful model for structure-assisted drug design (Wlodawer and Vondrasek 1998). Crystallography, NMR and computational studies have supported strategic improvements to each new inhibitor, and have

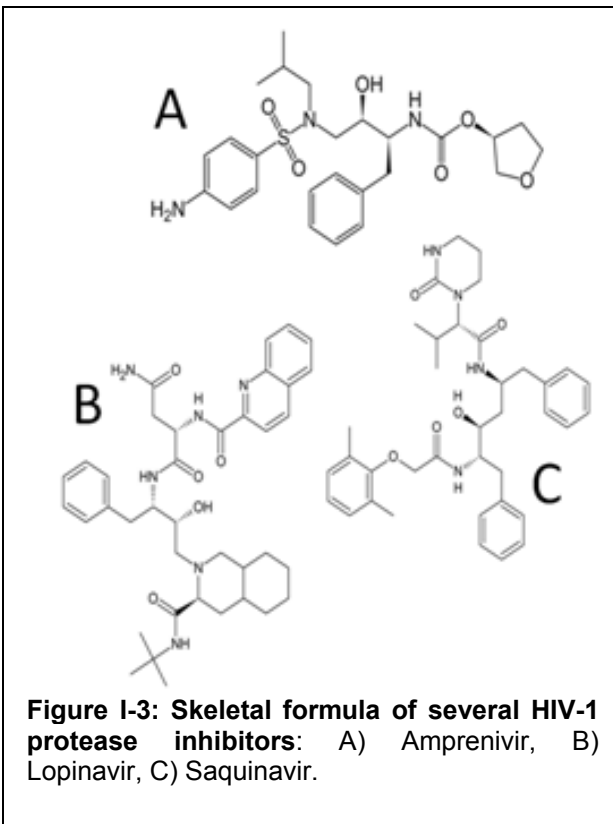
allowed researchers to identify the mechanisms of resistance exposed through protease mutation (Shenderovich, Kagan et al. 2003). However with the burgeoning global impact of HIV, there is a need to streamline the design and development of novel protease inhibitors that will resist mutational evasion.

## References

United Nations Programme on HIV/AIDS, World Health Organization (2005). "AIDS epidemic update."

The Henry J Kaiser Family Foundation (2008). "HIV/AIDS policy fact sheet."

Abagyan, R., M. Totrov, et al. (1994). "Icm - a New Method for Protein Modeling and Design - Applications to Docking and Structure Prediction from the Distorted Native Conformation." *Journal of Computational Chemistry* **15**(5): 488-506.



**Figure 1-3: Skeletal formula of several HIV-1 protease inhibitors: A) Amprenavir, B) Lopinavir, C) Saquinavir.**

- Adamson, C. S. and E. O. Freed (2007). "Human immunodeficiency virus type 1 assembly, release, and maturation." Advances in Pharmacology **55**: 347-387.
- Baxter, C. A., C. W. Murray, et al. (1998). "Flexible docking using Tabu search and an empirical estimate of binding affinity." Proteins-Structure Function and Genetics **33**(3): 367-382.
- Beddell, C. R., P. J. Goodford, et al. (1976). "Compounds designed to fit a site of known structure in human haemoglobin." British Journal of Pharmacology **57**(2): 201-209.
- Bennett, D. E., R. J. Camacho, et al. (2009). "Drug resistance mutations for surveillance of transmitted HIV-1 drug-resistance: 2009 update." PLoS One **4**(3): e4724.
- Bibette, J. (2012). "Gaining confidence in high-throughput screening." Proceedings of the National Academy of Sciences of the United States of America **109**(3): 649-650.
- Bohm, H. J. (1992). "LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads." Journal of Computer Aided Molecular Design **6**(6): 593-606.
- Bohm, H. J. (1994). "The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure." Journal of Computer Aided Molecular Design **8**(3): 243-256.
- Brooks, B. R., R. E. Bruccoleri, et al. (1983). "Charmm - a Program for Macromolecular Energy, Minimization, and Dynamics Calculations." Journal of Computational Chemistry **4**(2): 187-217.
- Brown, D. and G. Superti-Furga (2003). "Rediscovering the sweet spot in drug discovery." Drug Discovery Today **8**(23): 1067-1077.
- Burt, S., C. Hutchins, et al. (1997). "A Monte Carlo method for finding important ligand fragments from receptor data." Journal of Computer Aided Molecular Design **11**(3): 243-255.
- Chan D, K. P. (1998). ""HIV entry and its inhibition"." Cell **93**(5): 681-684.
- Cheng, T., X. Li, et al. (2009). "Comparative assessment of scoring functions on a diverse test set." Journal of Chemical Information and Modeling **49**(4): 1079-1093.
- Cozzini, P., G. E. Kellogg, et al. (2008). "Target flexibility: an emerging consideration in drug discovery and design." Journal of Medicinal Chemistry **51**(20): 6237-6255.
- Cushman, D. W., H. S. Cheung, et al. (1977). "Design of potent competitive inhibitors of angiotensin-converting enzyme. Carboxyalkanoyl and mercaptoalkanoyl amino acids." Biochemistry **16**(25): 5484-5491.

- Dakshnamurthy, S., M. Kim, et al. (2007). "In-silico fragment-based identification of novel angiogenesis inhibitors." Bioorganic and Medicinal Chemistry Letters **17**(16): 4551-4556.
- Davis, I. W. and D. Baker (2009). "RosettaLigand docking with full ligand and receptor flexibility." Journal of Molecular Biology **385**(2): 381-392.
- Dunbrack, R. L., Jr. and M. Karplus (1993). "Backbone-dependent rotamer library for proteins. Application to side-chain prediction." Journal of Molecular Biology **230**(2): 543-574.
- Durrant, J. D. and J. A. McCammon (2010). "Computer-aided drug-discovery techniques that account for receptor flexibility." Current Opinion in Pharmacology **10**(6): 770-774.
- Eisen, M. B., D. C. Wiley, et al. (1994). "Hook - a Program for Finding Novel Molecular Architectures That Satisfy the Chemical and Steric Requirements of a Macromolecule Binding-Site." Proteins-Structure Function and Genetics **19**(3): 199-221.
- Eldridge, M. D., C. W. Murray, et al. (1997). "Empirical scoring functions. 1. 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.
- Ewing, T. J. A., S. Makino, et al. (2001). "DOCK 4.0: Search strategies for automated molecular docking of flexible molecule databases." Journal of Computer-Aided Molecular Design **15**(5): 411-428.
- Fastier, F. N. (1964). "Modern Concepts in Relationship Between Structure and Biological Activity." Annual Review of Pharmacology **4**: 51-58.
- Frankel, A. D. and J. A. Young (1998). "HIV-1: fifteen proteins and an RNA." Annual Review of Biochemistry **67**: 1-25.
- Gane, P. J. and P. M. Dean (2000). "Recent advances in structure-based rational drug design." Current Opinion in Structural Biology **10**(4): 401-404.
- Ghose, A. K., E. P. Jaeger, et al. (1993). "Conformational Searching Methods for Small Molecules .1. Study of the Sybyl Search Method." Journal of Computational Chemistry **14**(9): 1050-1065.
- Gillet, V., A. P. Johnson, et al. (1993). "Sprout - a Program for Structure Generation." Journal of Computer-Aided Molecular Design **7**(2): 127-153.
- Gohlke, H., L. A. Kuhn, et al. (2004). "Change in protein flexibility upon complex formation: analysis of Ras-Raf using molecular dynamics and a molecular framework approach." Proteins **56**(2): 322-337.

- Goodford, P. J. (1984). "Drug design by the method of receptor fit." Journal of Medicinal Chemistry **27**(5): 558-564.
- Huang, S. Y. and X. Zou (2010). "Advances and challenges in protein-ligand docking." International Journal of Molecular Sciences **11**(8): 3016-3034.
- Hubbard, R. E. (1997). "Can drugs be designed?" Current Opinion in Biotechnology **8**(6): 696-700.
- Jenwitheesuk, E. and R. Samudrala (2003). "Improved prediction of HIV-1 protease-inhibitor binding energies by molecular dynamics simulations." BMC Structural Biology **3**: 2.
- Johnson, J. W., D. P. Evanoff, et al. (2008). "Cyclobutanone mimics of penicillins: effects of substitution on conformation and hemiketal stability." Journal of Organic Chemistry **73**(18): 6970-6982.
- Kaufmann, K. W., K. Glab, et al. (2008). Small Molecule Rotamers Enable Simultaneous Optimization of Small Molecule and Protein Degrees of Freedom in ROSETTALIGAND Docking. German Conference on Bioinformatics. A. Beyer and M. Schroeder. Dresden: 148-157.
- Kaufmann, K. W., G. H. Lemmon, et al. (2010). "Practically useful: what the Rosetta protein modeling suite can do for you." Biochemistry **49**(14): 2987-2998.
- Kendrew, J. C., G. Bodo, et al. (1958). "A three-dimensional model of the myoglobin molecule obtained by x-ray analysis." Nature **181**(4610): 662-666.
- Kuhlman, B., G. Dantas, et al. (2003). "Design of a novel globular protein fold with atomic-level accuracy." Science **302**(5649): 1364-1368.
- Langer, V., S. Li, et al. (2008). "Conformation of 1-aryl-2-aryloxy-1-propanols: threo-1-(4-hydroxy-3-methoxyphenyl)-2-(2-methoxyphenoxy)-1-propanol." Acta Crystallographica C **64**(Pt 10): o558-560.
- Lawrence, M. C. and P. C. Davis (1992). "CLIX: a search algorithm for finding novel ligands capable of binding proteins of known three-dimensional structure." Proteins **12**(1): 31-41.
- Lazaridis, T. and M. Karplus (1999). "Effective energy function for proteins in solution." Proteins **35**(2): 133-152.
- Lemmon, G., K. Kaufmann, et al. (2012). "Prediction of HIV-1 Protease/Inhibitor Affinity using RosettaLigand." Chemical Biology and Drug Design **79**(6): 888-896.
- Lemmon, G. and J. Meiler (2012). "Rosetta Ligand docking with flexible XML protocols." Methods in Molecular Biology **819**: 143-155.

- Lennard-Jones, J. E. (1924). "On the Determination of Molecular Fields. II. From the Equation of State of a Gas." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **106**: 463-477.
- Lewis, R. A., D. C. Roe, et al. (1992). "Automated Site-Directed Drug Design Using Molecular Lattices." Journal of Molecular Graphics **10**(2): 66-78.
- Louis, J. M., R. Ishima, et al. (2007). "HIV-1 protease: structure, dynamics, and inhibition." Advances in Pharmacology **55**: 261-298.
- Macarron, R., M. N. Banks, et al. (2011). "Impact of high-throughput screening in biomedical research." Nature Reviews Drug Discovery **10**(3): 188-195.
- Marshall, G. R. (1987). "Computer-aided drug design." Annul Reviews of Pharmacology and Toxicology **27**: 193-213.
- Martin, S. F. (2007). "Preorganization in biological systems: Are conformational constraints worth the energy?" Pure Applied Chemistry **79**(2): 193-200.
- Martinez-Ramirez, H. R., E. A. Jeurond, et al. (2008). "Dynamics of body protein deposition and changes in body composition after sudden changes in amino acid intake: I. Barrows." Journal of Animal Science **86**(9): 2156-2167.
- Mastrolorenzo, A., S. Rusconi, et al. (2007). "Inhibitors of HIV-1 protease: current state of the art 10 years after their introduction. From antiretroviral drugs to antifungal, antibacterial and antitumor agents based on aspartic protease inhibitors." Current Medicinal Chemistry **14**(26): 2734-2748.
- Matthews, D. A., R. A. Alden, et al. (1977). "Dihydrofolate reductase: x-ray structure of the binary complex with methotrexate." Science **197**(4302): 452-455.
- McMartin, C. and R. S. Bohacek (1997). "QXP: powerful, rapid computer algorithms for structure-based drug design." Journal of Computer Aided Molecular Design **11**(4): 333-344.
- Meiler, J. and D. Baker (2006). "ROSETTALIGAND: protein-small molecule docking with full side-chain flexibility." Proteins **65**(3): 538-548.
- Moon, J. B. and W. J. Howe (1991). "Computer design of bioactive molecules: a method for receptor-based de novo ligand design." Proteins **11**(4): 314-328.
- Morozov, A. V., T. Kortemme, et al. (2004). "Close agreement between the orientation dependence of hydrogen bonds observed in protein structures and quantum mechanical

- calculations." Proceedings of the National Academy of Sciences of the United States of America **101**(18): 6946-6951.
- Nicholson, L. K., T. Yamazaki, et al. (1995). "Flexibility and function in HIV-1 protease." Nature Structural Biology **2**(4): 274-280.
- Nishibata, Y. and A. Itai (1991). "Automatic Creation of Drug Candidate Structures Based on Receptor Structure - Starting Point for Artificial Lead Generation." Tetrahedron **47**(43): 8985-8990.
- Piana, S. and P. Carloni (2000). "Conformational flexibility of the catalytic Asp dyad in HIV-1 protease: An ab initio study on the free enzyme." Proteins **39**(1): 26-36.
- Rarey, M., B. Kramer, et al. (1996). "A fast flexible docking method using an incremental construction algorithm." Journal of Molecular Biology **261**(3): 470-489.
- Rhee, S. Y., W. J. Fessel, et al. (2005). "HIV-1 protease and reverse-transcriptase mutations: Correlations with antiretroviral therapy in subtype B isolates and implications for drug-resistance surveillance." Journal of Infectious Diseases **192**(3): 456-465.
- Rotstein, S. H. and M. A. Murcko (1993). "GenStar: a method for de novo drug design." Journal of Computer Aided Molecular Design **7**(1): 23-43.
- Rotstein, S. H. and M. A. Murcko (1993). "GroupBuild: a fragment-based method for de novo drug design." Journal of Medicinal Chemistry **36**(12): 1700-1710.
- Schneider, G. and H. J. Bohm (2002). "Virtual screening and fast automated docking methods." Drug Discovery Today **7**(1): 64-70.
- Shaikh, S. A., T. Jain, et al. (2007). "From drug target to leads-sketching a physicochemical pathway for lead molecule design in silico." Current Pharmaceutical Design **13**(34): 3454-3470.
- Shenderovich, M. D., R. M. Kagan, et al. (2003). "Structure-based phenotyping predicts HIV-1 protease inhibitor resistance." Protein Science **12**(8): 1706-1718.
- Simons, K. T., I. Ruczinski, et al. (1999). "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins." Proteins **34**(1): 82-95.
- Singh, J., J. Saldanha, et al. (1991). "A novel method for the modelling of peptide ligands to their receptors." Protein Engineering **4**(3): 251-261.
- Sousa, S. F., P. A. Fernandes, et al. (2006). "Protein-ligand docking: current status and future challenges." Proteins **65**(1): 15-26.



- Taylor, R. D., P. J. Jewsbury, et al. (2002). "A review of protein-small molecule docking methods." Journal of Computer-Aided Molecular Design **16**(3): 151-166.
- Trosset, J. Y. and H. A. Scheraga (1999). "PRODOCK: Software package for protein modeling and docking." Journal of Computational Chemistry **20**(4): 412-427.
- Veleg, H. F., H. Gohlke, et al. (2005). "DrugScore(CSD)-knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction." Journal of Medicinal Chemistry **48**(20): 6296-6303.
- Wang, R. X., L. Liu, et al. (1998). "SCORE: A new empirical method for estimating the binding affinity of a protein-ligand complex." Journal of Molecular Modeling **4**(12): 379-394.
- Waszkowycz, B., D. E. Clark, et al. (2011). "Outstanding challenges in protein-ligand docking and structure-based virtual screening." Wiley Interdisciplinary Reviews-Computational Molecular Science **1**(2): 229-259.
- Weiner, P. K. and P. A. Kollman (1981). "Amber - Assisted Model-Building with Energy Refinement - a General Program for Modeling Molecules and Their Interactions." Journal of Computational Chemistry **2**(3): 287-303.
- Welch, W., J. Ruppert, et al. (1996). "Hammerhead: Fast, fully automated docking of flexible ligands to protein binding sites." Chemistry & Biology **3**(6): 449-462.
- Wlodawer, A. and J. Vondrasek (1998). "Inhibitors of HIV-1 protease: a major success of structure-assisted drug design." Annual Review of Biophysics & Biomolecular Structure **27**: 249-284.
- Xiang, M. L., Y. Cao, et al. (2012). "Computer-Aided Drug Design: Lead Discovery and Optimization." Combinatorial Chemistry & High Throughput Screening **15**(4): 328-337.
- Zhao, Y. and M. F. Sanner (2007). "FLIPDock: docking flexible ligands into flexible receptors." Proteins **68**(3): 726-737.

## CHAPTER 2

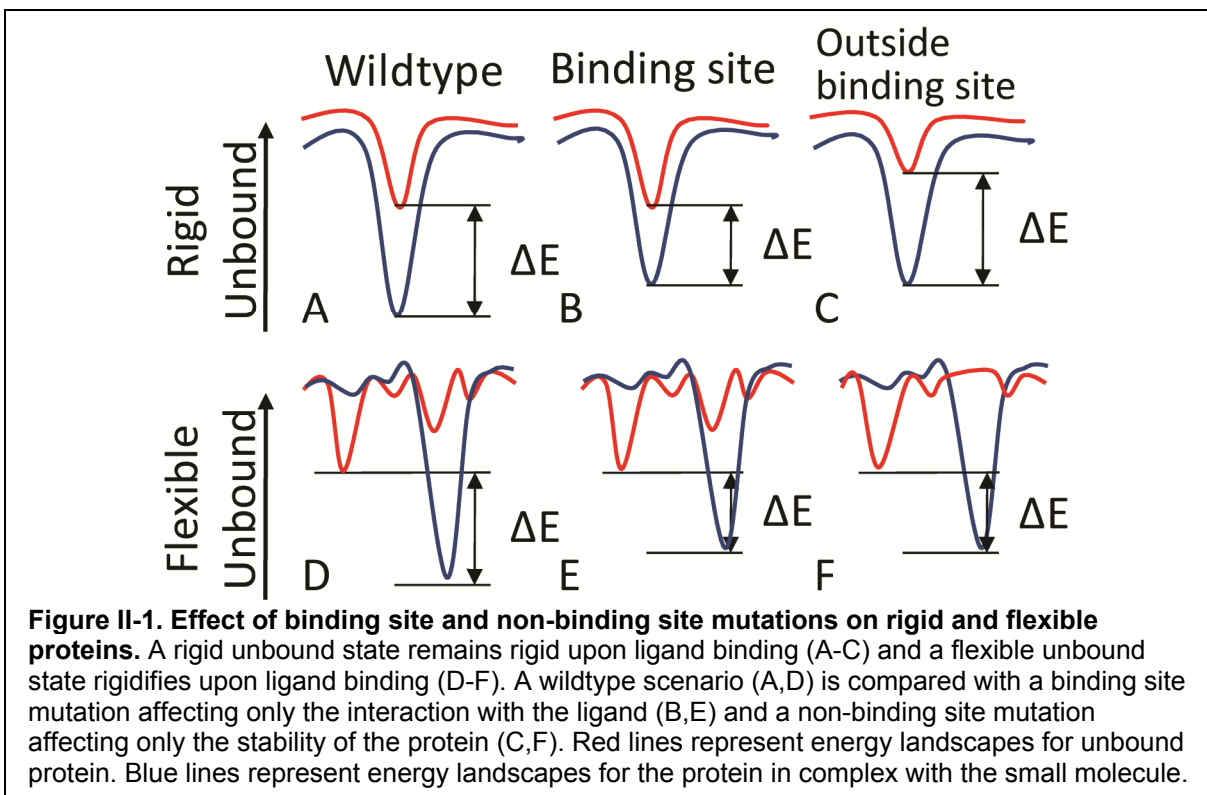
# PREDICTION OF HIV-1 PROTEASE/INHIBITOR AFFINITY USING ROSETTALIGAND

This chapter is adapted from an article published in *Chemical Biology and Drug Design* under the same title (Lemmon, Kaufmann et al. 2012).

### **Introduction**

The binding affinity of a drug to its protein target is defined by the free energy difference between the bound and unbound state. Mutation of the protein or chemical modification of the ligand can alter this energy difference directly – i.e. by adding or subtracting interactions between the two partners – or indirectly – i.e. by stabilizing or destabilizing protein or small molecule in either bound or unbound conformation (Shimotohno, Oue et al. 2001). For the unbound state, often ensembles of protein and small molecule need to be considered (Henzler-Wildman and Kern 2007) while the bound state is often considerably more rigid. HIV-1 protease (PR) interaction with its inhibitors is a model case for this scenario while examples for the opposite scenario – rigid protein increases flexibility upon binding – are also known (Gohlke, Kuhn et al. 2004; Martin 2007).

Current computational methods are capable of predicting direct effects reasonably well through an analysis of all interactions between protein and ligand. However, the same methods often fail to predict indirect effects. For instance it remains difficult to predict how mutations outside the binding pocket are propagated throughout the protein and to the binding site (Sousa, Fernandes et al. 2006). These indirect effects are likely to have greater destabilizing influence on

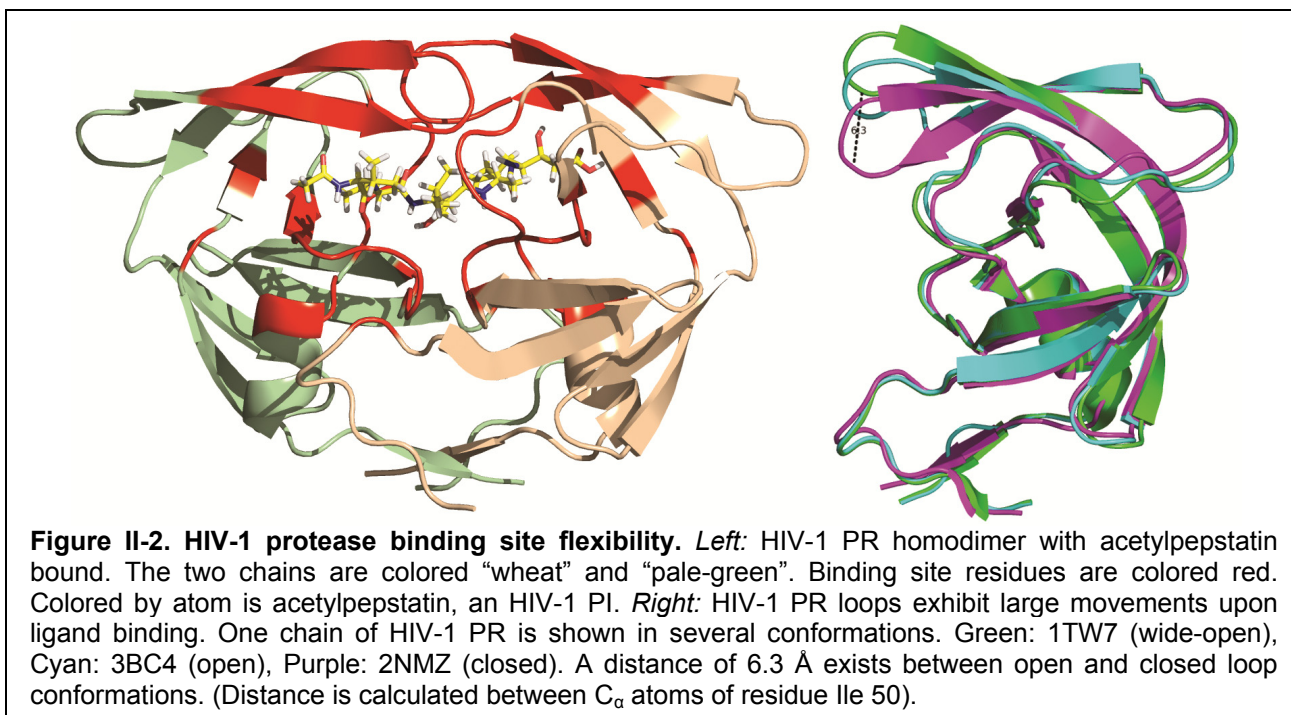


a rigid-bound state than on a flexible unbound state. Since the flexible unbound state is already unstable, mutation is unlikely to disrupt its structure or drastically alter its energy profile. In contrast, mutation within the rigid bound state has a greater potential to disrupt hydrogen bonds, van der Waals forces, etc.

We hypothesize that in the scenario of a rigid bound and flexible unbound state, prediction accuracy of indirect effects on binding affinity can be improved through a simple approximation. Figure II-1 summarizes the effects of mutations on binding free energy in two scenarios: The top row represents the scenario wherein the unbound state exists as one stable low energy conformation. The bottom row represents the rugged energy landscape (jagged red line) of a flexible unbound state with multiple energetic minima. In a thought experiment we compare a binding site mutation that is assumed to interfere only with direct interactions between ligand

and protein with a non-binding site mutation that is assumed to only affect stability of the protein, but does not change the protein-ligand interaction. In reality combinations of these two scenarios exist.

In the first scenario – a rigid unbound state engages the ligand and remains rigid, a mutation within the binding site that disrupts protein-small molecule interactions will lower the binding affinity (Figure II-1B). A mutation outside the binding pocket would have an equal effect on the free energy of bound and unbound conformation as they are identical. As a result the ligand affinity is unaltered (Figure II-1C). In the case of a flexible unbound state, mutations inside the binding pocket that interrupt protein-ligand interactions would again be expected to lower binding affinity (Figure II-1E). However, mutations outside the binding pocket are expected to have a greater destabilizing effect on the single rigid bound conformation than on the unbound state which consists of an ensemble of structures. On the other hand, mutations which affect low-energy structures that contribute to the unbound state will certainly affect the overall free energy of the unbound state. However, we hypothesize that this effect is small as mutations will affect only a fraction of the low-energy conformations the unbound state can assume. If the ensemble is large enough, influence on free energy will be small. This hypothesis suggests that the free energy of the unbound state can be approximated with a constant in this scenario. The result of this difference is a net change in binding energy due to mutation outside the binding pocket (Figure II-1F). It is obvious that this approximation is only valid for proteins that are very flexible in the unbound state and convert to a rigid bound conformation. HIV-1 PR is an example.



HIV-1 PR is a homodimer with a flexible binding site (Figure II-2). Over 200 high resolution crystal structures of HIV-1 PR mutants in complex with HIV-1 PR inhibitors (PIs) are deposited in the protein databank (PDB, resolution better than 2.0 Å) (Berman, Henrick et al. 2003). These mutants exhibit limited structural diversity verifying the well-defined rigid bound conformation of the protein (Louis, Ishima et al. 2007). However, the two flap regions exhibit up to 7Å of movement in the unbound state (Figure II-2) (Miller, Schneider et al. 1989; Galiano, Bonora et al. 2007). The unbound state is therefore best described as a large ensemble of structures (Ding, Layten et al. 2008). We hypothesize that it is for this reason that PR/PI docking studies have had difficulty predicting binding free energy ( $\Delta\Delta G_s$ ). The free energy of the unbound state ( $\Delta G_u$ ) is not accurately reflected by a single structure or a tight ensemble.

Cheng et al. assessed 16 scoring functions utilized in protein/ligand docking (Cheng, Li et al. 2009) for prediction of PR/PI  $\Delta\Delta G_s$ . Correlation coefficients ranged from R=0.17 to

R=0.34. RosettaLigand predicted  $\Delta\Delta G$ s with a correlation of R=0.41 (Meiler and Baker 2006). AutoDock predictions correlated with R=0.38 on a set of 25 HIV-1 PR/PI structures from the PDB, with binding data available (Jenwitheesuk and Samudrala 2003).

At the same time HIV PI therapies are greatly hampered by drug resistance mutations. Only recently, conformational ensembles were used to assist in designing PIs with broad enough specificity to avoid escape mutations (Sherman and Tidor 2008). The authors of this study evaluated chemical modifications to known PIs using electrostatic charge optimization. They chose not to include induced-fit effects or ligand flexibility.

In this study we use RosettaLigand to predict the effect of PR mutations inside and outside the binding pocket. Predicted  $\Delta\Delta G$ s are compared with experimentally determined  $\Delta\Delta G$ s. These include 34 HIV-1 PR mutants and eleven PIs. We demonstrate that by assuming the unbound state constant with respect to mutation we can achieve a correlation coefficient of R=0.71 over a wide array of PR/PI  $\Delta\Delta G$  data. Improved prediction of PR/PI binding affinity may help clinicians select the optimal PI for treatment and help design PIs with broad specificity that avoid resistance mutations.

## Materials and Methods

*176 experimental PR/PI binding energies have been collected.* PR/PI binding energies ( $\Delta\Delta G$ s) were obtained from the Binding Database ([www.bindingdb.org](http://www.bindingdb.org)) (Chen, Liu et al. 2001). These 176 binding energies include experimental conditions and HIV-1 PR mutant sequence information, but lack structural information. They include a total of eleven distinct PIs and 34 distinct PR sequences. 106 of these datapoints resulted from isothermal titration calorimetry (ITC) measurements. The remaining 70 datapoints are enzyme inhibition constants (K<sub>i</sub>s).

These  $K_i$ s were converted to binding energies using the equation  $\Delta G = RT \ln K_i$ , where  $R$  is the gas constant,  $8.314 \text{ J K}^{-1}\text{mol}^{-1}$ , and  $T$  is temperature in Kelvin.  $K_i$  values before and after conversion are summarized in Appendix A. Since temperatures were rarely reported, we assumed  $25^\circ\text{C}$  ( $298\text{K}$ ) for the conversion.

***171 high resolution template PR structures have been collected.*** 171 crystal structures of HIV-1 PR bound to various ligands were obtained from the PDB. These structures each have resolution better than  $2.0 \text{ \AA}$ . PDB codes, resolution, bound ligands, and citations for all 171 of these structures are listed in Appendix B. A multiple sequence alignment of these 171 structures is given as Appendix C.

***Threading of sequence onto structure for comparative modeling.*** 34 distinct sequences were associated with the 176 experimental PR/PI binding energy data points. The 3-letter residue codes found in each of the 171 backbones were replaced with 3-letter residue codes for each of the 34 sequences, thus generating 5,814 models. Missing side-chain coordinates were constructed using Rosetta.

***High resolution refinement of comparative models.*** Rosetta's high-resolution refinement protocol searches for low-energy structures in the conformational vicinity of the starting model (Bradley, Misura et al. 2005; Misura and Baker 2005). Backbone torsion angles are perturbed. Next side-chain rotamers are optimized (Dunbrack and Cohen 1997). Finally backbone and side-chain torsion angles are adjusted using a gradient-based energy minimization. This process is repeated multiple times, using a Monte Carlo accept/reject criterion (Li and Scheraga 1987).

***Low resolution initial placement of ligand.*** After a structural alignment was used to superimpose all comparative models, ligands were placed in the binding pockets of these models according to their positions in homologous crystal structures. Next ligands were randomly

translated up to 50 times or until the ligand centroid landed found non-clashing position. Then the ligand was rotated up to 1,000 times to find an orientation that has attractive and repulsive scores below a threshold. Finally a slide-together step moves the ligand toward the protein until the two are appropriately positioned for docking (Meiler and Baker 2006).

***Docking of PIs into comparative models.*** Six cycles of side-chain rotamer sampling were coupled with small (0.1 Å, 0.05 radians) ligand movements. Each cycle included minimization of ligand torsion angles with harmonic constraints (where 0.05 radians of movement is equal to one standard deviation). Each ligand torsion angle has a constraint score which is calculated as:  $f(x) = \left(\frac{x-x_0}{std\ dev}\right)^2$ . During cycles of docking, ligand flexibility is modeled only through small adjustments (minimization) of torsion angles rather than sampling of large-scale conformational changes. This is a limitation we allow for computational efficiency.

Amino acid side chains were repacked using a backbone-dependent rotamer library (Dunbrack and Karplus 1993). The structure resulting from each was accepted or rejected using a Monte Carlo approach. A soft repulsive scoring function was used during these 6 cycles so that small clashes would not lead to pose rejection. During the final minimization step Rosetta is able to resolve these small clashes.

The RosettaLigand scoring function with hard repulsive forces was used during the final minimization step. Hard repulsive forces produce very large scores for models with clashes thus allowing easy identification of infeasible poses. Final minimization includes with backbone flexibility, which is modeled through minimization of backbone torsion angles. Backbone torsion angles were minimized with harmonic constraints on the C $\alpha$  atom positions (0.2 Å standard deviation). Each C-alpha atom has a constraint score which is calculated as:  $f(x) = \left(\frac{x-x_0}{std\ dev}\right)^2$ .



The RosettaLigand score function includes the 6-12 Lennard-Jones potential (Lennard-Jones 1924), the Lazaridis-Karplus solvation model (Lazaridis and Karplus 1999), a side-chain rotamer score, based on the Dunbrack rotamer set (Dunbrack and Karplus 1993), a pair potential based on the probability of seeing two amino acids close together in space (Simons, Ruczinski et al. 1999), and an explicit orientation hydrogen bonding model (Kortemme, Morozov et al. 2003).

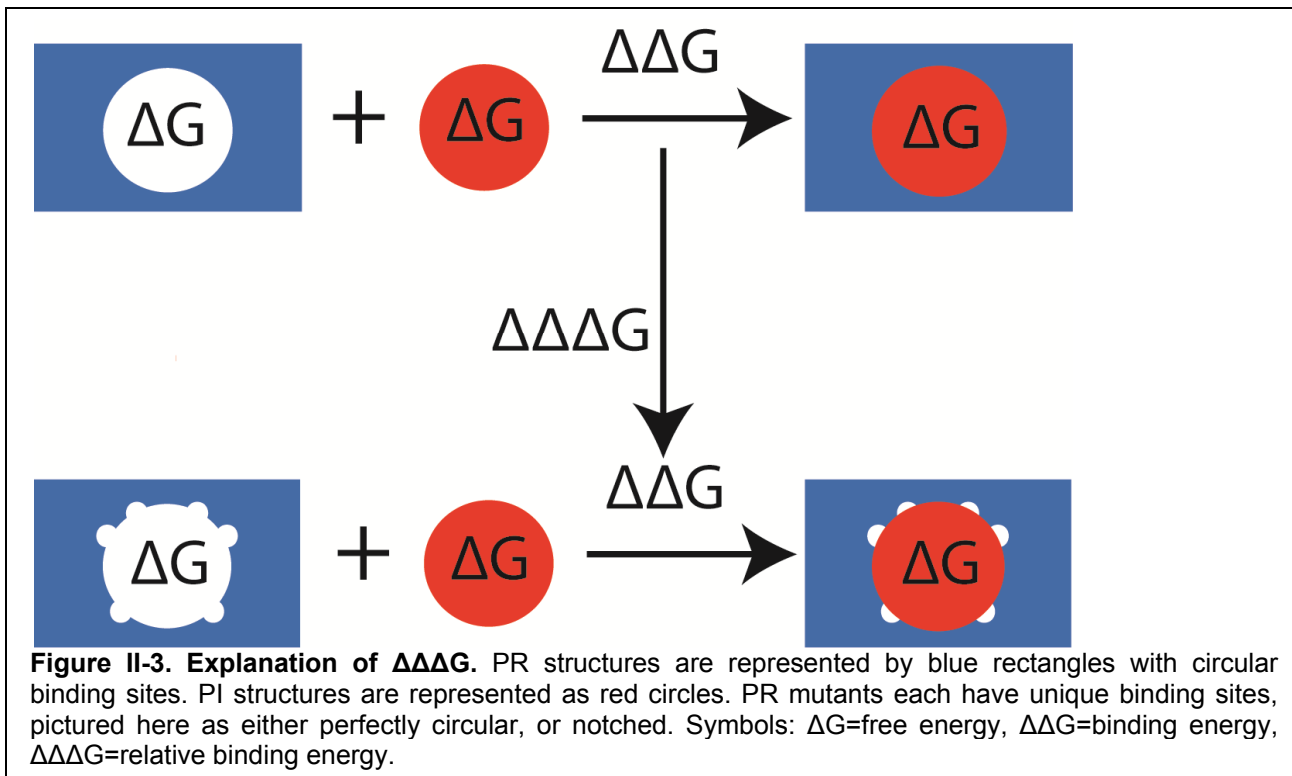
All computation was performed on the Vanderbilt University ACCRE cluster ([www.accre.vanderbilt.edu](http://www.accre.vanderbilt.edu)). Rosetta revision 32372 was used for all calculations. Command line arguments and input options are given in the Supporting Information.

***Predicting  $\Delta\Delta G$ s using the standard approach.*** The standard approach calculates  $\Delta\Delta G$ s as the difference between the free energy of a docked model ( $\Delta G_b$ ) and the free energy of the unbound model with equivalent sequence ( $\Delta G_u$ ) after energy minimization. This setup corresponds to Figure II-1A-C wherein the unbound state and bound state free energies are equally susceptible to disruption by mutation (Eq. I). For each of the 34 mutant PR sequences the lowest energy unbound comparative model was chosen to represent  $\Delta G_u$ . The lowest energy docked model for a given PR/PI pairing was chosen to represent  $\Delta G_b$ . The difference between these values was taken as a prediction of  $\Delta\Delta G$ .

***Predicting  $\Delta\Delta G$ s using the constant-unbound approach.*** The constant-unbound approach corresponds to Figure 1D-F and calculates  $\Delta\Delta G$  by assuming  $\Delta G_u$  to be unknown but invariant with mutation (Eq. II). The lowest energy docked model for a given PR/PI pairing was chosen to represent  $\Delta G_b$ .

$$\Delta\Delta G = \Delta G_b - \Delta G_u \text{ [I]}$$

$$\approx \Delta G_b - \text{const [II]}$$



Predicting  $\Delta\Delta\Delta G$  focuses on the influence of mutation on binding affinity. To determine how well RosettaLigand can predict changes in binding free energy ( $\Delta\Delta\Delta G$ , see Figure II-3) upon protein mutation  $i \rightarrow j$ , pairs of predicted or experimental  $\Delta\Delta G$ s sharing the same PI but different PR sequence were subtracted to obtain  $\Delta\Delta\Delta G$ s (Eqs. III, IV).  $\Delta\Delta\Delta G$ s predicted by Rosetta were compared with experimental  $\Delta\Delta\Delta G$ s to obtain  $\Delta\Delta\Delta G$  correlation. This strategy removes influences from the changes of the ligand thereby focusing on predicting the influence of mutations.

$$\begin{aligned} \Delta\Delta\Delta G &= \Delta\Delta G_i - \Delta\Delta G_j \\ &= (\Delta G_{i,b} - \Delta G_{i,u}) - (\Delta G_{j,b} - \Delta G_{j,u}) \quad \text{[III]} \\ &\approx \Delta G_{i,b} - \Delta G_{j,b} \quad \text{[IV]} \end{aligned}$$

***Optimization of RosettaLigand score term weights.*** The docking calculations performed so far were based on the original RosettaLigand scoring function (Meiler and Baker 2006) where the scoring term weights had been optimized across a set of diverse protein/ligand complexes. In the past it has been demonstrated that optimized scoring functions are needed to accurately predict free energies with Rosetta (Kortemme and Baker 2002). Therefore an optimized weight set for PR/PI complexes was developed. Score term weights were optimized separately for standard binding affinity predictions and constant-unbound predictions. Score term weights were also optimized separately for  $\Delta\Delta G$  predictions and  $\Delta\Delta\Delta G$  predictions. Hence, a total of four optimized weight sets were produced (Table II-1).

First, docking results were filtered by taking the top 5% of models by total energy and the top model by interface energy. A leave-one-out cross-validation analysis was used to determine the weights that produce the strongest correlation with experimental data. A multiple linear regression was used to determine weights that optimize the correlation between experimental and predicted binding affinity. The weight set was then applied to predict binding affinity of the omitted data-point. In a round robin scheme, each data point was left out. The correlation coefficients and standard deviations relate to the predictions made for these independent data points. The final optimal weight sets reported are average and standard over all cross-validation experiments (Table II-1). Bias is a simple constant that does not affect predicted/experimental correlation but is added to predicted values as an offset.

***Partitioning data by location of PR mutations.*** We partitioned the 34 sequences shown in Figure 4 into four distinct groups, based on the presence and location of “exceptional” mutations. Exceptional mutations are defined as amino acids that are uncommon or rare in a multiple sequence alignment – i.e. if 17 out of 34 sequences have an A in a position and the other

17 have a V, neither is an exceptional mutation. A sequence that has an S in the same position would be counted as an exceptional mutation A/V→S.

Exceptional mutations were selected using ClustalW alignment software (gray boxed residues in Figure II-4). The first group includes sequences with no exceptional mutations (sequences 4, 5, 22, and 26). The second group has only exceptional mutations within or near the binding site (red residues in Figure II-2) and includes sequences 1, 8, 16, 19, 21, 24, 29, 30, and 33. The third group has only exceptional mutations outside the binding pocket and includes sequences 2, 3, 9, 11, 12, 23, 27, and 28. The fourth includes sequences that have exceptional mutations within and outside the binding site (sequences 6, 7, 10, 13, 14, 15, 17, 18, 20, 25, 31, 32, and 34).

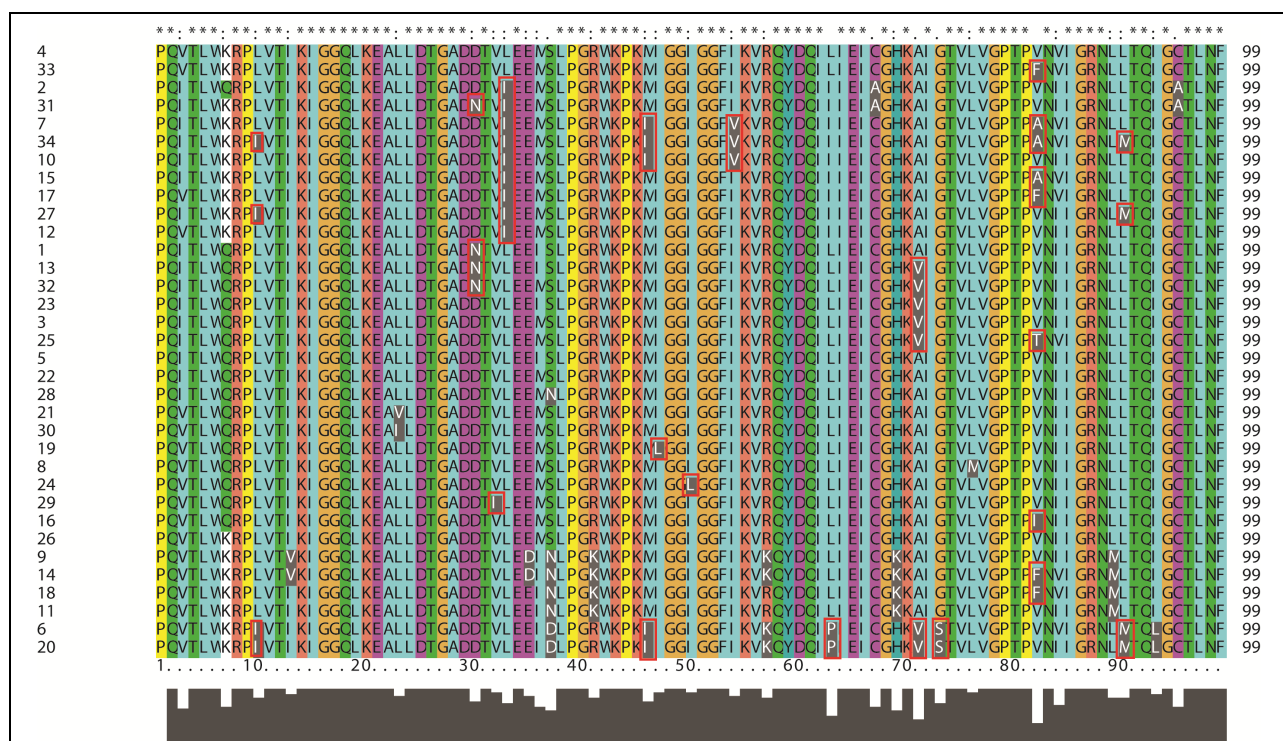
We also partitioned sequences based on whether exceptional mutations fell within or outside of the flexible flap region. We define this region as comprising residues 37-61 (Torbeev, Raghuraman et al. 2011). By this definition, 24% of PR lies in the flap region. Sequences with only exceptional mutation in the flap region include sequences 19 and 24. Sequences with only exceptional non-flap mutations include 1-3, 8, 9, 11-18, 20, 21, 23, 25, 27-33. Sequences with exceptional mutations in and out of the flap region include 6, 7, 10, 20 and 34.

## **Results/Discussion**

*Assessment of uncertainty in experimental binding affinity data.* As seen in Appendix A, for a few PR/PI pairs binding affinities have been determined multiple times. In these cases we use average values which reduces the total number of experimental ITC values from 106 to 99 while the total number of Ki datapoints is reduced from 70 to 62. We further use replicate data to estimate the accuracy of experimental values. The standard error for ITC replicates is

4.69 kJ/mol. The standard error for converted  $K_i$  replicates is 7.21 kJ/mol. We will use these numbers as estimates for the experimental uncertainty. As noted in the previous section, we assume a temperature of 25°C in order to convert  $K_i$ s to  $\Delta\Delta G$ s. This assumption introduces additional uncertainty for  $\Delta\Delta G$ s calculated from  $K_i$ s. Nevertheless, the standard deviation between  $\Delta\Delta G$  values converted from  $K_i$  data and matching ITC values is 1.07 kJ/mol, confirming the validity of the conversion.

**Building of comparative models from HIV-1 templates.** The 34 distinct mutant sequences found in our experimental data contained between 3 and 14 mutations per monomer to match the wild-type HIV-1 PR sequence (Ratner, Haseltine et al. 1985). These 34 mutant



**Figure II-4. HIV-1 PR Multiple sequence alignment using ClustalX 2.1.** 34 sequences were threaded onto each of 171 backbone templates. Aligned are the sequences from the 34 experimental binding energy datapoints. An asterisk ("\*") means that the residues or nucleotides in that column are identical in all sequences in the alignment. A colon (":") means that conserved substitutions have been observed. A period (".") means that semi-conserved substitutions are observed. Exceptional residues are colored gray. Positions enclosed in red boxes indicate residue positions with the potential to confer drug resistance (as suggested by Rhee et al. 2005) (Rhee, Fessel et al. 2005).

sequences were aligned and mutations at residues known to confer drug resistance are highlighted in red boxes (Figure II-4). Each of the 34 sequences was threaded onto the backbones of 171 template structures yielding 5,814 comparative models. These 171 HIV-1 PR/PI structures comprise PDB selections with  $<2.0$  Å resolution. These 5,814 ligand free structures were relaxed 10 times each using the Rosetta energy function (see methods). These 58,140 relaxed structures served as starting structures for RosettaLigand docking simulations.

***RosettaLigand docking protocol allows local flexibility.*** For each of the 176 experimentally determined PR/PI binding affinities, the 171 times 10 comparative models with matching sequence were docked with the respective ligand. A total of 300,960 unique input structures were used for ligand docking. Local induced-fit effects were considered through full PR and PI flexibility in the binding site: The RosettaLigand docking predictions allow ligand flexibility by minimizing ligand torsion angles. Backbone torsion angles near the PR/PI interface were also minimized. See Appendix G for specifics on Rosetta usage.

For each input, the docking protocol was repeated 20 times. For each set of predictions for a given PR/PI datapoint, docking results were filtered by taking the top 5% of models by total energy and the top model by interface energy (see Appendix E). Appendix F compares top scoring Rosetta models with experimental PR/PI complex structures from the PDB that share the same PI to confirm accuracy of the modeling procedure.

***Usage of experimental data for weight optimization.*** RosettaLigand uses a scoring function that has been optimized to give optimal docking results for a wide variety of ligands (Meiler and Baker 2006). For accurate prediction of free energies the weights of the scoring function need to be adjusted (Kortemme and Baker 2002). For the purposes of optimizing the RosettaLigand scoring function weights and then testing the predictive power, we split our

experimental datapoints into two groups. The 99 datapoints acquired by ITC were used to optimize weights because of their higher accuracy. Score term weights were optimized using leave-one-out cross-validation using 98 datapoints to fit the weights and predicting the 99th (see Table II-1). The 62  $K_i$  values converted to  $\Delta\Delta G$ s were used as a second independent test of the scoring function.

**Table 0-1. Score term weights optimized for HIV-1 PR binding affinity prediction.** A leave-one-out analysis was used to find score-term weights that optimize correlation between Rosetta predictions of  $\Delta\Delta G$  and 106 values determined using ITC. Standard deviations are shown.

Score Term	Rosetta Default Weights	$\Delta\Delta G$		$\Delta\Delta\Delta G$	
		Standard approach	Constant Unbound	Standard approach	Constant Unbound
Bias	N/A	-36.00±0.38	-1.19±12.9	-3.67±0.01	-0.26±0.01
attractive	0.8	0.82±0.02	0.76±0.01	0.20±0.00	0.72±0.00
repulsive	0.4	-0.01±0.02	0.08±0.01	0.11±0.00	0.003±0.00
solvation	0.6	0.78±0.03	1.39±0.03	0.10±0.00	1.32±0.00
dunbrack	0.4	0.33±0.01	-0.25±0.01	0.28±0.00	-0.24±0.00
pair	0.8	0.92±0.06	-2.76±0.06	0.52±0.01	-2.47±0.01
hbond_lr_bb	2.0	0.98±0.04	-0.28±0.05	0.07±0.00	0.18±0.01
hbond_bb_sc	2.0	0.10±0.03	0.32±0.03	-0.13±0.00	0.36±0.00
hbond_sc	2.0	-0.40±0.04	0.19±0.04	1.11±0.01	0.27±0.00

“Attractive” and “repulsive” are derived from the Lennard-Jones potential(Lennard-Jones 1924), “solvation” comes from a Lazaridis-Karplus model(Lazaridis and Karplus 1999), “dunbrack” is a side-chain rotamer score based on the Dunbrack rotamer set(Dunbrack and Karplus 1993), “pair” is a potential based on the probability of seeing two amino acids close together in space(Simons, Ruczinski et al. 1999), and “hbond” terms are based on an explicit orientation hydrogen bonding model(Kortemme, Morozov et al. 2003). sc: side-chain, bb: backbone, lr: long-range. Weight optimization was implemented in Mathematica (Wolfram Research 2010).

*Analysis of optimized score term weights.* Optimized score term weights are shown in Table II-1. The van der Waals attractive and solvation energies are given high weights across all optimized weight sets. Van der Waals attractive scores assess the shape complementarity of ligand and protein. The solvation score penalizes the burial of polar atoms not engaged in hydrogen bonds, thus favoring ligand poses that either expose their polar atoms or form hydrogen bonds with the protein. The repulsive score is down-weighted across all optimized

weight sets. This suggests that nature may be able to resolve certain steric clashes that RosettaLigand failed to resolve during the final gradient based minimization step.

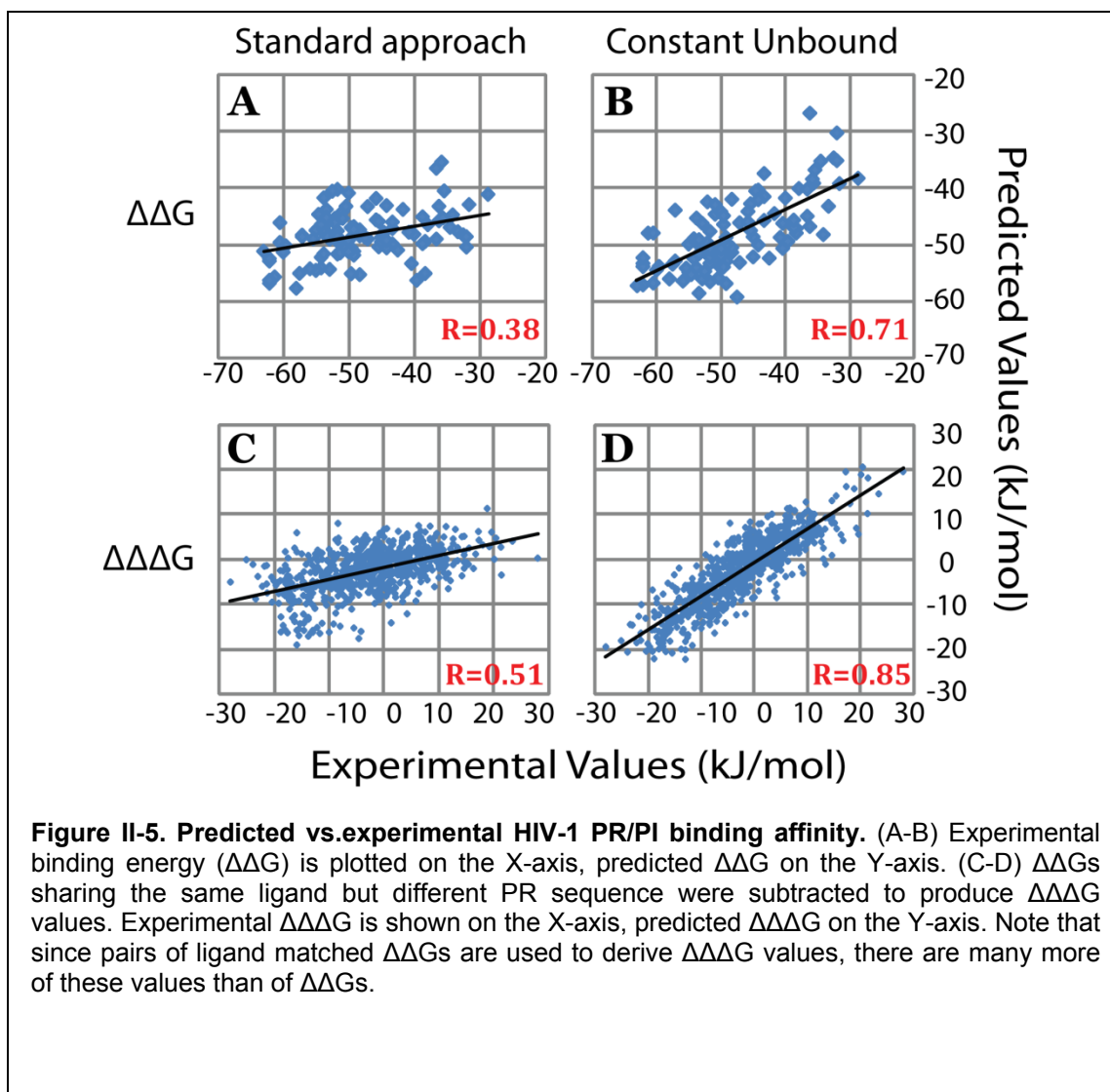
Score terms that capture protein/ligand hydrogen bonding effects were generally down-weighted from the default RosettaLigand weights. Because these terms include both protein/protein hydrogen bonding and protein/ligand hydrogen bonding it is difficult to grade RosettaLigand's ability to form native-like ligand hydrogen bonds. Creating a hydrogen bonding potential specific for protein/ligand and ligand/ligand interactions could potentially add more discriminatory power to the Rosetta score function.

Interestingly we find a significant negative constant-unbound weight for the amino acid pair potential. We attribute this negative weight to the fact that amino acid electrostatic interactions are disrupted in the PR binding site upon PI binding. The constant unbound approach however only involves the bound protease, so these disruptions are not accounted for. We find that removal of the amino acid pair potential from the scoring function does not result in significantly reduced prediction accuracy (data not shown).

Similarly, the Dunbrack score term has a negative weight within the constant unbound weight set. The Dunbrack score is based on the probability of seeing a given rotamer within the context of a residue's backbone environment (Dunbrack and Karplus 1993). This score term informs the standard approach since rotamers are likely to change upon ligand binding. However the term is less likely to inform ligand binding energy predictions using the constant-unbound approach.

***Predicting  $\Delta\Delta G$ s using the standard approach.*** The standard approach calculates  $\Delta\Delta G$ s as the difference between the free energy of a docked model ( $\Delta G_b$ ) and the free energy of the unbound model with equivalent sequence ( $\Delta G_u$ ) (see methods). Score terms were reweighted to





optimize predicted  $\Delta\Delta G$  correlation with experimental data (weights are shown in Table II-1, columns labeled “Standard Approach”). After reweighting, the predicted and experimental  $\Delta\Delta G$ s correlate with  $R=0.40$  (Figure II-5A), while  $\Delta\Delta\Delta G$ s correlate with  $R=0.47$  (Figure II-5C).

**Predicting  $\Delta\Delta G$ s using the constant-unbound approach.** The constant-unbound approach predicts  $\Delta\Delta G$  as a function of  $\Delta G_b$  alone. Assuming constant free energy for unbound PR, the  $\Delta\Delta G$  and  $\Delta\Delta\Delta G$  correlations improve to  $R=0.71$  and  $R=0.85$  (Figure II-5B, D) after score term reweighting (Table II-1, columns labeled “Constant Unbound”). The standard error of

prediction is with 5.91 kJ/mol and 4.49 kJ/mol, respectively, in range of the experimental uncertainty (4.69 kJ/mol, Table II-2).  $\Delta\Delta G$  correlations reported above are calculated by subtracting  $\Delta\Delta G$ s sharing the same PI but different PR sequence.  $\Delta\Delta G$  correlations calculated by subtracting  $\Delta\Delta G$ s sharing the same PR sequence but different PIs yield a correlation of  $R=0.61\pm 0.04$  with a standard error of 7.28 kJ/mol.

***Optimized score term weights predict binding affinity in independent data set.***

Optimized weight sets shown in Table II-1 were generated from ITC data only. In order to show that high correlation statistics were not an artifact of leave-one-out weight optimization, optimized weights were applied to  $\Delta\Delta G$  predictions for experimental  $K_i$  data. RosettaLigand predictions correlate well with the 62  $\Delta\Delta G$ s in this independent dataset ( $R=0.70$ , see Table II-2). The standard error in our predictions is 7.22 kJ/mol, which correlates with the previously determined experimental uncertainty for this dataset (7.21 kJ/mol).

***Analysis of data partitioned by location of PR mutations.*** We partitioned the experimental data according to whether mutations were found in the binding site of HIV-1 PR or

<b>Table 0-2. Correlations between RosettaLigand predictions and experimental data.</b> Pearson's correlation (RP) Spearman's rank correlation (RS) and standard errors (kJ/mol, kcal/mol) are shown.									
		$\Delta\Delta G$				$\Delta\Delta\Delta G$			
		n	RP	RS	kJ/mol	n	RP	RS	kJ/mol
ITC data*	Standard approach	99	0.38±0.09	0.51±0.09	7.82	591	0.51±0.03	0.51±0.03	7.29
	Constant unbound	99	0.71±0.05	0.69±0.05	5.91	591	0.85±0.01	0.86±0.01	4.49
Ki data†	Default weights	62	0.66±0.07	0.49±0.10	n/a	327	0.61±0.04	0.47±0.04	n/a
	Optimized weights	62	0.70±0.07	0.40±0.11	7.22	327	0.70±0.03	0.57±0.04	7.28

\*Correlation with ITC measurements after score term weight optimization (see table 1). †Correlation with  $\Delta\Delta G$ s converted from  $K_i$  data. The constant-unbound approach was used. Default weights are the RosettaLigand weights before optimization. Note that standard error is not relevant for default weights because the Rosetta score function is unit-less.

elsewhere. Averaging replicates reduces the total number of experimental  $\Delta\Delta G$  values from 176 to 149. These data points were assigned to one of the four groups. Group one contained no exceptional mutations and included 15 datapoints. Group 2 included 17 datapoints with only mutations in the binding site. Group 3 includes 44 datapoints with only mutations outside the binding site. Group 4 includes 73 datapoints with mutations inside and outside the binding site. Corresponding Rosetta predictions were reweighted using the previously optimized weights (weights from Table II-1, “constant-unbound”) and predicted  $\Delta\Delta G$  within each group were compared with experimental values.

Standard errors between Rosetta predicted  $\Delta\Delta G$  and experimental data are shown in Table II-2. Note that the small and variable sample size makes correlation coefficients unsuitable for comparison. Generally,  $\Delta\Delta\Delta G$  predictions outperform  $\Delta\Delta G$  predictions. Further, predictions are most accurate for sequences with no mutations or only non-binding site mutations. Accuracy decreases as binding site mutations occur. While the latter effect exemplifies the larger influence of binding site mutations for affinity, the former data point confirms our hypothesis that assuming PR  $\Delta G_u$  to be invariant with respect to mutation allows for accurate prediction of effects of non-binding site mutations on PR/PI affinity.

We also partitioned data based on whether mutations were found in the flexible flap region (residues 37-61) (Hornak, Okur et al. 2006). While our flap region definition comprised 24% of the protein, only 2 of the experimental data points contained only flap region mutations, 35 data points had mutations in flap and non-flap regions, and 97 data points contained only non-flap region mutations. It appears that predictions are more accurate for mutants that contain both, flap and non-flap mutations (Appendix D). This finding supports our hypothesis that assuming PR  $\Delta G_u$  to be invariant with respect to mutation allows for accurate prediction of effects of non-

binding site mutations on PR/PI affinity. The lack of only-flap region mutants complicates interpretation of this analysis.

## **Conclusion**

Both,  $\Delta\Delta G$  and  $\Delta\Delta\Delta G$  predictions improve for PR/PI complexes using the constant-unbound approach (to  $R=0.71$  and  $R=0.85$  respectively, after score term reweighting). This is expected since unbound HIV-1 PR exhibits a high degree of flexibility (Ding, Layten et al. 2008) and stabilizes upon ligand binding. Therefore the free energy of the unbound state is less sensitive to individual mutations. This result is significant because it demonstrates a simple way to improve binding free energy predictions for proteins with a flexible unbound state. By assuming differences in the unbound state of closely related structures are negligible, binding free energy prediction is possible considering the bound state of the protein only. This finding becomes even more important if one considers that a crystal structure of the unbound protein is often not available in such a scenario.

Clearly if it was possible to accurately predict the free energy of the unbound state, one could further improve binding affinity predictions. However, currently limited structural information is available to describe the conformational ensemble that represents unbound state of PR mutants.

As expected  $\Delta\Delta\Delta G$  predictions outperform  $\Delta\Delta G$  predictions. These relative binding energies focus on effects of mutations on the same ligand thereby removing the need to accurately predict differences in  $\Delta\Delta G$  among PIs. Because Rosetta scoring terms have been parameterized for optimizing amino acid side chain placement, Rosetta excels at  $\Delta\Delta\Delta G$  predictions.

Note that the standard approach that uses a single bound and unbound state resembles closely a lock-and-key paradigm with local induced fit in the binding site. The constant unbound approach resembles a conformational selection paradigm coupled with local induced fit in the binding site.

## **Future Directions**

During docking we allowed backbone flexibility within the binding site. A future study may need to incorporate global backbone flexibility during docking, to allow mutations outside the binding site to effect the conformation of the binding site. The Rosetta database only includes de-protonated aspartic acid. In a study by Wittayanarakul et al. the protonation state of the catalytic aspartate residues at position 25 was important for more accurate binding free energy calculations (Wittayanarakul, Hannongbua et al. 2008). The addition of protonated aspartate to the Rosetta residue type library should have a similar effect on Rosetta HIV-1 PR/PI  $\Delta\Delta G$  predictions.

Further, for several PIs, a water molecule mediates interaction with flap residues Ile-50 and Ile-50', stabilizing PR in the closed conformation (Wlodawer and Erickson 1993; Wlodawer and Vondrasek 1998). This water molecule is not modeled in the present study. However, given that both interactions are present in all PR/PI complexes cancellation of errors allows an accurate prediction of PR/PI affinity already with the setup presented here. Simultaneously optimizing the positioning of the PI and the bridging water molecule should lead to further improvements in RosettaLigand predictions of HIV-1 PR/PI interaction.

The ability to predict HIV-1 PR/PI  $\Delta\Delta G$ s based on PR sequence has clinical implications. Currently HIV-1 genotype assays assist physicians in prescribing the most effective inhibitor

(Tural, Ruiz et al. 2002). These assays are based on observed resistance patterns. When new inhibitors are developed, or when novel PR mutations are observed, predictions of binding affinity could be used to make decisions about which PI to prescribe. Future work will focus on demonstrating that Rosetta binding energy predictions correlation to clinical outcomes such as viral load measurements. Collecting quality data has made preliminary work in this area difficult (see Appendix H).

## Acknowledgements

Funding was provided through the Molecular Biophysics Training Grant at Vanderbilt University to G.L. and K.W.K. (NIH GM 08320) and an R01 grant to J.M. (NIH MH 090192).

## Supporting Information

A supplemental document is available online. It contains experimental  $\Delta\Delta G$  and  $K_i$  values used in this study; a description of each of the 171 template structures used for comparative modeling in this study correlations for partitioned by presence and location of exceptional mutations; a ClustalW multiple sequence alignment for the 171 template structures used in this study; images of Rosetta predictions superimposed on PDB structures; and a description of the options we used with Rosetta software.

## References

- Berman, H., K. Henrick, et al. (2003). "Announcing the worldwide Protein Data Bank." Nature Structural Biology **10**(12): 980.
- Bradley, P., K. M. Misura, et al. (2005). "Toward high-resolution de novo structure prediction for small proteins." Science **309**(5742): 1868-1871.
- Chen, X., M. Liu, et al. (2001). "BindingDB: a web-accessible molecular recognition database." Combinatorial Chemistry & High Throughput Screening **4**(8): 719-725.

- Cheng, T., X. Li, et al. (2009). "Comparative assessment of scoring functions on a diverse test set." Journal of Chemical Information and Modeling **49**(4): 1079-1093.
- Ding, F., M. Layten, et al. (2008). "Solution structure of HIV-1 protease flaps probed by comparison of molecular dynamics simulation ensembles and EPR experiments." Journal of the American Chemical Society **130**(23): 7184-7185.
- Dunbrack, R. L., Jr. and F. E. Cohen (1997). "Bayesian statistical analysis of protein side-chain rotamer preferences." Protein Science **6**(8): 1661-1681.
- Dunbrack, R. L., Jr. and M. Karplus (1993). "Backbone-dependent rotamer library for proteins. Application to side-chain prediction." Journal of Molecular Biology **230**(2): 543-574.
- Galiano, L., M. Bonora, et al. (2007). "Interflap distances in HIV-1 protease determined by pulsed EPR measurements." Journal of the American Chemical Society **129**(36): 11004-11005.
- Gohlke, H., L. A. Kuhn, et al. (2004). "Change in protein flexibility upon complex formation: analysis of Ras-Raf using molecular dynamics and a molecular framework approach." Proteins **56**(2): 322-337.
- Henzler-Wildman, K. and D. Kern (2007). "Dynamic personalities of proteins." Nature **450**(7172): 964-972.
- Hornak, V., A. Okur, et al. (2006). "HIV-1 protease flaps spontaneously open and reclose in molecular dynamics simulations." Proceedings of the National Academy of Sciences of the United States of America **103**(4): 915-920.
- Jenwitheesuk, E. and R. Samudrala (2003). "Improved prediction of HIV-1 protease-inhibitor binding energies by molecular dynamics simulations." BMC Structural Biology **3**: 2.
- Kortemme, T. and D. Baker (2002). "A simple physical model for binding energy hot spots in protein-protein complexes." Proceedings of the National Academy of Sciences of the United States of America **99**(22): 14116-14121.
- Kortemme, T., A. V. Morozov, et al. (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.
- Lazaridis, T. and M. Karplus (1999). "Effective energy function for proteins in solution." Proteins **35**(2): 133-152.
- Lemmon, G., K. Kaufmann, et al. (2012). "Prediction of HIV-1 Protease/Inhibitor Affinity using RosettaLigand." Chemical Biology and Drug Design **79**(6): 888-896.

- Lennard-Jones, J. E. (1924). "On the Determination of Molecular Fields. II. From the Equation of State of a Gas." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **106**: 463-477.
- Li, Z. and H. A. Scheraga (1987). "Monte Carlo-minimization approach to the multiple-minima problem in protein folding." Proceedings of the National Academy of Science, USA **84**(19): 6611-6615.
- Louis, J. M., R. Ishima, et al. (2007). "HIV-1 protease: structure, dynamics, and inhibition." Advances in Pharmacology **55**: 261-298.
- Martin, S. F. (2007). "Preorganization in biological systems: Are conformational constraints worth the energy?" Pure Applied Chemistry **79**(2): 193-200.
- Meiler, J. and D. Baker (2006). "ROSETTALIGAND: protein-small molecule docking with full side-chain flexibility." Proteins **65**(3): 538-548.
- Miller, M., J. Schneider, et al. (1989). "Structure of complex of synthetic HIV-1 protease with a substrate-based inhibitor at 2.3 Å resolution." Science **246**(4934): 1149-1152.
- Misura, K. M. and D. Baker (2005). "Progress and challenges in high-resolution refinement of protein structure models." Proteins **59**(1): 15-29.
- Ratner, L., W. Haseltine, et al. (1985). "Complete nucleotide sequence of the AIDS virus, HTLV-III." Nature **313**(6000): 277-284.
- Rhee, S. Y., W. J. Fessel, et al. (2005). "HIV-1 protease and reverse-transcriptase mutations: Correlations with antiretroviral therapy in subtype B isolates and implications for drug-resistance surveillance." Journal of Infectious Diseases **192**(3): 456-465.
- Sherman, W. and B. Tidor (2008). "Novel method for probing the specificity binding profile of ligands: applications to HIV protease." Chemical Biology and Drug Design **71**(5): 387-407.
- Shimotohno, A., S. Oue, et al. (2001). "Demonstration of the importance and usefulness of manipulating non-active-site residues in protein design." Journal of Biochemistry **129**(6): 943-948.
- Simons, K. T., I. Ruczinski, et al. (1999). "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins." Proteins **34**(1): 82-95.
- Sousa, S. F., P. A. Fernandes, et al. (2006). "Protein-ligand docking: current status and future challenges." Proteins **65**(1): 15-26.



- Torbeev, V. Y., H. Raghuraman, et al. (2011). "Protein conformational dynamics in the mechanism of HIV-1 protease catalysis." Proceedings of the National Academy of Sciences of the United States of America **108**(52): 20982-20987.
- Tural, C., L. Ruiz, et al. (2002). "Clinical utility of HIV-1 genotyping and expert advice: the Havana trial." AIDS **16**(2): 209-218.
- Wittayanarakul, K., S. Hannongbua, et al. (2008). "Accurate prediction of protonation state as a prerequisite for reliable MM-PB(GB)SA binding free energy calculations of HIV-1 protease inhibitors." Journal of Computational Chemistry **29**(5): 673-685.
- Wlodawer, A. and J. W. Erickson (1993). "Structure-based inhibitors of HIV-1 protease." Annual Review of Biochemistry **62**: 543-585.
- Wlodawer, A. and J. Vondrasek (1998). "Inhibitors of HIV-1 protease: a major success of structure-assisted drug design." Annual Review of Biophysics & Biomolecular Structure **27**: 249-284.
- Wolfram Research, I. (2010). Mathematica Edition: Version 8.0. Champaign, Illinois, Wolfram Research, Inc.

## CHAPTER 3

### ROSETTA LIGAND DOCKING WITH FLEXIBLE XML PROTOCOLS

The content of this chapter is largely taken from a book chapter published within the *Methods in Molecular Biology* series (Lemmon and Meiler 2012).

#### **Summary**

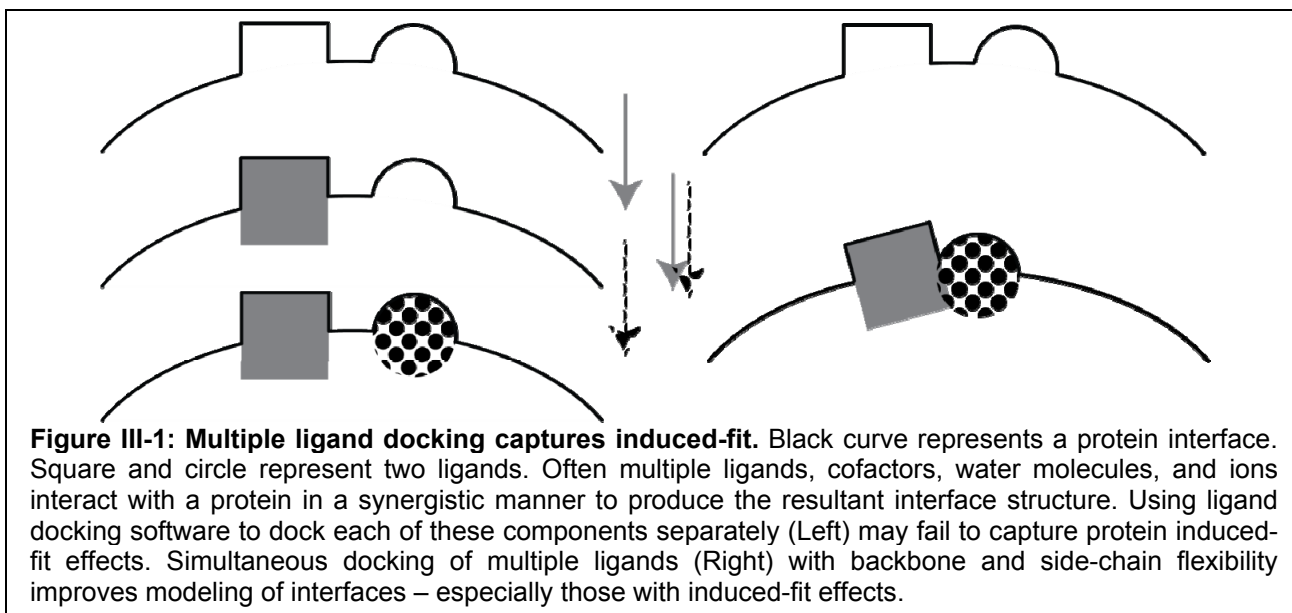
RosettaLigand is premier software for predicting how a protein and a small molecule interact. Benchmark studies demonstrate that 70% of the top scoring RosettaLigand predicted interfaces are within 2 Å RMSD from the crystal structure (Meiler and Baker 2006). The latest release of Rosetta ligand software includes many new features, such as (1) docking of multiple ligands simultaneously, (2) representing ligands as fragments for greater flexibility, (3) redesign of the interface during docking, and (4) an XML script based interface that gives the user full control of the ligand docking protocol.

#### **Introduction**

Rosetta is a suite of applications used in protein modeling (Kaufmann, Lemmon et al. 2010). These applications have proven themselves in the areas of protein structure prediction (Raman, Vernon et al. 2009), protein-protein docking (Chaudhury and Gray 2008), protein design (Jiang, Althoff et al. 2008), and protein-ligand docking (Meiler and Baker 2006). In 2006 RosettaLigand was introduced as software for modeling protein/small molecule interactions. RosettaLigand samples the rigid body position and orientation of the ligand as well as side-chain conformations using Monte Carlo minimization. Ensembles of ligand conformations and protein

backbones were used to sample conformational flexibility. The models produced by RosettaLigand conformational sampling are evaluated with a scoring function that includes an electrostatics model, an explicit orientation-dependent hydrogen bonding potential, an implicit solvation model, and van der Waals interactions (Meiler and Baker 2006). Default ligand-centric score term weights are provided through ‘ligand.wts’ and ‘ligand\_soft\_rep.wts’ (see the SCOREFXNS section of Figure III-2). However we have found that optimizing these score term weights for a particular class of protein/ligand complexes can greatly improve predictions (see Note 1).

RosettaLigand was later enhanced to allow receptor backbone flexibility as well as greater ligand flexibility (Lazaridis and Karplus 1999). Both ligand flexibility and backbone flexibility were shown to improve self-docking and cross-docking scores and lead to better performance than the open-source competitor AutoDock. Ligand flexibility was modeled by sampling ligand conformers and minimizing ligand torsion angles. Backbone flexibility included selecting stretches of residues near the ligand and sampling phi/psi angles for those residues, using a gradient based minimization (Davis and Baker 2009). Libraries of ligand conformers can be generated using methods presented by Kaufmann et al (Kaufmann, Glab et al. 2008). These features have enabled Rosetta to excel in predicting how pharmaceutically relevant compounds interact with their target (Davis, Raha et al. 2009).



In this chapter we present new features and enhancements to RosettaLigand. Multiple ligands, cofactors, ions, and key water molecules can now be docked simultaneously (Figure III-1). User provided ligand conformations are now sampled during docking, along with protein side-chain rotamer sampling. Interface residue identities can now be redesigned during docking. A new XML script format is used to describe the ligand docking protocol. This adds great flexibility for the user to customize their docking study.

## Materials

RosettaLigand is part of the Rosetta software suite for protein structure prediction. Visit <http://www.rosettacommons.org/> to obtain a license, download the latest release, and read the manual for help installing the software. The information in this tutorial applies to Rosetta version 3.2. Read the documentation about how to run Rosetta executables using command line or flag file options:

[http://www.rosettacommons.org/manuals/archive/rosetta3.2\\_user\\_guide/command\\_options.html](http://www.rosettacommons.org/manuals/archive/rosetta3.2_user_guide/command_options.html)

Read the tutorial entitled “Dock Design Parser Application”:

[http://www.rosettacommons.org/manuals/archive/rosetta3.2\\_user\\_guide/app\\_dock\\_design.html](http://www.rosettacommons.org/manuals/archive/rosetta3.2_user_guide/app_dock_design.html)

This guide describes an XML format that is now used for all aspects of ligand docking.

### ***Preparation of protein PDB input file***

Assure that the protein PDB has at least one backbone heavy atom present for each residue. Rosetta can add missing atoms to incomplete residues. If a residue is completely missing, use loop building to add its coordinates. Follow the loop building tutorial ([http://www.rosettacommons.org/manuals/archive/rosetta3.2\\_user\\_guide/app\\_loop.html](http://www.rosettacommons.org/manuals/archive/rosetta3.2_user_guide/app_loop.html)). Assure that residues are numbered in sequence. Rosetta will renumber residues if they are not. Assure that each ligand, cofactor, water molecule, or ion you wish to dock is assigned its own chain ID.

RosettaLigand has been successful in comparative modeling (Kaufmann, Dawson et al. 2009), where an experimental structure of the protein of interest is not available. In this case, a sequence alignment is made between the protein of interest and a homologous protein with similar sequence. The 3-letter codes in the PDB file of the homologous protein are replaced with the 3-letter codes of the protein of interest, according to the sequence alignment and side chain conformations are reconstructed using a rotamer library. If the protein of interest has insertions, loop modeling is used to fill in missing density.

Since ligand docking only repacks side-chain residues within the interface, we first repack all side-chain residues in the protein using the same score function that will be used in ligand docking. By optimizing unbound and bound protein structures using the same scoring function, we ensure that predicted binding affinity is based strictly on changes related to ligand

docking. The following XML code can be used for repacking the unbound structure within RosettaScripts.

```
<SCOREFUNCTION>
  <hard_rep weights=ligand>
</SCOREFUNCTION>
<MOVERS>
  <Repack name=repack score_function= hard_rep>
</MOVERS>
```

### ***Preparation of ligand PDB and “params” input files***

If you are starting with a ligand in PDB format, first convert it to .mol or .mol2 format. Use `<RosettaSource>/src/python/apps/mol_to_params.py` to generate a ligand params file and a ligand PDB file with Rosetta atom types. The .params file describes partial charges, atom types, bond lengths, bond angles, torsion angles, and atom types for each residue. Append the atoms in the generated ligand pdb file onto the end of the prepared protein PDB file.

If you are interested in large-scale ligand flexibility, generate conformations for your ligand using OpenEye’s Omega (<http://www.eyesopen.com/omega>) or MOE (<http://www.chemcomp.com>). These conformations should be in one PDB format separated by TER statements. Add the line “PDB\_ROTAMERS <location of PDB file with ligand conformations>” to the end of your .params file.

If your ligand has more than 7 rotatable bonds or if over 100 conformations are required to fully cover the conformational space of your ligand, split it into several smaller fragments. Specify split points at the bottom of your .mol or .mol2 file before running `molfile_to_params.py` in this fashion: “M SPLT <index 1> <index 2>” where indices 1 and 2 correspond to the atom number in the .mol or .mol2 file (the ATOM block line number). `molfile_to_params.py` will generate a .params file for each fragment.

### ***Relevant command line or flags file options***

Rosetta applications use a common set of options that can be specified either at the command line or in a file. Not all Rosetta options are relevant or accessed by each Rosetta application. The options below are most commonly used with ligand docking. An asterisk signifies a required option.

1. `-in:path:database` <path to Rosetta database>. The Rosetta database directory is downloaded from [www.rosettacommons.org](http://www.rosettacommons.org) and contains chemical descriptions of each amino acid as well default score term weights.\*
2. `-in:file:s` <space delimited list of PDB files containing protein and ligand(s)>. Alternatively use `-in:file:list`.\*
3. `-in:file:list` <text file with 2 or more PDB files listed on each line>. This option is especially useful for processing batches of proteins and ligands. PDBs on the same line are concatenated for docking.\*
4. `-in:file:extra_res_fa` <space delimited list of .params files for each ligand>. See section 2.2 for preparation of these .params files. Alternatively use `-in:file:extra_res_path`.\*
5. `-in:file:extra_res_path` <path to find .params files>. All files in this directory that end with '.param' or '.params' will be included in docking.\*
6. `-out:nstruct` <number of models to produce per input PDB>. Defaults to 1. See Note 2 on determining how many models to produce.
7. `-out:file:atom_tree_diff` <name of output file>. In `atom_tree_output` files only differences from a reference structure are recorded. Since output models usually only differ within the interface region, much less disk space is used by only recording differences.

8. `-parser:protocol <name of RosettaScripts XML file>`. This file allows the user to customize each step of ligand docking.\*
9. `-packing:ex1, packing:ex2`. These options provide larger (more fine-grained) rotamer libraries for conformational sampling of amino acid side chains. This can improve results but also increases compute times.

## **Methods**

The RosettaLigand protocol has been implemented as an XML script used with RosettaScripts (Fleishman, Leaver-Fay et al. 2011). Instead of providing a separate RosettaLigand executable, the user creates an XML script that describes each of the pieces of ligand docking, and passes this script to the RosettaScripts executable. This provides a large degree of flexibility to the user, and allows him or her to create novel approaches to ligand docking. The modifications to Rosetta C++ code that have made user defined ligand docking protocols possible are summarized in Appendix N.

In this section XML scriptable components directly related to ligand docking are described. After describing each component we combine them to demonstrate a complete ligand docking protocol that replicates the previously published protocol. Hundreds of additional components that are not ligand-centric are available and described in the RosettaScripts documentation found in the user guide. The XML components below are presented in the order in which they would be used during ligand docking.

### ***StartFrom***

Provide a list of possible xyz starting Coordinates for your ligand. One of these points is chosen at random and the ligand specified by chain is recentered at this position.



```
<StartFrom name=(string) chain=(string)/>  
  <Coordinates x=(float) y=(float) z=(float)/>  
</StartFrom>
```

### ***Translate***

Randomly move the ligand up to a specified distance in any direction from its starting position. If you are confident about your ligand's starting position and seek only to fine tune this position, consider selecting from a Gaussian distribution, where the specified angstroms represent 1 standard deviation from the starting point. If the random translation lands the ligand centroid on a point occupied by a protein atom, then try another random translation. Repeat this cycles number of times before giving up and leaving the ligand at the starting point. If the force option is specified as true, then even if no position lands the centroid in an empty location, the translation that led to the most ideal location (least overlapping) will be chosen. If comma separated tag\_along\_chains are specified, then the same translation that is applied to the ligand will be applied to these chains. This is useful in the case of waters and metals, where it is known that a water or metal should maintain a position close to the ligand. During initial placement of the ligand, tag\_along\_chains assures that related waters and metals stay with the ligand.

```
<Translate name=(string) chain=(string) distribution=[uniform|gaussian]  
angstroms=(float) cycles=(int) force=[true|false] tag_along_chains=(comma  
separated list of chains)/>
```

### ***Translates***

The Translates mover is similar to the translate mover only it applies translation separately to multiple molecules with the same chain identifier. This mover is generally used in conjunction with the command line flag:

“-in:file:treat\_residues\_in\_these\_chains\_as\_separate\_chemical\_entities <one letter chain IDs>”.

For instance if you are interested in allowing each explicit water molecule to move within a 1 Å sphere, you could first relabel all PDB waters of interest to have the chain ‘W’. Next, supply the following command-line flag:

```
“-in:file:treat_residues_in_these_chains_as_separate_chemical_entities W”.
```

Finally in your XML file, use Translates to move waters.

```
<Translates name=water_mover chain=W distribution=uniformgaussian angstroms=1 cycles=25 force=true/>
```

### ***CompoundTranslate***

CompoundTranslate is a special mover that takes Translate and Translates movers as children. If you are performing simultaneous docking of multiple ligands, you should use this mover. This mover first removes all the ligands specified by its children classes. It then places each ligand in the binding pocket in random order or in the order the child elements are listed.

```
<CompoundTranslate name=(string) randomize_order=[true|false]
allow_overlap=[true|false]>
  <Translate...>
  <Translates...>
</CompoundTranslate>
```

### ***Rotate***

Randomly rotate the ligand through all rotational degrees of freedom. Specify 360 degrees for full rotational freedom. ‘Cycles’ in this case is much more complicated than seen in Translate. Perform up to ‘cycles’ random rotations of the ligand. Only rotations that pass a Lennard-Jones attractive and repulsive score filter are stored. Also, rotations that are close in RMSD to other rotations are not stored. Once a minimum number of diverse structures are collected (this minimum is 5 times the number of ligand rotatable bonds) one of these structures is chosen at random as the starting structure. If no structures passed the attractive and repulsive filter just select the rotation with the best attractive and repulsive score.

This somewhat complicated rotation selection scheme is designed to enrich for hard to find ligand poses, such as those which access narrow and deep binding cavities. By storing only rotations that pass an energy filter we limit ourselves to rotations that are close to the protein but do not clash with it. By storing only poses with a minimum RMSD from each other, we increase the probability of selecting ‘hard to find’ poses (classes of similar ligand orientations that easily fit in the interface are only stored once). If you prefer to accept the first rotation, without filtering, just use cycles=1. If small molecules such as waters and metals are associated with the ligand, these can rotate together with the ligand by use of the tag\_along\_chains option.

```
<Rotate name=(string) chain=(string) distribution=[uniform|gaussian]
degrees=(int) cycles=(int) tag_along_chains=(comma separated list of
chains)/>
```

### ***Rotates***

The Rotates mover is similar to the Rotate mover only it applies rotation separately to multiple molecules with the same chain identifier. This mover is generally used in conjunction with the command line flag:

“-in:file:treat\_residues\_in\_these\_chains\_as\_separate\_chemical\_entities <one letter chain IDs>”.

```
<Rotates name=(string) chain=(string) distribution=[uniform|gaussian]
degrees=(int) cycles=(int) force=[true|false]/>
```

### ***SlideTogether***

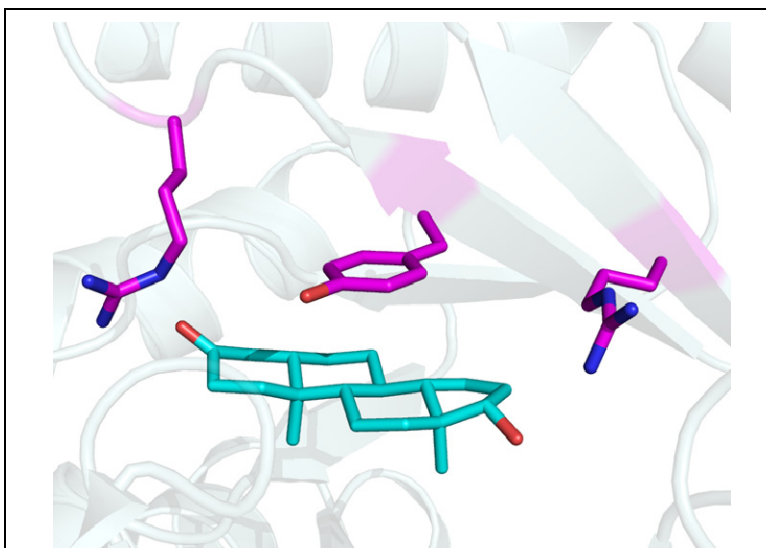
After an initial random positioning of the ligand, the ligand must be moved into close proximity to the protein. SlideTogether moves the ligand toward the protein, 2 Å at a time, until the two collide (as evidenced by a positive repulsive score). The step size is halved several times (1 Å, 0.5 Å, 0.25 Å) to minimize the distance between the ligand and the protein. This step proves to be crucial to Rosetta ligand docking. Without it interactions between amino acid side chains and the ligand are rare.

```
<SlideTogether name="&string" chain="&string"/>
```

### ***HighResDocker***

During high resolution docking, cycles of rotamer trials (sampling of side chain rotamers, one side chain at a time) and repacking (simultaneous sampling of rotamers for multiple side chains) are combined with small movements of the ligand(s). The size of these movements is described by the `high_res_angstroms` and `high_res_degrees` options of `LIGAND_AREAS` (see Note 3). `LIGAND_AREAS` are part of `INTERFACE_BUILDERS` (see Note 4) which are part of `MOVEMAP_BUILDERS` (see Note 5).

The `movemap_builder` describes which amino acid residues to include in rotamer trials, repacking, and minimization. If a ‘resfile’ is provided, interface residues are allowed to redesign (change amino acid identity), according to instructions provided in the specified file. Resfiles can also be specified through the command line flag “-packing:resfile”. Resfile support allows simultaneous optimization of ligand conformer, ligand pose, and protein interfaces constitution. This is useful in the case of protein therapeutics designed to sequester small molecules in vivo. For instance, proteins designed to bind circulating dihydrotestosterone



**Figure III-2. RosettaLigand design of HisF/DHT interface.** Dihydrotestosterone (DHT) was placed in the HisF binding site, using RosettaLigand to dock the ligand and design interface residues. The designed in residues (magenta) allow for hydrogen bonding contacts and structural stability to DHT (cyan). Figure contributed by Brittany Allison.

(DHF) could potentially be used to prevent prostate growth (Lund, Munson et al. 2004). Brittany Allison, a graduate student in the Meiler lab, used RosettaLigand XML with interface design enabled to predict mutations to the imidazole glycerol phosphate synthase (HisF) interface that could increase affinity for DHF (Figure III-2).

The HighResDocker allows the user to specify how many cycles of docking and how often to do a full repack (repack\_every\_Nth - only rotamer trials occur in the other cycles). After each cycle the structure is minimized. If minimize\_ligand values were specified in LIGAND\_AREAS, ligand torsion angles are minimized as well. Monte Carlo sampling is used with a Boltzmann criterion to determine whether to accept or reject the new structure after each cycle. If a tether\_ligand value greater than 0 is specified in LIGAND\_AREAS, the ligand will be restrained by the specified distance (in angstroms). The 'tether\_ligand' option prohibits multiple cycles of small translations in the same direction from moving the ligand farther than desired. Read the notes section on 'LIGAND\_AREAS' for more information about the 'tether\_ligand' option.

```
<HighResDocker name="string" cycles=(int) repack_every_Nth=(int)
scorefxn="string" movemap_builder="string" resfile="string"/>
```

### ***FinalMinimizer***

Minimize the structure docking protein/ligand complex. This includes off-rotamer side-chain torsion angle sampling. The movemap\_builder specifies which residues to minimize. If Calpha\_restraints were specified in LIGAND\_AREAS backbone  $\phi/\psi$  angles are minimized as well.

```
<FinalMinimizer name=(string) chain=(string) scorefxn=(string)
movemap_builder=(string)>
</FinalMinimizer>
```

### ***InterfaceScoreCalculator***

This component calculates a myriad of ligand specific scores and appends them to the output file. After scoring the complex the ligand is moved 1000 Å away from the protein. The model is then scored again. An interface score is calculated for each score term by subtracting separated energy from complex energy. If a native structure is specified, 4 additional score terms are calculated:

1. `ligand_centroid_travel`. The distance between the native ligand and the ligand in our docked model.
2. `ligand_radious_of_gyration`. An outstretched conformation would have a high radius of gyration. Ligands tend to bind in outstretched conformations.
3. `ligand_rms_no_super`. RMSD between the native ligand and the docked ligand.
4. `ligand_rms_with_super`. RMSD between the native ligand and the docked ligand after aligning the two in XYZ space. This is useful for evaluating how much ligand flexibility was sampled.

```
<InterfaceScoreCalculator name=(string) chains=(comma separated chars)
scorefxn=(string) native=(string)/>
```

### ***Putting it all together***

The following XML script replicates the protocol presented in Davis, 2009 (Davis and Baker 2009):

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <ligand_soft_rep weights=ligand_soft_rep>
      <Reweight scoretype=hack_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=hack_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>
single movers
  <StartFrom name=start_from chain=X>
    <Coordinates x=-1.731 y=32.589 z=-5.039/>
  </StartFrom>
  <CompoundTranslate name=compound_translate randomize_order=false
    allow_overlap=false>
    <Translate chain=X distribution=uniform angstroms=5.0 cycles=50
      force=true/>
    <Translate chain=Z distribution=uniform angstroms=5.0 cycles=50
      force=true/>
    <Translates chain=W distribution=uniform angstroms=5.0 cycles=50
      force=true/>
  </CompoundTranslate>
  <Rotate name=rotate chain=X distribution=uniform degrees=360 cycles=500/>
  <Rotates name=rotates chain=W distribution=uniform degrees=360
    cycles=15/> rotate each water molecule
  <SlideTogether name=slide_together chains=X,W,Z/>
  <HighResDocker name=high_res_docker cycles=6 repack_every_Nth=3
    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
    native="inputs/7cpa_native.pdb"/>

```

#### **compound movers**

```
<ParsedProtocol name=low_res_dock>  
  <Add mover_name=start_from/>  
  <Add mover_name=compound_translate/>  
  <Add mover_name=rotate/>  
  <Add mover_name=slide_together/>  
</ParsedProtocol>  
<ParsedProtocol name=high_res_dock>  
  <Add mover_name=high_res_docker/>  
  <Add mover_name=final/>  
</ParsedProtocol>  
</MOVERS>  
<PROTOCOLS>  
  <Add mover_name=low_res_dock/>  
  <Add mover_name=high_res_dock/>  
  <Add mover_name=add_scores/>  
</PROTOCOLS>  
</ROSETTASCRIPTS>
```

Because of the flexibility of ligand docking through RosettaScripts, it is easy to customize the above protocol. For instance, high throughput virtual screening of libraries of compounds can be accomplished by spending more time in low resolution docking. Results from low resolution docking can be filtering and used for high resolution docking. A variety of XML elements not specific to ligand docking can also be included as part of a docking study (see the Materials section).

A customized ligand docking protocol must take into consideration the number of desired output models (see Note 2), and the amount of time it will take to produce each model, given the available hardware (see Note 6). Best energy output models are then selected for further analysis (see Note 7), and used to generate testable hypotheses about protein/ligand interactions.



## Notes

### *Score Term reweighting*

The ligand weights specified in the database file “new.ligand.wts” perform well on a benchmark of diverse protein/ligand complexes. However results can be improved if weights are optimized for the class of protein/ligand interactions in which one is interested. We recently used a leave-one-out analysis to improve the correlation between experimental binding energy and Rosetta predicted binding energy for HIV-1 protease mutants bound to various protease inhibitors. Our leave-one-out weight optimization improves our correlation from 0.31 to 0.71 (Lemmon, Kaufmann et al. 2012).

### *How many models should I make?*

The number of models one should make is largely determined by how large of an interface one is sampling. For this reason carefully describing the size and shape of an interface can save much compute time. By adjusting the angstroms parameter of Translate and adding more StartFrom Coordinates, a user can restrict sampling to a smaller area. Another strategy is to create a limited number of models, then cluster the results based on RMSD (see section 4.4). Select several low energy clusters for further analysis. Select a model from each cluster. Use these models in ligand docking studies, after decreasing the size of angstroms in the Translate mover.

### ***LIGAND\_AREAS***

LIGAND\_AREAS describe parameters specific to each ligand, useful for multiple ligand docking studies (Figure I-1). "cutoff" is the distance in angstroms from the ligand an amino-acid's C-beta atom can be and that residue still be part of the interface. "all\_atom\_mode" can be

true or false. If `all_atom_mode` is true then if any ligand atom is within cutoff angstroms of the C-beta atom, that residue becomes part of the interface. If false, only the ligand neighbor atom is used to decide if the protein residue is part of the interface. "`add_nbr_radius`" increases the cutoff by the size of the ligand neighbor atom's radius specified in the ligand `.params` file. This size can be adjusted to represent the size of the ligand, without entering `all_atom_mode`. Thus `all_atom_mode` should not be used with `add_nbr_radius`.

Ligand minimization can be turned on by specifying a `minimize_ligand` value greater than 0. This value represents the size of one standard deviation of ligand torsion angle rotation (in degrees). By setting `Calpha_restraints` greater than 0, backbone flexibility is enabled. This value represents the size of one standard deviation of Calpha movement, in angstroms.

During high resolution docking, small amounts of ligand translation and rotation are coupled with cycles of rotamer trials or repacking. These values can be controlled by the '`high_res_angstrom`' and '`high_res_degrees`' values respectively. Cycles of small ligand translations can lead to a large translation. In some cases the ligand can “walk away from the protein”. The `tether_ligand` option prevents this by keeping the ligand close to its starting point during cycles of `high_res_docking`. This occurs via a harmonic distance constraint where the constraint function is of the form  $f(x) = \left(\frac{x-x_0}{std\ dev}\right)^2$ , where  $x-x_0$  represents the ligand centroid travel distance and standard deviation is supplied through the `tether_ligand` flag.

```
<[name_of_this_ligand_area] chain="&string" cutoff=(float)
add_nbr_radius=[true|false] all_atom_mode=[true|false] minimize_ligand=[float] Calpha_restraints=[float] high_res_angstroms=[float] high_res_degrees=[float] tether_ligand=[float]/>
```

## ***INTERFACE\_BUILDERS***

An interface builder describes how to choose residues that will be part of a protein-ligand interface. These residues are chosen for repacking, rotamer trials, and backbone minimization during ligand docking. The initial XML parameter is the name of the interface\_builder (for later reference). "ligand\_areas" is a comma separated list of strings matching LIGAND\_AREAS described previously. Finally 'extension\_window' surrounds interface residues with residues labeled as 'near interface'. This is important for backbone minimization, because a residue's backbone can't really move unless it is part of a stretch of residues that are flexible.

By specifying multiple ligand areas, multiple ligand docking is enabled. Simultaneous docking of multiple ligands, cofactors, water molecules and ions may capture synergistic effects overlooked by serial docking (Fig 2).

```
<[name_of_this_interface_builder] ligand_areas=(comma separated list of predefined ligand_areas) extension_window=(int)/>
```

## ***MOVEMAP\_BUILDERS***

A movemap builder constructs a movemap. A movemap is a 2xN table of true/false values, where N is the number of residues your protein/ligand complex. The two columns are for backbone and side-chain movements. The movemap builder combines previously constructed backbone and side-chain interfaces (see previous section). Leave out bb\_interface if you do not want to minimize the backbone. The minimize\_water option is a global option. If you are docking water molecules as separate ligands (multi-ligand docking) these should be described through LIGAND\_AREAS and INTERFACE\_BUILDERS.

```
<[name_of_this_movemap_builder] sc_interface=(string) bb_interface=(string) minimize_water=[true|false]/>
```

**Table 0-1: Relative speed for various components of RosettaLigand XML.** Carboxypeptidase A was docked with a phosphonate inhibitor (PDB Code: 7CPA). The ligand has 9 rotatable bonds. Each datapoint represents the average time in seconds for 10 runs. The combined protocol uses Rotate(360, 1000), HighResDocker with ligand flexibility and 6 cycles of packing (full repacks at cycles 1 and 4), and FinalMinimizer with backbone flexibility.

Amino acid rotamers Ligand Conformations	Standard rotamers				Extended Rotamers (ex1, ex2)			
	1	10	100	500	1	10	100	500
RosettaScripts startup	4.87	4.80	4.87	4.92	4.86	4.87	4.89	4.83
only setup movers	5.81	5.73	5.76	5.72	5.71	5.77	5.91	5.72
Start From	5.84	5.80	5.80	5.72	5.88	5.74	5.76	5.80
Translate (5, 50)	6.05	6.04	5.88	5.84	5.94	6.04	5.83	5.85
Rotate (360, 1)	6.42	6.37	4.74	6.27	6.40	6.40	4.44	6.27
Rotate (360, 1000)	76.32	44.81	78.42	40.50	82.94	42.31	68.18	39.71
SlideTogether	5.85	5.98	5.88	5.84	5.85	5.91	5.81	5.87
HighResDocker 1 RT	7.92	7.87	7.89	7.85	8.32	8.29	8.35	8.35
+ MinimizeLigand	8.23	8.21	8.22	8.43	8.32	8.26	8.20	8.34
HighResDocker 1 FR	6.37	6.30	6.38	6.33	11.93	11.85	12.00	11.81
+ ligand flexibility	6.43	6.38	6.38	6.33	11.77	11.70	11.91	11.84
FinalMinimizer	8.95	8.89	8.98	9.06	8.90	8.89	8.97	9.17
+ backbone flexibility	14.04	14.26	14.32	13.92	14.04	14.24	14.16	12.26
AddScores	6.02	5.87	5.84	5.95	5.88	5.87	5.77	6.05
Combined	86.77	87.20	95.88	83.35	104.19	98.40	68.36	53.46

***How long will this take to run?***

Of course this question depends on many factors: how fast your computer is, how many processors you have access to, how large is your protein? Increasing amino acid rotamers and ligand conformers can increase run-time. Protein backbone and ligand torsion angle minimization also add increase run-time. We have found that the majority of the time is spent in full-repack cycles of ligand docking. Table III-1 shows average times for modeling the interaction of Carboxypeptidase A with a phosphonate inhibitor. The XML script described under the heading “putting it all together” was used with the exception of modifications shown in column headings.

### ***How do I analyze my results***

When your docking study has finished you will have an output file (specified by the `–out:file:atom_tree_diff` option) which contains hundreds of models constructed and scored by Rosetta. You can extract these models to individual PDBs using RosettaScripts. Prepare an XML script that is essentially empty. Under `<PROTOCOLS>` include this line: `<Add mover_name=null/>`. Run the XML script with the following command line or flags file options:

5. `-in:file:atom_tree_diff <input file name>`
6. `-in:file:extra_res_fa <names of .params files>`
7. `–parser:protocol <name of XML file with null mover>`
8. `–database <directory of Rosetta Database>`

You may only be interested in the best models by interface score or by total score. You can list the TAGs of the models you wish to extract at the end of the command line. These tags are found in the `atom_tree_diff` output file after “POSE\_TAG”. You can search the file for lines that start with “SCORES”. By sorting these scores you can find the lowest energy models.

You can also use the Rosetta Cluster application to group your models by RMSD. Then you can choose one low energy model from several low energy clusters for further analysis. For more information, review the cluster documentation:

[http://www.rosettacommons.org/manuals/archive/rosetta3.1\\_user\\_guide/app\\_cluster.html](http://www.rosettacommons.org/manuals/archive/rosetta3.1_user_guide/app_cluster.html).

### ***Fragmentation for flexibility and design***

Rosetta code now supports ligands that consist internally as several independent ‘residues’. Analogous to the concept of side chain rotamers (Dunbrack and Karplus 1993), rotamer libraries can be generated for each of these ligand fragment residues. Appendix I reports

on our progress in this area. Appendix J extends the idea of ligand fragments to present design of small molecules through incremental construction.

## References

- Chaudhury, S. and J. J. Gray (2008). "Conformer selection and induced fit in flexible backbone protein-protein docking using computational and NMR ensembles." Journal of Molecular Biology **381**(4): 1068-1087.
- Davis, I. W. and D. Baker (2009). "RosettaLigand docking with full ligand and receptor flexibility." Journal of Molecular Biology **385**(2): 381-392.
- Davis, I. W., K. Raha, et al. (2009). "Blind docking of pharmaceutically relevant compounds using RosettaLigand." Protein Science **18**(9): 1998-2002.
- Dunbrack, R. L., Jr. and M. Karplus (1993). "Backbone-dependent rotamer library for proteins. Application to side-chain prediction." Journal of Molecular Biology **230**(2): 543-574.
- Fleishman, S. J., A. Leaver-Fay, et al. (2011). "RosettaScripts: a scripting language interface to the Rosetta macromolecular modeling suite." PLoS One **6**(6): e20161.
- Jiang, L., E. A. Althoff, et al. (2008). "De novo computational design of retro-aldol enzymes." Science **319**(5868): 1387-1391.
- Kaufmann, K. W., E. S. Dawson, et al. (2009). "Structural determinants of species-selective substrate recognition in human and Drosophila serotonin transporters revealed through computational docking studies." Proteins **74**(3): 630-642.
- Kaufmann, K. W., K. Glab, et al. (2008). Small Molecule Rotamers Enable Simultaneous Optimization of Small Molecule and Protein Degrees of Freedom in ROSETTALIGAND Docking. German Conference on Bioinformatics. A. Beyer and M. Schroeder. Dresden: 148-157.
- Kaufmann, K. W., G. H. Lemmon, et al. (2010). "Practically useful: what the Rosetta protein modeling suite can do for you." Biochemistry **49**(14): 2987-2998.
- Lazaridis, T. and M. Karplus (1999). "Effective energy function for proteins in solution." Proteins **35**(2): 133-152.
- Lemmon, G., K. Kaufmann, et al. (2012). "Prediction of HIV-1 Protease/Inhibitor Affinity using RosettaLigand." Chemical Biology and Drug Design **79**(6): 888-896.

- Lemmon, G. and J. Meiler (2012). "Rosetta Ligand docking with flexible XML protocols." Methods in Molecular Biology **819**: 143-155.
- Lund, T. D., D. J. Munson, et al. (2004). "Equol is a novel anti-androgen that inhibits prostate growth and hormone feedback." Biology of Reproduction **70**(4): 1188-1195.
- Meiler, J. and D. Baker (2006). "ROSETTALIGAND: protein-small molecule docking with full side-chain flexibility." Proteins **65**(3): 538-548.
- Raman, S., R. Vernon, et al. (2009). "Structure prediction for CASP8 with all-atom refinement using Rosetta." Proteins-Structure Function and Bioinformatics **77**: 89-99.

## CHAPTER 4

# TOWARDS LIGAND DOCKING INCLUDING EXPLICIT INTERFACE WATER MOLECULES

This chapter will be published as part of a PLoS ONE special collection focused on advances to Rosetta showcased at RosettaCon 2012.

### **Abstract**

Small molecule docking seeks to predict the interaction of a small molecule ligand with a protein at atomic-detail accuracy including position and conformation of the ligand but also conformational changes of the protein upon ligand binding. While successful in the majority of cases, leading docking algorithms including RosettaLigand fail in some cases to predict the correct protein/ligand complex structure. In this study we show that simultaneous docking of explicit interface water molecules greatly improves Rosetta's ability to distinguish correct from incorrect ligand poses. This result holds true for both protein-centric water docking, wherein waters are located relative to the protein binding site, and ligand-centric water docking, wherein waters move with the ligand during docking. Protein-centric docking is used to model 99 HIV-1 protease/protease inhibitor structures. We find protease inhibitor placement improving 9 times as often as it worsens when waters are also docked. Ligand-centric docking is applied to 341 structures from the CSAR benchmark of diverse protein/ligand complexes (Dunbar, Smith et al. 2011). Across this diverse dataset we see up to 56% recovery of failed docking studies when waters are also docked.



## Introduction

Small molecule docking methods seek to predict the structure of a protein/ligand complex (Huang and Zou 2010). Ligand docking generally consists of two components: *sampling* of the conformational space, and *scoring* of the resultant complex structures (Halperin, Ma et al. 2002). Sampling of the conformational space typically includes ligand position with respect to the protein (translation and rotation, often called ‘pose’), ligand conformation, and protein conformation. Scoring seeks to distinguish the correct from incorrect binding poses by comparing estimates of binding affinity. It is characterized by a trade-off between accuracy and speed (Halperin, Ma et al. 2002; Kim and Skolnick 2008). Myriad sampling and scoring algorithms have been developed and are reviewed elsewhere (Sousa, Fernandes et al. 2006). These approaches are often able to sample the correct binding pose, but satisfactory prediction of binding affinity has yet to be achieved (Kim and Skolnick 2008). One particular challenge in ligand docking studies is the positioning of interface water molecules (Sousa, Fernandes et al. 2006).

That interface water molecules play an important role in ligand binding is evidenced by the fact that many protein/ligand complexes contain structured water molecules that bridge protein and ligand. For instance in the CSAR dataset used in this paper, 299 out of 341 complexes include waters within hydrogen bonding distance of both protein and ligand atoms. These water molecules are often absent in experimental structures of the apo protein (Ni, Sotriffer et al. 2001). Water molecules stabilize protein/ligand interfaces by providing additional indirect interactions between protein and ligand through formation of hydrogen bonds with both partners (Sarkhel and Desiraju 2004). In an analysis of the geometric characteristics of hydrogen bonds found in complexes obtained from the PDB, Panigrahi and Desiraju determine that ligands

tend to prefer forming strong hydrogen bonds with protein residues and weaker interactions with water (Panigrahi and Desiraju 2007). This relationship between strong and weak interactions may help to fine-tune entropy-enthalpy requirements of ligand binding (Sarkhel and Desiraju 2004).

The addition of water molecules to protein/ligand docking studies has two opposing effects. Waters increase the conformational space – i.e. the number of possible protein/ligand/water interactions. At the same time however, waters reduce the ‘reasonable pose’ search space by presenting a more crowded binding site. This increases the chance for a protein to evolve to recognize a small molecule. In the fine-tuning of the thermodynamics and kinetics during evolution, water molecules might be removed or added. In contrast, human-designed interactions of proteins with ligands (drug discovery) might have fewer water molecules in the interface because in structure-based computer-aided drug design waters have typically been ignored (Schneider and Fechner 2005). Nevertheless, while computationally demanding, a number of drug design approaches now model water positioning (de Beer, Vermeulen et al. 2010).

Panigrahi and Desiraju find an average of 118 hydrogen bonds formed by water in the active site of each of 251 complexes studied. These included bonds between water and protein, water and ligand, and water and water (Panigrahi and Desiraju 2007). Similarly, we find that the number of water molecules in active sites within the CSAR dataset range from 0 to 15, with an average of 2.8 waters per interface. Networks of water/water hydrogen bonds can contribute to the stability of the complex by keeping bridging water molecules in the right position (Poornima and Dean 1995). Water molecules can also bridge protein/protein interactions, further stabilizing protein conformation (Ikura, Urakubo et al. 2004; Cameron, Short et al. 2007).

In scoring functions optimized to predict binding affinities (Bohm 1994; Jain 1996) components such as hydrogen bond energy have been weighted to *implicitly* account for the change in energy compared to hydrogen bonds formed with water (Rarey, Kramer et al. 1999). Similarly the “hydrophobic” score terms are used to *implicitly* represent desolvation of the protein receptor. Yet significant improvements have been seen in molecular dynamics-based binding affinity prediction when *explicit* waters are considered (Young, Abel et al. 2007; Deng and Roux 2008). These improvements suggest modeling of *explicit* waters may also improve binding affinity predictions in ligand docking studies.

For the present study we introduce the notions of a “protein-centric” approach that places water into the protein binding site prior to docking e.g. at polar groups and/or identified in crystallographic studies of the protein. In this approach the water location is tied to the protein. In the “ligand-centric” approach water is placed around polar groups of the ligand and moves with the ligand during the docking simulation. Depending on the scientific question asked, both approaches have merit: the protein-centric approach has the advantage that often likely water positions are known from crystallographic studies and can easily be incorporated. Water molecules often interact with multiple functional groups on the protein making it easier to predict possible water positions *de novo*. An advantage of a ligand-centric approach is that the surface of drug-like ligands is typically smaller with fewer polar groups when compared with the potential protein binding interface. Therefore fewer water positions need to be considered when placing them around ligand polar groups increasing sampling efficiency. So far, mostly protein-centric approaches have been tested.

In both self-docking (Roberts and Mancera 2008) and cross docking studies (Thilagavathi and Mancera 2010), correct ligand binding pose prediction can be greatly improved by the

presence of conserved crystallographic waters. For instance a FlexX prediction of an HIV-1 protease/protease inhibitor interface fails without the inclusion of a key water molecule known to be important for binding. Prepositioning this water at its known crystallographic coordinate leads to a practically perfect prediction (Kramer, Rarey et al. 1999). In this case the effect of water had little to do with *scoring* and everything to do with guiding the *sampling* algorithm. De Graaf et al. find RMSD accuracy improved 18% for AutoDock, 23% for FlexX, and 11% for GOLD when crystallographic waters were included (de Graaf, Pospisil et al. 2005) in Cytochrome P450 binding sites. Inclusion of crystallographic waters in the thymidine kinase binding site leads to 17% (Autodock), 35% (FlexX) and 0% (GOLD) improvements in RMSD prediction.

Nevertheless, explicit prediction of the *location* of key water molecules when docking ligands is not standard in current docking algorithms and limited to few specific examples: In a protein-centric approach, De Graaf et al. used GRID to preposition potential water positions within the binding pockets of 19 cytochrome P450 and 19 thymidine kinase crystal structures. These waters were present during docking predictions using AutoDock, FlexX, and GOLD. The authors found RMSD accuracy improved by 70% (Autodock), 32% (FlexX) and 7% (GOLD) for Cytochrome P450 docking 23% (Autodock), 12% (FlexX) and 23% (Gold) in RMSD placement for thymidine kinase.

In a *protein-centric* approach, the FlexX algorithm was extended to determine optimal placements of waters in protein active sites prior to ligand docking. These waters are added if they can form favorable hydrogen bonds with the ligand. A docking study including 200 protein/ligand complexes from the PDB shows mixed results. The average rank of the first model under 1Å drops from 23.9 to 14.8 but the average rank of the first model under 2Å increases from 6.4 to 10.2 (Rarey, Kramer et al. 1999).

Within the program GOLD, protein-centric waters can be turned on and off during docking and are rotated independently to optimize orientation. A constant penalty representing loss of entropy is added for waters that are switched on, thus rewarding displacement of water. A dataset of 28 protein/ligand complexes was studied where each complex contains one or more water molecules which form key hydrogen bonding interactions between protein and ligand. These crystallographic waters were input into the GOLD docking simulation and allowed to switch on and off. While 90% of the waters are correctly switched on, pose prediction success increases by just 2% (Chemscore) or 8% (Goldscore) (Verdonk, Chessari et al. 2005). When GOLD is challenged with displacing waters found in other crystal-structures of the same target, or when decoy waters are positioned using SuperStar, success rates for pose prediction decrease by 5-10% (Verdonk, Chessari et al. 2005).

The SLIDE approach is also protein-centric, and begins with crystallographic waters from an unliganded protein. Waters that are likely to be conserved are allowed to mediate protein-ligand interactions during docking. (Schnecke and Kuhn 2000). These waters can shift position or be removed when they collide with ligand atoms. Since the SLIDE approach focuses on the efficient screening of large databases of compounds it is not appropriate to directly compare results with other docking tools.

AutoDock can model protein flexibility as an ensemble of protein structures. Österberg et al. demonstrate that the inclusion of crystallographic water in these Autodock ensembles can lead to improved docking results. Specifically they collect 21 HIV-1 protease structures, 20 of which contain a key water, necessary for binding their respective inhibitors. One complex does not incorporate this water. The authors perform a cross-docking study where each of the 21 HIV-1 protease inhibitors are docked with each of the 21 protease structures using Autodock. In each

case, Autodock chooses a protease structure from the ensemble in which the presence of water is correctly determined. (Osterberg, Morris et al. 2002).

Lie et al. present a *ligand-centric* model for docking with waters. Waters are placed around and move with the ligand. These waters rotate along with flexible ligand torsions while the protein receptor is kept rigid. The authors chose 12 protein/ligand complexes in which docking studies without water failed and docking studies that consider all crystallographic water molecules succeed. Results from docking with ligand-centric waters demonstrate top ranked models with RMSD less than 2.0 Å in 6 out of 12 cases (Lie, Thomsen et al. 2011). Note that this study will not notice if addition of waters leads to failures in cases that were successful without addition of waters. This is a particularly important metric as the majority of docking simulations succeeds without considering water, i.e. worsening the performance here is a major concern.

In this study we present an extension to RosettaLigand software (Meiler and Baker 2006) that allows the inclusion of water molecules. The study pushes the boundary of ligand docking with water molecules in several ways: (1) RosettaLigand allows both *protein-centric* and *ligand-centric* water placement and therefore enables a comparison of results. (2) Protein flexibility and ligand flexibility are consistently considered. (3) The *a priori* knowledge of water positions from crystallographic studies is not required. (4) We also use a large (341) dataset of diverse protein/ligand complexes (Dunbar, Smith et al. 2011) to provide a more stringent and comprehensive benchmark than previous studies.

RosettaLigand has been proven effective at generating models of protein/ligand complexes at atomic-detail accuracy ( $< 2.0$  Å) (Das, Qian et al. 2007; Davis and Baker 2009). The RosettaLigand score terms include the 6-12 Lennard-Jones potential (Lennard-Jones 1924), the Lazaridis-Karplus solvation model (Lazaridis and Karplus 1999), a side-chain rotamer score,

based on the Dunbrack rotamer set (Dunbrack and Karplus 1993), a pair potential based on the probability of seeing two amino acids close together in space (Simons, Ruczinski et al. 1999), and an explicit orientation hydrogen bonding model (Kortemme, Morozov et al. 2003). RosettaLigand samples protein and ligand flexibility simultaneously (Davis and Baker 2009). Protein flexibility includes sampling from ensembles of protein backbones, rapid side-chain rotamer sampling, and minimizing backbone  $\phi/\psi$  angles of residues near the ligand. Ligand flexibility is modeled by sampling pre-generated ligand conformers and local minimization of ligand torsion angles.

The RosettaLigand algorithm pairs low resolution sampling of the ligand pose within the protein binding site with high resolution refinement, thus allowing for speed and accuracy. Recent updates to RosettaLigand software have allowed for docking multiple small molecules (including waters, metals, and cofactors) simultaneously (Lemmon and Meiler 2012). In this paper we demonstrate improvements to RosettaLigand docking results when water molecules are included in the interface. We benchmark Rosetta using (1) a set of HIV-1 protease/inhibitor co-crystal structures from the protein data bank (PDB, <http://www.rcsb.org>), and (2) the CSAR benchmark dataset (Dunbar, Smith et al. 2011) (<http://www.csardock.org/>).

HIV-1 protease (PR) plays an essential role in the HIV-1 lifecycle and thus is an important target for drug therapy (Adamson and Freed 2007). PR is the classic success story of structure-assisted drug design (Wlodawer and Vondrasek 1998). The binding of most HIV-1 protease inhibitors (PIs) is mediated by a key water molecule that forms hydrogen bonds between the PI and the PRs flexible loop regions (Louis, Ishima et al. 2007). This interaction is necessary for binding and stabilizes the loops in the closed-conformation (Hornak, Okur et al. 2006). We select 11 protease structures from the PDB, each containing a different protease

inhibitor and slightly different protease sequences (due to mutation). We perform cross-docking studies between each pair of protease structures. We do so using standard docking as well as docking with protein-centric waters with positions identified through crystallographic studies. Our results demonstrate significant improvement in binding pose prediction when water docking is included.

In addition to a homogeneous and well-understood benchmark, we assess the effect of water docking on a benchmark of heterogeneous protein/ligand complexes. The CSAR benchmark includes 341 protein/ligand complexes experimentally determined structures of protein/ligand complexes. Each CSAR datapoint also contains structural waters and  $K_d$  values. CSAR data was prepared for the uniform evaluation of methods for prediction of ligand binding mode and binding affinity (Dunbar, Smith et al. 2011). In 195 of these structures, we find between 1 and 8 water molecules positioned to directly interact with both protein and ligand. Unlike the HIV-1 PR dataset, wherein extensive structural and biochemical studies have confirmed the importance of the key water molecule studied, the waters we study within the CSAR dataset are chosen simply based on their crystallographic coordinates. In this paper the CSAR dataset is subjected to both standard docking and docking with ligand-centric waters. We find significant improvement in model ranking when waters are added. Inhibitor RMSDs are also improved.

## **Materials and Methods**

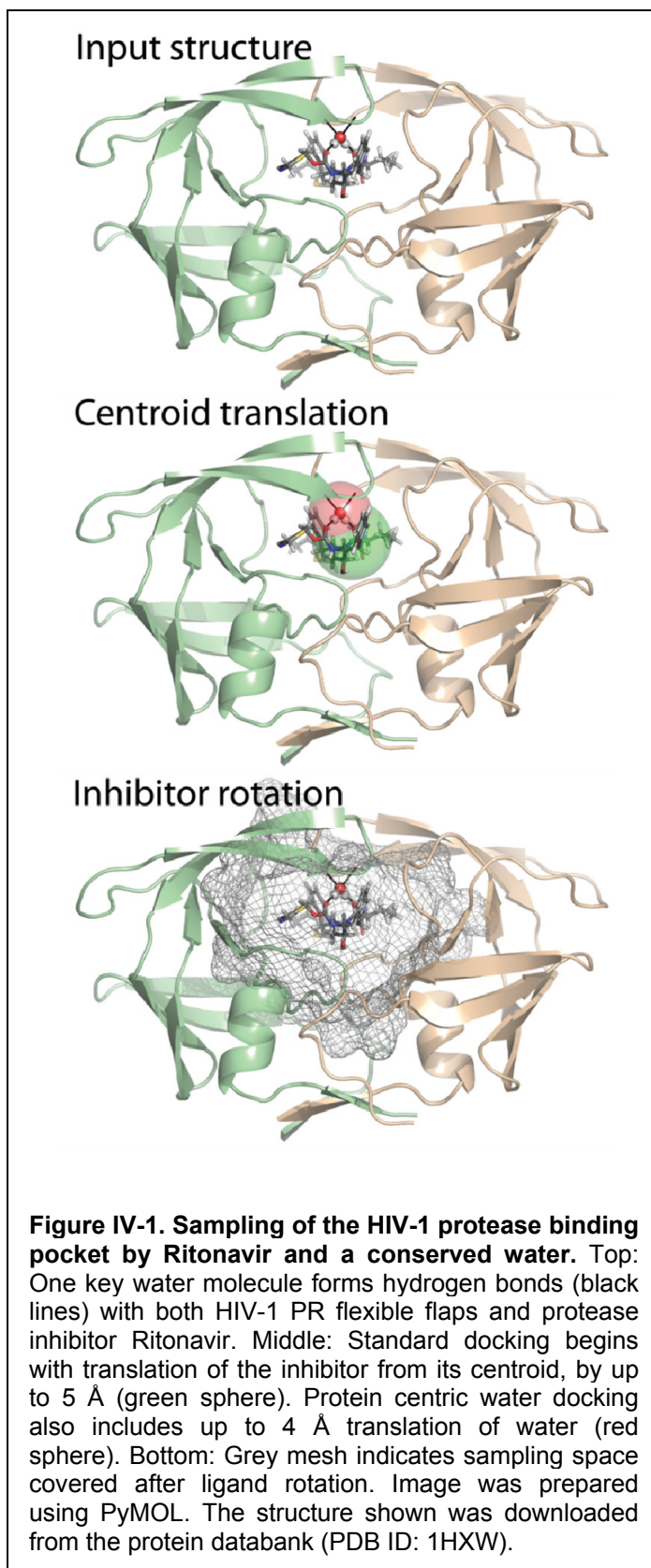
*Preparation of HIV-1 PR inputs for cross docking.* Eleven HIV-1 PR crystal structures representing 9 unique PR sequences and 11 unique protease inhibitors (PIs) were obtained from the PDB. Each of these structures includes a conserved water molecule, known to be important



for stabilizing loop regions during binding (Figure IV-1, top panel). The PR sequences represented herein differ from one another by up to 14 residues per 99 residue chain. Each PI was combined with each protease backbone, producing 99 input structures. Cross docking consists of combining the PI and PR sequence from one complex with the backbone coordinates of another complex. Rosetta ligand docking is challenged to correctly predict PI pose, given incorrect backbone and side-chain starting coordinates. In our study each of the 99 input structures is docked with and without the inclusion/docking of the conserved water molecule mentioned above.

***Preparation of CSAR dataset.*** First we extracted inhibitor atom coordinates from the input files. Rosetta software ships with a script, ``mol_file_to_params.py`` which was used to prepare `.params` files describing the chemical properties of each inhibitor and assigning each inhibitor atom a Rosetta atom type. We wrote scripts that use BioPython to right align residue names, convert non-canonical residues to their canonical base residues, and remove neutralizing caps from N-terminal and C-terminal ends of CSAR input structures. Protein chains were relabeled alphabetically, as they appear in the PDB. The inhibitor was given the chain 'X' and residue code 'INH'. All waters were given the chain ID 'W' and the residue code 'WAT'. We wrote scripts that use PyMOL (Schrodinger 2010) to select interface waters from among all waters in the crystal structure. 'Loose waters' were defined as those with oxygen atoms within 3.0Å of at least one protein and one inhibitor atom. 'Tight waters' have oxygen atoms within 3.0Å of at least 2 protein and two inhibitor atoms. Finally for each inhibitor we used the BCL ([http://www.meilerlab.org/index.php/bclcommons/show/b\\_apps\\_id/1](http://www.meilerlab.org/index.php/bclcommons/show/b_apps_id/1)) to determine LogP, molecular weight, number of rotatable bonds, number of hydrogen bond acceptors, and number of hydrogen bond donors.

*Standard docking – low resolution sampling.* Docking without waters entails first placing the ligand in the putative binding site. Next the ligand is translated randomly within a 5 Å radius sphere (Figure IV-1, middle panel, green sphere). This is repeated up to 50 times, or until the ligand centroid does not clash with the protein in the new inhibitor position. If after 50 cycles of movements no non-clashing placement is identified, the placement with the lowest score is accepted. Next comes a rotation step in which the inhibitor is rotated randomly up to 1000 times to identify a rotation that does not lead to clashes with the protein. Unlike the translation step, which only looks at the ligand centroid, the rotation step affirms that no inhibitor atoms clash with protein atoms. The “slide together” step then slides the inhibitor toward the protein until the two collide. Then the



inhibitor is slid back away from the protein a small amount. This step ensures the inhibitor is in close enough proximity to the protein to allow the high affinity contacts to be formed during high resolution docking. The space sampled by this low resolution protocol is represented as mesh, Figure IV-1, bottom panel.

***Standard docking – high resolution refinement.*** High resolution docking involves small inhibitor translations of up to 0.1 Å and rotations of up to 5°. These movements are coupled with either rotamer trials (sampling of rotamers, one residue at a time) or repacking (sampling rotamers at multiple positions simultaneously). Both rotamer trials and repacking are restricted to residues within 6 Å of an inhibitor. Next a gradient based minimization is applied, which allows for interface side-chain torsion angle adjustments, along with adjustment of inhibitor torsion angles. High resolution docking is repeated 6 times, using a Monte Carlo approach. During a final minimization step, backbone  $\phi/\psi$  angles within 7 Å of the inhibitor are minimized as well. The XML describing low resolution and high resolution standard docking is presented as Appendix K.

***HIV-1 PR/PI cross-docking with a protein-centric water.*** The 99 HIV-1 PR/PI cross-docking inputs (described above) were subjected to a docking protocol in which one key water moves independent of the ligand. In this protein-centric water docking scheme, the interface water is initialized at its conserved coordinates. During translation, this water is allowed to move within a 4Å radius sphere (Figure IV-1, middle panel, red sphere). During rotation, the water is allowed to fully rotate. As hydrogen is generally not resolved in X-ray crystal structures, Rosetta adds hydrogen to the water molecule prior to translation or rotation. The interface definition used to select residues for side-chain repacking and for backbone minimization was extended to include the conserved water molecule. During high resolution docking, this water is allowed to

move in the same fashion as the ligand – 0.1 Å translations and 5° rotations. The XML describing low resolution and high resolution standard docking is presented as Appendix L. Table IV-1 compares protein-centric water docking with other protocols used in this study.

<b>Table IV-1. Comparison of protein-centric and ligand-centric water docking.</b> Protein-centric water docking was used with HIV-1 protease data while ligand-centric water docking was used with CSAR data.		
	<b>Protein Centric Water docking (HIV-1 PR/PIs)</b>	<b>Ligand Centric Waters (CSAR benchmark)</b>
Input preparation	Crystallographic waters within 3.0 Å of protein and inhibitor are included in the docking study	
Inhibitor Translation	<b>Inhibitor</b> moves up to 5 Å, finding a non-clashing location.	<b>Inhibitor &amp; water</b> move together up to 5 Å, finding a non-clashing location.
Water Translation	Up to 50 cycles of 1 Å water movement, first non-clashing move is accepted.	
Inhibitor Rotation	<b>Inhibitor</b> rotates up to 1000 times to optimize attractive & repulsive scores	<b>Inhibitor &amp; water</b> rotate together up to 1000 times to optimize attractive & repulsive scores
Water Rotation	Waters rotate together up to 100 times to optimize attractive & repulsive scores	
High Resolution docking	6 Cycles of inhibitor & water translation (0.1 Å) and rotation (5°). Each cycle coupled with side-chain rotamer sampling & gradient based minimization of side-chain and inhibitor torsion angles.	
Final minimization	Gradient based minimization of backbone and side chain degrees of freedom around the inhibitor and waters.	

**CSAR self-docking with ligand-centric waters.** The 341 CSAR inputs (described above) were subjected to a docking protocol in which waters are translated and rotated along with the ligand before they are allowed smaller independent movements. As in protein-centric water docking, hydrogen is first added to water molecules by Rosetta. Waters then move with the inhibitor up to 5 Å. Next waters are allowed independent movements of up to 1 Å. The inhibitor and waters are then allowed full rotation as a single rigid body. Finally, waters rotate independently. High resolution docking occurs as described for *protein-centric* waters. The XML describing low resolution and high resolution standard docking is presented as Appendix M. Table IV-1 describes the differences between ligand-centric and protein-centric waters.

***Creation of ligand conformers.*** MOE (Molecular Operating Environment) (2011) was used to generate (where possible) 10 diverse, low energy conformations per inhibitor. The program was run with the following criteria: Method: LowModeMD, Rejection Limit: 100, Iteration Limit 1000, RMS Gradient 0.008, MM Iteration Limit: 300, Conformation limit: 10, Energy window: 11, RMSD limit: 2Å. Where 10 conformations were not identified, the RMSD limit was reduced to 1Å, then 0.5Å, then 0.25Å.

***Placement of decoy waters.*** An XYZ-grid was created with gridpoints spaced 0.15 Å apart. The inhibitor coordinates were translated to the grid origin. Around each inhibitor atom was drawn a sphere with a radius equal to its Van der Waals radius. Grid points within these spheres were marked as occupied. Next around each inhibitor hydrogen bond donor or acceptor was drawn a ring with an inner radius of 2.75 Å and an outer radius of 2.9 Å. The sets of grid points that fell within these rings were filtered to remove those grid points occupied by other atoms. Finally, for each set of filtered grid points, the grid point with the shortest distance to all remaining gridpoints in the set was chosen as the coordinate for water placement.

***CSAR self-docking with ligand conformers and water decoys.*** A subset of CSAR data was tested with a protocol that attempts to predict water positions. Ligand-centric water positions are pre-computed around each inhibitor hydrogen bond donor or acceptor atom, as described above. This process is repeated for each conformer. Low resolution docking begins with ligand translation in the absence of water. We simplify ligand centric docking in this study by only considering complexes with one water. After each cycle of rotation one precomputed water is added to the inhibitor or conformer. If this inhibitor/water complex clashes with the protein, the water is removed, the inhibitor undergoes another random rotation, and another water is added. After this rotate step is finalized waters rotate independently of the ligand. During the slide

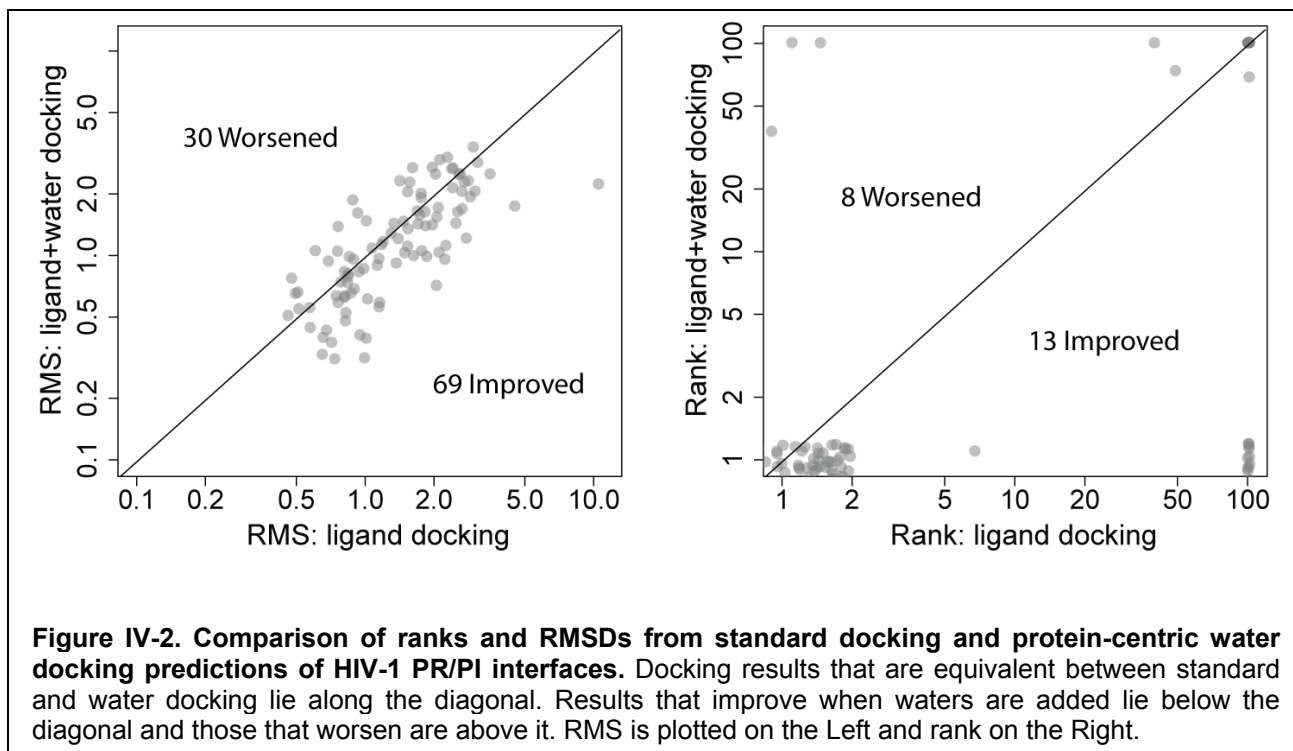
together step waters slide with the ligand toward the protein and are included in the test for clashes. High resolution docking occurs as described for *protein-centric* waters. Table 1 compares the three docking protocols showcased in this paper.

***Docking Model Production and Analysis.*** For both HIV-1 protease and the CSAR dataset a similar approach was used. Regardless of whether we used standard docking, *protein-centric* water docking, or *ligand-centric* water docking, 1000 models were produced per input complex. The top 100 by total Rosetta energy score were selected from among these models. Next, the top model by inhibitor interface score was chosen. When waters were added after clustering ligand conformer docking results, 100 models were produced per placed water.

Computation was split between the Vanderbilt University ACCRE cluster ([www.accre.vanderbilt.edu](http://www.accre.vanderbilt.edu)) and the Center for Structural Biology piranha cluster ([structbio.vanderbilt.edu/comp/hw/piranha](http://structbio.vanderbilt.edu/comp/hw/piranha)). Rosetta revision 49194 was used for all calculations. The additional time per output model necessary to perform protein-centric and ligand-centric water docking is insignificant. However additional sampling (additional output models) may be needed to adequately sample combinations of inhibitor and water placements.

***Ranking metrics.*** Sets of 100 top scoring models described in the previous paragraph were sorted by interface score. Their order of appearance in this sorted list represents their rank. Ranking metrics used in this paper include (1) whether the top ranked model has an RMSD under 2.0Å, (2) whether there exists a model under 2.0Å RMSD within the top 10 ranked structures, (3) the change in rank between top scoring models from two separate studies.

***RMSD calculations.*** The accuracy of models created by RosettaLigand docking was determined by comparing them to the experimentally determined structures, via root-mean-square deviation (RMSD) calculations. We calculate RMSD by (1) summing over the squared



distance for each pair of matching inhibitor atoms between experimental and predicted structures, (2) dividing by the total number of inhibitor atoms, and (3) determining the square root. Note that all RMSDs reported herein are between inhibitor atoms only.

## Results/Discussion

*Protein-centric water docking improves placement of HIV-1 PR inhibitors.* HIV-1 PR/PI docking is mediated by a conserved water that hydrogen bonds between the PI and the PR flexible flap regions (Figure IV-1). The 99 cross-docking PR/PI input structures were subjected to standard docking (without water) and protein-centric water docking, which involves sampling the position and orientation of the conserved water within a 4 Å sphere centered at the crystallographic coordinate. In 69 out of 99 cross docking studies, the addition of water led to top scoring models where the inhibitor was placed more accurately (as measured by RMSD, see

**Table IV-2. Change in RMSD of top Rosetta model of HIV-1 PR/Pis when water is docked.** RMSDs are calculated between top scoring Rosetta model and experimentally determined structure. In green are studies where adding water improved RMSD by greater than 1 Å.

Ligand/Protein	1HXW	1KZK	1LZQ	1OHR	1SDT	1T7J	2NMW	2O4S	5HVP
1HXW-Ritona	-0.102	-0.947	-0.880	0.309	-0.073	-0.194	0.295	-1.285	-0.214
1KZK-AG1776	0.628	-0.120	-0.554	-0.598	-0.147	-0.854	-0.162	-0.440	-0.334
1KZK-KNI272	0.464	-8.299	-0.179	-0.576	0.139	-0.717	-0.434	-1.061	-0.161
1KZK-KNI764	0.678	0.653	-0.262	-0.543	-0.690	-0.498	0.066	-0.426	0.048
1LZQ-Ethyle	0.025	-2.767	-0.030	-0.020	-0.191	-1.552	-0.580	0.197	0.171
1OHR-Nelfin	-0.598	0.785	-0.204	-0.415	0.340	-1.145	-1.323	-0.062	0.042
1SDT-Indina	-0.162	-0.971	-0.198	-0.228	-0.250	-0.637	-0.537	-0.392	-0.324
1T7J-Ampren	-0.018	-0.104	0.989	0.472	-0.116	-0.426	0.040	-1.065	0.259
2NMW-Saquin	-0.336	0.846	0.341	0.907	0.563	-0.195	0.279	1.067	-0.064
2O4S-Lopina	-0.193	-0.971	-1.091	-0.178	-0.336	-0.088	-0.498	-0.352	0.161
5HVP-Acetyl	-0.044	-0.289	0.745	-0.037	0.049	-0.221	0.350	0.478	-0.062

Figure IV-2). When focusing on only significant changes in RMSD larger than 1 Å in magnitude, we observe 9 to 1 ratio of improved to worsened cases (Table IV-2).

***Protein-centric water docking improves ranking of HIV-1 PR inhibitors:*** One metric used to gauge success in docking is the rank of the first model with RMSD under 2.0 Å. That is, in a list of models sorted by Rosetta predicted interface energy, what is the position of the first model in that list with an inhibitor less than 2.0 Å RMSD from the native coordinates? By this metric 13 ranks improve and 8 get worse when water docking is included in modeling of the PR/PI interface (Figure IV-2). Another metric for successful docking is whether the top scoring model has a ligand placed within 2.0 Å RMSD from the experimentally determined position. By this metric standard docking correctly places PIs in 71 out of 99 cases. With *protein-centric* water docking 77 out of 99 cases are successful. Twelve failed studies became successes upon addition of water. Yet 6 successful standard docking studies failed when waters were added



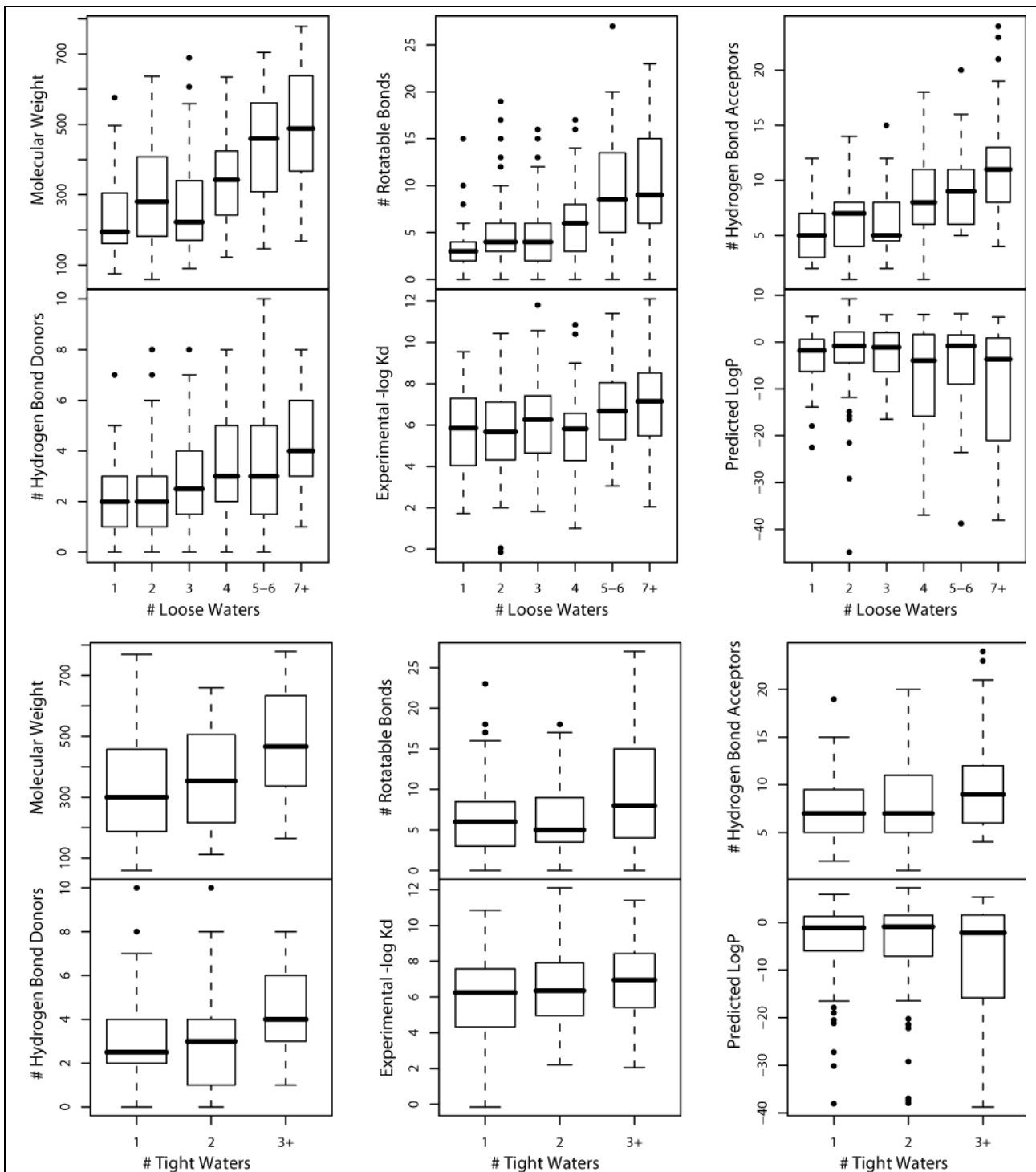
(Table IV-3). Thus RosettaLigand *protein-centric* docking is twice as likely to improve docking results in this particular benchmark. Because of the homogeneous nature of this dataset, differences between success and failure are not due to differences receptor structure, but rather the result of the stochastic nature of our sampling methods. Increased sampling should improve the results of both standard and protein-centric water docking.

**Table IO-3. Change in whether top scoring model of HIV-1 PR/PI is <2.0 Å RMSD when water is docked.** RMSD is calculated between predicted and experimental inhibitor coordinates. A single check mark indicates success under both standard and protein-centric water docking. A single 'X' indicates failures in both cases. Arrows indicate a change from success to failure or vice versa.

Ligand/Protein	1HXW	1KZK	1LZQ	1OHR	1SDT	1T7J	2NMW	2O4S	5HVP
1HXW-Ritona	✓	x→✓	x→✓	✓	✓	✓	✓→x	x→✓	✓
1KZK-AG1776	✓	x	✓	✓	✓	✓	✓	✓	✓
1KZK-KNI272	✓	x	✓	✓	✓	✓	✓	x→✓	✓
1KZK-KNI764	✓	✓→x	✓	✓	✓	✓	✓	✓	✓
1LZQ-Ethyle	✓	x→✓	✓	✓	✓	x→✓	x	✓	✓
1OHR-Nelfin	✓	✓→x	✓	✓	✓	x→✓	x→✓	✓	✓
1SDT-Indina	✓	x→✓	✓	✓	✓	✓	x	x→✓	✓
1T7J-Ampren	✓	x	✓	✓	✓	✓	✓	x→✓	✓
2NMW-Saquin	✓	x	x	✓→x	✓→x	✓	x	✓→x	✓
2O4S-Lopina	✓	x	x	✓	✓	✓	x→✓	x	✓
5HVP-Acetyl	✓	x	x	✓	✓	x	x	x	✓

**Analysis of CSAR dataset.** The CSAR dataset contains 341 protein/inhibitor complex crystal structures, each with a reported binding affinity ( $K_d$ ). The proteins range in size from 119 residues to 2228 residues. The ligands range in size from 9 atoms to 118 atoms. Other properties are summarized in Table IV-4. We filtered crystallographic water molecules based on two criteria: ‘loose waters’ are within 3.0 Å of both a protein and an inhibitor atom; ‘tight waters’ are within 3.0 Å of at least 2 protein and 2 inhibitor atoms. The tight water subset includes an average of 1.1 waters per complex, while the loose water subset retains 3.3 waters per complex on average. Figure IV-3 reveals how various inhibitor properties trend with number of interface

waters within the loose and tight subsets, respectively. As expected, the size of the inhibitor (as measured by molecular mass, number of rotatable bonds, or number of hydrogen bond donors or acceptors) correlates with the number of water molecules that form interactions with the inhibitor and the protein.



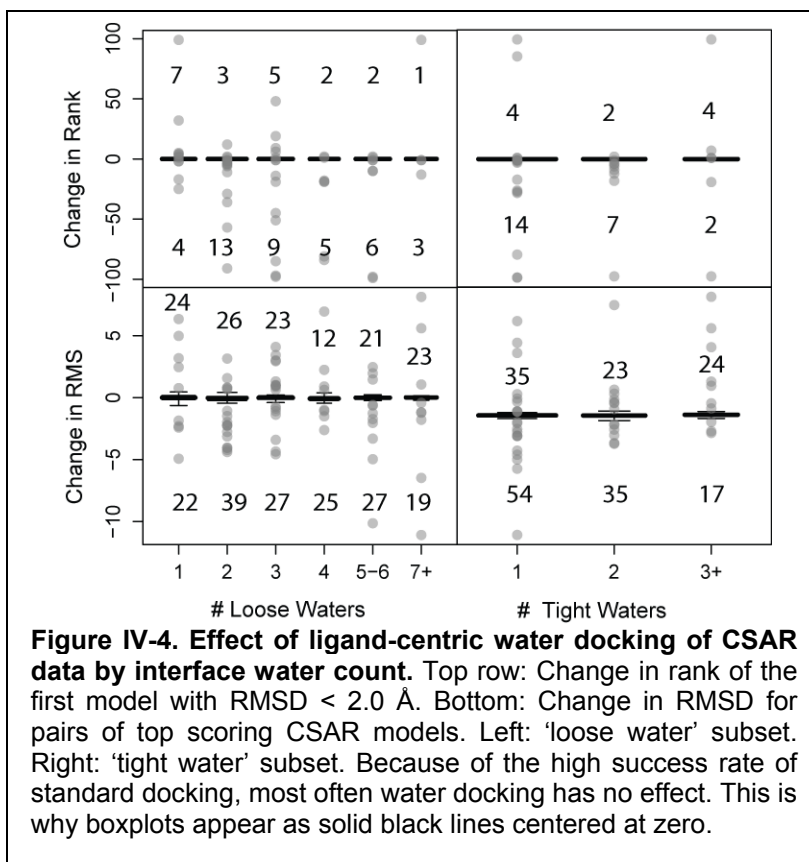
**Figure IV-3. CSAR inhibitor properties by number of interface waters.** The width of each bar indicates the number of CSAR datapoints the bar summarizes. Number of interface waters is indicated on the X-axis. The solid black line within the box represents the median. The top and bottom of the box represent the 25<sup>th</sup> and 75<sup>th</sup> percentile, the dotted lines extend to the min and max values. Outliers are plotted as black dots and calculated as values less than  $Q1 - 1.5 \times IQR$  or greater than  $Q3 + 1.5 \times IQR$ . On the Y-axis, various inhibitor properties are shown.

	<u>Min</u>	<u>Max</u>	<u>Mean</u>	<u>Std Dev</u>	<u>Median</u>
# of protein residues	119	2228	495	267	366
# of protein atoms	1756	32736	7664	4069	5661
# of inhibitor atoms	9	118	42	12.5	37
inhibitor molecular weight	59.1	779	332	108.6	304
# of inhibitor rotatable bonds	0	27	6	2.8	5
# of inhibitor H-bond acceptors	1	24	7	4.1	6
# of inhibitor H-bond donors	0	14	3	2.9	3
LogP	-44.9	9.2	-4.6	11.5	-1.0
Tight waters	0	8	1	1.7	1
Loose waters	0	19	3	2.1	3

***Ligand-centric water docking improves CSAR inhibitor placement.*** As described in the Methods section, crystallographic waters within 3.0 Å of an inhibitor atom and a protein atom were included in ligand-centric docking of CSAR data. These waters initially moved with the ligand and are subsequently allowed to translate and rotate independent of the ligand. Table IV-5 shows average scores and RMSD of top scoring Rosetta models with and without water docking. With both tight and loose water subsets, Rosetta energy scores decrease when water is docked. No significant change is seen in average inhibitor placement accuracy (RMSD) when water is

<u>Waters</u>	<u>Protocol</u>	<u>n</u>	<u>Total score</u>	<u>Inhibitor</u>	<u>RMSD</u>	<u>Water</u>	<u>W RMSD</u>
Tight	Standard dock	194	-1192±954	-17.93±6.5	1.06±1.79		
	Water dock	194	-1197±953	-20.80±7.4	1.18±2.26	-3.56±2.29	1.48±1.48
	Water – Standard	194	-4.6±14.8	-2.87±2.39	0.12±1.97		
	Per water effect	194	-1.67±8.13	-1.61±1.28	-0.01±1.47	-2.49±2.48	0.98±1.12
Loose	Standard Dock	299	-1184±968	-17.28±6.3	1.24±1.86		
	Water dock	299	-1193±968	-21.11±7.6	1.09±1.80	-3.20±1.86	1.60±1.38
	Water – Standard	299	-8.8±16.4	-3.83±3.36	-0.15±1.69		
	Per water effect	299	-1.86±5.71	-1.04±0.92	-0.04±0.78	-1.32±1.29	0.65±0.84

also docked. However, counts of the number of improved RMSDs and worsened RMSDs (Bottom rows of Figures IV-5 and IV-6) demonstrate that ligand-centric water docking is more likely to improve inhibitor placement than to make it worse. Ratios of improved to worsened RMSDs for the tight and loose subsets are 106:82 and 159:129 respectively.



**Ligand-centric water docking improves CSAR inhibitor ranking.** The rank of the first Rosetta model (by interface score) under 2 Å RMSD is a common measure of prediction quality.

**Table IV-6.: Ranking metrics for CSAR docking studies.** Rows 3 and 7 represent the difference between standard docking and ligand-centric water docking. Since high baseline success rates limit room for improvement, the % of possible improvement that was achieved (Water-Standard / N-standard dock) is shown in rows 4 & 8.

Waters	Protocol	N	Top model is < 2 Å RMSD		A < 2 Å RMSD model exists among top 10 models	
			#	%	#	%
Tight	Standard dock	194	164	84.5	176	90.7
	Water dock	194	167	86.1	186	95.9
	Water – Standard	194	3	1.6	10	5.2
	<b>% Improvement</b>					<b>56</b>
Loose	Standard Dock	299	237	79.3	264	88.3
	Water dock	299	249	83.3	278	93.0
	Water – Standard	299	12	4	14	4.7
	<b>% Improvement</b>					<b>40</b>

Figure IV-4 (top row) demonstrates that ligand-centric water docking of tight waters improves ranks 24 times for every 9 times it makes them worse. Loose water docking improves ranks twice as often as it worsens them. These results are reiterated in Figure IV-5, where ranks with and without water docking are plotted.

*Ligand-centric water docking improves CSAR docking “success rates”.*

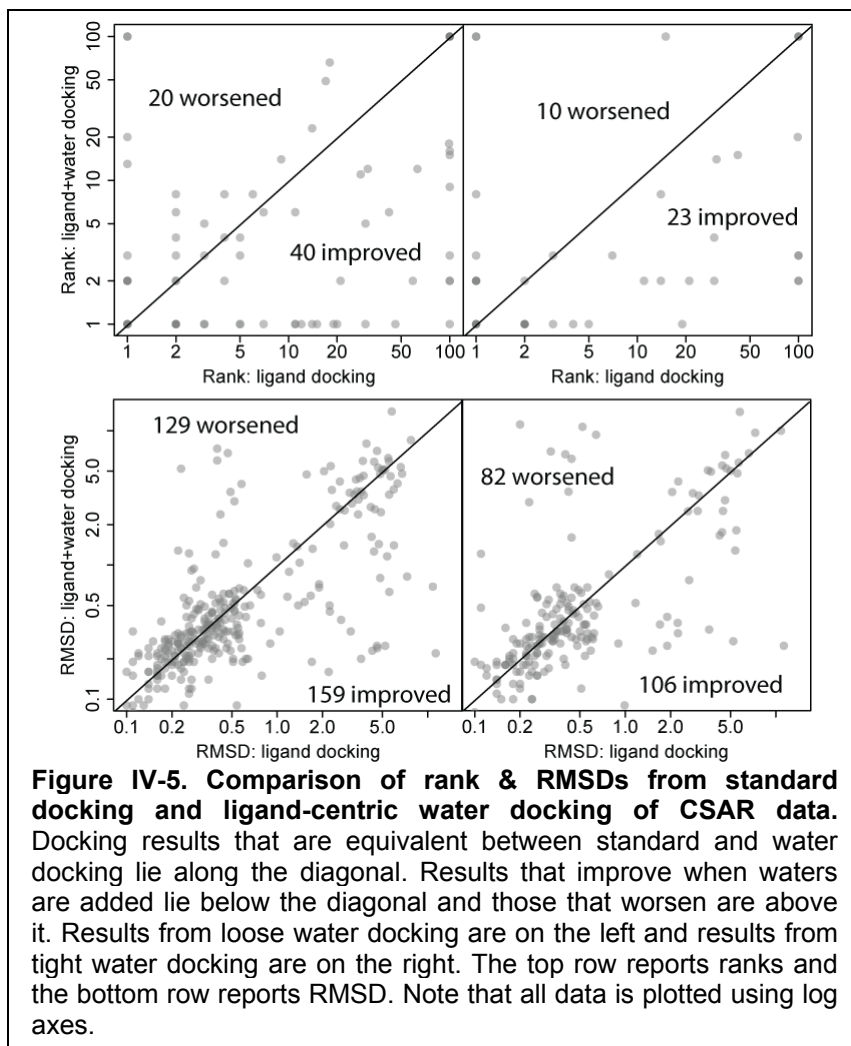
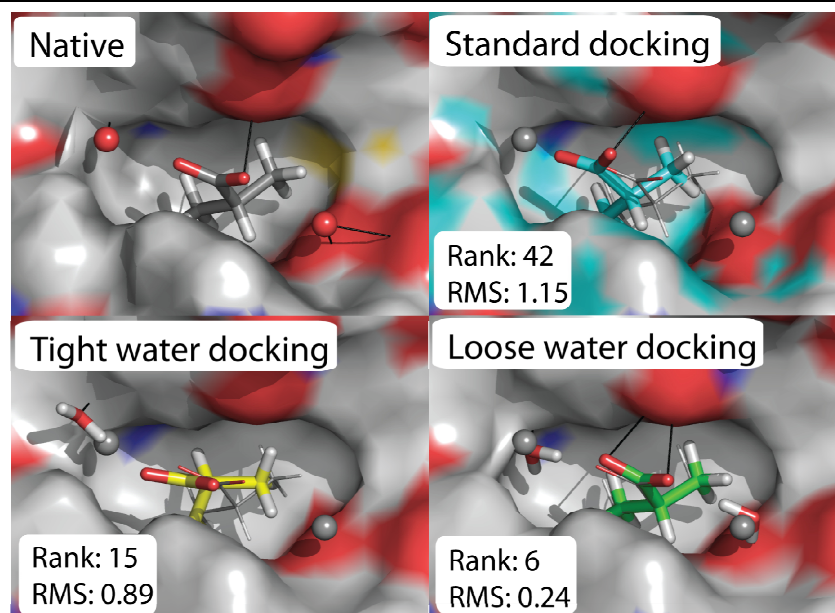


Table IV-6 reports on two metrics of successful docking: (1) whether the binding pose of the top scoring model is within 2.0 Å RMSD from the ligand coordinates reported in the CRYSTAL structure, and (2) whether there is a model with accuracy under 2.0 Å RMSD among the top ten models by score. For both the tight water subset and the loose water subset, we find ligand-centric water docking increases inhibitor docking success rates. Since standard docking is already quite successful we calculate the % of possible improvement that is achieved by water docking (Table IV-6, rows 4 and 8). This percentage reveals that ligand-centric water docking is



**Figure IV-6. Example CSAR docking result where ligand-centric waters improve inhibitor placement.** Docking results for CSAR complex 'set1\_120'. Top left: Experimental structure of PDB: 1IUP, coded as CSAR datapoint 'set1\_120'. Waters (Oxygen only) are shown as red spheres. Black lines represent polar contacts predicted by PyMOL. Top right and bottom row: native inhibitor (lines) and waters (spheres) are shown in grey for comparison. Docked waters are shown as sticks (note that Rosetta adds hydrogens). Docked inhibitors are shown in cyan, yellow, and green. For each study the models were sorted by total score, then interface energy. The first model with RMSD < 2.0 Å is depicted. Its position in the sorted list (rank) and its RMSD to native are shown.

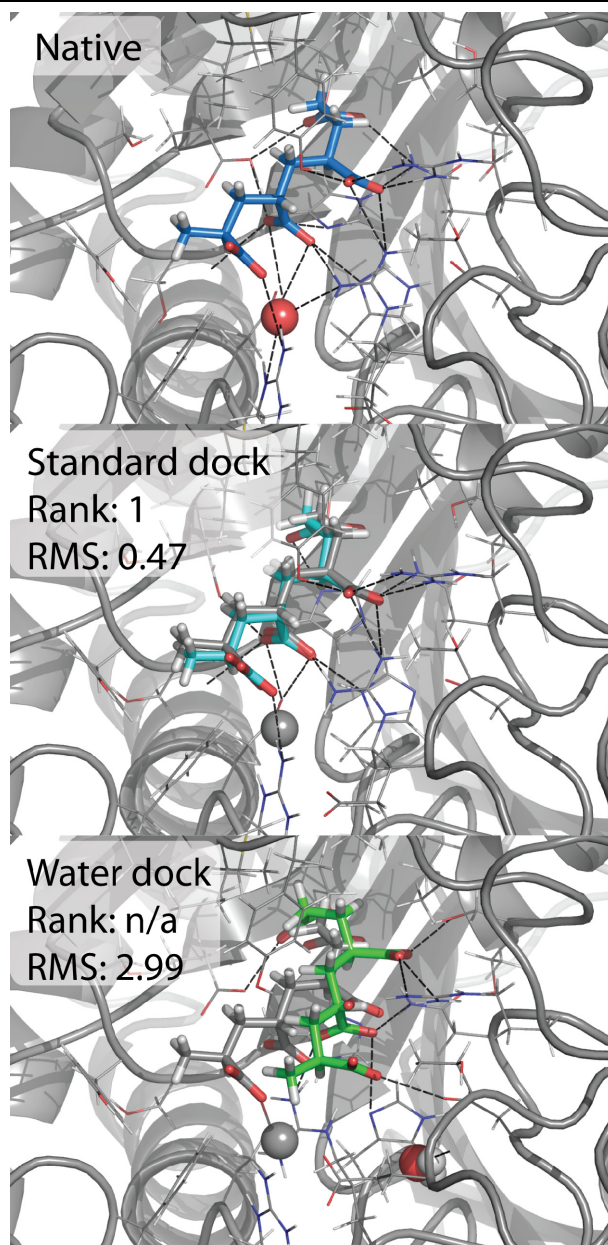
highly effective at recovering failed docking studies (56% recovery with tight waters and 40% recovery with loose waters).

*Analysis of successes and failures in CSAR water docking.* Figure IV-6 illustrates a case in which ligand-centric waters (which move with the ligand during ligand translation and rotation) restrict sampling of incorrect ligand binding poses and lead to improved ranking. In this case, water reduces the availability of reasonable, non-clashing poses, thus increasing the likelihood of finding the correct pose. In contrast, Figure IV-7 shows a case where standard docking succeeds and water docking fails. In this case the native structure contains 22 PyMOL (Schrodinger 2010) predicted polar contacts. The complexity of the hydrogen bonding network makes side chain rotamer packing especially reliant on the initial positions of inhibitor and

water. Since water docking adds 6 additional degrees of freedom to low resolution placement, additional low resolution sampling may be necessary to correctly place both water and inhibitor.

In Figure IV-8 we plot change in rank as a function of the crowdedness of the binding interface. For tight and loose water subsets we find a correlation of 0.50. This suggests that water docking is more likely to improve ranks in spacious binding pockets. However crowded interfaces require a combined accuracy of water and inhibitor placement that may cause water docking to hinder inhibitor placement.

Thus water docking

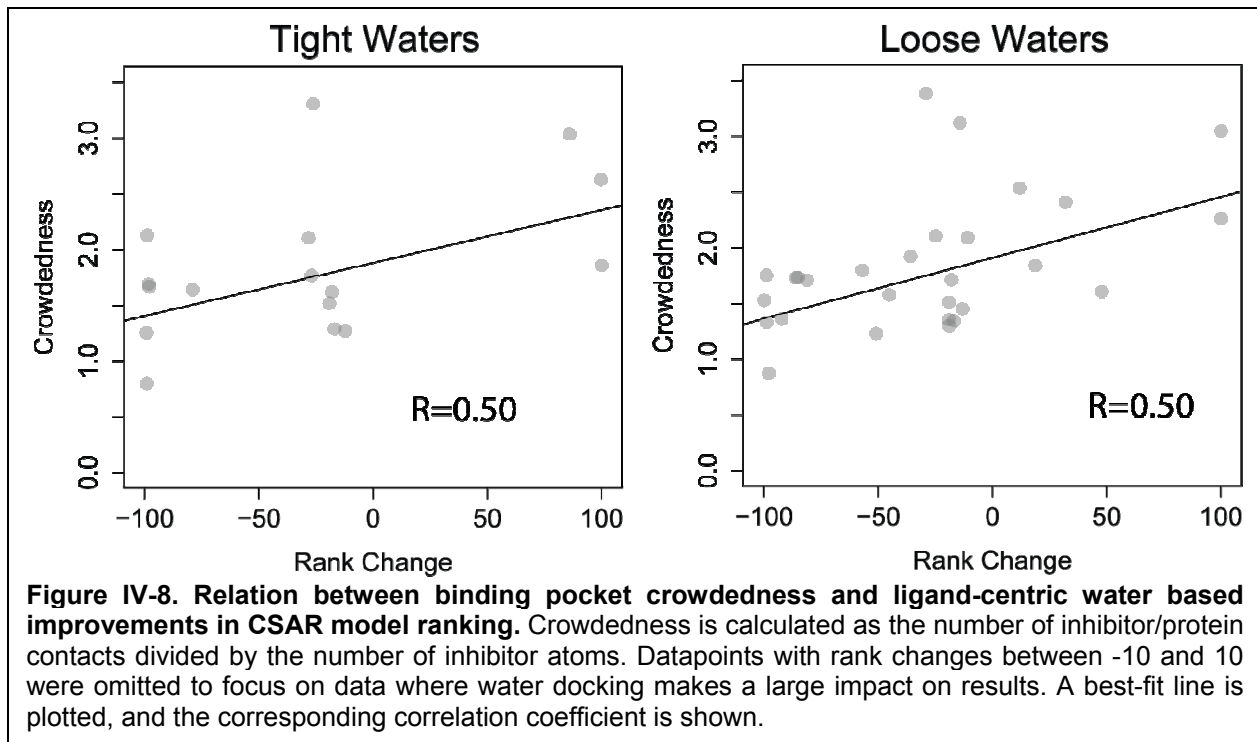


**Figure IV-7. Example CSAR docking result where ligand-centric waters worsen inhibitor placement.** Docking results for CSAR complex set1\_181. Top: experimental structure with inhibitor in blue, water as a red sphere, and polar contacts as black dashed lines. 22 polar contacts are predicted by PyMOL, 4 of which contact the water molecule. Middle: Top scoring model from docking without water. Native inhibitor and water in grey, Rosetta model in cyan. PyMOL predicts 16 polar contacts. Bottom: Lowest RMSD model from docking with loose waters. Rosetta model shown in green. No model within the top 100 by total energy score has RMSD < 2.0 Å (hence rank is 'n/a'). Shown is the lowest RMSD structure. PyMOL predicts 11 polar contacts (1 with water).



presents a trade-off between reducing the reasonable pose sampling space, while at the same time increasing the degrees of freedom in the docking study. Throughout this paper we produce 1000 models per input, regardless of how many waters are present. Scaling low resolution sampling to account for the number of waters being modeled is expected to be a more successful approach.

*Docking of CSAR interface waters does not improve binding affinity predictions.* R values between experimental and predicted binding affinity for the ‘tight’ subset are 0.54 (standard dock) and 0.51 (water dock). For the ‘loose’ subset these values are 0.54 (standard dock) and 0.46 (water dock). Thus while RMSD and rank metrics improve, binding affinity does not. This may be due to the fact that Rosetta score terms weights have already been adjusted to account for the effects of water. For instance hydrogen bond weights have been optimized to



account for the change in energy compared to hydrogen bonds formed with water (Rarey, Kramer et al. 1999). Similarly the “hydrophobic” score terms are used to represent desolvation of the protein receptor.

A future direction includes re-optimizing the Rosetta score function to appropriately evaluate the effects of explicit water on free energy. We suggest that successes in RMSD and rank metrics are gained mainly because of improved sampling, rather than improved scoring. Water most likely plays an indirect role in improving Rosetta scoring (see Table IV-5), by leading Rosetta to more accurate ligand placement and subsequent selection of side chain conformations.

*Docking with ligand conformers is improved by a water placement algorithm.* We filtered CSAR data to find complexes with one tight interface water – 92 complexes met this criterion. Up to ten conformations were generated for each ligand. On average the number of conformations generated was 9. Standard self-docking (Davis and Baker 2009) with ligand conformers was performed on each of these 92 complexes. Ligand conformers are randomly chosen during low resolution docking and randomly sampled during high resolution docking as well. 1000 models were generated for each 92 inputs. In 68 out of 92 cases, Rosetta successfully positions top scoring ligand poses within 2.0 Å RMSD from the crystallographic position.

The remaining 24 failed studies each contained models under 2.0 Å RMSD, but these models were not top scoring models. Less accurate models scored better in these 24 cases. In order to recover failed docking studies, we first cluster Rosetta models by inhibitor RMSD. We used the Rosetta Cluster application to group each set of 1000 Rosetta models by ligand pose similarity. For each of the 24 studies we selected the top model by interface score from each of

the top 5 lowest energy clusters for further analysis. In 11 of these 24 sets of 5 models, exists a model under 2.0 Å RMSD.

We sought to determine whether adding water molecules could decipher the correct binding pose from these 11 failed models. For each of the 5 models in each of the 11 aforementioned studies, waters were placed around each hydrogen bond donor and acceptor atom (see methods) of each ligand conformer. The ligand was docked with each of its waters (100 models per water) using an algorithm with only very small movements of inhibitor and water. The result is that in 7 of these 11 failed cases, the top scoring model is now better than 2 Å RMSD. Thus in 7 out of 11 cases where it was possible for water to help Rosetta differentiate the correct pose from several top scoring poses, water did so.

This last study is very much a preliminary result. Other studies in this paper each begin with prior knowledge about the locations of waters relative to the protein and/or the ligand. This study attempts to ‘brute force’ the identification of good water positions by trying all positions around polar inhibitor atoms. Future work will focus on the development of an algorithm to intelligently predict water molecule locations.

In conclusion, where comparative models or experimental data sheds light on the rough position of interface waters relative to a protein or a ligand, including those waters in Rosetta docking studies can significantly improve prediction results. Using Rosetta to predict the presence and position of water without prior knowledge has potential. However the extensive sampling required relative to the mixed results demonstrates that a more sophisticated placement algorithm is needed.

## References

- Chemical Computing Group Inc. (2011). "Molecular Operating Environment (MOE)."
- Adamson, C. S. and E. O. Freed (2007). "Human immunodeficiency virus type 1 assembly, release, and maturation." Advances in Pharmacology **55**: 347-387.
- Bohm, H. J. (1994). "The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure." Journal of Computer Aided Molecular Design **8**(3): 243-256.
- Cameron, I. L., N. J. Short, et al. (2007). "Verification of simple hydration/dehydration methods to characterize multiple water compartments on Tendon Type 1 Collagen." Cell Biology International **31**(6): 531-539.
- Das, R., B. Qian, et al. (2007). "Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home." Proteins **69 Suppl 8**: 118-128.
- Davis, I. W. and D. Baker (2009). "RosettaLigand docking with full ligand and receptor flexibility." Journal of Molecular Biology **385**(2): 381-392.
- de Beer, S. B., N. P. Vermeulen, et al. (2010). "The role of water molecules in computational drug design." Current Topics in Medicinal Chemistry **10**(1): 55-66.
- de Graaf, C., P. Pospisil, et al. (2005). "Binding mode prediction of cytochrome p450 and thymidine kinase protein-ligand complexes by consideration of water and rescoring in automated docking." Journal of Medicinal Chemistry **48**(7): 2308-2318.
- Deng, Y. and B. Roux (2008). "Computation of binding free energy with molecular dynamics and grand canonical Monte Carlo simulations." Journal of Chemical Physics **128**(11): 115103.
- Dunbar, J. B., Jr., R. D. Smith, et al. (2011). "CSAR benchmark exercise of 2010: selection of the protein-ligand complexes." Journal of Chemical Information and Modeling **51**(9): 2036-2046.
- Dunbrack, R. L., Jr. and M. Karplus (1993). "Backbone-dependent rotamer library for proteins. Application to side-chain prediction." Journal of Molecular Biology **230**(2): 543-574.
- Halperin, I., B. Y. Ma, et al. (2002). "Principles of docking: An overview of search algorithms and a guide to scoring functions." Proteins-Structure Function and Genetics **47**(4): 409-443.

- Hornak, V., A. Okur, et al. (2006). "HIV-1 protease flaps spontaneously close to the correct structure in simulations following manual placement of an inhibitor into the open state." Journal of the American Chemical Society **128**(9): 2812-2813.
- Huang, S. Y. and X. Zou (2010). "Advances and challenges in protein-ligand docking." International Journal of Molecular Sciences **11**(8): 3016-3034.
- Ikura, T., Y. Urakubo, et al. (2004). "Water-mediated interaction at a protein-protein interface." Chemical Physics **307**(2-3): 111-119.
- Jain, A. N. (1996). "Scoring noncovalent protein-ligand interactions: a continuous differentiable function tuned to compute binding affinities." Journal of Computer Aided Molecular Design **10**(5): 427-440.
- Kim, R. and J. Skolnick (2008). "Assessment of programs for ligand binding affinity prediction." Journal of Computational Chemistry **29**(8): 1316-1331.
- Kortemme, T., A. V. Morozov, et al. (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.
- Kramer, B., M. Rarey, et al. (1999). "Evaluation of the FLEXX incremental construction algorithm for protein-ligand docking." Proteins-Structure Function and Genetics **37**(2): 228-241.
- Lazaridis, T. and M. Karplus (1999). "Effective energy function for proteins in solution." Proteins **35**(2): 133-152.
- Lemmon, G. and J. Meiler (2012). "Rosetta Ligand docking with flexible XML protocols." Methods in Molecular Biology **819**: 143-155.
- Lennard-Jones, J. E. (1924). "On the Determination of Molecular Fields. II. From the Equation of State of a Gas." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **106**: 463-477.
- Lie, M. A., R. Thomsen, et al. (2011). "Molecular docking with ligand attached water molecules." Journal of Chemical Information and Modeling **51**(4): 909-917.
- Louis, J. M., R. Ishima, et al. (2007). "HIV-1 protease: structure, dynamics, and inhibition." Advances in Pharmacology **55**: 261-298.
- Meiler, J. and D. Baker (2006). "ROSETTALIGAND: protein-small molecule docking with full side-chain flexibility." Proteins **65**(3): 538-548.

- Ni, H., C. A. Sotriffer, et al. (2001). "Ordered water and ligand mobility in the HIV-1 integrase-5CITEP complex: a molecular dynamics study." Journal of Medicinal Chemistry **44**(19): 3043-3047.
- Osterberg, F., G. M. Morris, et al. (2002). "Automated docking to multiple target structures: incorporation of protein mobility and structural water heterogeneity in AutoDock." Proteins **46**(1): 34-40.
- Panigrahi, S. K. and G. R. Desiraju (2007). "Strong and weak hydrogen bonds in the protein-ligand interface." Proteins **67**(1): 128-141.
- Poornima, C. S. and P. M. Dean (1995). "Hydration in drug design. 1. Multiple hydrogen-bonding features of water molecules in mediating protein-ligand interactions." Journal of Computer Aided Molecular Design **9**(6): 500-512.
- Rarey, M., B. Kramer, et al. (1999). "The particle concept: placing discrete water molecules during protein-ligand docking predictions." Proteins **34**(1): 17-28.
- Roberts, B. C. and R. L. Mancera (2008). "Ligand-protein docking with water molecules." Journal of Chemical Information and Modeling **48**(2): 397-408.
- Sarkhel, S. and G. R. Desiraju (2004). "N-H...O, O-H...O, and C-H...O hydrogen bonds in protein-ligand complexes: strong and weak interactions in molecular recognition." Proteins **54**(2): 247-259.
- Schnecke, V. and L. A. Kuhn (2000). "Virtual screening with solvation and ligand-induced complementarity." Perspectives in Drug Discovery and Design **20**(1): 171-190.
- Schneider, G. and U. Fechner (2005). "Computer-based de novo design of drug-like molecules." Nature Reviews Drug Discovery **4**(8): 649-663.
- Schrodinger, LLC (2010). The PyMOL Molecular Graphics System, Version 1.3r1.
- Simons, K. T., I. Ruczinski, et al. (1999). "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins." Proteins **34**(1): 82-95.
- Sousa, S. F., P. A. Fernandes, et al. (2006). "Protein-ligand docking: current status and future challenges." Proteins **65**(1): 15-26.
- Thilagavathi, R. and R. L. Mancera (2010). "Ligand-protein cross-docking with water molecules." Journal of Chemical Information and Modeling **50**(3): 415-421.
- Verdonk, M. L., G. Chessari, et al. (2005). "Modeling water molecules in protein-ligand docking using GOLD." Journal of Medicinal Chemistry **48**(20): 6504-6515.

- Verdonk, M. L., G. Chessari, et al. (2005). "Modeling water molecules in protein-ligand docking using GOLD." Journal of Medicinal Chemistry **48**(20): 6504-6515.
- Wlodawer, A. and J. Vondrasek (1998). "Inhibitors of HIV-1 protease: a major success of structure-assisted drug design." Annual Review of Biophysics & Biomolecular Structure **27**: 249-284.
- Young, T., R. Abel, et al. (2007). "Motifs for molecular recognition exploiting hydrophobic enclosure in protein-ligand binding." Proceedings of the National Academy of Sciences of the United States of America **104**(3): 808-813.

## CHAPTER 5

### CONCLUSION & FUTURE DIRECTIONS

In this dissertation we have presented several advances in small molecule docking using the RosettaLigand software package. In chapter 2 we show how RosettaLigand binding affinity predictions can be improved. These improvements are particularly helpful in the case of flexible proteins that become rigid upon inhibitor binding. In chapter 3 we present a user's guide to the new XML-script interface to RosettaLigand. This new interface allows the user to fully customize ligand docking, effectively creating unique ligand docking protocols on-the-fly. Along with this new interface, RosettaLigand has many increased capabilities, including simultaneous docking of multiple ligands, protein interface design during ligand docking, and docking of ligands composed of multiple 'residues'. In chapter 4 we demonstrate the usefulness of simultaneously docking small molecules and waters in protein binding sites. The docking of waters is shown to improve the placement and ranking of inhibitors.

#### **RosettaLigand binding affinity prediction**

RosettaLigand is now capable of accurate predictions of HIV-1 protease/protease inhibitor binding affinity ( $\Delta\Delta G$ ). These predictions will allow chemists to evaluate new protease inhibitor prototypes before synthesis and experimental validation. First however, our method should be validated using dataset containing inhibitors not present in our training data. Improved predictions of relative binding affinity ( $\Delta\Delta\Delta G$ ) can be used to understand mechanisms of PI drug resistance. During inhibitor design, RosettaLigand can predict the effect of common drug resistance mutations on the efficacy of the new inhibitor. A future direction we envisioned is



development of webservers that (1) allow investigators to upload HIV-1 PR sequence and inhibitor structure and return predicted binding affinity, and (2) predict the effect of mutation on binding affinity.

We also propose that HIV-1 genotype data could be used to make decisions about which PI to prescribe. Our attempts to correlate binding affinity with viral load failed. A future direction would be to compare Rosetta binding affinity predictions with the statistical models currently used in clinical PI decision making. These models are based on observed genotype/phenotype (genotype compared with in vitro viral replication) or genotype/patient outcome data. If these results are favorable, structural data could guide PI choice. This would mark the beginnings of *structure-based personalized medicine*.

## **Advances to RosettaLigand software**

The RosettaLigand protocol was rewritten as a customizable XML script. The new code allows multiple ligand docking as well as protein interface design. The XML framework allows the user to seamlessly combine ligand docking with homology modeling, loop building, protein-protein docking. Multiple simultaneous ligand docking is a novel feature that has hitherto not been possible in ligand docking software. This allows for including explicit interface waters, metal ions, and cofactors in docking simulations. In the case of enzymes that break or form chemical bonds, Rosetta can now model the 2 or 3 member complex before and after catalysis.

RosettaLigand docking with interface design allows an investigator to design a protein that binds to a small molecule without knowing a priori the exact binding pose the small molecule will assume. Proteins designed to bind small molecules represent a class of protein therapeutics that have been used in oncology, treatment of arthritis, and selective drug delivery

(Leader, Baca et al. 2008). RosettaLigand's newfound abilities in the area of interface design have the potential to expedite the development of new protein therapeutics.

My PhD research focused on improving RosettaLigand sampling. Future work must focus on improving RosettaLigand scoring. This will entail implementing Rosetta atom types that are unambiguously derived from orbital assignments. Current Rosetta atom types derive from amino acid definitions, such as 'c-alpha' and 'N-tryptophan'. Special transition state orbital-based atom types could be coupled with Rosetta docking with interface design thereby allowing improvements in design of enzymes that stabilize the transition state (Richter, Leaver-Fay et al. 2011).

## **Modeling of explicit waters within RosettaLigand**

We demonstrate multiple ligand docking via explicit placement of interface waters. RosettaLigand was challenged to simultaneously determine inhibitor pose and position and orientation of each interface water molecule. Water docking led to significant improvements in ranking of docked models. Similar to what others have shown, we find that this effect is due to improved sampling rather than improved scoring (Kramer, Rarey et al. 1999). However our results were shaded by the fact that water placement was restricted to a 4 Å (HIV-1 protease dataset) or 1 Å (CSAR dataset) radius sphere around the crystallographic water coordinate, and Rosetta was not given the choice of whether or not to include the waters.

Future work involves developing a more sophisticated algorithm for determining the presence and positions of interface waters without prior knowledge. Such an algorithm should consider the entropic cost of displacing waters present before ligand docking (Verdonk, Chessari et al. 2005). Because interface waters led to improvements in rank, we suggest in silico high

throughput screening as an area that could benefit substantially from water placement. This is because in silico high throughput screening relies on the software's ability to correctly rank screened compounds.

## References

- Kramer, B., M. Rarey, et al. (1999). "Evaluation of the FLEXX incremental construction algorithm for protein-ligand docking." Proteins-Structure Function and Genetics **37**(2): 228-241.
- Leader, B., Q. J. Baca, et al. (2008). "Protein therapeutics: a summary and pharmacological classification." Nature Reviews Drug Discovery **7**(1): 21-39.
- Richter, F., A. Leaver-Fay, et al. (2011). "De Novo Enzyme Design Using Rosetta3." PLoS One **6**(5).
- Verdonk, M. L., G. Chessari, et al. (2005). "Modeling water molecules in protein-ligand docking using GOLD." Journal of Medicinal Chemistry **48**(20): 6504-6515.

## APPENDICES

### A. Experimental $\Delta\Delta G$ & $K_i$ values for HIV-1 PR/PI binding.

Each datapoint measures binding affinity between a particular PR mutant and a particular PI. 11 unique PIs and 34 unique sequences were used. All values are reported in J/mol.  $K_i$  measurements were converted to  $\Delta\Delta G$  (original values are reported in brackets). Sequences are numbered 1-34 (Figure 4).

Sequence	Acetylpepstatin	AG1776	Amprenavir	Indinavir	KNI-272	KNI-764	Lopinavir	Nelfinavir	Ritonavir	Saquinavir	Ethylenamine- Peptidomimetic- Inhibitor
1		0.11 (-56.8)		5.10 (-47.3)	1.70 (-50.1)			6.00 (-46.9)	4.50 (-47.6)		
2		0.02 (-60.9)									
3		0.02 (-60.9)		2.80 (-48.8)	0.37 (-53.8)			0.22 (-55.1)	0.50 (-53.1)		
4				-49.7				-51.4	-50.1	-50.1	
4				4.40 (-47.7)				1.90 (-49.8)	0.91 (-51.6)	2.00 (-49.7)	
5		0.11 (-56.8)		2.70 (-48.9)	0.60 (-52.6)			0.90 (-51.6)	0.15 (-56.1)		
6				-32.6				-34.6	-32.1	-32.1	
6				1102 (-34.0)				722 (-35.1)	2107 (-32.4)	1948 (-32.6)	
7	-31.7		-48.4	-43.4	-45.5	-57.2	-53.5	-45.9	-44.7	-41.8	
8				0.11 (-56.8)							

9			-48.4		-61.4		-52.2	-55.1	-51	
10	-36.7		-53.5	-51.0	-54.3	-60.6	-62.2	-51.8	-53.5	-49.7
11			-50.5		-62.2		-52.6	-56.0	-51.8	
12	-36.3		-55.1	-51.8	-55.5	-59.7	-63.1	-53.5	-57.2	-54.3
12			-49.3				-47.6		-49.3	
13		0.32 (-54.2)	7.60 (-46.3)	1.70 (-50.1)			26 (-43.3)	13.0 (-45.0)		
14			-38.4		-53.0		-44.3	-40.5	-43.4	
15	-33.4		-51	-45.1	-48.4	-57.6	-58.1	-51.8	-49.7	-49.3
16			0.05 (-58.7)							
17			-38.4				-40.5		-43.4	
18			-39.7		-53.9		-45.1	-40.9	-44.3	
19			3.11 (-48.6)							
19			6.62 (-46.7)							
20			-35.7				-35.9	-36.7	-36.3	
20			481 (-36.1)				445 (-36.3)	340 (-36.9)	755 (-34.9)	
21			2.81 (-48.8)							
22										1.50 (-50.4)
23		0.03 (-60.1)	2.00 (-49.7)	0.38 (-53.8)			0.63 (-52.5)	1.10 (-51.1)		
24			32 (-54.2)							
25										4.10 (-47.9)
26			-53.0		-62.2		-54.7	-60.1	-53.5	
27	-34.2		-51.8	-49.3	-53.9	-60.6	-62.2	-50.5	-53.9	-50.1
28			0.36 (-53.9)	0.01 (-66.9)			0.25 (-54.8)	0.03 (-60.3)	0.46 (-53.3)	
28			0.54 (-52.9)	0.58 (-52.7)			1.20 (-50.9)	0.70 (-52.3)	0.90 (-53.7)	
28			3.10 (-48.6)	0.74 (-54.2)						
29			2.64 (-49.0)							
30			0.59 (-52.7)							
31		0.08 (-57.5)	4.90 (-47.4)	0.90 (-51.6)			6.80 (-46.6)	0.23 (-55.0)		

32		0.02 (-60.9)		5.10 (-47.3)	1.10 (-51.1)			6.10 (-46.9)	2.40 (-49.2)		
33				-42.6		-54.3		-47.2	-45.5	-45.9	
34	-28.8		-44.3	-38.8	-38	-52.2	-47.2	-40.1	-38.8	-35.5	

## B. Template structures used for comparative modeling.

Each of 34 sequences found in binding energy data was threaded onto these 171 template structures. PDB codes, resolution, and citations for each structure in this table were extracted directly from PDB files.

PDB Code	Resolution (angstroms)	Citation
1A30	2.00	JRNL REF BIOCHEMISTRY V. 37 2105 1998
1A8K	2.00	JRNL REF EUR.J.BIOCHEM. V. 249 523 1997
1A94	2.00	JRNL REF BIOCHEMISTRY V. 37 4518 1998
1AJV	2.00	JRNL REF J.MED.CHEM. V. 40 898 1997
1AJX	2.00	JRNL REF J.MED.CHEM. V. 40 898 1997
1AXA	2.00	JRNL REF PROTEIN SCI. V. 7 300 1998
1BV7	2.00	JRNL REF BIOCHEMISTRY V. 37 15042 1998
1BV9	2.00	JRNL REF BIOCHEMISTRY V. 37 15042 1998
1BWA	1.90	JRNL REF BIOCHEMISTRY V. 37 15042 1998
1BWB	1.80	JRNL REF BIOCHEMISTRY V. 37 15042 1998
1D4H	1.81	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1D4I	1.81	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1D4J	1.81	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1D4Y	1.97	JRNL REF J.MED.CHEM. V. 39 4349 1996
1DAZ	1.55	JRNL REF EUR.J.BIOCHEM. V. 263 238 1999
1DIF	1.70	JRNL REF J.MOL.BIOL. V. 255 321 1996
1DW6	1.88	JRNL REF EUR.J.BIOCHEM. V. 263 238 1999
1EBK	2.06	JRNL REF EUR.J.BIOCHEM. V. 263 238 1999
1EBW	1.81	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1EBY	2.29	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1EBZ	2.01	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1ECO	1.79	JRNL REF EUR.J.BIOCHEM. V. 271 4594 2004
1EC1	2.10	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1EC2	2.00	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1EC3	1.80	JRNL REF EUR.J.BIOCHEM. V. 270 1746 2003
1F7A	2.00	JRNL REF J.MOL.BIOL. V. 301 1207 2000
1FEJ	1.78	JRNL REF PROTEINS V. 43 455 2001

1FF0	1.85	JRNL REF PROTEINS V. 43 455 2001
1FFF	1.90	JRNL REF PROTEINS V. 43 455 2001
1FFI	1.70	JRNL REF PROTEINS V. 43 455 2001
1FG6	1.80	JRNL REF PROTEINS V. 43 455 2001
1FG8	1.85	JRNL REF PROTEINS V. 43 455 2001
1FGC	1.90	JRNL REF PROTEINS V. 43 455 2001
1G2K	1.95	JRNL REF J.MED.CHEM. V. 44 155 2001
1G35	1.80	JRNL REF J.MED.CHEM. V. 44 155 2001
1HIV	2.00	JRNL REF PROTEIN SCI. V. 1 1061 1992
1HPV	1.90	JRNL REF J.AM.CHEM.SOC. V. 117 1181 1995
1HPX	2.00	JRNL REF STRUCTURE V. 3 581 1995
1HSG	2.00	JRNL REF J.BIOL.CHEM. V. 269 26344 1994
1HSH	1.90	JRNL REF J.BIOL.CHEM. V. 269 26344 1994
1HTE	2.80	JRNL REF BIOCHEMISTRY V. 33 8417 1994
1HTG	2.00	JRNL REF BIOCHEMISTRY V. 33 8417 1994
1HVH	1.80	JRNL REF J.MED.CHEM. V. 41 1446 1998
1HVH	1.80	JRNL REF J.MED.CHEM. V. 41 1446 1998
1HVI	1.80	JRNL REF J.AM.CHEM.SOC. V. 116 847 1994
1HVJ	2.00	JRNL REF J.AM.CHEM.SOC. V. 116 847 1994
1HVK	1.80	JRNL REF J.AM.CHEM.SOC. V. 116 847 1994
1HVL	1.80	JRNL REF J.AM.CHEM.SOC. V. 116 847 1994
1HVR	1.80	JRNL REF SCIENCE V. 263 380 1994
1HWR	1.80	JRNL REF J.BIOL.CHEM. V. 273 12325 1998
1HXW	1.80	JRNL REF PROC.NATL.ACAD.SCI.USA V. 92 2484 1995
1IDA	1.70	JRNL REF STRUCTURE V. 3 33 1995
1IIQ	1.83	JRNL REF J.MED.CHEM. V. 45 1432 2002
1IZH	1.90	JRNL REF J.MOL.BIOL. V. 324 739 2002
1K2B	1.70	JRNL REF PROTEINS V. 48 107 2002
1K6V	2.00	JRNL REF PROTEIN SCI. V. 11 418 2002
1K6V	2.00	JRNL REF PROTEIN SCI. V. 11 418 2002
1KJ7	2.00	JRNL REF STRUCTURE V. 10 369 2002
1KJF	2.00	JRNL REF STRUCTURE V. 10 369 2002
1KJG	2.00	JRNL REF STRUCTURE V. 10 369 2002
1KJH	2.00	JRNL REF STRUCTURE V. 10 369 2002
1KZK	1.09	JRNL REF BIOCHEMISTRY V. 41 4582 2002
1LZQ	2.20	JRNL REF J.MED.CHEM. V. 46 1636 2003
1MER	1.90	JRNL REF BIOCHEMISTRY V. 36 1573 1997
1MES	1.90	JRNL REF BIOCHEMISTRY V. 36 1573 1997
1MET	1.90	JRNL REF BIOCHEMISTRY V. 36 1573 1997
1MEU	1.90	JRNL REF BIOCHEMISTRY V. 36 1573 1997



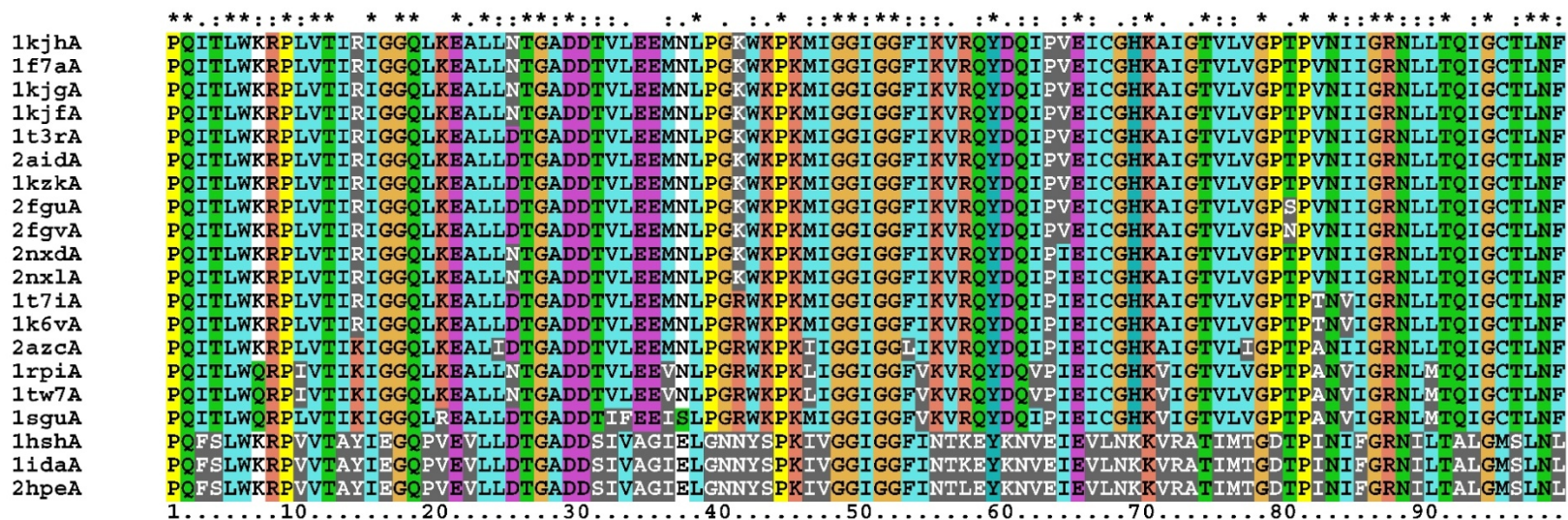
1MRW	2.00	JRNL REF PROTEINS V. 55 594 2004
1MRX	2.00	JRNL REF PROTEINS V. 55 594 2004
1MSM	2.00	JRNL REF PROTEINS V. 55 594 2004
1MSN	2.00	JRNL REF PROTEINS V. 55 594 2004
1MT7	1.90	JRNL REF J.VIROL. V. 77 1305 2003
1MT8	2.15	JRNL REF J.VIROL. V. 77 1305 2003
1MT9	2.00	JRNL REF J.VIROL. V. 77 1305 2003
1NH0	1.03	JRNL REF J.MED.CHEM. V. 47 2030 2004
1NPA	2.00	JRNL REF J.MED.CHEM. V. 40 2440 1997
1NPV	2.00	JRNL REF J.MED.CHEM. V. 46 1831 2003
1NPW	2.00	JRNL REF J.MED.CHEM. V. 46 1831 2003
1ODW	2.10	JRNL REF PROTEIN PEPT.LETT. V. 3 399 1996
1ODX	2.00	JRNL REF PROTEIN PEPT.LETT. V. 3 399 1996
1ODY	2.00	JRNL REF PROTEIN SCI. V. 7 2314 1998
1OHR	2.10	JRNL REF J.MED.CHEM. V. 40 3979 1997
1PRO	1.80	JRNL REF J.MED.CHEM. V. 39 392 1996
1QBR	1.80	JRNL REF J.MED.CHEM. V. 40 181 1997
1QBU	1.80	JRNL REF J.MED.CHEM. V. 40 181 1997
1RL8	2.00	JRNL REF TO BE PUBLISHED 2005
1RPI	1.86	JRNL REF J.VIROL. V. 78 3123 2004
1SGU	1.90	JRNL REF BIOCHEMISTRY V. 43 12141 2004
1SP5	1.80	JRNL REF TO BE PUBLISHED
1T3R	1.20	JRNL REF J.MED.CHEM. V. 48 1813 2005
1T7I	1.35	JRNL REF J.MED.CHEM. V. 48 1813 2005
1TSQ	2.00	JRNL REF J.VIROL. V. 78 12446 2004
1TW7	1.30	JRNL REF STRUCTURE V. 13 1887 2005
1W5V	1.80	JRNL REF EUR.J.BIOCHEM. V. 271 4594 2004
1W5W	1.80	JRNL REF EUR.J.BIOCHEM. V. 271 4594 2004
1W5X	1.90	JRNL REF EUR.J.BIOCHEM. V. 271 4594 2004
1W5Y	1.90	JRNL REF EUR.J.BIOCHEM. V. 271 4594 2004
1WBK	2.00	JRNL REF TO BE PUBLISHED
1WBM	2.00	JRNL REF TO BE PUBLISHED
1XL2	1.50	JRNL REF ANGEW.CHEM.INT.ED.ENGL. V. 44 3140 2005
1XL5	1.73	JRNL REF ANGEW.CHEM.INT.ED.ENGL. V. 44 3140 2005
1ZPA	2.02	JRNL REF CHEMBIOCHEM V. 6 1167 2005
2A1E	1.30	JRNL REF CHEMMEDCHEM V. 1 186 2006
2A4F	1.90	JRNL REF BIOORG.MED.CHEM.LETT. V. 15 5499 2005
2AID	1.90	JRNL REF J.BIOL.CHEM. V. 268 15343 1993
2AOC	1.30	JRNL REF FEBS J. V. 272 5265 2005
2AOD	1.40	JRNL REF FEBS J. V. 272 5265 2005

2AOE	1.54	JRNL REF FEBS J. V. 272 5265 2005
2AOH	1.42	JRNL REF FEBS J. V. 272 5265 2005
2AQU	2.00	JRNL REF BIOCHEMISTRY V. 45 5468 2006
2AVM	1.10	JRNL REF J.MOL.BIOL. V. 354 789 2005
2AVO	1.10	JRNL REF J.MOL.BIOL. V. 354 789 2005
2AVO	1.10	JRNL REF J.MOL.BIOL. V. 354 789 2005
2AVQ	1.30	JRNL REF J.MOL.BIOL. V. 354 789 2005
2AVS	1.10	JRNL REF J.MOL.BIOL. V. 354 789 2005
2AVV	1.50	JRNL REF J.MOL.BIOL. V. 354 789 2005
2AZC	2.01	JRNL REF J.MOL.BIOL. V. 356 967 2006
2BB9	1.35	JRNL REF BIOORG.MED.CHEM.LETT. V. 16 859 2006
2BBB	1.70	JRNL REF BIOORG.MED.CHEM.LETT. V. 16 859 2006
2BPV	1.90	JRNL REF ACTA CRYSTALLOGR.,SECT.D V. 54 1053 1998
2BPY	1.90	JRNL REF ACTA CRYSTALLOGR.,SECT.D V. 54 1053 1998
2CEM	1.80	JRNL REF J.MED.CHEM. V. 49 1828 2006
2CEN	1.70	JRNL REF J.MED.CHEM. V. 49 1828 2006
2F3K	1.60	JRNL REF ANTIMICROB.AGENTS CHEMOTHER. V. 50 1518 2006
2F3K	1.60	JRNL REF ANTIMICROB.AGENTS CHEMOTHER. V. 50 1518 2006
2F80	1.45	JRNL REF J.MED.CHEM. V. 49 1379 2006
2F80	1.45	JRNL REF J.MED.CHEM. V. 49 1379 2006
2FGU	2.00	JRNL REF J.VIROL. V. 80 6906 2006
2FGV	1.50	JRNL REF J.VIROL. V. 80 6906 2006
2FLE	1.90	JRNL REF TO BE PUBLISHED
2FNS	1.85	JRNL REF J.VIROL. V. 80 3607 2006
2FNT	1.44	JRNL REF J.VIROL. V. 80 3607 2006
2FXD	1.60	JRNL REF TO BE PUBLISHED
2FXE	1.80	JRNL REF TO BE PUBLISHED
2G69	1.35	JRNL REF J.MOL.BIOL. V. 358 1191 2006
2HB3	1.35	JRNL REF J.MED.CHEM. V. 49 5252 2006
2HB4	2.15	JRNL REF ACTA CRYSTALLOGR.,SECT.D V. 63 866 2007
2HPE	2.00	JRNL REF TO BE PUBLISHED
2HS2	1.22	JRNL REF J.MOL.BIOL. V. 363 161 2006
2I0A	1.80	JRNL REF J.MED.CHEM. V. 49 7342 2006
2I0D	1.95	JRNL REF J.MED.CHEM. V. 49 7342 2006
2I4D	1.50	JRNL REF J.MOL.BIOL. V. 363 635 2006
2I4U	1.50	JRNL REF J.MOL.BIOL. V. 363 635 2006
2I4V	1.50	JRNL REF J.MOL.BIOL. V. 363 635 2006
2I4W	1.55	JRNL REF J.MOL.BIOL. V. 363 635 2006
2I4X	1.55	JRNL REF J.MOL.BIOL. V. 363 635 2006
2IEO	1.53	JRNL REF J.MOL.BIOL. V. 338 341 2004

2JE4	1.07	JRNL REF J.AM.CHEM.SOC. V. 129 11480 2007
2NNK	1.25	JRNL REF PROTEINS V. 67 232 2007
2NPH	1.65	JRNL REF PROC.NATL.ACAD.SCI.USA V. 103 18464 2006
2NXD	2.00	JRNL REF PROTEINS 2007
2NXL	2.00	JRNL REF PROTEINS 2007
2O4K	1.60	JRNL REF J.VIROL. V. 81 5144 2007
2O4K	1.60	JRNL REF J.VIROL. V. 81 5144 2007
2O4L	1.33	JRNL REF J.VIROL. V. 81 5144 2007
2O4L	1.33	JRNL REF J.VIROL. V. 81 5144 2007
2O4N	2.00	JRNL REF J.VIROL. V. 81 5144 2007
2O4P	1.80	JRNL REF J.VIROL. V. 81 5144 2007
2O4S	1.54	JRNL REF J.VIROL. V. 81 5144 2007
2O4S	1.54	JRNL REF J.VIROL. V. 81 5144 2007
2PC0	1.40	JRNL REF ACTA CRYSTALLOGR.,SECT.D V. 63 866 2007
2PK6	1.45	JRNL REF CHEM.BIOL.DRUG.DES. V. 69 413 2007
2PSU	1.93	JRNL REF CHEM.BIOL.DRUG.DES. V. 69 298 2007
2PSV	1.75	JRNL REF CHEM.BIOL.DRUG.DES. V. 69 298 2007
2PSV	1.75	JRNL REF CHEM.BIOL.DRUG.DES. V. 69 298 2007
2Q3K	2.00	JRNL REF J.VIROL. 2007
2Q54	1.85	JRNL REF J.MED.CHEM. V. 50 4316 2007
2Q55	1.90	JRNL REF J.MED.CHEM. V. 50 4316 2007
2Q5K	1.95	JRNL REF J.MED.CHEM. V. 50 4316 2007
5HVP	2.00	JRNL REF J.BIOL.CHEM. V. 265 14209 1990
7UPJ	2.00	JRNL REF J.MED.CHEM. V. 40 1149 1997

## C. Sequence Alignment of 171 HIV-1 PR backbone templates

Multiple sequence alignment using ClustalX 2.1. 34 sequences were threaded onto each of 171 backbone templates. Shown are the sequences from the 171 backbone templates. Colors correspond to amino acid type as shown in the color key below. An asterisk ("\*") means that the residues or nucleotides in that column are identical in all sequences in the alignment. A colon (":") means that conserved substitutions have been observed. A period (".") means that semi-conserved substitutions are observed. Exceptional residues are colored gray









#### D. HIV-1 PR/PI data partitioned by location of mutations.

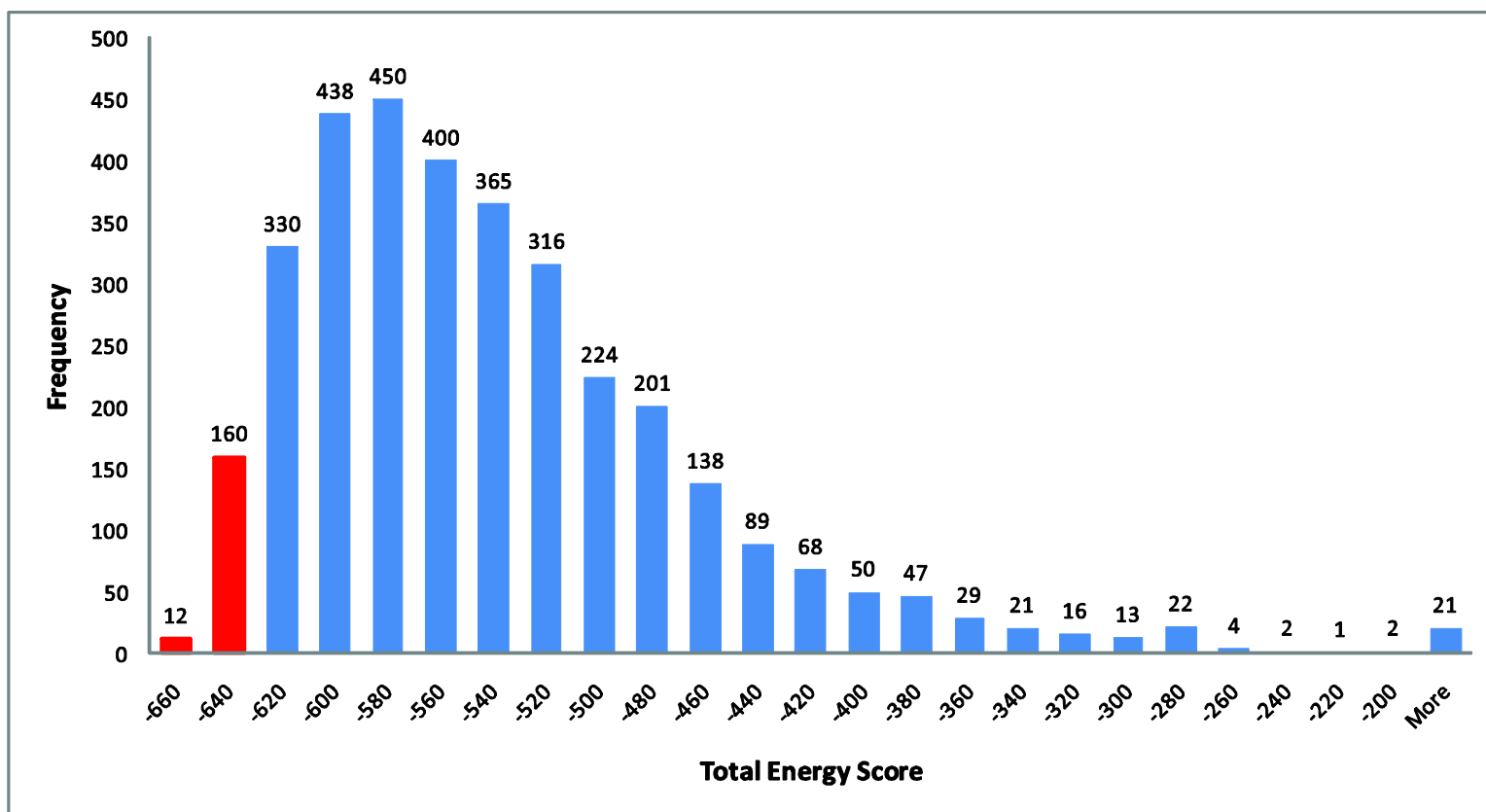
Sequences from the Binding Database were grouped based on the presence and location of exceptional mutations. A ClustalW alignment was used to identify exceptional mutations. The “constant-unbound” approach was used. Note that the small and variable sample size (n) makes comparing these values suspect. R-values and standard errors (kJ/mol, kcal/mol) between RosettaLigand predictions and experimental data are shown. Data is grouped by whether mutations are present in the flap region. A future study with a larger sample size is needed to differentiate between mutations in flexible and rigid regions of HIV-1 protease.

		$\Delta\Delta G$					$\Delta\Delta\Delta G$				
		n	$R_p$	$R_s$	kJ/mol	kcal/mol	n	$R_p$	$R_s$	kJ/mol	kcal/mol
Datapoints with no exceptional mutations		15	0.39±0.22	0.31±0.24	4.29,	1.03	10	-0.24±0.31	-0.07±0.33	4.68,	1.12
Mutations grouped by proximity to protease inhibitor	Only binding site mutations	17	0.09±0.25	0.01±0.25	8.12,	1.94	38	0.37±0.14	0.36±0.14	6.64,	1.59
	Only non-binding site mutations	44	0.36±0.13	0.27±0.14	6.10,	1.46	95	0.24±0.10	0.25±0.10	4.61,	1.10
	Mutations in and out of binding site	73	0.65±0.06	0.67±0.07	8.90,	2.13	280	0.78±0.02	0.79±0.02	5.83,	1.39
Mutations grouped by whether they are in flap region	Only flap region mutations	2	n/a	n/a	4.33,	1.03	1	n/a	n/a	0.91	0.22
	Only non-flap mutations	97	0.37±0.09	0.34±0.09	8.29,	1.91	584	0.28±0.04,	0.30±0.04	7.29	1.74
	Mutations in and out of flap region	35	0.76±0.07	0.78±0.07	7.24,	1.73	56	0.89±0.03,	0.88±0.03	4.41	1.05



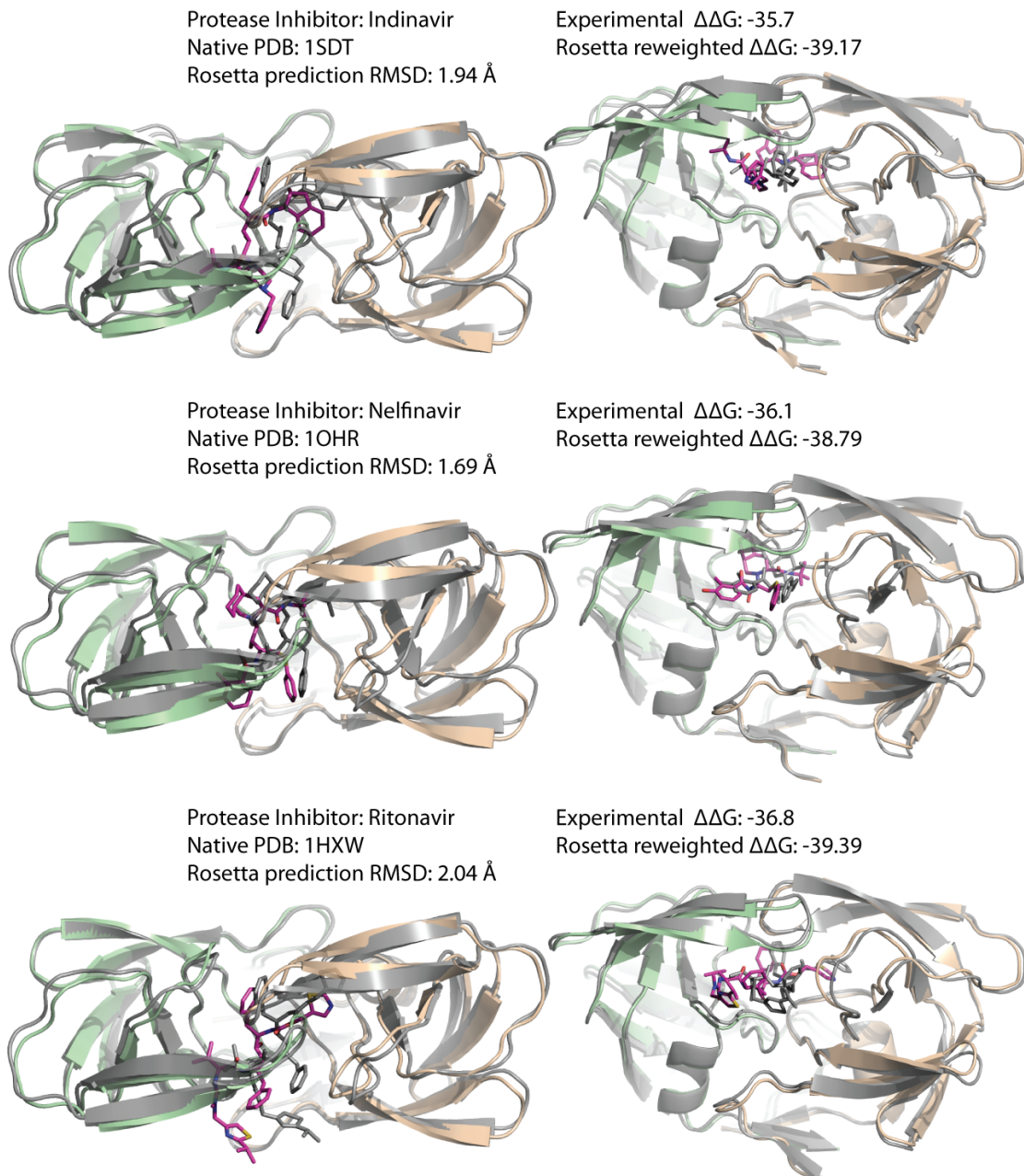
### E. Distribution of Rosetta energy scores for a set of HIV-1 PR/PI models

Histogram of Rosetta scores for prediction of Ritonavir binding to PR mutant sequence 1. A total of 3420 predicted energy scores are plotted (20 models produced for each of 171 backbone templates). We filter these models by selecting the top 5% (shown in red) by total energy score, and from these, the top model by interface score.



## F. Comparison of top Rosetta models and experimental structures

Gray structures are from the PDB ('native'), while colored structures are Rosetta predictions. All 3 Rosetta predictions are based on protease sequence 20. No experimentally determined structure is available for sequence 20. The the experimental ("native") structures shown are not based on sequence 20, but on other PR sequences. Thus ligand RMSD values here are a combination of the deviation between the modeled sequence and the experimental sequence, and the inaccuracy of the model.



## **G. HIV-1 PR/PI docking protocol**

### ***Preliminary Relax Step***

Two Rosetta executables are used in this analysis. Input structures without ligands are first relaxed using the fast relax protocol. We use all default options. Relaxing the input structure within the Rosetta force-field places the protein in a Rosetta energy minimum. This assures change in energy between unbound and bound structures is due to ligand binding, rather than minimization of the protein structure within the Rosetta forcefield.

#### **High resolution refinement**

*Revision of Rosetta used:* 32372

*Executable:* fast\_relax.linuxgccrelease

*Options used:* n/a

### ***Ligand docking step***

Ligand docking can be split into a low resolution and high resolution step. During low resolution, initial placement of the ligand, we allowed only 0.1 Å translational movement (`uniform_trans 0.1`), since the location of the binding pocket for these ligands is well defined. We rotate the ligand randomly up to 1000 times (`improve_orientation 1000`), searching for rotations with good attractive and repulsive scores.

The high resolution protocol includes 6 cycles of docking. Each cycle includes small movements of the ligand and sampling of side chain rotamers (`protocol abbrev2`). By tethering the ligand, we keep the ligand from moving too far over consecutive cycles of docking

(tether\_ligand 0.1). During high resolution docking we minimize the torsion angles within the ligand (minimize\_ligand) applying a harmonic constraint, where  $5^\circ$  is one standard deviation (harmonic\_torsions 5).

Until the final minimization the repulsive score term is down-weighted so that small clashes are allowed (soft\_rep). These clashes are resolved during the final minimization. The final minimization includes minimization backbone  $\phi/\psi$  angles (minimize\_backbone) with harmonic constraints on the  $C\alpha$  atoms, where  $0.2 \text{ \AA}$  is one standard deviation (harmonic\_Calphas 0.2).

### **RosettaLigand docking**

*Executable:* ligand\_dock.linuxgccrelease

*Options used:*

```
-docking  
  
    -uniform_trans 0.1  
  
    -ligand  
  
        -improve_orientation 1000  
  
        -minimize_ligand  
  
        -harmonic_torsions 5  
  
        -minimize_backbone  
  
        -harmonic_Calphas 0.2  
  
        -soft_rep  
  
        -old_estat  
  
        -protocol abbrev2  
  
        -tether_ligand 0.1
```

## H. Prediction of clinical outcomes

We hypothesized that change in viral load upon change in protease inhibitor (PI) prescription should correlate with change in PI binding affinity ( $\Delta\Delta\Delta G$ ). Thus we sought to compare RosettaLigand predicted  $\Delta\Delta\Delta G$ s with changes in viral load resulting from change in drug regimen.

Through collaboration with Dr. Richard D'Aquila and Marie Pia De Pasquale, we acquired viral load measurements for 4 patients infected with HIV-1. This data included dates the viral loads were taken, information about the protease sequence, and a list of prescribed anti-retroviral drugs at the time of each viral load measurement. This data is shown in the table below. Bold mutations are known to confer drug resistance. Other mutations are secondary mutations, meaning they do not themselves confer drug resistance but may be compensatory mutations, rendering the mutant virus more fit. Bold drugs are protease inhibitors. "IDs" are assigned to each of the 4 patients. Viral loads were measured before and after change of drug regimen.

ID	date	seq	Viral load	current prescription
<b>101</b>	9/26/03	<b>D30N N88D</b> A71T	21,000	AZT 3TC <b>NLF</b>
	9/08/04		16,000	ddI TDF <b>LPV</b>
	3/04/06		4,800	ddI TDF <b>LPV</b>
<b>102</b>	12/01/03	<b>I54V V82A I84V L90M</b> L10I V11VI A71V L89IV	390,000	3TC ddI TDF <b>LPV</b> D4T TDF <b>RTV fosAMP</b>
	2/24/04	<b>I54V V82A I84V L90M</b> L10I V11VI A71V L89IV	580	T20
<b>216</b>	1/06/04	L10V L63PA	23,000	3TC D4T <b>NLF</b>
	8/19/04		4,600	3TC D4T <b>NLF</b>
<b>218</b>	2/05/04	<b>D30N N88D L90LM</b> A71V	67,000	ABC TDF <b>LPV</b>
	11/23/04		80	ABC TDF <b>LPV</b>

In the case of patient 102, sequence data reveals that at two positions multiple polymorphisms are present (V11VI and L89IV). Thus 4 PR sequences were possible. We predicted Rosetta binding affinities for all four sequences. Similarly patient 216 contained two variants at position 63 (L63PA) and patient 218 contained two variants at position 90 (L90LM). Thus 2 PR sequences were possible for patients 216 and 218.

We threaded these 9 sequences onto each of our 171 template backbone structures (see Appendices B & D). These threaded models were relaxed 10 times each using the Rosetta energy function (see methods section of chapter 2). Each relaxed structure was repacked and minimized 10 times. The top scoring model from each set of 100 models was chosen, yielding 9x171 Rosetta input proteins. For patients 216 and 218 the protease inhibitor was not changed between viral load measurements, thus these datapoints were omitted from further analysis.

RosettaLigand was used to predict differences in binding affinity between the pairs of PIs taken by patient 101 and 102. Rosetta predictions were reweighted using the constant unbound  $\Delta\Delta G$  weights shown in Table II-1. For patient 101, the switch from lopinavir to nelfinavir accompanied a predicted increase in binding affinity of -0.3 kcal/mol. For patient 102 the switch from Lopinavir to Ritonavir accompanied a predicted decrease in binding affinity of 1.42 kcal/mol.

Unfortunately, patients 101 and 102 had multiple modifications in drug regimen. Patients 101 and 102 both had changes to their nucleotide analog reverse transcriptase inhibitor (NRTI) prescriptions. Patient 2 was also prescribed an HIV fusion inhibitor (T-20). Thus it is not possible to tell what portion of drop in viral load results from change in PI prescription.

I worked with a rotation student, Rebecca Levinson, to perform a similar experiment on a larger dataset acquired from the Treatment-Change Episode database on the Stanford University

HIV Drug Resistance Database website (<http://hivdb.stanford.edu/TCEs/>). Together we wrote a python script that parsed the database XML and identified datapoints where patients that had viral load measurements taken before and after change in protease inhibitor and without change in any other prescribed anti-retroviral drug. These filters left us with 22 pairs viral load measurements with HIV-1 PR sequence data. In the table below, columns titled “base VL” and “pose VL” report viral loads before and after change in protease inhibitor. Column “ $\Delta$  VL” reports the difference between “base VL” and “pose VL”.

We used RosettaLigand to predict changes in binding affinity that result from change in PI. We applied optimized weights from Table II-1, as described previously. Binding affinities before (“base  $\Delta\Delta G$ ”) and after (“post  $\Delta\Delta G$ ”) change in protease inhibitor are shown in the table below. The difference between these values is labeled as “ $\Delta\Delta\Delta G$ ”. Unfortunately we found only weak correlation between Rosetta predicted  $\Delta\Delta\Delta G$ s and  $\Delta$  VL ( $R = 0.24$ ). We believe that this is in part because of the inconsistencies in the dataset. Some datapoints were measured days apart, others were measured months apart from each other. Thus further research is needed in this area. As a first step we propose comparing Rosetta  $\Delta\Delta\Delta G$ s predictions results from in vitro viral replication assays.

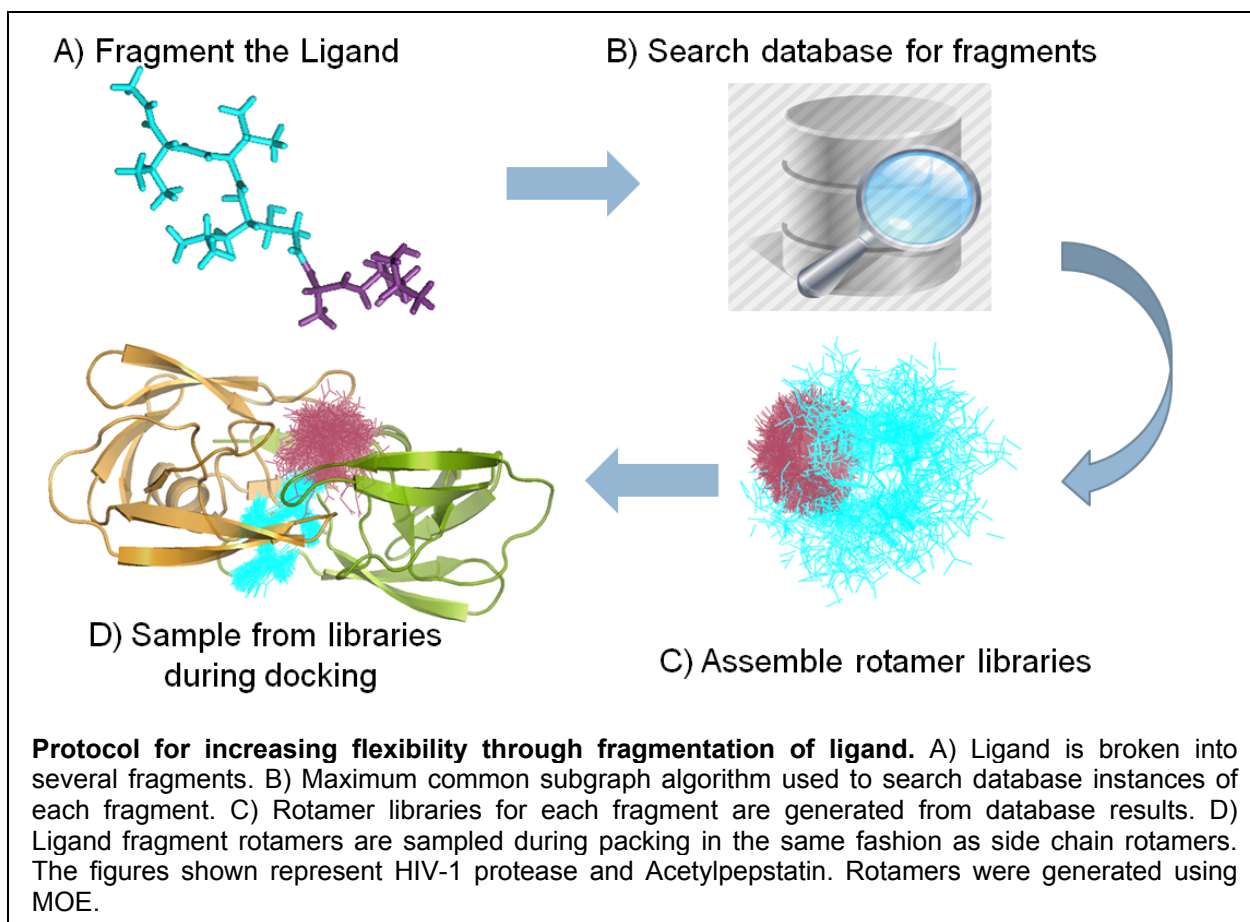
Patient #	Data from the Stanford TCE			Rosetta Predictions		
	base VL	post VL	$\Delta$ VL	base $\Delta\Delta$ G	post $\Delta\Delta$ G	$\Delta\Delta\Delta$ G
1	125893	31623	94270	-4.73	-11.18	6.45
52	199526	398107	-198581	-8.17	-1.51	-6.66
154	100000	251	99749	-11.43	-10.91	-0.52
1331	1585	79	1506	-10.5	-9.21	-1.29
70	5012	3162	1850	-9.44	-1.22	-8.22
306	63096	158489	-95393	-11.04	-13.58	2.54
1212	19953	251	19702	-6.66	-12.79	6.13
755	25119	200	24919	-9.58	-11.93	2.35
922	63096	1000	62096	-12.77	-11.04	-1.73
500	1585	79	1506	-6.78	-7.1	0.32
528	3162	631	2531	-10.75	-13.03	2.28
1278	19953	50119	-30166	-9.79	-10.5	0.71
964	31623	50	31573	-6.65	-10.59	3.94
113	5012	50119	-45107	-12.38	-14.57	2.19
77	15849	19953	-4104	-8.73	-1.8	-6.93
787	3981	200	3781	-11.37	-11.93	0.56
125	5012	1585	3427	-11.66	-11.75	0.09
958	63096	6310	56786	-11.57	-11.02	-0.55
1130	12589	501	12088	-10.62	-2.01	-8.61
365	19953	158	19795	-11.02	-0.94	-10.08
727	3981	79	3902	-10.86	-12.69	1.83
653	12589	251	12338	-11.31	-13.31	2
Correlation between $\Delta$ VL & $\Delta\Delta$ G						<b>0.244123</b>



## **I. Ligand docking through incremental construction**

The Figure below (next page) represents our vision for improved modeling of ligand flexibility within RosettaLigand. Prior to this work ligand flexibility within Rosetta was modeled through sampling ligand rotatable bonds during docking, and/or sampling of user provided conformations of ligands. Davis and Baker point out that when ligands present more than 7 rotatable bonds, the conformational space becomes computationally infeasible for RosettaLigand to efficiently sample (Davis and Baker 2009). This is true of many ligand docking applications (Erickson, Jalaie et al. 2004). The problem is reduced by using pregenerated conformations based on torsion profiles for each atom type pairing found in the Cambridge Structural Database (Kaufmann, Glab et al. 2008). However for large flexible ligands such as peptidomimetics, the number of feasible conformations can still be intractable.

Thus we sought to implement a fragment-based strategy for ligand flexibility. By splitting a ligand into multiple fragments, creating conformers for each fragment, and docking fragments one at a time, the competing goals of flexibility and efficiency can be balanced. Incremental construction strategies have been implemented within several ligand docking programs including FlexX (Rarey, Kramer et al. 1996), DOCK 4.0 (80 (Makino and Kuntz 1997), ADAM (Mizutani, Tomioka et al. 1994), Hammerhead (Welch, Ruppert et al. 1996), and SLIDE (Schnecke and Kuhn 2000) and are reviewed by Taylor et al (Taylor, Jewsbury et al. 2002).



We modified Rosetta code to allow ligands to consist of multiple fragments. Within Rosetta these fragments are analogous to amino acid residues (Figure 1A). This required overcoming several assumptions within Rosetta, namely that (1) a residue can only connect to a maximum of 3 other residues (2) a residue can only connect to another residue through one connection point; and (3) an atom can only be a member of one connection. Now residues can connect to any number of other residues, can connect to the same residue through multiple connections, and can connect to several residues through the same atom.

As a tool constructed originally for protein folding, Rosetta code contained an assumption that most residues, as part of a peptide chain, would connect to two other residues, upstream and downstream, and a few would connect to 3 residues (including a disulfide bond).

This assumption was present throughout Rosetta's scoring methods. In particular Rosetta caches atom/atom energies in energy tables in order to avoid repeating computations for atom pairs that maintain their relative positions between Rosetta sampling steps. In order to perform fast and efficient rotamer sampling, rotamer energies are stored in a "trie" data structure for fast lookup (Leaver-Fay, Kuhlman et al. 2005). These lookup functions assume that residues are connected to up to 3 other residues. The Overcoming this assumption required writing 250 additional lookup functions, as presented in revision 23336. The large number of functions has to do with the efficiency gain sought from avoiding polymorphism. This approach is called 'type resolution'.

Jeff Mendenhall and Sandeep Kothiwale have written algorithms to fragment ligands and to search the CSD for rotamers of a given fragment. A maximal common subgraph algorithm (Shen, Lange et al. 2008) was implemented in order to find fragments within larger molecules. This involves converting molecules in the CSD into a library of graphs. Each search fragment is then converted to a graph representation and compared with all CSD graphs. The set of matching subgraphs is returned. This work was accomplished using the BCL, a cheminformatics software package developed in the Meiler Lab.

Assembling ligand rotamer libraries for use with Rosetta (Figure 1C) will involve converting matching CSD subgraphs into inputs recognizable to Rosetta. These inputs could simply be PDB files containing each instance of the fragment found in the CSD. Alternatively rotamer libraries can be represented by files that list the combinations of torsion angles for each rotatable bond along with the propensity of those sets of torsions within the CSD. Sets of torsions that are more commonly seen in the PDB can be given more favorable Rosetta energies.

Figure 1D shows a real example of docking two fragment Acetylpepstatin into HIV-1 protease. This example demonstrates the added flexibility we hoped to achieve through fragmentation. However the protocol currently limits itself to two pre-assembled fragments. The atoms that form the connection point between the two fragments remain fixed, and fragment conformers are aligned onto these atoms. Future work entails writing an algorithm that docks an initial fragment with its conformers, and then connects additional fragments through multiple rounds of ligand conformer docking (see pseudocode below).

```
fragment molecule into pieces each with no more than 4 rotatable bonds
generate fragment conformers from database of known structures
for each fragment
    sample starting position, fragment conformers and side chain rotamers
    predict binding affinity
keep strongest binding affinity fragment as starting fragment
connect each remaining fragment using the original connectivity
    sample bond angles for the newly connected fragment
    sample fragment conformers and side chain rotamers
    sample rigid body position for extended molecule
minimize backbone, side-chain, and ligand torsion angles
```

## References

- Davis, I. W. and D. Baker (2009). "RosettaLigand docking with full ligand and receptor flexibility." Journal of Molecular Biology **385**(2): 381-392.
- Erickson, J. A., M. Jalaie, et al. (2004). "Lessons in molecular recognition: The effects of ligand and protein flexibility on molecular docking accuracy." Journal of Medicinal Chemistry **47**(1): 45-55.
- Kaufmann, K. W., K. Glab, et al. (2008). Small Molecule Rotamers Enable Simultaneous Optimization of Small Molecule and Protein Degrees of Freedom in ROSETTALIGAND Docking. German Conference on Bioinformatics. A. Beyer and M. Schroeder. Dresden: 148-157.
- Leaver-Fay, A., B. Kuhlman, et al. (2005). "Rotamer-pair energy calculations using a trie data structure." Algorithms in Bioinformatics, Proceedings **3692**: 389-400.
- Makino, S. and I. D. Kuntz (1997). "Automated flexible ligand docking method and its application for database search." Journal of Computational Chemistry **18**(14): 1812-1825.

- Mizutani, M. Y., N. Tomioka, et al. (1994). "Rational Automatic Search Method for Stable Docking Models of Protein and Ligand." Journal of Molecular Biology **243**(2): 310-326.
- Rarey, M., B. Kramer, et al. (1996). "A fast flexible docking method using an incremental construction algorithm." Journal of Molecular Biology **261**(3): 470-489.
- Schnecke, V. and L. A. Kuhn (2000). "Virtual screening with solvation and ligand-induced complementarity." Perspectives in Drug Discovery and Design **20**(1): 171-190.
- Shen, Y., O. Lange, et al. (2008). "Consistent blind protein structure generation from NMR chemical shift data." Proceedings of the National Academy of Sciences of the United States of America **105**(12): 4685-4690.
- Taylor, R. D., P. J. Jewsbury, et al. (2002). "A review of protein-small molecule docking methods." Journal of Computer-Aided Molecular Design **16**(3): 151-166.
- Welch, W., J. Ruppert, et al. (1996). "Hammerhead: Fast, fully automated docking of flexible ligands to protein binding sites." Chemistry & Biology **3**(6): 449-462.

## **J. Ligand design through incremental construction**

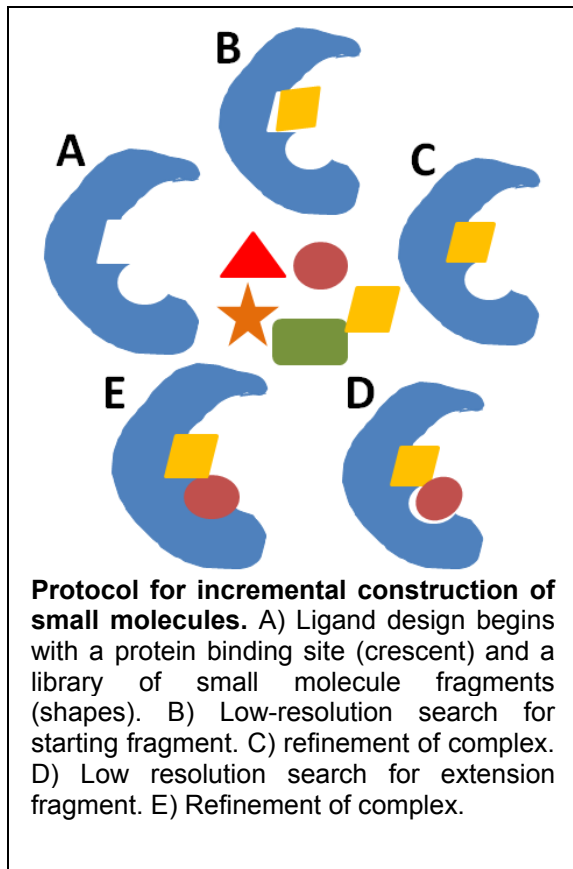
The advances to Rosetta code described in the previous section have direct application in the area of small molecule design. One can consider ligand docking through incremental construction a degenerate case of ligand design in which the fragments under consideration are constrained to only form the connections that rebuild the desired small molecule. Because of its atomic detail accuracy in modeling protein structure, we believe small molecule design using Rosetta will be a unique tool, especially tailored to improve design of molecules within flexible protein interfaces (e.g. inhibitors of HIV-1 protease).

### ***General Rosetta code modifications***

In addition to assumptions listed in the previous chapter, two additional Rosetta code assumptions no longer present limitations to ligand design. These are that (1) all of a residue's connection points are connected to other residues when Rosetta is initialized; and (2) residues are aware of the bond lengths and bond angles to form between themselves and their connecting partners.

Based on our improvements to RosettaLigand (see revision 37281), residues no longer require connection partners to be present. Our strategy at the time of revision 37281 required a residue's connection point to know the bond length, bond angle, and atom type that connected it to another residue. Only residues with overlapping connecting atom-types could be connected. As of revision 49115 this atom-type specific restriction has been removed. Revision 49115 implements an ideal bond length lookup table that builds connections with ideal bond lengths for pairs of atom types.

We have put in place the essential elements of Rosetta ligand design. These elements include code that (1) grows the small molecule by connecting fragments randomly chosen from a library of building blocks; (2) docks and scores the growing ligand, accepting or rejecting new growths based on Rosetta energy scores; (3) specifies growth termination criteria including molecular mass, number of hydrogen bond acceptors, number of hydrogen bond donors, number of heavy atoms, or total number of atoms; (4) adds hydrogen to unsatisfied valences to terminate ligand growth.



With these elements in place we sought to implement the following algorithm, presented as pseudo-code...

```

generate fragment library with conformers for each fragment
for fragment in fragment library
    dock fragment sampling fragment conformers and side chain rotamers
    predict binding affinity
keep strongest binding affinity fragment as starting fragment
while user defined growth cutoffs have not been met
    connect random fragment at random connection point
    sample fragment conformers and side chain rotamers
    sample rigid body position for extended molecule
    accept or reject new growth using a Monte Carlo approach
minimize backbone, side-chain, and ligand torsion angles

```

### ***Creation of ligand fragments***

Before ligand design can occur, a library of small molecule fragments must be generated. The script `mol_to_params.py` has been modified to allow creation of these small molecule fragments. Creation of small molecule fragments begins with the identification of MOL or MOL2/MDL files of molecules containing the fragments the user would like to add to his/her fragment library. To the bottom of these files are added lines of the format “M SPLT <ID\_1> <ID\_2>”. ID\_1 and ID\_2 should be replaced with the atom ID in the MOL or MDL file between which you would like to create fragments. If this is a hydrogen bond, then only one fragment will be created.

### ***GrowLigand***

Ligand design begins with the docking of an initial molecule fragment from a library of fragments. This step proceeds using the standard docking procedure outlined in chapter 3. Next the initial fragment is extended using the GrowLigand XML element. GrowLigand at this point does not have any scoring functionality. It simply selects at random a fragment from the provided fragment library and attaches that fragment to the ligand with the specified chain. The attachment is made by randomly selecting a connection point on the growing ligand and on the fragment to be connected. Docking and scoring of the extended ligand occurs via ligand docking XML described in chapter 2.

```
<GrowLigand name="string" chain="string"/>
```

### ***Filters for terminating ligand design growth***

Hydrogen bond acceptor and donor filters stop growth when a provided cutoff is surpassed. The HeavyAtom filter terminates growth when the number of non-hydrogen atoms has reached a cutoff value, while the AtomCount filter relies on total number of atoms including



hydrogens. Similarly the MolecularMass filter and MolarMass filters stop small molecule extension when their limits are exceeded. The CompleteConnections filter stops growth if there are no growths possible. The ChainExistsFilter is useful to ensure that ligand design only occurs for those starting fragments that have been positioned in the binding pocket.

```
<HBondAcceptor name="string" chain="string" hbond_acceptor_limit=<int>/>
<HBondDonor name="string" chain="string" hbond_donor_limit=<int>/>
<HeavyAtom name="string" chain="string" heavy_atom_limit=<int>/>
<AtomCount name="string" chain="string" atom_limit=<int>/>
<MolecularMass name="string" chain="string" mass_limit=<int>/>
<MolarMass name="string" chain="string" mass_limit=<int>/>
<CompleteConnections name="string" chain="string"/>
<ChainExists name="string" chain="string"/>
```

### ***AddHydrogens***

After terminating growth of a small molecule, it is likely that the molecule contains connection points that are not connected to other small molecule fragments. These connecting atoms retain geometry that suggests atoms are missing. The AddHydrogens code adds hydrogens to these unsatisfied connection points.

```
<AddHydrogens name="string" chain="string"/>
```

### ***Ligand design XML***

Below is described a complete XML algorithm for designing a small molecule fragment-by-fragment using the new Rosetta code. This algorithm has not been tested or optimized. It is designed as a starting point for future users and developers to improve upon. In short, this algorithm docks a starting fragment, extends, docks, and extends again until cutoff criteria are reached. Finally ligand specific scores are appended to the output PDB.

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <ligand_soft_rep weights=ligand_soft_rep>
      <Reweight scoretype=hack_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=hack_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>
<FILTERS>
  <CompleteConnections name=connections chain="string"/>
  <HBondAcceptor name=acceptors chain=X hbond_acceptor_limit=10/>
  <HBondDonor name=donors chain=X hbond_donor_limit=5/>
  <AtomCount name=atoms chain=X atom_limit=70/>
  <MolecularMass name=mass chain=X mass_limit=500/>
  <CompoundStatement name=all_filters>
    <NOT filter_name=connections/>
    <ANDNOT filter_name=acceptors/>
    <ANDNOT filter_name=donors/>
    <ANDNOT filter_name=atoms/>
    <ANDNOT filter_name=mass/>
  </CompoundStatement>
</FILTERS>
<MOVERS>
single movers
  <StartFrom name=start_from chain=X>
    <Coordinates x=-1.731 y=32.589 z=-5.039/>
  </StartFrom>

```

```

<Translate name=translate chain=X distribution=uniform angstroms=1.0
    cycles=20/>
<Rotate name=rotate chain=X distribution=uniform degrees=40 cycles=100/>
<SlideTogether name=slide_together chains=X/>
<HighResDocker name=high_res_docker cycles=6 repack_every_Nth=3
    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/>
<GrowLigand name="string" chain="string"/>
<AddHydrogens name="string" chain="string"/>
compound movers
<ParsedProtocol name=low_res_dock>
  <Add mover_name=start_from/>
  <Add mover_name=translate/>
  <Add mover_name=rotate/>
  <Add mover_name=slide_together/>
</ParsedProtocol>
<ParsedProtocol name=high_res_dock>
  <Add mover_name=high_res_docker/>
  <Add mover_name=final/>
</ParsedProtocol>
<ParsedProtocol name= complete_dock>
  <Add mover_name=low_res_dock/>
  <Add mover_name=high_res_dock/>
</ParsedProtocol>
<ParsedProtocol name=grow_dock>
  <Add mover_name=grow/>
  <Add mover_name=high_res_dock/>
</ParsedProtocol>
<LoopOver name=grow_loop mover_name=grow_dock filter_name=all_filters>
</MOVERS>
<PROTOCOLS>
Dock the starting fragment
  <Add mover_name=complete_dock/>
Grow and dock in a loop, until cutoff filters are hit
  <Add mover_name=grow_loop/>
Add final ligand scores
  <Add mover_name=add_scores/>
</PROTOCOLS>
</ROSETTASCRIPTS>

```

### *Future directions*

While many of the essential elements of RosettaLigandDesign have been implemented, the effectiveness of RosettaLigandDesign has not been demonstrated. This will first require the preparation of a database of design fragments. For each fragment a rotamer library similar to

amino acid rotamer libraries should be prepared. Rotamer libraries will allow flexibility through conformational sampling during design.

In order to evaluate RosettaLigand designs, it will be necessary to develop a design benchmark. This will consist of a collection of 20 protein/ligand complexes of known structure. Each ligand in this dataset will be split into fragments. The first benchmark test will require Rosetta to reassemble the fragments using the correct connectivity and recovering the correct binding pose. The second benchmark will build upon the first by mixing all the fragments from the 20 ligand benchmark and requiring Rosetta to select the correct fragments from among this set. Finally Rosetta will be required to design small molecules using fragments from the complete ligand fragment library with rotamers. Results will be evaluated based on the number of key contacts that are recapitulated. Between each test it will be necessary to reevaluate and optimize the design algorithm. This will involve finding a balance between efficiency and accuracy. Fast grid based low-resolution screening of fragments will be used to increase efficiency.

## K. XML used with standard docking (chapter 4)

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <ligand_soft_rep weights=ligand_soft_rep>
      <Reweight scoretype=hack_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=hack_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>
    <inhibitor_dock_sc chain=X cutoff=6.0 add_nbr_radius=true all_atom_mode=true/>
    <inhibitor_final_sc chain=X cutoff=6.0 add_nbr_radius=true all_atom_mode=true/>
    <inhibitor_final_bb 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=inhibitor_dock_sc/>
    <side_chain_for_final ligand_areas=inhibitor_final_sc/>
    <backbone ligand_areas=inhibitor_final_bb extension_window=3/>
  </INTERFACE_BUILDERS>
  <MOVEMAP_BUILDERS>
    <docking sc_interface=side_chain_for_docking minimize_water=false/>
    <final sc_interface=side_chain_for_final bb_interface=backbone
  minimize_water=false/>
  </MOVEMAP_BUILDERS>
  <MOVERS>
    <Translate name=translate chain=X distribution=uniform angstroms="%%BIG%%"
  cycles=50 force=true/> first place the ligand
    <Rotate name=rotate_x chain=X distribution=uniform degrees=360 cycles=800/>
    <SlideTogether name=slide_together chain=X/>
    <HighResDocker name=high_res_docker cycles=6 repack_every_Nth=3
  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/>
  </MOVERS>
</ROSETTASCRIPTS>
```

```

    <Add mover_name=translate/>
    <Add mover_name=rotate_x/>
    <Add mover_name=slide_together/>
  <Add mover_name=high_res_docker/>
    <Add mover_name=final/>
    <Add mover_name=add_scores/>
</PROTOCOLS>
</ROSETTASCRIPTS>

```

## L. XML used with protein-centric docking

```

<ROSETTASCRIPTS>
  <SCOREFXNS>
    <ligand_soft_rep weights=ligand_soft_rep>
      <Reweight scoretype=hack_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=hack_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>
    <inhibitor_dock_sc chain=X cutoff=6.0 add_nbr_radius=true all_atom_mode=true/>
    <water_dock_sc chain=Y cutoff=2.0 add_nbr_radius=true all_atom_mode=true/>
    <inhibitor_final_sc chain=X cutoff=6.0 add_nbr_radius=true all_atom_mode=true/>
    <water_final_sc chain=Y cutoff=2.0 add_nbr_radius=true all_atom_mode=true/>
    <inhibitor_final_bb chain=X cutoff=7.0 add_nbr_radius=false all_atom_mode=true
  Calpha_restraints=0.3/>
    <water_final_bb chain=Y cutoff=2.5 add_nbr_radius=false all_atom_mode=true
  Calpha_restraints=0.3/>
  </LIGAND_AREAS>
  <INTERFACE_BUILDERS>
    <side_chain_for_docking ligand_areas=inhibitor_dock_sc,water_dock_sc/>
    <side_chain_for_final ligand_areas=inhibitor_final_sc,water_final_sc/>
    <backbone ligand_areas=inhibitor_final_bb,water_final_bb 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>
    <Translate name=translate_x chain=X distribution=uniform angstroms=5.0
cycles=50/>
    <Translate name=translate_y chain=Y distribution=uniform angstroms=4.0
cycles=50/>
    <ParsedProtocol name=translate>
      <Add mover_name=translate_x/>
      <Add mover_name=translate_y/>
    </ParsedProtocol>
    <ParsedProtocol name=translate_three_fourths mode=single_random>
      <Add mover_name=translate/>
      <Add mover_name=translate/>
      <Add mover_name=translate/>
      <Add mover_name=null/>
    </ParsedProtocol>
    <Rotate name=rotate_x chain=X distribution=uniform degrees=360 cycles=800/>
    <Rotate name=rotate_y chain=Y distribution=uniform degrees=360 cycles=100/>
    <SlideTogether name=slide_together chain=X/>
    <HighResDocker name=high_res_docker cycles=6 repack_every_Nth=3
scorefxn=ligand_soft_rep movemap_builder=docking/>
    <FinalMinimizer name=final scorefxn=hard_rep movemap_builder=final/>
    <InterfaceScoreCalculator name=add_scores chains=X,Y scorefxn=hard_rep/>
    <ReportToDB name=report_scores db="%%output_db%%" sample_source=job_data>
      <feature name=JobDataFeatures/>
    </ReportToDB>
  </MOVERS>
  <PROTOCOLS>
    <Add mover_name=translate_three_fourths/>
    <Add mover_name=rotate_x/>
    <Add mover_name=rotate_y/>
    <Add mover_name=slide_together/>
    <Add mover_name=high_res_docker/>
    <Add mover_name=final/>
    <Add mover_name=add_scores/>
    <Add mover_name=report_scores/>
  </PROTOCOLS>
</ROSETTASCRIPTS>

```

## M. XML used with ligand-centric docking

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <ligand_soft_rep weights=ligand_soft_rep>
      <Reweight scoretype=hack_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=hack_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>
    <inhibitor_dock_sc chain=X cutoff=6.0 add_nbr_radius=true all_atom_mode=true/>
    <water_dock_sc chain=W cutoff=2.0 add_nbr_radius=true all_atom_mode=true/>
    <inhibitor_final_sc chain=X cutoff=6.0 add_nbr_radius=true all_atom_mode=true/>
    <water_final_sc chain=W cutoff=2.0 add_nbr_radius=true all_atom_mode=true/>
    <inhibitor_final_bb chain=X cutoff=7.0 add_nbr_radius=false all_atom_mode=true
Calpha_restraints=0.3/>
    <water_final_bb chain=W cutoff=2.5 add_nbr_radius=false all_atom_mode=true
Calpha_restraints=0.3/>
  </LIGAND_AREAS>
  <INTERFACE_BUILDERS>
    <side_chain_for_docking ligand_areas=inhibitor_dock_sc,water_dock_sc/>
    <side_chain_for_final ligand_areas=inhibitor_final_sc,water_final_sc/>
    <backbone ligand_areas=inhibitor_final_bb,water_final_bb 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>
    <Translate name=translate_X chain=X distribution=uniform
angstroms="%%BIG%%" cycles=50 force=true tag_along_chains=W/> first place the ligand
    <CompoundTranslate name=compound_translate randomize_order=true
allow_overlap=false>
    <Translates chain=W distribution=uniform angstroms="%%SMALL%%"
cycles=50 force=true/> then the water molecules
```



```

    </CompoundTranslate>
    <Rotate name=rotate_x chain=X distribution=uniform degrees=360 cycles=800/>
    <Rotates name=rotate_w chain=W distribution=uniform degrees=360 cycles=100/>
    <SlideTogether name=slide_together chains=X,W/>
    <HighResDocker name=high_res_docker cycles=6 repack_every_Nth=3
scorefxn=ligand_soft_rep movemap_builder=docking/>
    <FinalMinimizer name=final scorefxn=hard_rep movemap_builder=final/>
    <InterfaceScoreCalculator name=add_scores chains=X,W scorefxn=hard_rep/>
</MOVERS>
<PROTOCOLS>
    <Add mover_name=translate_X/>
    <Add mover_name=compound_translate/>
    <Add mover_name=rotate_x/>
    <Add mover_name=rotate_w/>
    <Add mover_name=slide_together/>
    <Add mover_name=high_res_docker/>
    <Add mover_name=final/>
    <Add mover_name=add_scores/>
</PROTOCOLS>
</ROSETTASCRIPTS>

```

## N. Summary of my commits to the Rosetta SVN server

Sdfasdf

### 2012-06-19.

Fixing a bug in the ligand rotation code that caused way too many rotations to occur  
ligand\_dock\_script integration test is expected to fail.

M rosetta\_source/src/protocols/ligand\_docking/Rotate.cc

### 2012-06-13.

The list of includes was not correct in the template file.

M rosetta\_source/src/pilot\_apps.src.settings.template

### 2012-05-31.

- 1) Removing duplicated code (CompleteConnectionsFilter)
- 2) Adding new filters: MolarMass, MolecularMass
- 3) Modifying ligand Rotate code so that little things that tag\_along (water, metal)  
can rotate with the ligand
- 4) Adding some unit tests.

No integration test changes are expected

M rosetta\_source/test/core/chemical/ResidueTypeSetTests.cxxtest.hh

M rosetta\_source/src/protocols/ligand\_docking/Rotates.cc

M rosetta\_source/src/protocols/init/init.FilterCreators.ihh

A rosetta\_source/test/core/chemical/ElementSet.cxxtest.hh

A rosetta\_source/src/protocols/ligand\_docking/MolecularMassFilter.hh

M rosetta\_source/src/protocols/ligand\_docking/ChainExistsFilter.cc

```

M rosetta_source/src/protocols/init/init.FilterRegistrators.ihh
D rosetta_source/src/protocols/filters/CompleteConnectionsFilter.cc
M rosetta_source/src/protocols/ligand_docking/Rotate.cc
A rosetta_source/src/protocols/ligand_docking/MolarMassFilterCreator.hh
A rosetta_source/test/core/chemical/ideal_bond_lengths.txt
A rosetta_source/src/protocols/ligand_docking/MolarMassFilter.cc
A rosetta_source/test/core/chemical/element_properties.txt
M rosetta_source/src/protocols/ligand_docking/MinimizeBackbone.cc
D rosetta_source/src/protocols/filters/CompleteConnectionsFilter.hh
A rosetta_source/src/protocols/ligand_docking/MolecularMassFilterCreator.hh
M rosetta_source/src/protocols/ligand_docking/Rotate.hh
A rosetta_source/src/protocols/ligand_docking/MolecularMassFilter.cc
M rosetta_source/src/protocols_h.4.src.settings
A rosetta_source/src/protocols/ligand_docking/MolarMassFilter.hh
A rosetta_source/test/core/chemical/IdealBondLengthSet.cxxtest.hh

```

#### 2012-05-30.

Jump now has an additional setter that takes a vector of length 6 (3 translational, 3 rotational degrees of freedom)

Its gaussian move now returns the move that was made.

RigidBodyRandomizeMover can now remember the random move that was made and apply that same move again.

```

M rosetta_source/src/protocols/rigid/RigidBodyMover.hh
M rosetta_source/src/core/kinematics/Jump.hh
M rosetta_source/src/protocols/rigid/RigidBodyMover.cc
M rosetta_source/src/core/kinematics/Jump.cc

```

#### 2012-05-28.

Modified the integration script so that --host option can work in the Meiler lab. The reason it doesn't work is because we use tcsh. Many of us would rather use bash but that's the way it is. The "horrible hack" found in previous versions of 'integration.py' assumes a bash shell is being used...

```
'PATH='
```

instead of...

```
'set PATH='
```

for tcsh.

I now add this special PATH setting line to the command.sh file, then call that file as usual:

```
bash command.sh
```

In addition, you can now specify how many nodes on each host to use in this fashion:

```
<host>/<num_procs>
```

So for meilerlab people...

```
./integration.py --host=hydrogen/4 --host=manganese/6 ...
```

```
M rosetta_tests/integration/integration.py
```

#### 2012-05-25.

Pose's append residue by bond function now has an optional argument "bool lookup\_bond\_length". If set to true, keep the same geometry but adjust the bond length based on a table lookup. This is useful for small molecule design. No integration tests are expected to fail.

```

M rosetta_source/src/core/chemical/Element.hh
M rosetta_source/src/core/conformation/util.cc
M rosetta_source/src/core/pose/Pose.cc
A rosetta_source/src/core/chemical/IdealBondLengthSet.hh
M rosetta_source/src/core/chemical/ChemicalManager.hh
M rosetta_source/src/core/chemical/ResidueType.hh

```

M rosetta\_source/src/core/pose/util.cc  
 M rosetta\_source/src/core/conformation/util.hh  
 M rosetta\_source/src/protocols/ligand\_docking/GrowLigand.cc  
 M rosetta\_source/src/core/pose/Pose.hh  
 M rosetta\_source/src/core/conformation/Conformation.cc  
 M rosetta\_source/src/core/chemical/residue\_io.cc  
 A rosetta\_source/src/core/chemical/IdealBondLengthSet.fwd.hh  
 A rosetta\_database/chemical/atom\_type\_sets/fa\_standard/ideal\_bond\_lengths.txt  
 M rosetta\_source/src/core/pose/util.hh  
 M rosetta\_source/src/core/chemical/Element.cc  
 M rosetta\_source/src/core/chemical/ResidueConnection.hh  
 M rosetta\_source/src/protocols/ligand\_docking/GrowLigand.hh  
 M rosetta\_source/src/core.2.src.settings  
 A rosetta\_source/src/core/chemical/IdealBondLengthSet.cc  
 M rosetta\_source/src/core/chemical/ChemicalManager.cc  
 M rosetta\_source/src/core/conformation/Conformation.hh  
 M rosetta\_source/src/core/chemical/ResidueType.cc

**2012-04-30.**

oops. forgot my return statement.

M rosetta\_source/src/protocols/features/PoseConformationFeatures.cc

**2012-04-30.**

Fixing a memory error in report\_features that Rocco pointed out. It was in an else clause that hadn't yet been triggered. Also removing 1146 warnings in FoldTree.hh

M rosetta\_source/src/protocols/features/PoseConformationFeatures.hh  
 M rosetta\_source/src/protocols/features/PoseConformationFeatures.cc  
 M rosetta\_source/src/core/kinematics/FoldTree.hh

**2012-04-19.**

Function declared, but not defined, broke the PyRosetta build. Removing unused function. No test changes expected

M rosetta\_source/src/protocols/ligand\_docking/SlideTogether.hh

**2012-04-17.**

1) New ChainExists filter filters based on presence of a chain  
 2) Uniform Sphere mover can now remember the random move it last applied so it can apply the same random move to other chains  
 3) Ligand docking files updated to allow advanced movements of several ligands/waters/metals, etc. For instance it is now possible to first translate a ligand with a large translation and have waters or metals around it move with it. Next smaller translations are applied to the waters and metals. SlideTogether mover also slides the ligand and associated waters and metals.

Integration tests that will change:

ligand\_dock\_script

ligand\_dock\_7cpa

ligand\_dock\_grid

kinemage\_grid\_output

ligand\_database\_io

M rosetta\_source/src/protocols/ligand\_docking/HighResDocker.cc  
 M rosetta\_source/src/protocols/ligand\_docking/Rotates.cc  
 A rosetta\_source/src/protocols/ligand\_docking/ChainExistsFilterCreator.hh  
 M rosetta\_source/src/protocols/init/init.FilterCreators.ihh  
 M rosetta\_source/src/protocols/rigid/RigidBodyMover.cc  
 A rosetta\_source/src/protocols/ligand\_docking/ChainExistsFilter.cc  
 M rosetta\_source/src/protocols/init/init.FilterRegistrators.ihh  
 M rosetta\_tests/integration/tests/ligand\_dock\_grid/ligand\_dock.xml

```

M rosetta_source/src/protocols/ligand_docking/Translate.cc
M rosetta_tests/integration/tests/kinemage_grid_output/ligand_dock.xml
M rosetta_source/src/protocols/rigid/RigidBodyMover.hh
M rosetta_source/src/protocols/ligand_docking/SlideTogether.cc
M rosetta_source/src/protocols/ligand_docking/CompoundTranslate.cc
A rosetta_source/src/protocols/ligand_docking/ChainExistsFilter.hh
M rosetta_tests/integration/tests/ligand_database_io/ligand_dock.xml
M rosetta_source/src/protocols/ligand_docking/InterfaceScoreCalculator.cc
M rosetta_source/src/protocols/ligand_docking/MinimizeBackbone.cc
M rosetta_source/src/protocols/ligand_docking/InterfaceBuilder.cc
M rosetta_source/src/protocols/ligand_docking/Translate.hh
M rosetta_source/src/protocols/ligand_docking/MoveMapBuilder.cc
M rosetta_tests/integration/tests/ligand_dock_script/ligand_dock.xml
M rosetta_source/src/protocols/ligand_docking/SlideTogether.hh
M rosetta_source/src/protocols_h.4.src.settings

```

#### 2012-04-17.

Now the FaDockingSlideIntoContact mover can take more than 1 jump\_id. Everything downstream of these jumps is moved together during apply. No test changes expected.

```

M rosetta_source/src/protocols/docking/DockingInitialPerturbation.hh
M rosetta_source/src/protocols/docking/DockingInitialPerturbation.cc

```

#### 2012-04-05.

Replacing 2 functions in utility/string\_util.cc...

```

std::vector split(std::string const &)
std::vector string_split(std::string const &, char)

```

with

```

utility::vector1 split(std::string const &)
utility::vector1 string_split(std::string const &, char)

```

Refactoring code that uses these functions. All integration tests passed on my end, although there were several numerical instability sort of issues (e.g. a value in the tenth decimal place changing from a 7 to an 8)

```

M rosetta_source/src/protocols/enzdes/enzdes_util.cc
M rosetta_source/src/protocols/jd2/JobOutputter.cc
M rosetta_source/src/core/pose/util.cc
M rosetta_source/src/protocols/jd2/message_listening/DbMoverMessageListener.cc
M rosetta_source/src/protocols/features/util.cc
M rosetta_source/src/protocols/optimize_weights/OptEDData.cc
M rosetta_source/src/utility/string_util.hh
M rosetta_source/src/protocols/toolbox/match_enzdes_util/EnzdesLoopsFile.cc
M rosetta_source/src/core/conformation/symmetry/SymDof.cc
M rosetta_source/src/protocols/match/output/PDBWriter.cc
M rosetta_source/src/protocols/match/downstream/ScoringSecMatchRPE.cc
M rosetta_source/src/protocols/match/MatcherTask.cc
M rosetta_source/src/core/conformation/symmetry/VirtualCoordinate.hh
M
rosetta_source/src/protocols/toolbox/match_enzdes_util/MatchConstraintFileInfo.cc
M rosetta_source/src/utility/file/file_sys_util.cc
M rosetta_source/src/core/chemical/sdf/mol_util.cc
M rosetta_source/src/core/conformation/symmetry/VirtualCoordinates.cc
M
rosetta_source/src/core/scoring/electron_density_atomwise/ElectronDensityAtomwise.cc
M rosetta_source/src/protocols/dna/util.cc
M rosetta_source/src/core/scoring/methods/SequenceDependentRefEnergy.cc
M rosetta_source/src/core/scoring/SS_Killhairpins_Info.cc

```

```

M rosetta_source/src/apps/pilot/dekim/score_nonlocal_frags.cc
M rosetta_source/src/core/chemical/orbitals/OrbitalType.cc
M rosetta_source/src/core/chemical/sdf/ctab_base.cc
M rosetta_source/src/core/chemical/sdf/v3_parser.cc
M rosetta_source/src/protocols/jobdist/Jobs.cc
M
rosetta_source/src/protocols/fltdsgn/filters/ParallelBetaPairingPreferenceFilter.cc
M rosetta_source/src/apps/pilot/mike/mini_rosetta_native_client.cc
M
rosetta_source/src/protocols/comparative_modeling/hybridize/HybridizeProtocol.cc
M rosetta_source/src/protocols/simple_moves/MinMover.cc
M rosetta_source/src/core/pose/util.hh
M rosetta_source/src/protocols/ligand_docking/InterfaceScoreCalculator.cc
M rosetta_source/src/protocols/relax/FastRelax.cc
M rosetta_source/src/protocols/qsar/scoring_grid/polarizGrid.cc
M rosetta_source/src/core/conformation/symmetry/SymDof.hh
M rosetta_source/src/core/conformation/symmetry/SymmData.cc
M rosetta_source/src/protocols/fltdsgn/topology/HelixPairing.cc
M rosetta_source/src/devel/helixAssembly/NativeResidueReader.cc
M rosetta_source/src/apps/pilot/mike/evolution.cc
M rosetta_source/src/protocols/jd2/parser/Blueprint.cc
M
rosetta_source/src/protocols/toolbox/task_operations/RestrictToInterfaceVectorOperation.cc
M rosetta_source/src/utility/file/file_sys_util.hh
M rosetta_source/src/numeric/interpolation/util.cc
M rosetta_source/src/protocols/loophash/BackboneDB.cc
M rosetta_source/src/protocols/fltdsgn/topology/StrandPairing.cc
M rosetta_source/src/protocols/fltdsgn/potentials/SetAACompositionPotential.cc
M rosetta_source/src/protocols/wum/DatabaseEntryWorkUnit.cc
M rosetta_source/src/core/chemical/sdf/v3_parser.hh
M rosetta_source/src/apps/pilot/nobuyasu/pick_bab.cc
M rosetta_source/src/core/chemical/sdf/ctab_parser.cc
M rosetta_source/src/protocols/ligand_docking/InterfaceScoreCalculator.hh
M rosetta_source/src/protocols/dna/DnaDesignDef.cc
M rosetta_source/src/protocols/loophash/LoopHashSampler.cc
M rosetta_source/src/protocols/loops/LoopsFileIO.cc
M rosetta_source/src/utility/string_util.cc
M rosetta_source/src/core/conformation/symmetry/VirtualCoordinate.cc
M rosetta_source/src/protocols/fltdsgn/topology/HSSTriplet.cc
M rosetta_source/src/core/scoring/electron_density/ElectronDensity.cc
M rosetta_source/src/core/kinematics/MoveMap.cc
M rosetta_source/src/protocols/motifs/motif_utils.cc
M rosetta_source/src/protocols/optimize_weights/NestedEnergyTermOptEData.cc
M rosetta_source/src/core/scoring/orbitals/OrbitalsLookup.cc
M rosetta_source/src/protocols/ligand_docking/InterfaceBuilder.cc
M rosetta_source/src/core/import_pose/import_pose.cc
M
rosetta_source/src/protocols/protein_interface_design/movers/DockAndRetrieveSidechains.cc
M rosetta_source/src/protocols/dna/DnaInterfacePacker.cc

```

2011-11-23.

When docking ligands with RosettaScripts each ligand needs its own 1-letter PDB chain. A new command line flag tells rosetta to turn each Residue within specified 1-letter PDB chain characters into separate chains (incrementing the chain\_id).

"-in:file:treat\_residues\_in\_these\_chains\_as\_separate\_chemical\_entities"

Ligand Rotation and Translation code updated to reflect this new feature.

Integration test changes expected for:

ligand\_dock\_script

features

M rosetta\_source/src/protocols/features/StructureFeatures.cc

D

rosetta\_tests/integration/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_input.pdb

M rosetta\_source/src/basic/options/options\_rosetta.py

A rosetta\_source/src/protocols/ligand\_docking/Rotates.hh

M rosetta\_source/src/core/pose/util.cc

M rosetta\_source/src/core/io/pdb/file\_data.cc

D

rosetta\_tests/integration/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_native.pdb.gz

M rosetta\_source/src/protocols/ligand\_docking/Translate.cc

M rosetta\_source/src/protocols/features/DatabaseStatements.cc

M rosetta\_source/src/protocols/ligand\_docking/Rotate.cc

M rosetta\_source/doc/options.dox

M rosetta\_source/src/basic/options/keys/OptionKeys.cc.gen1.hh

M rosetta\_source/src/basic/options/keys/in.OptionKeys.gen.hh

M rosetta\_source/src/basic/options/keys/OptionKeys.cc.gen3.hh

D

rosetta\_tests/integration/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_input.pdb.gz

M rosetta\_source/src/protocols/init.cc

M rosetta\_source/src/basic/options/option.cc.gen.hh

M rosetta\_source/src/protocols/ligand\_docking/CompoundTranslate.hh

M

rosetta\_database/chemical/residue\_type\_sets/fa\_standard/residue\_types/metal\_ions/K.params

D

rosetta\_tests/integration/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_native.pdb

A rosetta\_source/src/protocols/ligand\_docking/Rotates.cc

M rosetta\_source/src/protocols/features/FeaturesReporter.cc

M rosetta\_database/chemical/residue\_type\_sets/fa\_standard/residue\_types.txt

M rosetta\_tests/integration/tests/ligand\_dock\_script/command

M rosetta\_source/src/basic/options/keys/OptionKeys.cc.gen0.hh

M rosetta\_source/src/protocols/ligand\_docking/CompoundTranslate.cc

M rosetta\_source/src/basic/options/keys/OptionKeys.cc.gen2.hh

M rosetta\_source/src/protocols.src.settings

M rosetta\_source/src/core/pose/util.hh

M rosetta\_source/src/protocols/ligand\_docking/Translate.hh

M rosetta\_source/src/protocols/ligand\_docking/Rotate.hh

A rosetta\_source/src/protocols/ligand\_docking/Rotates.fwd.hh

M rosetta\_tests/integration/tests/ligand\_dock\_script/flags

A rosetta\_source/src/protocols/ligand\_docking/RotatesCreator.hh

M rosetta\_tests/integration/tests/ligand\_dock\_script/ligand\_dock.xml

M rosetta\_tests/integration/tests/ligand\_dock\_script/inputs/7cpa.params

**2011-10-27.**

Reverting 2 foreach loops within Pose and FoldTree as requested by Christopher Miles. These changes led to a disruption of Qi for some developers.

```

M rosetta_source/src/core/kinematics/FoldTree.cc
M rosetta_source/src/core/pose/Pose.cc
2011-10-27.
Upon Andrew's behest I am rolling back changes from 45556 and 45540 which replaced
'for' with 'foreach'. 'foreach' is being kept in code written by Meilerlab members
(including 'features', 'orbitals', and 'sdf' code), as well as parser code (as Sarel
requested). No integration test changes expected
M
rosetta_source/src/protocols/constraints_additional/SequenceCouplingConstraint.cc
M rosetta_source/src/basic/Tracer.cc
M rosetta_source/src/protocols/moves/AddCavitiesMover.cc
M rosetta_source/src/protocols/filters/PackerNeighborGraphFilter.cc
M rosetta_source/src/protocols/docking/metrics.cc
M rosetta_source/src/protocols/docking/DockingPrepackProtocol.cc
M rosetta_source/src/protocols/docking/DockingLowRes.cc
M rosetta_source/src/protocols/docking/DockingHighResLegacy.cc
M rosetta_source/src/protocols/docking/DockingEnsemblePrepackProtocol.cc
M rosetta_source/src/protocols/moves/MetropolisHastingsMover.cc
M rosetta_source/src/core/conformation/ResidueKinWriter.cc
M rosetta_source/src/protocols/moves/MetricRecorder.cc
M rosetta_source/src/protocols/moves/InterfaceAnalyzerMover.cc
M rosetta_source/src/protocols/docking/DockTaskFactory.cc
M
rosetta_source/src/protocols/constraints_additional/AmbiguousMultiConstraint.cc
M rosetta_source/src/protocols/docking/DockMinMover.cc
M rosetta_source/src/protocols/filters/ShapeComplementarityFilter.cc
2011-10-26.
Fixing a really stupid bug that broke the clang build but didn't show up elsewhere
M rosetta_source/src/protocols/jd2/parser/FragSetLoader.cc
2011-10-26.
Rolling back change to adduct.cc which broke the dna_interface_design integration
test. Sorry for that :\. A few changes to database related stuff. Also, replaced
iterators with the lovely boost FOREACH. The following integration test changes are
expected:
dna_interface_design
features
database_jd2_io
M rosetta_source/src/protocols/jd2/parser/ScoringGridLoader.cc
M
rosetta_source/src/protocols/constraints_additional/SequenceCouplingConstraint.cc
M rosetta_source/src/core/import_pose/pose_stream/PDBPoseInputStream.cc
M rosetta_source/src/protocols/jd2/AtomTreeDiffJobInputter.cc
M rosetta_source/src/core/import_pose/atom_tree_diffs/atom_tree_diff.hh
M rosetta_source/src/protocols/filters/BasicFilters.cc
M rosetta_source/src/protocols/moves/AddCavitiesMover.cc
M rosetta_source/src/protocols/docking/metrics.cc
M rosetta_source/src/protocols/jd2/parser/TaskOperationLoader.cc
M rosetta_source/src/protocols/moves/MinMover.cc
M rosetta_source/src/core/chemical/ResidueTypeSet.cc
M rosetta_source/src/core/pose/datacache/cacheable_observers.cc
M rosetta_source/src/core/scoring/rms_util.tpl.hh
M rosetta_source/src/protocols/jd2/parser/FragSetLoader.cc
M rosetta_source/src/protocols/filters/Filter.cc

```

```

M rosetta_source/src/core/kinematics/FoldTree.cc
M rosetta_source/src/protocols/moves/MetropolisHastingsMover.cc
M rosetta_source/src/core/import_pose/atom_tree_diffs/atom_tree_diff.cc
M rosetta_source/src/protocols/moves/KinematicMover.cc
M rosetta_source/src/protocols/features/HBondFeatures.cc
M rosetta_source/src/protocols/moves/InterfaceAnalyzerMover.cc
M rosetta_source/src/core/chemical/adduct_util.cc
M rosetta_source/src/protocols/filters/PackerNeighborGraphFilter.cc
M rosetta_source/src/protocols/jd2/parser/ScoreFunctionLoader.cc
M rosetta_source/src/protocols/docking/DockingPrepackProtocol.cc
M rosetta_source/src/protocols/docking/DockingLowRes.cc
M rosetta_source/src/protocols/jd2/AtomTreeDiffJobInputter.hh
M rosetta_source/src/protocols/moves/ParsedProtocol.cc
M rosetta_source/src/protocols/docking/DockingHighResLegacy.cc
M rosetta_source/src/protocols/docking/DockingEnsemblePrepackProtocol.cc
M rosetta_source/src/protocols/moves/MetricRecorder.cc
M rosetta_source/src/core/import_pose/import_pose.cc
M rosetta_source/src/protocols/jd2/parser/MonteCarloLoader.cc
M rosetta_source/src/protocols/jd2/DockDesignParser.cc
M rosetta_source/src/protocols/docking/DockTaskFactory.cc
M rosetta_source/src/core/pose/Pose.fwd.hh
M
rosetta_source/src/protocols/constraints_additional/AmbiguousMultiConstraint.cc
  M rosetta_source/src/protocols/filters/ShapeComplementarityFilter.cc
  M rosetta_source/src/protocols/docking/DockMinMover.cc

```

#### 2011-10-25.

3 integration tests are expected to fail:

- \* features
- \* database\_jd2\_io
- \* ligand\_dock\_script

This commit changes the schema of the database slightly, and fixes some database bugs. Score types are now added to the database with protocol information. Also, bugs were found in a few for loops using iterators. These were replaced with foreach loops. Other non-buggy for loops with iterators were replaced with foreach loops because they are so wonderful.

```

M rosetta_source/src/protocols/features/StructureFeatures.cc
M rosetta_source/src/core/chemical/sdf/sdf_parser.cc
M rosetta_source/src/protocols/moves/ReportToDB.cc
M rosetta_source/src/protocols/features/ProtocolFeatures.cc
M rosetta_source/src/protocols/features/JobDataFeatures.cc
A rosetta_source/src/protocols/features/ScoreTypeFeatures.fwd.hh
M rosetta_source/src/core/chemical/adduct_util.hh
M rosetta_source/src/core/chemical/ResidueTypeSet.cc
M rosetta_source/src/protocols/features/FeaturesReporter.hh
M rosetta_source/src/protocols/features/DatabaseStatements.cc
M rosetta_source/src/core/conformation/Conformation.cc
A rosetta_source/src/protocols/features/ScoreTypeFeatures.hh
M rosetta_source/src/protocols/features/StructureScoresFeatures.hh
M rosetta_source/src/protocols/jd2/DatabaseJobOutputter.cc
M rosetta_source/src/core/chemical/adduct_util.cc
M rosetta_source/src/core/chemical/sdf/ctab_parser.cc
M rosetta_source/src/core/io/pdb/pose_io.cc
M rosetta_source/src/basic/Tracer.cc

```



M rosetta\_source/src/protocols/features/FeaturesReporter.cc  
 M rosetta\_source/src/core/pose/Pose.cc  
 M rosetta\_source/src/protocols/features/ProteinSilentReport.hh  
 M rosetta\_source/src/protocols/features/ProtocolFeatures.hh  
 M rosetta\_source/src/core/chemical/sdf/v3\_parser.cc  
 A rosetta\_source/src/protocols/features/ScoreTypeFeatures.cc  
 M rosetta\_source/src/protocols/features/StructureScoresFeatures.cc  
 M rosetta\_source/src/protocols/features/OrbitalsFeatures.cc  
 M rosetta\_source/src/core/chemical/residue\_io.cc  
 M rosetta\_source/src/core/conformation/ResidueKinWriter.cc  
 M rosetta\_source/src/protocols.src.settings  
 M rosetta\_source/src/protocols/features/ResidueTypesFeatures.cc  
 M rosetta\_source/src/core/conformation/Residue.cc  
 M rosetta\_source/src/core/chemical/sdf/MolData.cc  
 M rosetta\_source/src/basic/Emitter.cc  
 M rosetta\_source/src/core/chemical/sdf/mol\_writer.cc  
 M rosetta\_source/src/protocols/features/ProteinSilentReport.cc  
 M rosetta\_source/src/protocols/features/DatabaseFilters.cc

**2011-10-20.**

Replacing for loops with BOOST\_FOREACH uncovered a few bugs. 1 Integration test expected (ligand\_dock\_script)

M rosetta\_source/src/protocols/ligand\_docking/LigandArea.fwd.hh  
 M rosetta\_source/src/protocols/ligand\_docking/StartFrom.cc  
 M rosetta\_source/src/protocols/ligand\_docking/InterfaceBuilder.hh  
 M rosetta\_source/src/protocols/ligand\_docking/HighResDocker.cc  
 M rosetta\_source/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.cc  
 M rosetta\_source/src/protocols/ligand\_docking/LigandDesign.cc  
 M rosetta\_source/src/protocols/ligand\_docking/GrowLigand.cc  
 M rosetta\_source/src/protocols/ligand\_docking/ResidueTorsionRestrains.cc  
 M rosetta\_source/src/protocols/ligand\_docking/Rotate.cc  
 M rosetta\_source/src/protocols/ligand\_docking/LigandArea.hh  
 M rosetta\_source/src/protocols/ligand\_docking/CompoundTranslate.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/GridManager.cc  
 M rosetta\_source/src/protocols/ligand\_docking/InterfaceScoreCalculator.cc  
 M rosetta\_source/src/protocols/ligand\_docking/LigandDockingLoaders.cc  
 M rosetta\_source/src/protocols/ligand\_docking/InterfaceBuilder.cc  
 M rosetta\_source/src/protocols/ligand\_docking/ligand\_options/Interface.cc  
 M rosetta\_source/src/protocols/ligand\_docking/AddHydrogens.cc  
 M rosetta\_source/src/protocols/ligand\_docking/MoveMapBuilder.cc  
 M rosetta\_source/src/protocols/ligand\_docking/ligand\_scores.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/GridManager.hh

**2011-10-19.**

Refactored ProteinSilentReport and ProteinSilentReport\_util.

Now the ProteinSilentReport apply method has a "database filter".

There is a database filter base class and child classes which include:

TopPercentOfEachInput

TopPercentOfAllInputs

TopCountOfEachInput

TopCountOfAllInputs

Instead of many different flags file options, you now just use the

out:database\_filter <FilterName> option. Database filters in turn call upon functions in DatabaseStatements.hh to access the database. I also added an error check to

from\_string in string\_util.hh. No integration tests fail

D rosetta\_source/src/protocols/features/ProteinSilentReport\_util.hh  
 M rosetta\_source/src/protocols/features/JobDataFeatures.cc  
 M rosetta\_source/src/protocols/features/ProteinSilentReport.hh  
 A rosetta\_source/src/protocols/features/DatabaseFilters.hh  
 M rosetta\_source/src/basic/options/options\_rosetta.py  
 A rosetta\_source/src/protocols/features/DatabaseStatements.cc  
 M rosetta\_source/doc/options.dox  
 M rosetta\_source/src/utility/string\_util.hh  
 M rosetta\_source/src/basic/options/keys/OptionKeys.cc.gen0.hh  
 M rosetta\_source/src/basic/options/keys/OptionKeys.cc.gen1.hh  
 M rosetta\_source/src/basic/options/keys/out.OptionKeys.gen.hh  
 M rosetta\_source/src/basic/options/keys/OptionKeys.cc.gen2.hh  
 M rosetta\_source/src/basic/options/keys/OptionKeys.cc.gen3.hh  
 M rosetta\_source/src/protocols.src.settings  
 M rosetta\_source/src/protocols/ligand\_docking/MinimizeBackbone.cc  
 A rosetta\_source/src/protocols/features/DatabaseStatements.hh  
 A rosetta\_source/src/protocols/features/DatabaseFilters.fwd.hh  
 D rosetta\_source/src/protocols/features/ProteinSilentReport\_util.cc  
 M rosetta\_source/src/basic/options/option.cc.gen.hh  
 M rosetta\_source/src/protocols/features/ProteinSilentReport.cc  
 A rosetta\_source/src/protocols/features/DatabaseFilters.cc

**2011-10-11.**

Now the database inputter reads in the string, string\_string, and string\_real data. When extracting poses from a database to PDB we should keep this extra data.

M rosetta\_source/src/protocols/features/JobDataFeatures.cc  
 M rosetta\_source/src/protocols/features/PoseConformationFeatures.cc  
 M rosetta\_source/src/protocols/features/ProteinSilentReport.cc  
 M rosetta\_source/src/protocols/features/JobDataFeatures.hh

**2011-10-10.**

Modification to the features report\_features was necessary to fix sqlite database IO problems. Complete residue type info (including variant type) was not being stored in the DB, so residues could not be built properly when reading in structs from the DB. No integration test changes expected.

M rosetta\_source/src/protocols/qsar/scoring\_grid/HbaGrid.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/polarizGrid.hh  
 M rosetta\_source/src/core/pose/Pose.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/HbdGrid.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/VdwGrid.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/RepGrid.hh  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/AtrGrid.hh  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/HbaGrid.hh  
 M rosetta\_source/src/protocols/ligand\_docking/Translate.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/HbdGrid.hh  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/VdwGrid.hh  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/GridBase.cc  
 M rosetta\_source/src/protocols/features/PoseConformationFeatures.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/polarizGrid.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/GridBase.hh  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/AtrGrid.cc  
 M rosetta\_source/src/protocols/qsar/scoring\_grid/RepGrid.cc

**2011-09-13.**

Sorry everyone, I forgot to add this file on my last commit. See commit message 44655

A rosetta\_source/src/protocols/qsar/qsarMoverCreator.hh

**2011-09-12.**

Integration test changes expected for:

ligand\_dock\_script

ligand\_dock\_grid

#1. Grid functions that place ligands in low-res docking now allow ligands to see other ligands so they don't land on top of each other.

#2. The grid manager is now controlled through Rosetta Scripts more whole-heartedly

M rosetta\_source/src/protocols/qsar/scoring\_grid/HbdGridCreator.hh

M rosetta\_source/src/protocols/jd2/parser/ScoringGridLoader.cc

M rosetta\_source/src/protocols/ligand\_docking/Transform.hh

M rosetta\_source/src/protocols/qsar/scoring\_grid/HbdGrid.cc

M rosetta\_source/src/protocols/ligand\_docking/grid\_functions.hh

M rosetta\_source/src/protocols/qsar/qsarMap.hh

M rosetta\_source/src/protocols/qsar/scoring\_grid/VdwGrid.cc

M rosetta\_source/src/protocols/qsar/qsarMover.cc

M rosetta\_source/src/protocols/ligand\_docking/Translate.cc

M rosetta\_source/src/protocols/ligand\_docking/Rotate.cc

M rosetta\_source/src/protocols/qsar/scoring\_grid/polarizGridCreator.hh

M rosetta\_source/src/protocols/qsar/scoring\_grid/AtrGridCreator.hh

M rosetta\_source/src/protocols/qsar/scoring\_grid/GridFactory.hh

M rosetta\_source/src/protocols/ligand\_docking/Transform.cc

M rosetta\_source/src/protocols/qsar/scoring\_grid/GridManager.hh

M rosetta\_source/src/protocols/ligand\_docking/grid\_functions.cc

M rosetta\_source/src/protocols/qsar/scoring\_grid/RepGrid.cc

M rosetta\_source/src/protocols/ligand\_docking/InterfaceScoreCalculator.hh

M rosetta\_source/src/protocols/qsar/scoring\_grid/HbaGrid.cc

M rosetta\_source/src/protocols/qsar/scoring\_grid/VdwGridCreator.hh

M rosetta\_source/src/protocols/ligand\_docking/ligand\_scores.hh

M rosetta\_source/src/apps/public/idealize.cc

M rosetta\_source/src/protocols/qsar/scoring\_grid/GridFactory.cc

M rosetta\_source/src/protocols/qsar/scoring\_grid/GridCreator.hh

M rosetta\_source/src/protocols/ligand\_docking/CompoundTranslate.cc

M rosetta\_source/src/protocols/ligand\_docking/SlideTogether.cc

M rosetta\_source/src/protocols/qsar/qsarMover.hh

M rosetta\_source/src/protocols/qsar/scoring\_grid/GridManager.cc

M rosetta\_source/src/core/grid/CartGrid.hh

M rosetta\_source/src/protocols/ligand\_docking/InterfaceScoreCalculator.cc

M rosetta\_source/src/protocols/ligand\_docking/Translate.hh

M rosetta\_source/src/protocols/ligand\_docking/Rotate.hh

M rosetta\_source/src/protocols/qsar/scoring\_grid/polarizGrid.cc

M rosetta\_source/src/protocols/qsar/scoring\_grid/RepGridCreator.hh

M rosetta\_source/src/protocols/ligand\_docking/ligand\_scores.cc

M rosetta\_source/src/protocols/qsar/scoring\_grid/HbaGridCreator.hh

M rosetta\_source/src/protocols/qsar/scoring\_grid/AtrGrid.cc

**2011-08-09.**

Commenting out #includes of files in "src/protocols/jobdist" that are not being used.

Also removing "using namespace protocols::jobdist" so it is easier to deprecate jobdist stuff

M mini/src/apps/benchmark/benchmark.cc

M mini/src/devel/cycpep/CycPepMover.hh

M mini/src/apps/pilot/lemmon/protonatePack.cc

M mini/src/devel/denovo\_protein\_design/CreateStartingStructureMover.cc

M mini/src/apps/benchmark/Design.bench.hh  
M mini/src/devel/denovo\_protein\_design/util.cc  
M mini/src/apps/benchmark/ShearMover.bench.hh  
M mini/src/apps/public/rosetta\_scripts/rosetta\_scripts.cc  
M mini/src/apps/public/scenarios/ca\_to\_allatom.cc  
M mini/src/apps/benchmark/SmallMover.bench.hh  
M mini/src/apps/public/antibody/antibody\_mode.cc  
M mini/src/apps/public/match/gen\_lig\_grids.cc  
M mini/src/apps/public/enzdes/enzyme\_design.cc  
M mini/src/apps/benchmark/LigandDockScript.bench.hh  
M mini/src/apps/public/match/gen\_apo\_grids.cc  
M mini/src/apps/public/comparative\_modeling/score\_aln.cc  
M mini/src/core/io/raw\_data/RawStruct.cc  
M mini/src/apps/public/analysis/score.cc  
M mini/src/apps/public/flexpep\_docking/FlexPepDocking.cc  
M mini/src/apps/benchmark/LigandDock.bench.hh

**2011-07-19.**

The ligand\_dock\_script has been failing with very small differences in the 12th decimal place. For now I "fixed" the problem by outputting PDBs instead of atom\_tree\_diffs. I haven't really worked with atom\_tree\_diff code much, so I'm not sure what was going on there. Sorry for the inconvenience.

M mini/test/integration/tests/ligand\_dock\_script/flags  
M mini/test/integration/tests/ligand\_dock\_script/command

**2011-07-11.**

another instance where I named a score function with the same name as a default score function RosettaScripts was ignoring my score function.

M mini/test/integration/tests/ligand\_dock\_script/ligand\_dock.xml

**2011-07-11.**

loading unbound rotamers to improve ligand\_dock\_scripts test results. No integration test changes expected. Also changing the name of my score function. Turns out that RosettaScripts SILENTLY ignores user-specified score functions if the name is the same as one of its pre-programmed score functions.

M mini/test/scientific/cluster/ligand\_dock\_scripts/rotate.xml  
M mini/test/scientific/cluster/ligand\_dock\_scripts/translate\_rotate.xml

**2011-07-07.**

Rocco pointed out a mistake I made in my last commit. I unintentionally replaced ligand\_soft\_rep with soft\_rep in the scoring functions specified in a few XML scripts. The "ligand\_dock\_script" integration test is expected to change.

M mini/test/scientific/cluster/ligand\_dock\_scripts/rotate.xml  
M mini/test/integration/tests/ligand\_dock\_script/ligand\_dock.xml  
M mini/test/scientific/cluster/ligand\_dock\_scripts/translate\_rotate.xml

**2011-07-06.**

This commit brings score functions used by these tests inline with those used in the original ligand\_dock code. Reweight tags are used in the XML to accomplish this. Only ligand\_dock\_script and ligand\_dock\_grid are expected to change.

M mini/test/scientific/cluster/ligand\_dock\_scripts/rotate.xml  
M mini/test/integration/tests/ligand\_dock\_script/ligand\_dock.xml  
M mini/test/integration/tests/ligand\_dock\_grid/ligand\_dock.xml  
M mini/test/scientific/cluster/ligand\_dock\_scripts/translate\_rotate.xml

**2011-06-16.**

Removing "-multiple\_processes\_writing\_to\_one\_directory", added by Sergey, which I believe is causing only 100 structs to be output amongst 6 files even though what I want is 100 output structs per file

M mini/test/scientific/cluster/ligand\_dock\_scripts/analyze.py

M mini/test/scientific/cluster/ligand\_dock\_scripts/submit.py

**2011-06-01.**

Updating ligand\_dock\_script scientific test to match latest code

No changes except in the scientific test results for "ligand\_dock\_script"

M mini/test/scientific/cluster/ligand\_dock\_scripts/rotate.xml

M mini/test/scientific/cluster/ligand\_dock\_scripts/translate\_rotate.xml

**2011-05-12.**

Adding ElementSets which are handled by the ChemicalManager. Element symbols correspond to AtomType names. Also, adding a molecular\_weight\_ to ResidueType. Database files were just committed. Make sure you update your database as well.

M mini/test/core/chemical/automorphism.cxxtest.hh

M mini/src/core.2.src.settings

A mini/src/core/chemical/ElementSet.fwd.hh

M mini/test/protocols/enzdes/PredockRotCenter.cxxtest.hh

M mini/test/protocols/enzdes/MatchConstraintFileInfo.cxxtest.hh

A mini/src/protocols/ligand\_docking/HBondDonorFilter.hh

M mini/src/core/chemical/ChemicalManager.cc

A mini/src/protocols/ligand\_docking/AtomCountFilter.cc

A mini/src/core/chemical/Element.hh

A mini/src/protocols/ligand\_docking/HBondAcceptorFilter.cc

M mini/test/protocols/enzdes/LigInterfaceConstraints.cxxtest.hh

M mini/test/protocols/enzdes/HBondCalculatorsTest.cxxtest.hh

M mini/test/protocols/match/RigidLigandBuilder.cxxtest.hh

M mini/src/core/chemical/ResidueType.hh

M mini/src/core/scoring/constraints/ResidueTypeConstraint.cc

M mini/src/core/pose/util.cc

M mini/src/core/chemical/ResidueTypeSet.cc

A mini/src/core/chemical/ElementSet.hh

A mini/src/protocols/ligand\_docking/HBondDonorFilterCreator.hh

M mini/test/protocols/ligand\_docking/ResidueTorsionRestrains.cxxtest.hh

A mini/src/protocols/ligand\_docking/HBondDonorFilter.cc

M mini/src/protocols/ligand\_docking/HeavyAtomFilter.cc

M mini/test/protocols/enzdes/ResfileAutoRemap.cxxtest.hh

M mini/src/core/chemical/sdf/mol\_parser.hh

A mini/src/core/chemical/Element.cc

M mini/src/core/scoring/constraints/FourPointsFunc.cc

A mini/src/protocols/ligand\_docking/AtomCountFilterCreator.hh

A mini/src/protocols/ligand\_docking/HBondAcceptorFilterCreator.hh

M mini/test/protocols/match/LigandConformer.cxxtest.hh

M mini/test/core/scoring/rms\_util.cxxtest.hh

M mini/test/core/io/silent/symmetric\_binary\_protein\_silent.cxxtest.hh

M mini/src/core/chemical/residue\_io.hh

M mini/src/core/chemical/ResidueType.cc

M mini/src/core/pack/dunbrack/SingleResidueDunbrackLibrary.cc

M mini/src/core/conformation/Residue.hh

M mini/src/protocols/ligand\_docking/HighResDocker.cc

A mini/src/core/chemical/ElementSet.cc

M mini/test/core/io/silent/binary\_protein\_silent.cxxtest.hh

M mini/src/core/chemical/ChemicalManager.hh

A mini/src/protocols/ligand\_docking/AtomCountFilter.hh

A mini/src/protocols/ligand\_docking/HBondAcceptorFilter.hh

M mini/src/core/chemical/sdf/mol\_parser.cc

M mini/test/apps/public/ligand\_docking/ligand\_dock.cxxtest.hh  
M mini/src/core/scoring/constraints/ConstraintFactory.cc  
A mini/src/core/chemical/Element.fwd.hh  
M mini/src/core/scoring/constraints/ResidueTypeConstraint.hh  
M mini/src/core/chemical/residue\_io.cc  
M mini/src/core/pose/util.hh  
M mini/src/core/chemical/ResidueTypeSet.hh  
M mini/src/protocols.src.settings

**2011-05-09.**

Sergey added the "-run::multiple\_processes\_writing\_to\_one\_directory" flag (Rev 41846)  
This led to...

"ERROR: ambiguous, cannot have both -out::overwrite and -  
run::multiple\_processes\_writing\_to\_one\_directory"  
in the ligand\_dock\_scripts scientific benchmark. I have removed the overwrite flag  
M mini/test/scientific/cluster/ligand\_dock\_scripts/flags.txt

**2011-03-02.**

fixing a little bug where I was casting an XML tag as a char instead of a float  
M mini/src/protocols/ligand\_docking/LigandArea.cc

**2011-01-19.**

just a little naming bug, no test changes expected

M

mini/src/protocols/protein\_interface\_design/movers/FavorNonNativeResiduePreCycle.cc

**2011-01-18.**

FoldTree inherits from ReferenceCount, yet "ReferenceCount()" was not found in the  
initializer list for its constructors. This led to a difficult to find bug related to  
my use of FoldTree owner pointers. The bug goes away when I added "ReferenceCount()"  
to the initializer list. Also included in this commit are changes that replace .hh  
with .fwd.hh. No tests are expected to fail.

M mini/src/protocols/ligand\_docking/MinimizeBackbone.cc  
M mini/src/protocols/init.cc  
M mini/src/protocols/ligand\_docking/GrowLigand.hh  
M mini/src/protocols/ligand\_docking/AddHydrogen.cc  
M mini/src/protocols/ligand\_docking/LigandDesign.cc  
M mini/src/protocols/ligand\_docking/AddHydrogens.cc  
M mini/src/core/kinematics/FoldTree.fwd.hh  
M mini/src/core/kinematics/FoldTree.hh  
M mini/src/protocols/ligand\_docking/FinalMinimizer.cc  
M mini/src/protocols/ligand\_docking/GrowLigand.cc  
M mini/src/protocols/ligand\_docking/MinimizeBackbone.hh  
M mini/src/protocols/ligand\_docking/AddHydrogen.hh  
M mini/src/protocols/ligand\_docking/LigandDesign.hh

**2011-01-08.**

adding TetherLigand functionality to ligand docking code, just like the good old  
days.

M mini/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.cc  
M mini/src/protocols/init.cc  
M mini/src/protocols/ligand\_docking/LigandArea.cc  
M mini/src/protocols/ligand\_docking/TetherLigand.cc  
M mini/src/protocols/ligand\_docking/HighResDocker.hh  
M mini/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.hh  
M mini/src/protocols/ligand\_docking/HighResDocker.cc  
M mini/src/protocols/ligand\_docking/LigandArea.hh  
A mini/src/protocols/ligand\_docking/TetherLigand.fwd.hh

D mini/src/protocols/ligand\_docking/TetherLigandCreator.hh  
M mini/src/protocols/ligand\_docking/TetherLigand.hh  
M mini/src/protocols.src.settings

**2011-01-08.**

Thanks to Christopher Miles for fixing my mistake (I added a new file but didn't commit my protocols.src.settings file). Also I didn't commit this new xml script that goes with the integration test. only integration test change expected for ligand\_dock\_script

M mini/test/integration/tests/ligand\_dock\_script/ligand\_dock.xml

**2011-01-07.**

committing new 'LigandArea' class for use with rosetta\_scripts. Define ligand specific features for use with other rosetta\_scripts movers.

ligand\_dock\_script integration test changes expected.

M mini/src/protocols/ligand\_docking/MinimizeBackbone.cc  
M mini/src/protocols/ligand\_docking/LigandDockingLoaders.cc  
M mini/src/protocols/ligand\_docking/InterfaceBuilder.cc  
M mini/src/protocols/init.cc  
M mini/src/protocols/ligand\_docking/MinimizeLigand.hh  
M mini/src/protocols/ligand\_docking/MoveMapBuilder.cc  
M mini/src/protocols/ligand\_docking/FinalMinimizer.cc  
A mini/src/protocols/ligand\_docking/LigandArea.fwd.hh  
M mini/src/protocols/ligand\_docking/LigandDockingLoaders.hh  
M mini/src/protocols/ligand\_docking/InterfaceBuilder.hh  
M mini/src/protocols/ligand\_docking/HighResDockers.cc  
M mini/src/protocols/loophash/LoopHashLibrary.cc  
M mini/src/protocols/ligand\_docking/MoveMapBuilder.hh  
M mini/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.cc  
A mini/src/protocols/ligand\_docking/LigandArea.cc  
D mini/src/protocols/ligand\_docking/MinimizeLigandCreator.hh  
M mini/src/protocols/ligand\_docking/HighResDockers.hh  
M mini/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.hh  
M mini/src/protocols/ligand\_docking/MinimizeLigand.cc  
A mini/src/protocols/ligand\_docking/LigandArea.hh  
M mini/src/protocols/ligand\_docking/LigandDockingLoaderCreators.hh

**2010-12-04.**

fixing the unit tests. sorry for the trouble.

M mini/src/core/pose/PDBInfo.hh  
M mini/src/core/pose/PDBInfo.cc  
M mini/src/core/pose/symmetry/util.cc

**2010-12-03.**

Removing errors introduced by Sergey's virtual inheritance of ReferenceCount

ligand\_dock\_script integration test changes.

M mini/src/protocols/ligand\_docking/MinimizeBackbone.cc  
M mini/src/protocols/ligand\_docking/InterfaceBuilder.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Interface.cc  
M mini/src/protocols/ligand\_docking/MoveMapBuilder.cc  
M mini/src/protocols/ligand\_docking/AddHydrogens.cc  
M mini/src/protocols/ligand\_docking/FinalMinimizer.cc  
M mini/src/protocols/ligand\_docking/HeavyAtomFilter.hh  
M mini/src/protocols/ligand\_docking/RandomConformers.cc  
M mini/src/protocols/ligand\_docking/MinimizeBackbone.hh  
M mini/src/protocols/ligand\_docking/StartFrom.cc  
M mini/src/protocols/ligand\_docking/HighResDockers.cc

M mini/src/protocols/ligand\_docking/InterfaceBuilder.hh  
M mini/src/protocols/ligand\_docking/LigandDockProtocol.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Interface.hh  
M mini/src/protocols/ligand\_docking/AddHydrogens.hh  
M mini/src/protocols/ligand\_docking/MoveMapBuilder.hh  
M mini/src/protocols/ligand\_docking/LigandBaseProtocol.hh  
M mini/src/protocols/ligand\_docking/FinalMinimizer.hh  
M mini/test/integration/tests/ligand\_dock\_script/ligand\_dock.xml  
M mini/src/protocols/ligand\_docking/AddHydrogen.cc  
M mini/src/protocols/ligand\_docking/CompleteConnectionsFilter.hh  
M mini/src/protocols/ligand\_docking/HighResDocker.hh  
M mini/src/protocols/ligand\_docking/Translate.cc  
M mini/src/protocols/ligand\_docking/Rotate.cc  
M mini/src/protocols/ligand\_docking/MinimizeLigand.cc  
M mini/src/protocols/ligand\_docking/CompoundTranslate.cc

**2010-12-03.**

support for Favoring (or disfavoring with a negative score) non-native residues.

M mini/src/protocols/protein\_interface\_design/design\_utils.cc

A

mini/src/protocols/protein\_interface\_design/movers/FavorNonNativeResiduePreCycle.hh

A

mini/src/protocols/protein\_interface\_design/movers/FavorNonNativeResiduePreCycleCreator.hh

A

mini/src/protocols/protein\_interface\_design/movers/FavorNonNativeResiduePreCycle.cc

M mini/src/protocols/protein\_interface\_design/design\_utils.hh

M mini/src/protocols.src.settings

**2010-12-03.**

Adding a constraint to favor changes in residue identity

M mini/src/core/pose/PDBInfo.hh

M mini/src/core.3.src.settings

A mini/src/core/scoring/constraints/NonResidueTypeConstraint.fwd.hh

M mini/src/core/pose/PDBInfo.cc

A mini/src/core/scoring/constraints/NonResidueTypeConstraint.hh

M mini/src/core/pose/symmetry/util.cc

M mini/src/core/scoring/constraints/ResidueTypeConstraint.hh

A mini/src/core/scoring/constraints/NonResidueTypeConstraint.cc

**2010-12-01.**

Now XML scripts that use options that the code does not access will fail with an informative message. 3 integration tests were using options that were not in the code. I removed those options or corrected them in the cases where I could tell what the author had meant to write. tests that change:

hotspot\_graft

place\_simultaneously

ligand\_dock\_script

M mini/test/integration/tests/ligand\_dock\_script/ligand\_dock.xml

M mini/src/protocols/jd2/DockDesignParser.cc

M mini/test/integration/tests/hotspot\_graft/two\_native\_stubs.xml

M mini/test/integration/tests/place\_simultaneously/four\_stubs\_Oct09\_new.xml

M mini/src/utility/tag/Tag.hh

**2010-11-15.**

removing this Scientific/cluster test. Now use ligand\_dock\_scripts

D mini/test/scientific/cluster/multi\_residue\_ligand\_docking



**2010-10-29.**

extract\_atom\_tree\_diffs executable no longer necessary. Instead, just run rosetta\_scripts with an XML file that has nothing in it but a null mover. Use in:file:atom\_tree\_diff. JD2 will choose the AtomTreeDiffInputter and each job will be printed out as a PDB. You can use the tag option as before to specify tags you want to extract. Scores from the atom\_tree\_diff file are appended to the bottom of the PDB.

- M mini/src/protocols/ligand\_docking/InterfaceScoreCalculator.cc
- M mini/src/protocols/init.cc
- M mini/src/protocols/ligand\_docking/ligand\_options/Interface.cc
- M mini/src/apps/pilot/ian/ligdock\_confidence.cc
- M mini/src/protocols/jd2/DockDesignParser.cc
- M mini/src/protocols/ligand\_docking/LigandBaseProtocol.cc
- M mini/src/protocols/ligand\_docking/ligand\_scores.cc
- M mini/src/protocols/ligand\_docking/InterfaceScoreCalculator.hh
- M mini/src/protocols/jd2/JobDistributorFactory.cc
- M mini/src/core/io/atom\_tree\_diffs/atom\_tree\_diff.cc
- M mini/src/protocols/ligand\_docking/HighResDocker.cc
- M mini/src/protocols/jd2/AtomTreeDiffJobInputter.cc
- M mini/src/apps/public/ligand\_docking/extract\_atomtree\_diffs.cc
- M mini/src/apps/benchmark/Docking.bench.hh
- M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.cc
- D mini/src/protocols/ligand\_docking/MultiResidueLigandDock.fwd.hh
- M mini/src/protocols/ligand\_docking/LigandBaseProtocol.hh
- M mini/src/protocols/ligand\_docking/ligand\_scores.hh
- M mini/src/apps/pilot/ian/cluster\_ligand\_poses.cc
- M mini/src/core/io/atom\_tree\_diffs/atom\_tree\_diff.hh
- M mini/src/apps/pilot/ian/select\_best\_unique\_ligand\_poses.cc
- D mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc
- M mini/src/protocols/ligand\_docking/HighResDocker.hh
- M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.hh
- D mini/src/protocols/ligand\_docking/MultiResidueLigandDock.hh

**2010-10-12.**

Adding a show() method to this Constraint.

- M mini/src/core/scoring/constraints/ResidueTypeConstraint.cc
- M mini/src/core/scoring/constraints/ResidueTypeConstraint.hh

**2010-10-06.**

Commenting out about 3456 unused #includes found in the src/apps/benchmark folder.

- M mini/src/apps/benchmark/SmallMover.bench.hh
- M mini/src/apps/benchmark/benchmark.cc
- M mini/src/apps/benchmark/Minimizer.bench.hh
- M mini/src/apps/benchmark/LigandDockScript.bench.hh
- M mini/src/apps/benchmark/Design.bench.hh
- M mini/src/apps/benchmark/ShearMover.bench.hh
- M mini/src/apps/benchmark/score.bench.hh
- M mini/src/apps/benchmark/Docking.bench.hh
- M mini/src/apps/benchmark/LigandDock.bench.hh

**2010-10-06.**

adding performance benchmark for ligand\_dock\_scripts (a rosetta\_script).

No integration test changes expected

- M mini/src/apps/benchmark/benchmark.cc
- A mini/src/apps/benchmark/ligand\_dock/ligand\_dock\_script.xml
- A mini/src/apps/benchmark/LigandDockScript.bench.hh

```

A mini/src/apps/benchmark/ligand_dock/ligand_dock_script_flags.txt
M mini/test/profile/tests/ligand_dock_script/flags
M mini/test/scientific/cluster/ligand_dock_scripts/analyze.py
M mini/test/scientific/cluster/ligand_dock_scripts/flags.txt
M mini/test/profile/tests/ligand_dock_script/ligand_dock.xml
2010-10-05.
updating ligand_dock_script tests. No changes expected
except for the scientific cluster test, "ligand_dock_scripts"
M mini/src/protocols/ligand_docking/InterfaceScoreCalculator.cc
M mini/test/scientific/cluster/ligand_dock_scripts/rotate.xml
M mini/test/scientific/cluster/ligand_dock_scripts/submit.py
M mini/test/scientific/cluster/ligand_dock_scripts/translate_rotate.xml
2010-09-21.
Adding a scriptable mover that adds special ligand docking score terms.
Adding ligand_dock_script tests. "multi_residue_ligand_dock" tests are
removed and ligand_dock_script tests take their place.
A mini/test/scientific/cluster/ligand_dock_scripts/input/2ctc.params
A mini/test/scientific/cluster/ligand_dock_scripts/input/1aql_confs.pdb.gz
M mini/src/protocols/ligand_docking/InterfaceBuilder.cc
A mini/demo/lemmongh/ligand_dock_script/122/AA1.params
A mini/demo/lemmongh/ligand_dock_script/112/ligand_options.txt
A mini/test/scientific/cluster/ligand_dock_scripts/input/1pph_confs.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/native/4tim_6tim.pdb.gz
A mini/demo/lemmongh/ligand_dock_script/111/ligand_dock.xml
A mini/demo/lemmongh/ligand_dock_script/112/ligand_dock.xml
A mini/demo/lemmongh/ligand_dock_script
D mini/src/protocols/ligand_docking/LigandAreaCreator.hh
A mini/test/profile/tests/ligand_dock_script/inputs/ZN1.params
A mini/test/scientific/cluster/ligand_dock_scripts/input/1p8d.params
A mini/test/scientific/cluster/ligand_dock_scripts/yaml/events.py
A mini/test/scientific/cluster/ligand_dock_scripts/input/4tim_6tim.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/yaml/cyaml.py
A mini/test/scientific/cluster/ligand_dock_scripts/targets.py
A mini/test/integration/tests/ligand_dock_script/inputs/7cpa_7cpa_native.pdb
M mini/src/protocols/ligand_docking/StartFrom.hh
M mini/src/protocols/ligand_docking/HighResDocker.hh
A mini/test/scientific/cluster/ligand_dock_scripts/yaml/serializer.py
A
mini/test/scientific/cluster/ligand_dock_scripts/unbound_from_kwk/2prg_1fm9.pdb.gz
A mini/src/protocols/ligand_docking/InterfaceScoreCalculator.fwd.hh
A
mini/test/scientific/cluster/ligand_dock_scripts/unbound_from_kwk/1pq6_1pqc.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/unbound_from_kwk
A mini/test/scientific/cluster/ligand_dock_scripts/translate_rotate.xml
M mini/src/protocols/ligand_docking/MinimizeBackbone.cc
A mini/test/profile/tests/ligand_dock_script
A mini/test/scientific/cluster/ligand_dock_scripts/native/2prg_1fm9.pdb.gz
A mini/test/integration/tests/ligand_dock_script/command
A mini/test/scientific/cluster/ligand_dock_scripts/native/1pq6_1pqc.pdb.gz
A mini/test/profile/tests/ligand_dock_script/inputs/7cpa_confs.pdb
A mini/demo/lemmongh/ligand_dock_script/122/ligand_dock.xml
A mini/test/scientific/cluster/ligand_dock_scripts/input/2prg_1fm9.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/input/1dbj_confs.pdb.gz

```

A mini/test/scientific/cluster/ligand\_dock\_scripts/input/2prg.params  
A mini/demo/lemmongh/ligand\_dock\_script/111/Acetyl.params  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pq6\_1pq6.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1fm9.params  
A mini/test/scientific/cluster/ligand\_dock\_scripts/rotate.xml  
A mini/demo/lemmongh/ligand\_dock\_script/112/1.pdb  
A  
mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/7cpa\_2ctc.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/7cpa\_confs.pdb.gz  
A mini/demo/lemmongh/ligand\_dock\_script/122/flags.txt  
A mini/test/profile/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_input.pdb.gz  
A  
mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1pq6\_1pq6.pdb.gz  
A  
mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/6tim\_4tim.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1dwd\_confs.pdb.gz  
A mini/test/integration/tests/ligand\_dock\_script/inputs/7cpa\_confs.pdb  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1dm2\_confs.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/native/7cpa\_2ctc.pdb.gz  
A  
mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/2dbl\_1dbj.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/composer.py  
A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1pq6\_1pq6.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/native/6tim\_4tim.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1p8d\_confs.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/7cpa\_2ctc.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/\_\_init\_\_.py  
A mini/demo/lemmongh/ligand\_dock\_script/111  
A mini/test/integration/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_input.pdb.gz  
A mini/demo/lemmongh/ligand\_dock\_script/112  
A mini/test/profile/tests/ligand\_dock\_script/command  
A mini/test/scientific/cluster/ligand\_dock\_scripts/native/2dbl\_1dbj.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pq6\_1pq6.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/6tim\_4tim.pdb.gz  
A mini/demo/lemmongh/ligand\_dock\_script/112/proteinAndLigand.pdb.gz  
A mini/test/profile/tests/ligand\_dock\_script/inputs  
A mini/test/scientific/cluster/ligand\_dock\_scripts  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pq6\_confs.pdb.gz  
A  
mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1fm9\_2prg.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/submit  
A mini/test/profile/tests/ligand\_dock\_script/inputs/7cpa\_confs.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/2dbl\_1dbj.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/flags.txt  
M mini/src/protocols/init.cc  
A  
mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1dbj\_2dbl.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1fm9\_2prg.pdb.gz  
A  
mini/test/integration/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_native.pdb.gz  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1ppc.params  
A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pq6.params  
A mini/src/protocols/ligand\_docking/InterfaceScoreCalculator.hh

A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1fm9\_confs.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/4tim\_confs.pdb.gz  
 M mini/src/protocols/ligand\_docking/StartFrom.cc  
 M mini/src/protocols/ligand\_docking/HighResDocker.cc  
 A mini/demo/lemmongh/ligand\_dock\_script/112/README  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1fm9\_2prg.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/6tim\_confs.pdb.gz  
 A mini/demo/lemmongh/ligand\_dock\_script/121  
 A mini/demo/lemmongh/ligand\_dock\_script/122  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1pqc\_1p8d.pdb.gz  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1p8d\_1pqc.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1dbj\_2dbl.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/2dbl\_confs.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/4tim.params  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input  
 A mini/demo/lemmongh/ligand\_dock\_script/122/proteinAndLigand.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/emitter.py  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1dbj\_2dbl.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/2ctc\_confs.pdb.gz  
 A mini/demo/lemmongh/ligand\_dock\_script/111/command  
 A mini/demo/lemmongh/ligand\_dock\_script/112/command  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1dwd\_1dwc.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1pqc\_1p8d.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1p8d\_1pqc.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/7cpa.params  
 A mini/demo/lemmongh/ligand\_dock\_script/112/2.params  
 A mini/demo/lemmongh/ligand\_dock\_script/111/ligand\_design.xml  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/error.py  
 A mini/demo/lemmongh/ligand\_dock\_script/122/ligand\_options.txt  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1p8d\_1pqc.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pqc\_1p8d.pdb.gz  
 A mini/demo/lemmongh/ligand\_dock\_script/README  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1aq1.params  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/dumper.py  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1dwd\_1dwc.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1ppc\_1pph.pdb.gz  
 A mini/test/integration/tests/ligand\_dock\_script/inputs/7cpa\_confs.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/resolver.py  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/submit.py  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1dwd\_1dwc.pdb.gz  
 M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.hh  
 A mini/demo/lemmongh/ligand\_dock\_script/111/flags.txt  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/analyze  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1ppc\_1pph.pdb.gz  
 D mini/src/protocols/ligand\_docking/LigandArea.hh  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1dwc\_confs.pdb.gz  
 A mini/demo/lemmongh/ligand\_dock\_script/121/command

A mini/demo/lemmongh/ligand\_dock\_script/122/command  
 A mini/test/integration/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_input.pdb  
 A mini/demo/lemmongh/ligand\_dock\_script/112/1.params  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1ppc\_1pph.pdb.gz  
 A mini/test/integration/tests/ligand\_dock\_script/inputs/ZN1.params  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pqc\_confs.pdb.gz  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1aq1\_1dm2.pdb.gz  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1dm2\_1aq1.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/parser.py  
 A mini/test/profile/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_input.pdb  
 M mini/src/protocols/ligand\_docking/InterfaceBuilder.hh  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1pph\_1ppc.pdb.gz  
 M mini/doc/apps/public/ligand\_dock.dox  
 A mini/test/integration/tests/ligand\_dock\_script/ligand\_dock.xml  
 A mini/demo/lemmongh/ligand\_dock\_script/112/2.pdb  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1aq1\_1dm2.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1dm2\_1aq1.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/loader.py  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/nodes.py  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1pph\_1ppc.pdb.gz  
 A mini/demo/lemmongh/ligand\_dock\_script/122/AA2.confs  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1dm2\_1aq1.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1aq1\_1dm2.pdb.gz  
 A mini/src/protocols/ligand\_docking/InterfaceScoreCalculator.cc  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pph\_1ppc.pdb.gz  
 A mini/test/profile/tests/ligand\_dock\_script/inputs/7cpa\_7cpa\_native.pdb  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/constructor.py  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1pq6\_1p8d.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1dwd.params  
 A mini/test/integration/tests/ligand\_dock\_script/inputs  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1dm2.params  
 M mini/src/protocols/ligand\_docking/MinimizeBackbone.hh  
 D mini/src/protocols/ligand\_docking/LigandArea.fwd.hh  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pph.params  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/ZN1.params  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/analyze.py  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/native/1pq6\_1p8d.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/yaml/tokens.py  
 A mini/test/integration/tests/ligand\_dock\_script/inputs/7cpa.params  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/2ctc\_7cpa.pdb.gz  
 A mini/demo/lemmongh/ligand\_dock\_script/112/flags.txt  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pq6\_1p8d.pdb.gz  
 A mini/test/scientific/cluster/ligand\_dock\_scripts/input/1pq6.params  
 A  
 mini/test/scientific/cluster/ligand\_dock\_scripts/unbound\_from\_kwk/1dwc\_1dwd.pdb.gz  
 A mini/test/profile/tests/ligand\_dock\_script/flags  
 A mini/test/profile/tests/ligand\_dock\_script/ligand\_dock.xml  
 A mini/test/profile/tests/ligand\_dock\_script/inputs/7cpa.params

```

A mini/test/scientific/cluster/ligand_dock_scripts/input/1ppc_confs.pdb.gz
D mini/demo/lemmongh/multi_ligand_dock
A mini/test/scientific/cluster/ligand_dock_scripts/native/2ctc_7cpa.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/input/1dwc.params
A
mini/test/scientific/cluster/ligand_dock_scripts/unbound_from_kwk/1p8d_1pq6.pdb.gz
A mini/demo/lemmongh/ligand_dock_script/122/AA2.params
A mini/test/scientific/cluster/ligand_dock_scripts/yaml/scanner.py
A mini/test/scientific/cluster/ligand_dock_scripts/input/2prg_confs.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/native/1dwc_1dwd.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/input/2ctc_7cpa.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/yaml/representer.py
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.cc
A mini/test/scientific/cluster/ligand_dock_scripts/yaml/reader.py
A mini/test/scientific/cluster/ligand_dock_scripts/native/1p8d_1pq6.pdb.gz
A mini/test/profile/tests/ligand_dock_script/inputs/7cpa_7cpa_native.pdb.gz
A mini/demo/lemmongh/ligand_dock_script/111/Acetyl.pdb.gz
D mini/src/protocols/ligand_docking/LigandArea.cc
A mini/test/integration/tests/ligand_dock_script/flags
A mini/test/scientific/cluster/ligand_dock_scripts/input/1dwc_1dwd.pdb.gz
A mini/test/scientific/cluster/ligand_dock_scripts/input/6tim.params
A mini/test/scientific/cluster/ligand_dock_scripts/input/1dbj.params
A mini/test/integration/tests/ligand_dock_script
A mini/test/scientific/cluster/ligand_dock_scripts/input/2dbl.params
A mini/demo/lemmongh/ligand_dock_script/122/AA1.confs
A mini/test/scientific/cluster/ligand_dock_scripts/input/1p8d_1pq6.pdb.gz
A mini/src/protocols/ligand_docking/InterfaceScoreCalculatorCreator.hh
A
mini/test/scientific/cluster/ligand_dock_scripts/unbound_from_kwk/4tim_6tim.pdb.gz
M mini/src/protocols.src.settings

```

**2010-08-19.**

This commit adds a multi-arg constructor to Ian's LigandDockProtocol mover so Sergey can add it to PyRosetta

```

M mini/src/protocols/ligand_docking/LigandDockProtocol.hh
M mini/src/protocols/ligand_docking/LigandDockProtocol.cc

```

**2010-08-18.**

demonstration for setting up ligands and proteins for ligand\_docking. Uses Ian davis's ARLS script and ligand\_dock code

```

A mini/demo/ligand_dock/ligand_dock/inputs
A mini/demo/ligand_dock/ligand_setup/ligand.mol2
A mini/demo/ligand_dock/ligand_dock/prep_files/7cpa.pdb
A mini/demo/ligand_dock/ligand_dock/inputs/7cpa_confs.pdb.gz
A mini/demo/ligand_dock/ligand_dock/prep_files
A mini/demo/ligand_dock/ligand_dock/prep_files/ligand.pdb
A mini/demo/ligand_dock/ligand_dock/flags
A mini/demo/ligand_dock/ligand_setup/7cpa_input.pdb
A mini/demo/ligand_dock/ligand_setup/command
A mini/demo/ligand_dock/ligand_dock/inputs/7cpa.params
A mini/demo/ligand_dock/ligand_dock
A mini/demo/ligand_dock/ligand_dock/prep_files/ligand.mdl
A mini/demo/ligand_dock
A mini/demo/ligand_dock/ligand_dock/inputs/7cpa_native.pdb.gz
A mini/demo/ligand_dock/ligand_setup/list.txt

```

```

A mini/demo/ligand_dock/ligand_dock/readme.txt
A mini/demo/ligand_dock/ligand_dock/inputs/ZN1.params
A mini/demo/ligand_dock/ligand_dock/inputs/7cpa_input.pdb.gz
A mini/demo/ligand_dock/ligand_setup/zn.mol2
A mini/demo/ligand_dock/ligand_setup
2010-08-12.
Examples of ligand-centric movers and filters used with rosetta_scripts
M mini/demo/lemmongh/multi_ligand_dock/111/ligand_dock.xml
A mini/demo/rosetta_scripts/ligand_design.xml
A mini/demo/rosetta_scripts/ligand_dock.xml
A mini/demo/lemmongh/multi_ligand_dock/111/ligand_design.xml
D mini/demo/lemmongh/multi_ligand_dock/111/ligand_options.txt
M mini/demo/lemmongh/multi_ligand_dock/111/flags.txt
2010-08-12.
fixing integration test problem I seem to have created. sorry.
M mini/test/integration/integration.py
2010-08-11.
modified integration.py to delete from 'ref' integration tests no longer in 'tests'
M mini/test/integration/integration.py
2010-08-11.
removing link from ligand_dock_7cpa and replacing with files
A mini/test/integration/tests/ligand_dock_7cpa/inputs/7cpa_confs.pdb.gz
A mini/test/integration/tests/ligand_dock_7cpa/inputs/7cpa_7cpa_input.pdb.gz
A mini/test/integration/tests/ligand_dock_7cpa/inputs/7cpa_confs.pdb
A mini/test/integration/tests/ligand_dock_7cpa/inputs/7cpa_7cpa_input.pdb
M mini/test/integration/tests/ligand_dock_7cpa/flags
A mini/test/integration/tests/ligand_dock_7cpa/inputs/ZN1.params
D mini/test/integration/tests/ligand_dock_7cpa/input_link
A mini/test/integration/tests/ligand_dock_7cpa/inputs/7cpa.params
A mini/test/integration/tests/ligand_dock_7cpa/inputs/7cpa_7cpa_native.pdb.gz
A mini/test/integration/tests/ligand_dock_7cpa/inputs/7cpa_7cpa_native.pdb
A mini/test/integration/tests/ligand_dock_7cpa/inputs
2010-08-11.
using embedded scones instead of local scones
M mini/test/run.py
2010-08-11.
use local scones not embedded scones
M mini/test/run.py
2010-08-11.
this test is timing out on the test server even though it works fine
on my local machine. Removing the test for now.
D mini/test/integration/tests/multi_residue_ligand_dock
2010-08-10.
This brings the current ligand dock code in line with what I showed at RosettaCon,
namely, the new RosettaLigand is a collection of movers ran by RosettaScripts.
Only 1 integration test change, multi_residue_ligand_dock. The tests and demos will
be updated soon, and documentation will be added to the RosettaScripts wiki page.
A mini/src/protocols/ligand_docking/FinalMinimizer.fwd.hh
A mini/src/protocols/ligand_docking/InterfaceBuilder.cc
A mini/src/protocols/ligand_docking/FinalMinimizerCreator.hh
M mini/src/protocols/ligand_docking/Translate.hh
D mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.hh
D mini/src/protocols/ligand_docking/ligand_options/Mutate_same_name3.hh

```

A mini/src/protocols/ligand\_docking/MinimizeLigand.hh  
A mini/src/protocols/ligand\_docking/MoveMapBuilder.cc  
A mini/src/protocols/ligand\_docking/LigandAreaCreator.hh  
D mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_ligand.hh  
A mini/src/protocols/ligand\_docking/TetherLigand.hh  
M mini/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.cc  
M mini/src/protocols/ligand\_docking/RandomConformers.hh  
D mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.hh  
D mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.cc  
M mini/src/protocols/ligand\_docking/StartFrom.hh  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc  
A mini/src/protocols/ligand\_docking/HighResDocker.hh  
D mini/src/protocols/ligand\_docking/ligand\_options/Tether\_ligand.hh  
M mini/src/protocols/ligand\_docking/GrowLigand.cc  
M mini/test/integration/tests/multi\_residue\_ligand\_dock/flags  
M mini/src/protocols/ligand\_docking/Rotate.cc  
D mini/src/protocols/ligand\_docking/ligand\_options/BaseOptions.cc  
A mini/src/protocols/ligand\_docking/LigandArea.hh  
A mini/src/protocols/ligand\_docking/TetherLigandCreator.hh  
A mini/src/protocols/ligand\_docking/MinimizeBackbone.cc  
M mini/src/protocols/ligand\_docking/RandomConformersCreator.hh  
A mini/src/protocols/ligand\_docking/FinalMinimizer.cc  
D mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.cc  
A mini/src/protocols/ligand\_docking/InterfaceBuilder.hh  
A mini/src/protocols/ligand\_docking/MoveMapBuilder.hh  
D mini/src/protocols/ligand\_docking/MultiResidueLigandDockCreator.hh  
M mini/src/protocols/ligand\_docking/TranslateCreator.hh  
D mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.fwd.hh  
A mini/src/protocols/ligand\_docking/MinimizeLigand.fwd.hh  
M mini/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.hh  
M mini/src/protocols/ligand\_docking/CompoundTranslate.cc  
A mini/src/protocols/ligand\_docking/SlideTogether.cc  
D mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.cc  
D mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.hh  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.hh  
M mini/src/protocols/ligand\_docking/Rotate.hh  
D mini/src/protocols/ligand\_docking/ligand\_options/BaseOptions.hh  
D mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.fwd.hh  
D mini/src/protocols/ligand\_docking/ligand\_options/Slide\_together.cc  
D mini/src/apps/pilot/lemmon/multi\_residue\_ligand\_dock\_impl.cc  
M mini/src/protocols/jd2/DockDesignParser.cc  
A mini/src/protocols/ligand\_docking/HighResDocker.fwd.hh  
A mini/src/protocols/ligand\_docking/HighResDockerCreator.hh  
A mini/src/protocols/ligand\_docking/MinimizeBackbone.hh  
A mini/src/protocols/ligand\_docking/LigandArea.fwd.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/Interface.hh  
A mini/src/protocols/ligand\_docking/FinalMinimizer.hh  
M mini/src/protocols/ligand\_docking/AddHydrogen.cc  
D mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.hh  
D mini/src/apps/pilot/lemmon/multi\_residue\_ligand\_dock.cc  
A mini/src/protocols/ligand\_docking/MinimizeLigandCreator.hh  
M mini/src/protocols/ligand\_docking/Translate.cc  
M mini/test/integration/tests/multi\_residue\_ligand\_dock/ligand\_dock.xml



```

D mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.cc
D mini/src/protocols/ligand_docking/ligand_options/Mutate_same_name3.cc
A mini/src/protocols/ligand_docking/MinimizeLigand.cc
A mini/src/protocols/ligand_docking/MinimizeBackboneCreator.hh
A mini/src/protocols/ligand_docking/InterfaceBuilder.fwd.hh
M mini/src/pilot_apps.src.settings.all
A mini/src/protocols/ligand_docking/MoveMapBuilder.fwd.hh
M mini/src/protocols/init.cc
D mini/src/protocols/ligand_docking/ligand_options/Minimize_ligand.cc
A mini/src/protocols/ligand_docking/TetherLigand.cc
A mini/src/protocols/ligand_docking/SlideTogether.hh
M mini/src/protocols/ligand_docking/CompoundTranslate.hh
D mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.hh
M mini/src/protocols/ligand_docking/RandomConformers.cc
D mini/src/protocols/ligand_docking/ligand_options/InterfaceBuilder.cc
M mini/src/protocols/ligand_docking/StartFrom.cc
A mini/src/protocols/ligand_docking/HighResDocker.cc
D mini/src/protocols/ligand_docking/ligand_options/Tether_ligand.cc
D mini/src/protocols/ligand_docking/ligand_options/Slide_together.hh
A mini/src/protocols/ligand_docking/LigandArea.cc
D mini/src/protocols/ligand_docking/ligand_options/BaseOptions.fwd.hh
M mini/src/protocols/ligand_docking/LigandDesign.cc
A mini/src/protocols/ligand_docking/SlideTogetherCreator.hh
A mini/src/protocols/ligand_docking/CompoundTranslateCreator.hh
A mini/src/protocols/ligand_docking/MinimizeBackbone.fwd.hh
M mini/src/protocols.src.settings

```

**2010-07-05.**

I've refactored the "low resolution" steps of RosettaLigand so now they are a part of rosetta\_scripts. This is half of the refactoring project. In the end RosettaLigand will just be an XML script, run with the scripter, no executable of its own. Only one integration test failed: multi\_residue\_ligand\_dock

```

A mini/src/protocols/ligand_docking/Translate.hh
A mini/test/scientific/cluster/multi_residue_ligand_docking/rotate.xml
A mini/demo/lemmogh/multi_ligand_dock/122/ligand_dock.xml
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.hh
M mini/src/protocols/ProteinInterfaceDesign/DockDesign.cc
D mini/src/protocols/ligand_docking/ligand_options/Random_conformer.hh
D mini/src/protocols/ligand_docking/ligand_options/Translate.hh
A mini/src/protocols/ligand_docking/DistributionMap.hh
A mini/src/protocols/ligand_docking/RandomConformers.hh
D mini/src/protocols/ligand_docking/ligand_options/Start_from.cc
M mini/src/protocols/ligand_docking/MultiResidueLigandDock.cc
A mini/src/protocols/ligand_docking/StartFrom.hh
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.hh
M mini/test/integration/tests/multi_residue_ligand_dock/flags
A mini/src/protocols/ligand_docking/Rotate.cc
M mini/demo/lemmogh/multi_ligand_dock/112/ligand_options.txt
M mini/demo/lemmogh/multi_ligand_dock/122/ligand_options.txt
M mini/test/profile/tests/multi_residue_ligand_dock/ligand_options.txt
A mini/src/protocols/ligand_docking/RandomConformersCreator.hh
M mini/demo/lemmogh/multi_ligand_dock/112/flags.txt
A

```

mini/test/scientific/cluster/multi\_residue\_ligand\_docking/translate\_rotate.xml

```

A mini/src/protocols/ligand_docking/Translate.fwd.hh
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.fwd.hh
A mini/src/protocols/ligand_docking/TranslateCreator.hh
M mini/test/integration/tests/multi_residue_ligand_dock/ligand_options.txt
M mini/demo/lemmongh/multi_ligand_dock/111/ligand_options.txt
A mini/src/protocols/ligand_docking/CompoundTranslate.cc
D mini/src/protocols/ligand_docking/ligand_options/Start_from.hh
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.cc
M mini/src/protocols/ligand_docking/AddHydrogensCreator.hh
M mini/test/scientific/cluster/multi_residue_ligand_docking/submit.py
M mini/src/protocols/ligand_docking/GrowLigand.hh
D mini/src/protocols/ligand_docking/ligand_options/Rotate.cc
A mini/src/protocols/ligand_docking/Rotate.hh
A mini/src/protocols/ligand_docking/StartFromCreator.hh
M mini/test/integration/tests/multi_residue_ligand_dock/command
M mini/test/scientific/cluster/multi_residue_ligand_docking/dock_options.txt
M mini/test/scientific/cluster/multi_residue_ligand_docking/dock_flags.txt
M mini/demo/lemmongh/multi_ligand_dock/111/command
A mini/src/protocols/ligand_docking/RotateCreator.hh
M mini/demo/lemmongh/multi_ligand_dock/112/command
M mini/src/protocols/ligand_docking/AddHydrogen.cc
M mini/src/apps/pilot/lemmon/multi_residue_ligand_dock.cc
M mini/test/profile/tests/multi_residue_ligand_dock/command
A mini/src/protocols/ligand_docking/Translate.cc
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.cc
A mini/test/integration/tests/multi_residue_ligand_dock/ligand_dock.xml
D mini/src/protocols/ligand_docking/ligand_options/Random_conformer.cc
A mini/demo/lemmongh/multi_ligand_dock/111/ligand_dock.xml
D mini/src/protocols/ligand_docking/ligand_options/Translate.cc
A mini/demo/lemmongh/multi_ligand_dock/112/ligand_dock.xml
M mini/src/protocols/init.cc
M mini/test/profile/tests/multi_residue_ligand_dock/flags
A mini/test/profile/tests/multi_residue_ligand_dock/ligand_dock.xml
A mini/src/protocols/ligand_docking/CompoundTranslate.hh
A mini/src/protocols/ligand_docking/DistributionMap.cc
M mini/src/protocols/ligand_docking/grid_functions.cc
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.hh
A mini/src/protocols/ligand_docking/RandomConformers.cc
M mini/test/scientific/cluster/multi_residue_ligand_docking/no_dock_flags.txt
A mini/src/protocols/ligand_docking/StartFrom.cc
D mini/src/protocols/ligand_docking/ligand_options/Rotate.hh
M mini/src/protocols/ligand_docking/LigandDesignCreator.hh
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.cc
M mini/demo/lemmongh/multi_ligand_dock/111/flags.txt
M mini/src/protocols/ligand_docking/GrowLigandCreator.hh
A mini/src/protocols/ligand_docking/Rotate.fwd.hh
M mini/demo/lemmongh/multi_ligand_dock/121/command
M mini/src/protocols/ligand_docking/RandomConformerMover.cc
M mini/src/protocols.src.settings

```

**2010-06-25.**

4 integration tests fail. multi\_residue\_ligand\_dock integration test failures are meaningful. The other 3 are cosmetic (metalloprotein\_broker, ligand\_dock\_7cpa, fold\_and\_dock) These changes introduce the rudiments of ligand design using Rosetta.

They allow "fragments", which are 1 residue poses with open connections. These connection points are sampled during ligand design. Also multi\_residue\_ligand dock memory footprint should be improved, by using lists of residues instead of poses.

```
A mini/src/protocols/ligand_docking/GrowLigand.hh
M mini/src/protocols/ligand_docking/ligand_options/Rotate.cc
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.hh
M mini/src/protocols/jd2/DockDesignParser.cc
M mini/src/protocols/ligand_docking/ligand_options/Slide_together.cc
A mini/src/protocols/ligand_docking/LigandDesign.fwd.hh
M mini/src/core/scoring/rms_util.hh
A mini/src/protocols/ligand_docking/AddHydrogens.hh
A mini/src/protocols/ligand_docking/AddHydrogen.cc
A mini/src/apps/pilot/lemmon/ligand_design.cc
A mini/src/protocols/filters/HeavyAtomFilter.hh
M mini/src/core/chemical/ResidueType.hh
M mini/src/protocols/ligand_docking/ligand_options/Start_from.cc
M mini/src/protocols/ligand_docking/MultiResidueLigandDock.cc
M mini/src/core/io/pdb/file_data.cc
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.hh
A mini/src/protocols/ligand_docking/GrowLigand.cc
M mini/src/core/scoring/rms_util.tmpl.hh
A mini/src/protocols/ligand_docking/LigandDesign.hh
M mini/src/core/conformation/Conformation.cc
M mini/src/apps/pilot/lemmon/deepMove.cc
A mini/src/protocols/filters/CompleteConnectionsFilter.hh
M mini/src/core/chemical/ResidueConnection.hh
M mini/src/protocols/init.cc
M mini/src/core/scoring/rms_util.cc
A mini/src/protocols/ligand_docking/AddHydrogens.cc
M mini/test/run.py
M mini/src/core/chemical/ResidueType.cc
M mini/src/apps/pilot/lemmon/pack.cc
A mini/src/protocols/filters/HeavyAtomFilter.cc
M mini/src/protocols/ligand_docking/ligand_options/Rotate.hh
A mini/src/protocols/ligand_docking/AddHydrogen.fwd.hh
A mini/src/protocols/ligand_docking/LigandDesignCreator.hh
M mini/demo/lemmongh/multi_ligand_dock/111/flags.txt
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.cc
A mini/src/protocols/ligand_docking/MultiResidueLigandDockCreator.hh
A mini/src/protocols/ligand_docking/GrowLigand.fwd.hh
A mini/src/protocols/ligand_docking/GrowLigandCreator.hh
A mini/src/protocols/ligand_docking/LigandDesign.cc
A mini/src/protocols/filters/CompleteConnectionsFilter.cc
M mini/demo/lemmongh/multi_ligand_dock/111/ligand_options.txt
A mini/src/protocols/ligand_docking/AddHydrogen.hh
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.cc
M mini/src/protocols/ligand_docking/ligand_options/Start_from.hh
A mini/src/protocols/ligand_docking/AddHydrogens.fwd.hh
M mini/src/core/chemical/residue_io.cc
A mini/src/protocols/ligand_docking/AddHydrogensCreator.hh
M mini/src/protocols/ligand_docking/MultiResidueLigandDock.hh
M mini/src/protocols.src.settings
```

2010-05-19.

Updated documentation for ligand\_dock  
M mini/doc/apps/public/ligand\_dock.dox  
**2010-04-28.**  
Changing output levels for scientific benchmarks  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/no\_dock\_flags.txt  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/dock\_flags.txt  
**2010-03-30.**  
lowering output level for tests  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/no\_dock\_flags.txt  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/dock\_flags.txt  
**2010-03-30.**  
need more output in the log files for these scientific tests  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/no\_dock\_flags.txt  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/dock\_flags.txt  
**2010-03-17.**  
now Multiresidue ligand dock will use automorphic RMSD for single residue ligands  
M mini/src/protocols/ligand\_docking/ligand\_scores.hh  
M mini/src/protocols/ligand\_docking/ligand\_scores.cc  
**2010-03-12.**  
Fixed a few bugs in MultiResidueLigandDock.  
Integration test change expected for multi\_residue\_ligand\_dock  
M mini/src/protocols/ligand\_docking/ligand\_options/Interface.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Slide\_together.cc  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc  
M mini/src/protocols/jd2/AtomTreeDiffJobOutputter.hh  
M mini/src/protocols/jd2/AtomTreeDiffJobOutputter.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.cc  
**2010-02-25.**  
fixing cluster test  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/analyze.py  
**2010-02-25.**  
replacing yaml link with yaml copy  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml  
**2010-02-25.**  
replacing link with a copy of yaml files  
D mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml  
**2010-02-25.**  
fixing cluster test  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/analyze  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/analyze.py  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/submit.py  
**2010-02-22.**  
cluster test changes  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/ZN1.params  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/submit.py  
**2010-02-15.**  
MultiResidueLigandDocking cluster test was failing. The overwrite flag hopefully fixes this.  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/no\_dock\_flags.txt  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/analyze.py  
M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/dock\_flags.txt  
**2010-02-02.**  
A few changes to fix the ligand\_docking scientific cluster test

M mini/test/scientific/cluster/ligand\_docking/FLAGS.txt  
 M mini/test/scientific/cluster/ligand\_docking/submit

**2009-11-30.**

Changes to multi\_ligand\_docking scientific benchmark, no integration test changes expected

D mini/test/scientific/cluster/multi\_residue\_ligand\_docking/postprocess.py  
 M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/analyze  
 M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/no\_dock\_flags.txt  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/analyze.py  
 M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/dock\_flags.txt  
 M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/submit.py

**2009-10-15.**

Hope to fix my cluster test

M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/submit

**2009-09-17.**

forgot to svn add...

A mini/src/protocols/jd2/AtomTreeDiffJobInputter.fwd.hh  
 A mini/src/protocols/jd2/AtomTreeDiffJobInputter.hh  
 A mini/src/protocols/jd2/AtomTreeDiffJobInputter.cc

**2009-09-17.**

AtomTreeDiffJobInputter allows user to continue building models starting with an atom\_tree\_diff file. Option in:file:atom\_tree\_diff added. The multi\_residue\_ligand\_dock integration test changes.

M mini/src/core/options/keys/OptionKeys.cc.gen.hh  
 M mini/src/core/options/keys/in.OptionKeys.gen.hh  
 M mini/src/core/io/atom\_tree\_diffs/atom\_tree\_diff.hh  
 M mini/src/protocols/jd2/AtomTreeDiffJobOutputter.hh  
 M mini/doc/options.dox  
 M mini/src/core/io/atom\_tree\_diffs/atom\_tree\_diff.cc  
 M mini/src/protocols/jd2/JobDistributorFactory.cc  
 M mini/src/core/options/option.cc.gen.hh  
 M mini/src/protocols/jd2/JobInputter.fwd.hh  
 M mini/src/core/options/options\_rosetta.py  
 M mini/src/core/options/option.cc.include.gen.hh  
 M mini/src/protocols/jd2/AtomTreeDiffJobOutputter.cc  
 M mini/src/protocols.src.settings

**2009-09-14.**

Changes to multi\_residue\_ligand\_docking scientific benchmark

M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/postprocess.py  
 M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/analyze  
 M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/submit  
 M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/no\_dock\_flags.txt  
 M mini/test/scientific/cluster/multi\_residue\_ligand\_docking/dock\_flags.txt

**2009-09-10.**

Residue's chain\_member is now a core::Size instead of an int Associated casting of ints to core::Sizes were removed. no test changes expected

M mini/src/devel/AnchoredDesign/InterfaceAnalyzerMover.cc  
 M mini/src/protocols/forge/build/ConnectRight.cc  
 M mini/src/core/pose/Pose.hh  
 M mini/src/core/kinematics/AtomTree.cc  
 M mini/src/core/pose/PDBInfo.cc  
 M mini/src/core/conformation/Residue.hh  
 M mini/src/core/io/pdb/pose\_io.cc

M mini/src/core/pose/Pose.cc  
**2009-09-10.**  
Adding profile tests for ligand\_dock and multi\_residue\_ligand\_dock  
A mini/test/profile/tests/ligand\_dock\_7cpa/inputs/7cpa\_confs.pdb  
A mini/test/profile/tests/ligand\_dock\_7cpa/inputs/7cpa\_7cpa\_input.pdb  
D mini/test/integration/tests/ligand\_dock\_7cpa/7cpa\_confs.pdb  
A  
mini/test/integration/tests/multi\_residue\_ligand\_dock/inputs/7cpa\_7cpa\_input.pdb  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/README  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/inputs/ZN1.params  
M mini/test/integration/tests/ligand\_dock\_7cpa/flags  
A  
mini/test/profile/tests/multi\_residue\_ligand\_dock/inputs/7cpa\_7cpa\_native.pdb  
D mini/test/integration/tests/multi\_residue\_ligand\_dock/7cpa\_confs.pdb  
A mini/test/integration/tests/ligand\_dock\_7cpa/inputs/7cpa\_confs.pdb  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/inputs/ZN1.params  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/command  
D mini/test/integration/tests/ligand\_dock\_7cpa/7cpa.params  
M mini/test/integration/tests/multi\_residue\_ligand\_dock/flags  
A mini/test/profile/tests/ligand\_dock\_7cpa/inputs/7cpa\_7cpa\_native.pdb  
D mini/test/integration/tests/ligand\_dock\_7cpa/7cpa\_7cpa\_native.pdb  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/inputs  
A mini/test/integration/tests/ligand\_dock\_7cpa/inputs  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/ligand\_options.txt  
D mini/test/integration/tests/ligand\_dock\_7cpa/7cpa\_7cpa\_input.pdb  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/inputs/7cpa\_confs.pdb  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/flags  
A mini/test/profile/tests/ligand\_dock\_7cpa/inputs/ZN1.params  
D mini/test/integration/tests/ligand\_dock\_7cpa/ZN1.params  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/inputs/7cpa.params  
A mini/test/profile/tests/ligand\_dock\_7cpa/command  
D mini/test/integration/tests/multi\_residue\_ligand\_dock/7cpa.params  
A mini/test/profile/tests/multi\_residue\_ligand\_dock  
A mini/test/integration/tests/ligand\_dock\_7cpa/inputs/7cpa.params  
D mini/test/integration/tests/multi\_residue\_ligand\_dock/7cpa\_7cpa\_native.pdb  
A mini/test/profile/tests/ligand\_dock\_7cpa/inputs  
A mini/test/integration/tests/ligand\_dock\_7cpa/inputs/7cpa\_7cpa\_native.pdb  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/silent.out  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/inputs  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/inputs/7cpa\_confs.pdb  
A mini/test/profile/tests/multi\_residue\_ligand\_dock/inputs/7cpa\_7cpa\_input.pdb  
D mini/test/integration/tests/multi\_residue\_ligand\_dock/7cpa\_7cpa\_input.pdb  
A mini/test/profile/tests/ligand\_dock\_7cpa/flags  
A mini/test/integration/tests/ligand\_dock\_7cpa/inputs/7cpa\_7cpa\_input.pdb  
D mini/test/integration/tests/multi\_residue\_ligand\_dock/ZN1.params  
A mini/test/profile/tests/ligand\_dock\_7cpa/inputs/7cpa.params  
A mini/test/integration/tests/ligand\_dock\_7cpa/inputs/ZN1.params  
A mini/test/profile/tests/ligand\_dock\_7cpa  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/inputs/7cpa.params  
A  
mini/test/integration/tests/multi\_residue\_ligand\_dock/inputs/7cpa\_7cpa\_native.pdb  
A mini/test/profile/tests/ligand\_dock\_7cpa/silent.out  
**2009-09-10.**

Several includes to fix the build.

```
M mini/src/protocols/ligand_docking/ligand_options/Rotate.cc
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.hh
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.cc
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.cc
M mini/src/protocols/ligand_docking/ligand_options/Rotate.hh
```

**2009-09-09.**

This commit replaces .hh with .fwd.hh includes in header files. The ligand\_dock and multi\_residue\_ligand\_dock integration tests fail with with minor numerical differences

```
M mini/src/protocols/ligand_docking/ligand_options/Rotate.cc
M mini/src/protocols/ligand_docking/ResidueTorsionRestrains.hh
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.hh
M mini/src/protocols/ligand_docking/ligand_options/BaseOptions.hh
A mini/src/protocols/ligand_docking/ligand_options/InterfaceBuilder.fwd.hh
M mini/src/protocols/ligand_docking/ligand_options/Random_conformer.hh
M mini/src/protocols/ligand_docking/ligand_options/Translate.hh
M mini/src/protocols/ligand_docking/LigandDockProtocol.cc
M mini/src/protocols/ligand_docking/ligand_options/Minimize_ligand.hh
M mini/src/protocols/ligand_docking/ligand_options/Interface.hh
M mini/test/scientific/cluster/multi_residue_ligand_docking/dock_options.txt
M
mini/src/protocols/ligand_docking/ligand_options/interface_distance_functions.hh
M mini/test/scientific/cluster/multi_residue_ligand_docking/dock_flags.txt
M mini/src/protocols/ligand_docking/UnconstrainedTorsionsMover.cc
M mini/src/protocols/ligand_docking/ligand_options/Start_from.cc
M mini/src/protocols/ligand_docking/ligand_options/InterfaceBuilder.hh
M mini/src/protocols/ligand_docking/ligand_options/Soft_rep.cc
M mini/src/protocols/ligand_docking/ligand_options/Tether_ligand.hh
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.hh
M mini/src/protocols/ligand_docking/ResidueTorsionRestrains.cc
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.cc
M mini/src/protocols/ligand_docking/ligand_options/BaseOptions.cc
M mini/test/protocols/ligand_docking/ResidueTorsionRestrains.cxxtest.hh
M mini/src/protocols/ligand_docking/ligand_options/Random_conformer.cc
M mini/src/protocols/ligand_docking/ligand_options/Translate.cc
M mini/src/protocols/ligand_docking/ligand_options/Minimize_ligand.cc
M mini/src/protocols/ligand_docking/ligand_options/Interface.cc
M
mini/src/protocols/ligand_docking/ligand_options/interface_distance_functions.cc
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.cc
A mini/src/protocols/ligand_docking/UnconstrainedTorsionsMover.fwd.hh
M mini/src/protocols/ligand_docking/ligand_options/InterfaceBuilder.cc
M
mini/test/scientific/cluster/multi_residue_ligand_docking/no_dock_options.txt
M mini/src/protocols/ligand_docking/ligand_options/Rotate.hh
M mini/test/scientific/cluster/multi_residue_ligand_docking/no_dock_flags.txt
M mini/src/protocols/ligand_docking/ligand_options/Tether_ligand.cc
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.cc
A mini/src/protocols/ligand_docking/ResidueTorsionRestrains.fwd.hh
M mini/src/protocols/ligand_docking/LigandDockProtocol.hh
```

```
M mini/src/protocols/ligand_docking/UnconstrainedTorsionsMover.hh
A mini/src/protocols/ligand_docking/ligand_options/Interface.fwd.hh
M mini/src/protocols/ligand_docking/ligand_options/Start_from.hh
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.cc
M mini/src/protocols/ligand_docking/ligand_options/Soft_rep.hh
```

#### 2009-08-27.

This is expected to change the multi\_residue\_ligand\_dock integration test. This commit brings the code up to date with what I showed at RosettaCon, namely (1) support for resfiles, (2) better support for docking multiple ligands simultaneously (the atr/rep grid is aware of other ligands now).

```
M mini/src/protocols/ligand_docking/ligand_options/Translate.cc
M mini/src/protocols/ligand_docking/grid_functions.hh
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.hh
M mini/src/protocols/ligand_docking/MultiResidueLigandDock.cc
M mini/src/protocols/ligand_docking/ligand_scores.cc
M mini/src/protocols/ligand_docking/grid_functions.cc
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.hh
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.hh
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.cc
M mini/src/core/conformation/Residue.hh
M mini/src/protocols/ligand_docking/ligand_options/Translate.hh
M mini/src/protocols/ligand_docking/LigandDockProtocol.cc
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.cc
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.cc
M mini/src/apps/public/design/fixbb.cc
```

#### 2009-06-09.

Fixing broken build due to map::at not supported on old compiler

```
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.cc
```

#### 2009-06-08.

Refactor and Debugging of the multi\_residue\_ligand\_dock code.

Integration tests changes expected for the multi\_residue\_ligand\_dock test.

```
M mini/src/protocols/ligand_docking/ligand_options/Rotate.cc
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_ligand.cc
A mini/src/protocols/ligand_docking/ligand_options/Interface.cc
A
```

```
mini/src/protocols/ligand_docking/ligand_options/interface_distance_functions.cc
```

```
M mini/src/protocols/ligand_docking/grid_functions.cc
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.cc
M mini/src/protocols/ligand_docking/ligand_options/InterfaceBuilder.cc
M mini/src/protocols/ligand_docking/ligand_options/Rotate.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_ligand.hh
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.cc
A mini/src/protocols/ligand_docking/ligand_options/Interface.hh
A
```

```
mini/src/protocols/ligand_docking/ligand_options/interface_distance_functions.hh
```

```
M mini/src/protocols/ligand_docking/grid_functions.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.hh
M mini/src/protocols/ligand_docking/ligand_options/InterfaceBuilder.hh
M mini/src/protocols/ligand_docking/ligand_options/Soft_rep.cc
M mini/src/protocols/ligand_docking/ligand_options/chain_functions.hh
```



M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.hh  
M mini/src/protocols.src.settings

**2009-06-03.**

3 warnings have been removed.

M mini/src/protocols/jd2/AtomTreeDiffJobOutputter.cc

**2009-05-26.**

I hereby comply with Kristian's most noble goal of getting code documented.

May we all find the fortitude to do likewise

A mini/doc/apps/pilot/lemmon

A mini/doc/apps/pilot/lemmon/multi\_residue\_ligand\_dock.dox

M mini/src/pilot\_apps.src.settings.all

**2009-05-24.**

Fixing the multi\_residue\_ligand\_dock integration test by adding the correct jd2 specific egrep -v options

M mini/test/integration/tests/multi\_residue\_ligand\_dock/command

**2009-05-21.**

Adding the ligand\_dock performance benchmark, based on Ian Davis's ligand\_dock code. The OptionCollection was modified so that the ligand\_dock benchmark can read a flags file with ligand\_dock specific options after core::init has been called. Also fixing the atom\_tree\_diffs unit test

M mini/src/apps/benchmark/benchmark.cc

A mini/src/apps/benchmark/extra\_params/ZN1.params

D mini/src/apps/benchmark/ligand\_dock/CP1\_conformers.pdb

M mini/src/utility/options/OptionCollection.hh

A mini/test/core/io/atom\_tree\_diffs

A mini/src/apps/benchmark/extra\_params

M mini/src/apps/benchmark/ligand\_dock/ligand\_dock\_flags.txt

D mini/src/apps/benchmark/ligand\_dock/ZN1.params

A mini/test/core/io/atom\_tree\_diffs/atom\_tree\_diff.cxxtest.hh

A mini/src/apps/benchmark/extra\_params/CP1.params

M mini/performance-benchmark.py

D mini/test/core/io/silent/atomtree\_diff.cxxtest.hh

M mini/test/core.test.settings

D mini/src/apps/benchmark/ligand\_dock/CP1.params

M mini/src/utility/options/OptionCollection.cc

A mini/src/apps/benchmark/extra\_params/CP1\_conformers.pdb

M mini/src/apps/benchmark/LigandDock.bench.hh

**2009-05-20.**

Few files I forgot. Sorry.

A mini/src/protocols/jd2/AtomTreeDiffJobOutputter.fwd.hh

A mini/src/protocols/jd2/AtomTreeDiffJobOutputter.hh

A mini/src/protocols/jd2/AtomTreeDiffJobOutputter.cc

**2009-05-20.**

multi\_residue\_ligand\_dock now uses JD2. AtomTreeDiffJobDistributor uses Ian's atom\_tree\_diff code to produce output. atom\_tree\_diff files were relocated to their own directory/namespace, since they have nothing to do with silent files. A new out:file:atom\_tree\_diff option was created, since out:file:silent shouldn't have multiple meanings.

M mini/src/apps/pilot/ian/redo\_rms\_atomtree\_diffs.cc

M mini/src/protocols/jd2/Job.hh

M mini/src/protocols/ligand\_docking/ligand\_options/Rotate.cc

M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.hh  
M mini/src/apps/pilot/rhiju/rna\_protein\_test.cc  
M mini/src/apps/pilot/lemmon/multi\_residue\_ligand\_dock\_impl.cc  
M mini/src/protocols/ligand\_docking/LigandBaseProtocol.cc  
A mini/src/protocols/ligand\_docking/ligand\_scores.cc  
A mini/src/apps/benchmark/ligand\_dock/ligand\_dock\_flags.txt  
A mini/src/apps/benchmark/ligand\_dock/ZN1.params  
M mini/src/apps/public/ligand\_docking/extract\_atomtree\_diffs.cc  
A mini/src/core/io/atom\_tree\_diffs  
M mini/demo/lemmongh/multi\_ligand\_dock/111/command  
M mini/src/core/options/keys/OptionKeys.cc.gen.hh  
M mini/src/apps/pilot/lemmon/multi\_residue\_ligand\_dock.cc  
M mini/test/core.test.settings  
M mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.cc  
M mini/src/apps/pilot/ian/select\_best\_unique\_ligand\_poses.cc  
A mini/src/core/io/atom\_tree\_diffs/atom\_tree\_diff.hh  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc  
A mini/src/apps/benchmark/ligand\_dock/CP1.params  
M mini/test/integration/tests/multi\_residue\_ligand\_dock/flags  
M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.cc  
M mini/doc/options.dox  
D mini/src/core/io/silent/atomtree\_diff.hh  
M mini/src/core/options/options\_rosetta.py  
A mini/src/apps/benchmark/LigandDock.bench.hh  
M mini/demo/lemmongh/multi\_ligand\_dock/README  
M mini/src/apps/pilot/ian/ligand\_evis.cc  
M mini/src/apps/benchmark/benchmark.cc  
M mini/src/core.src.settings  
M mini/src/core/options/keys/out.OptionKeys.gen.hh  
M mini/src/apps/pilot/ian/ligdock\_confidence.cc  
A mini/src/apps/benchmark/ligand\_dock/CP1\_conformers.pdb  
M mini/src/protocols/ligand\_docking/ligand\_dock\_impl.cc  
A mini/src/core/io/atom\_tree\_diffs/atom\_tree\_diff.cc  
M mini/src/protocols/jd2/JobDistributorFactory.cc  
M mini/src/core/options/option.cc.gen.hh  
M mini/demo/lemmongh/multi\_ligand\_dock/111/flags.txt  
D mini/src/core/io/silent/atomtree\_diff.cc  
M mini/src/protocols/ligand\_docking/LigandBaseProtocol.hh  
A mini/src/protocols/ligand\_docking/ligand\_scores.hh  
A mini/src/apps/benchmark/ligand\_dock  
M mini/src/apps/pilot/ian/cluster\_ligand\_poses.cc  
M mini/src/protocols/jobdist/standard\_mains.cc  
A mini/src/apps/benchmark/ligand\_dock/7cpa\_native.pdb  
M mini/test/apps/public/ligand\_docking/ligand\_dock.cxxtest.hh  
M mini/demo/lemmongh/multi\_ligand\_dock/111/ligand\_options.txt  
M mini/src/core/options/option.cc.include.gen.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.cc  
A mini/src/apps/benchmark/ligand\_dock/7cpa\_input.pdb  
M mini/src/protocols/jobdist/JobDistributors.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.hh  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.hh  
M mini/src/protocols.src.settings

2009-05-09.

These changes fix output file naming problems. For instance myProtein.pdb.gz before would be lead to output such as myProtein.pdb\_0001.pdb.gz. This is fixed now. These changes also changes also allow multiple PDBs to be combined and have a combined output name.

```
M mini/src/protocols/jd2/JobOutputter.cc
M mini/src/protocols/jobdist/Jobs.hh
M mini/src/utility/file/FileName.hh
M mini/src/protocols/jobdist/Jobs.cc
M mini/src/utility/file/FileName.cc
M mini/src/protocols/jd2/JobOutputter.hh
```

#### **2009-05-09.**

A few formatting corrections. Only one ";" per line (unless in a for loop)

```
M mini/src/protocols/enzdes/enzdes_util.hh
M mini/src/core/io/pdb/pdb_dynamic_reader.cc
```

#### **2009-05-07.**

Gets rid of string\_util warning. Refactors pose\_from\_pdb stuff

```
M mini/src/protocols/ligand_docking/ligand_options/Translate.cc
M mini/src/utility/string_util.cc
M mini/src/core/io/pdb/pose_io.hh
M mini/src/core/io/pdb/pose_io.cc
```

#### **2009-05-07.**

These changes allow a FileName to be created from a list of FileNames. This is useful for multiple PDBs combined into one pose.

```
M mini/src/utility/file/file_sys_util.hh
M mini/src/utility/string_util.cc
M mini/src/utility/file/FileName.hh
M mini/src/utility/file/file_sys_util.cc
M mini/src/utility/string_util.hh
```

#### **2009-04-21.**

Now standard mains has been updated to process the in:file:list option

```
M mini/src/protocols/jobdist/standard_mains.cc
```

#### **2009-04-21.**

The JobDistributorFactory now recognizes the new in:file:list option. Use this option to provide files that multiple PDBs on each line. Multiple PDBs on a single line are combined into one pose.

```
M mini/src/protocols/jd2/Job.hh
M mini/src/protocols/jd2/JobDistributorFactory.cc
```

#### **2009-04-20.**

pdb\_from\_pose() now can take either a string that represents one PDB filename or a string of multiple PDB filenames separated by spaces. In the latter case, all PDBs are concatenated and one pose is created from them.

```
M mini/src/core/io/pdb/pose_io.hh
M mini/src/core/io/pdb/pose_io.cc
```

#### **2009-04-20.**

In addition to l and s, start\_files() now parses "list", which is a list of fileNames where each file has PDB filenames on each line. All PDBs on a line are combined to form one pose. This is useful for ligand docking and protein docking, etc.

```
M mini/src/core/options/util.cc
```

#### **2009-04-20.**

Adding a join function for strings. Modifying the slurp function to append, not overwrite. Should not modify integration tests.

```
M mini/src/utility/string_util.cc
M mini/src/utility/string_util.hh
```

**2009-04-17.**

Adding the in:file:list option. in:file:l is now deprecated because in:file:list handles all the functionality of in:file:l with added features. The format of files listed in in:file:list is as follows. Each line contains a list of PDB file names separated by spaces. PDBs on the same line will be combined to form one pose.

```
M mini/src/core/options/keys/OptionKeys.cc.gen.hh
M mini/src/core/options/keys/in.OptionKeys.gen.hh
M mini/src/core/options/option.cc.gen.hh
M mini/src/core/options/options_rosetta.py
```

**2009-04-06.**

fixing a little bug

```
M mini/src/protocols/ligand_docking/MultiResidueLigandDock.cc
```

**2009-04-04.**

a cluster benchmark test for multi\_residue\_ligand\_dock This test replicates the ligand\_docking benchmark. It should produce similar results. This test does not include new multi\_residue functionality or multi\_ligand functionality.

```
A
mini/test/scientific/cluster/multi_residue_ligand_docking/input/1pq6_1ppc.pdb.gz
  A mini/test/scientific/cluster/multi_residue_ligand_docking/yaml/loader.py
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/unbound_from_kwk/2ctc_7cpa.pdb.gz
  A mini/test/scientific/cluster/multi_residue_ligand_docking/yaml
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/unbound_from_kwk/1fm9_2prg.pdb.gz
  A mini/test/scientific/cluster/multi_residue_ligand_docking/analyze
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/native/1ppc_1pph.pdb.gz
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/input/1dwc_confs.pdb.gz
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/input/1aq1_1dm2.pdb.gz
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/input/1dm2_1aq1.pdb.gz
  A mini/test/scientific/cluster/multi_residue_ligand_docking/input/1pq6.params
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/unbound_from_kwk/1dwc_1dwd.pdb.gz
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/input/1pph_1ppc.pdb.gz
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/input/1pqc_confs.pdb.gz
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/native/2prg_1fm9.pdb.gz
  A mini/test/scientific/cluster/multi_residue_ligand_docking/input/1dwc.params
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/unbound_from_kwk/1dbj_2dbl.pdb.gz
  A
mini/test/scientific/cluster/multi_residue_ligand_docking/unbound_from_kwk/1p8d_1pq6.pdb.gz
  A mini/test/scientific/cluster/multi_residue_ligand_docking/input/1ppc.params
```

A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1pq6\_1ppc.pdb.gz  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1ppc.params  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/7cpa\_2ctc.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dbj\_confs.pdb.gz  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/resolver.py  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1p8d\_1ppc.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1ppc\_1p8d.pdb.gz  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/tokens.py  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1ppc\_1pq6.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/6tim\_4tim.pdb.gz  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/4tim.params  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1dm2\_1aq1.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1aq1\_1dm2.pdb.gz  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/6tim.params  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dbj.params  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/submit  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1pph\_1ppc.pdb.gz  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2dbl.params  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/7cpa\_confs.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1pq6\_1p8d.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2dbl\_1dbj.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/4tim\_6tim.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1dwd\_1dwc.pdb.gz  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dwd\_confs.pdb.gz  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/postprocess.py  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/7cpa.params  
   A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2ctc.params  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dm2\_confs.pdb.gz  
   A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/constructor.py

A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/7cpa\_2ctc.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/error.py  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1pqc\_1pq6.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1aq1.params  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/6tim\_4tim.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1p8d\_confs.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1ppc\_1pph.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2ctc\_7cpa.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1fm9\_2prg.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/reader.py  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/2dbl\_1dbj.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1pq6\_1p8d.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1p8d.params  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/cyaml.py  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dwc\_1dwd.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1pq6\_confs.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/2prg\_1fm9.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dbj\_2dbl.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1p8d\_1pq6.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1pq6\_1pqc.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/dock\_options.txt  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/dock\_flags.txt  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1ppc\_confs.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/2ctc\_7cpa.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1fm9\_2prg.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/composer.py  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1pqc\_1p8d.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1p8d\_1pqc.pdb.gz

A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1aq1\_1dm2.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1dm2\_1aq1.pdb.gz  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/scanner.py  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2prg\_confs.pdb.gz  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/dumper.py  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1fm9\_confs.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1dwc\_1dwd.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/4tim\_confs.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1pph\_1ppc.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/representer.py  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/\_\_init\_\_.py  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/ZN1.params  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/6tim\_confs.pdb.gz  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2prg.params  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1dbj\_2dbl.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1p8d\_1pq6.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2dbl\_confs.pdb.gz  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1fm9.params  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/events.py  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/4tim\_6tim.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dwd\_1dwc.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/7cpa\_2ctc.pdb.gz  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/nodes.py  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/emitter.py  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/serializer.py  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2ctc\_confs.pdb.gz  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/no\_dock\_options.txt  
 A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/no\_dock\_flags.txt  
 A  
 mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1pqc\_1pq6.pdb.gz

A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/6tim\_4tim.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1p8d\_1pqc.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1pqc\_1p8d.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1aq1\_confs.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1ppc\_1pph.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/1pq6\_1p8d.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/unbound\_from\_kwk/2dbl\_1dbj.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dwd.params  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1pph\_confs.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1dm2.params  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/4tim\_6tim.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/yaml/parser.py  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/native/1dwd\_1dwc.pdb.gz  
A  
mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/2prg\_1fm9.pdb.gz  
A mini/test/scientific/cluster/multi\_residue\_ligand\_docking/input/1pph.params

**2009-04-04.**  
a change to the multi\_residue\_ligand\_dock integration test is expected.  
M mini/src/protocols/ligand\_docking/ligand\_options/Translate.cc

**2009-03-11.**  
Fixing indentation. Please follow coding guidelines.  
M mini/src/protocols/jd2/SilentFileJobOutputter.cc  
M mini/src/core/chemical/ResidueTypeSet.cc  
M mini/src/protocols/jd2/PDBJobOutputter.cc  
M mini/src/protocols/moves/Mover.hh

**2009-03-03.**  
Getting rid of a warning  
M mini/src/core/io/pdb/pose\_io.hh  
M mini/src/core/io/pdb/pose\_io.cc

**2009-03-02.**  
A scientific\_biweekly test to compare with Ian's ligand dock code. While this code handles multiple ligands, this test only uses single ligands. It is supposed to be just like Ian's scientific test. Results should be similar.  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dwc\_1dwd.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1pq6\_confs.pdb.gz



A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/2prg\_1fm9.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1p8d\_1pq6.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dbj\_2dbl.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1pq6\_1pqc.pdb.gz  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/dock\_options.txt  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/dock\_flags.txt  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1ppc\_confs.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/2ctc\_7cpa.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1fm9\_2prg.pdb.gz  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/composer.py  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1aq1\_1dm2.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1p8d\_1pqc.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1pqc\_1p8d.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1dm2\_1aq1.pdb.gz  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/scanner.py  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/2prg\_confs.pdb.gz  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/dumper.py  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1fm9\_confs.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1dwc\_1dwd.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/4tim\_confs.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1pph\_1ppc.pdb.gz  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/representer.py  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/\_\_init\_\_.py  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/ZN1.params  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/6tim\_confs.pdb.gz  
 A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/2prg.params  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1dbj\_2dbl.pdb.gz  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1p8d\_1pq6.pdb.gz

```

    A mini/test/scientific/biweekly_tests/multi_ligand_docking/yaml/events.py
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/input/1fm9.params
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/input/2dbl_confs.pdb.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/input/4tim_6tim.pdb.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/input/1dwd_1dwc.pdb.gz
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/yaml/nodes.py
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/unbound_from_kwk/7cpa_2ctc.p
db.gz
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/yaml/serializer.py
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/yaml/emitter.py
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/no_dock_options.txt
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/input/2ctc_confs.pdb.gz
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/no_dock_flags.txt
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/unbound_from_kwk/1pqc_1pq6.p
db.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/unbound_from_kwk/6tim_4tim.p
db.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/native/1pqc_1p8d.pdb.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/native/1p8d_1pqc.pdb.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/input/1ppc_1pph.pdb.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/input/1aq1_confs.pdb.gz
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/command
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/unbound_from_kwk/2dbl_1dbj.p
db.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/unbound_from_kwk/1pq6_1p8d.p
db.gz
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/input/1dwd.params
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/input/1pph_confs.pdb.gz
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/input/1dm2.params
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/yaml/parser.py
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/native/4tim_6tim.pdb.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/native/1dwd_1dwc.pdb.gz
    A
mini/test/scientific/biweekly_tests/multi_ligand_docking/input/2prg_1fm9.pdb.gz
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/input/1pph.params
    A mini/test/scientific/biweekly_tests/multi_ligand_docking/README

```

A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1pq6\_1ppc.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/loader.py  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/2ctc\_7cpa.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1fm9\_2prg.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1ppc\_1pph.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1pq6.params  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dm2\_1aq1.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1aq1\_1dm2.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dwc\_confs.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1dwc\_1dwd.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1pph\_1ppc.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1ppc\_confs.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dwc.params  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/2prg\_1fm9.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1dbj\_2dbl.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1p8d\_1pq6.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1ppc.params  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1pq6\_1ppc.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1ppc.params  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/7cpa\_2ctc.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dbj\_confs.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/resolver.py  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1p8d\_1ppc.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1ppc\_1p8d.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/tokens.py  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1ppc\_1pq6.pdb.gz

A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/6tim\_4tim.pdb.gz  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/4tim.params  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1aq1\_1dm2.pdb.gz  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1dm2\_1aq1.pdb.gz  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/6tim.params  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dbj.params  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/7cpa\_confs.pdb.gz  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/2dbl.params  
 A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1pph\_1ppc.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/2dbl\_1dbj.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1pq6\_1p8d.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/4tim\_6tim.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dwd\_confs.pdb.gz  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1dwd\_1dwc.pdb.gz  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/7cpa.params  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/postprocess.py  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/2ctc.params  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1dm2\_confs.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/7cpa\_2ctc.pdb.gz  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/constructor.py  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/error.py  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1aq1.params  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1pqc\_1pq6.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/6tim\_4tim.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1p8d\_confs.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/2ctc\_7cpa.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/unbound\_from\_kwk/1ppc\_1pph.pdb.gz  
   A  
 mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1fm9\_2prg.pdb.gz  
   A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/reader.py

A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/1pq6\_1p8d.pdb.gz  
A  
mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/native/2dbl\_1dbj.pdb.gz  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/input/1p8d.params  
A mini/test/scientific/biweekly\_tests/multi\_ligand\_docking/yaml/cyaml.py  
**2009-03-02.**  
Fixing const correctness. Functions should not return const built-in types. Warning under GCC 4.3.2  
M mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.cc  
**2009-02-28.**  
EXPECTED to BREAK ligand\_dock integration test This change makes pose.split\_by\_chain(chain id) fast  
M mini/src/core/io/pdb/pose\_io.hh  
M mini/src/core/pose/Pose.hh  
M mini/src/core/io/pdb/pose\_io.cc  
M mini/src/core/pose/Pose.cc  
**2009-02-25.**  
Checking for a boundary case. Fixing a logic bug  
M mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.cc  
**2009-02-18.**  
Cleaning up some tracer output  
M mini/src/protocols/ligand\_docking/ligand\_options/Start\_from.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.cc  
**2009-02-17.**  
Just fixing a bug  
M mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.cc  
**2009-02-17.**  
Changing UTracers to debug level  
M mini/src/protocols/ligand\_docking/ligand\_options/Translate.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Rotate.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Start\_from.cc  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Tether\_ligand.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Start\_from.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.cc  
**2009-02-17.**  
Fixing integration test problems, might take one more commit. Sorry.  
M mini/test/integration/tests/multi\_residue\_ligand\_dock/flags  
**2009-02-16.**  
A new integration test for multi\_residue\_ligand docking. This test mimicks Ian's ligand\_docking integration test. If all is well, results should be similar.  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/7cpa\_confs.pdb  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/7cpa\_7cpa\_input.pdb  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/ligand\_options.txt  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/ZN1.params  
A mini/test/integration/tests/multi\_residue\_ligand\_dock/7cpa.params

```
A mini/test/integration/tests/multi_residue_ligand_dock/flags
A mini/test/integration/tests/multi_residue_ligand_dock/README
A mini/test/integration/tests/multi_residue_ligand_dock/7cpa_7cpa_native.pdb
A mini/test/integration/tests/multi_residue_ligand_dock/command
A mini/test/integration/tests/multi_residue_ligand_dock
```

#### 2009-02-16.

Fixing some bugs, const-correct issues, etc.

```
M mini/src/protocols/ligand_docking/ligand_options/Translate.cc
M mini/src/protocols/ligand_docking/ligand_options/Rotate.cc
M mini/src/protocols/ligand_docking/ligand_options/Start_from.cc
M mini/src/protocols/ligand_docking/MultiResidueLigandDock.cc
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.cc
M mini/src/protocols/ligand_docking/ligand_options/Translate.hh
M mini/src/protocols/ligand_docking/ligand_options/Rotate.hh
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.cc
M mini/src/protocols/ligand_docking/ligand_options/Start_from.hh
M mini/src/protocols/ligand_docking/MultiResidueLigandDock.hh
```

#### 2009-02-16.

Adding function to remove all constraints

```
M mini/src/core/pose/Pose.hh
M mini/src/core/pose/Pose.cc
```

#### 2009-02-12.

Fixing error caused by not flushing tracer buffer with endl.

```
M mini/src/core/scoring/constraints/ConstraintSet.cc
```

#### 2009-02-12.

Got rid of map::operator[], using safer map::find() now

```
M mini/src/protocols/ligand_docking/ligand_options/Translate.cc
M mini/src/protocols/ligand_docking/ligand_options/Start_from.cc
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.cc
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.cc
M mini/src/protocols/ligand_docking/ligand_options/Tether_ligand.cc
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.cc
```

#### 2009-02-05.

zipping up my PDBs

```
M mini/demo/lemmongh/multi_ligand_dock/112/command
A mini/demo/lemmongh/multi_ligand_dock/121/command
M mini/demo/lemmongh/multi_ligand_dock/122/command
A mini/demo/lemmongh/multi_ligand_dock/111/Acetyl.pdb.gz
D mini/demo/lemmongh/multi_ligand_dock/111/Acetyl.pdb
A mini/demo/lemmongh/multi_ligand_dock/112/proteinAndLigand.pdb.gz
D mini/demo/lemmongh/multi_ligand_dock/112/proteinAndLigand.pdb
A mini/demo/lemmongh/multi_ligand_dock/122/proteinAndLigand.pdb.gz
D mini/demo/lemmongh/multi_ligand_dock/122/proteinAndLigand.pdb
M mini/demo/lemmongh/multi_ligand_dock/111/flags.txt
M mini/demo/lemmongh/multi_ligand_dock/112/flags.txt
M mini/demo/lemmongh/multi_ligand_dock/README
M mini/demo/lemmongh/multi_ligand_dock/122/flags.txt
```

#### 2009-02-05.

Some examples of how to use my code, more to follow

```
A mini/demo/lemmongh/multi_ligand_dock
A mini/demo/lemmongh/multi_ligand_dock/112/README
A mini/demo/lemmongh/multi_ligand_dock/111
A mini/demo/lemmongh/multi_ligand_dock/121
```

A mini/demo/lemmongh/multi\_ligand\_dock/112  
 A mini/demo/lemmongh/multi\_ligand\_dock/122  
 A mini/demo/lemmongh/multi\_ligand\_dock/111/Acetyl.params  
 A mini/demo/lemmongh/multi\_ligand\_dock/112/1.pdb  
 A mini/demo/lemmongh/multi\_ligand\_dock/112/2.pdb  
 A mini/demo/lemmongh/multi\_ligand\_dock/111/flags.txt  
 A mini/demo/lemmongh/multi\_ligand\_dock/122/AA1.conf  
 A mini/demo/lemmongh/multi\_ligand\_dock/112/flags.txt  
 A mini/demo/lemmongh/multi\_ligand\_dock/122/flags.txt  
 A mini/demo/lemmongh/multi\_ligand\_dock/122/AA2.conf  
 A mini/demo/lemmongh/multi\_ligand\_dock/111/command  
 A mini/demo/lemmongh/multi\_ligand\_dock/112/command  
 A mini/demo/lemmongh/multi\_ligand\_dock/122/command  
 A mini/demo/lemmongh/multi\_ligand\_dock/112/1.params  
 A mini/demo/lemmongh/multi\_ligand\_dock/112/2.params  
 A mini/demo/lemmongh/multi\_ligand\_dock/111/Acetyl.pdb  
 A mini/demo/lemmongh  
 A mini/demo/lemmongh/multi\_ligand\_dock/122/AA1.params  
 A mini/demo/lemmongh/multi\_ligand\_dock/111/ligand\_options.txt  
 A mini/demo/lemmongh/multi\_ligand\_dock/112/proteinAndLigand.pdb  
 A mini/demo/lemmongh/multi\_ligand\_dock/122/AA2.params  
 A mini/demo/lemmongh/multi\_ligand\_dock/112/ligand\_options.txt  
 A mini/demo/lemmongh/multi\_ligand\_dock/122/proteinAndLigand.pdb  
 A mini/demo/lemmongh/multi\_ligand\_dock/122/ligand\_options.txt  
 A mini/demo/lemmongh/multi\_ligand\_dock/README

**2009-02-04.**

Got rid of a several bugs including an earwig, an earth worm, and a dust mite (those ones are really hard to find since they are so small)

M mini/src/protocols/ligand\_docking/ligand\_options/Rotate.cc  
 M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/Rotate.hh  
 M mini/src/protocols/ligand\_docking/ligand\_options/Tether\_ligand.cc

**2009-02-04.**

Fixing some logic

M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.hh  
 M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.cc

**2009-02-03.**

Fixing a few bugs, adhering to coding guidelines better. Const corrected stuff

M mini/src/protocols/ligand\_docking/ligand\_options/Translate.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/Rotate.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.hh  
 M mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_ligand.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/Mutate\_same\_name3.hh  
 M mini/src/protocols/ligand\_docking/LigandBaseProtocol.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/Random\_conformer.hh  
 M mini/src/protocols/ligand\_docking/grid\_functions.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.hh  
 M mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/Translate.hh  
 M mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.cc  
 M mini/src/protocols/ligand\_docking/ligand\_options/Rotate.hh

M mini/src/protocols/ligand\_docking/ligand\_options/Tether\_ligand.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.fwd.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/Start\_from.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.cc  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Tether\_ligand.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Mutate\_same\_name3.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Random\_conformer.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/Start\_from.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.hh  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.hh

**2009-02-03.**  
a little const correctness  
M mini/src/protocols/moves/MinMover.hh

**2009-02-03.**  
a little const correctness  
M mini/src/protocols/geometry/RB\_geometry.hh  
M mini/src/protocols/geometry/RB\_geometry.cc

**2009-02-03.**  
Adding a function to ensure that chemical edges have atom info (this was a bug I had)  
M mini/src/core/kinematics/FoldTree.hh  
M mini/src/core/kinematics/FoldTree.cc

**2009-01-24.**  
Forgot a few files  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc  
M mini/src/protocols/ligand\_docking/MultiResidueLigandDock.hh

**2009-01-24.**  
no comment...  
M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.cc

**2009-01-24.**  
One more try...  
M mini/src/protocols/ligand\_docking/ligand\_options/chain\_functions.cc

**2009-01-24.**  
Removing old file  
D mini/src/protocols/ligand\_docking/ligand\_options/Improve\_orientation.cc

**2009-01-24.**  
Removing old file  
D mini/src/protocols/ligand\_docking/ligand\_options/Improve\_orientation.hh

**2009-01-24.**  
Sorry for breaking the build.  
M mini/src/protocols.src.settings

**2009-01-24.**  
Refactoring  
A mini/src/protocols/ligand\_docking/ligand\_options/Translate.cc  
A mini/src/protocols/ligand\_docking/ligand\_options/Rotate.cc  
M mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.hh  
M mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_ligand.cc



```

M mini/src/protocols/ligand_docking/ligand_options/BaseOptions.hh
M mini/src/protocols/ligand_docking/ligand_options/Slide_together.cc
M mini/src/protocols/ligand_docking/ligand_options/Mutate_same_name3.hh
M mini/src/protocols/ligand_docking/ligand_options/Random_conformer.hh
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.cc
A mini/src/protocols/ligand_docking/ligand_options/Translate.hh
M mini/src/protocols/ligand_docking/ligand_options/InterfaceBuilder.cc
A mini/src/protocols/ligand_docking/ligand_options/Rotate.hh
M mini/src/protocols/ligand_docking/ligand_options/Tether_ligand.cc
M mini/src/protocols/ligand_docking/ligand_options/Minimize_ligand.hh
M mini/src/protocols/ligand_docking/ligand_options/Slide_together.hh
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.fwd.hh
M mini/src/protocols/ligand_docking/ligand_options/Minimize_backbone.hh
M mini/src/protocols/ligand_docking/ligand_options/BaseOptions.fwd.hh
M mini/src/protocols/ligand_docking/ligand_options/Start_from.cc
M mini/src/protocols/ligand_docking/ligand_options/InterfaceBuilder.hh
M mini/src/protocols/ligand_docking/ligand_options/Soft_rep.cc
M mini/src/protocols/ligand_docking/ligand_options/Tether_ligand.hh
M mini/src/protocols/ligand_docking/ligand_options/LigandOptionMap.cc
M mini/src/protocols/ligand_docking/ligand_options/BaseOptions.cc
M mini/src/protocols/ligand_docking/ligand_options/Mutate_same_name3.cc
M mini/src/protocols/ligand_docking/ligand_options/ProtocolOption.cc
M mini/src/protocols/ligand_docking/ligand_options/Random_conformer.cc
M mini/src/protocols/ligand_docking/ligand_options/Soft_rep.hh

```

**2009-01-24.**

These functions would be less stupid if my calls to `pose.conformation().chain_begin(chain_id)` and `pose.conformation().chain_end(chain_id)` returned iterators instead of `core::Size`

```

A mini/src/protocols/ligand_docking/ligand_options/chain_functions.hh
A mini/src/protocols/ligand_docking/ligand_options/chain_functions.cc

```

**2009-01-24.**

Get the attractive and repulsive scores for a "rigid\_body" (a multi-residue ligand in my case), in a less ugly manner

```

M mini/src/protocols/ligand_docking/grid_functions.hh
M mini/src/protocols/ligand_docking/grid_functions.cc

```

**2009-01-24.**

Now one can specify to the rigid body mover how much movement to apply, giving phi and psi angles.

```

M mini/src/protocols/moves/RigidBodyMover.hh
M mini/src/protocols/moves/RigidBodyMover.cc

```

**2009-01-24.**

Now one can specify the range of random reorientation he/she prefers, rather than always 360 degrees

```

M mini/src/protocols/geometry/RB_geometry.hh
M mini/src/protocols/geometry/RB_geometry.cc

```

**2008-09-19.**

Forgot a few #includes

```

M mini/src/protocols/ligand_docking/MultiResidueLigandDock.cc

```

**2008-09-19.**

Oops, forgot a file...

```

M mini/src/protocols.src.settings

```

**2008-09-19.**

New options for my multi-ligand, multi-residue-ligand dock application. Release of the application itself (mostly taken from Ian's ligand dock application).

Addition of this application to pilot\_apps\_all

- M mini/src/core/options/keys/OptionKeys.cc.gen.hh
- A mini/src/apps/pilot/lemmon/multi\_residue\_ligand\_dock.cc
- A mini/src/apps/pilot/lemmon/multi\_residue\_ligand\_dock\_impl.cc
- M mini/src/core/options/option.cc.gen.hh
- M mini/src/core/options/keys/OptionKeys.hh.gen.hh
- M mini/src/core/options/options\_rosetta.py
- M mini/src/pilot\_apps.src.settings.all

**2008-09-19.**

Addition of a method "split\_by\_chain(Size chain\_id) const" which returns a Pose unlike the previous split\_by\_chain, this one doesn't return a vector of Poses from each chain, but just the one that you asked for

- M mini/src/core/pose/Pose.cc
- M mini/src/core/pose/Pose.hh

**2008-09-19.**

This sizeable commit represents the pre-alpha release of the support code for my new application for multi-ligand, multi-ligand-residue docking. This code will (one day) allow the user to dock any number of ligands. Additionally each ligand can be represented as a collection of ligand "residues". Each residue in turn can be represented as a collection of 'rotamers'. Thus ligands can be treated similar to proteins. I loosely followed Ian's ligand\_dock protocol in the creation of this code. The use of this application requires an additional ligand\_options.txt file. I did this because the option flags system gets quite cumbersome if you want to specify many options for different ligands. I wrote my code so that the options that Ian provided are available in my code for each ligand separately. Soon I will provide more documentation concerning how to use this application.

- A mini/src/protocols/ligand\_docking/ligand\_options/Slide\_together.hh
- A mini/src/protocols/ligand\_docking/MultiResidueLigandDock.fwd.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.fwd.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/BaseOptions.fwd.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/Start\_from.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.cc
- A mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.hh
- A mini/src/protocols/ligand\_docking/MultiResidueLigandDock.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/Improve\_orientation.cc
- A mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/BaseOptions.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/Slide\_together.cc
- A mini/src/protocols/ligand\_docking/ligand\_options/Mutate\_same\_name3.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/Random\_conformer.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_ligand.hh
- A mini/src/protocols/ligand\_docking/ligand\_options
- A mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/Start\_from.cc
- A mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/Soft\_rep.cc
- A mini/src/protocols/ligand\_docking/MultiResidueLigandDock.cc
- A mini/src/protocols/ligand\_docking/ligand\_options/Tether\_ligand.hh
- A mini/src/protocols/ligand\_docking/ligand\_options/LigandOptionMap.cc
- A mini/src/protocols/ligand\_docking/ligand\_options/BaseOptions.cc
- A mini/src/protocols/ligand\_docking/ligand\_options/Mutate\_same\_name3.cc

A mini/src/protocols/ligand\_docking/ligand\_options/Random\_conformer.cc  
A mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_ligand.cc  
A mini/src/protocols/ligand\_docking/ligand\_options/ProtocolOption.hh  
A mini/src/protocols/ligand\_docking/ligand\_options/Minimize\_backbone.cc  
A mini/src/protocols/ligand\_docking/ligand\_options/Improve\_orientation.hh  
A mini/src/protocols/ligand\_docking/ligand\_options/InterfaceBuilder.cc  
A mini/src/protocols/ligand\_docking/ligand\_options/Tether\_ligand.cc

**2008-09-19.**

Addition of a constructor that takes a set of ResidueTorsionRestrains instead of a vector of ResidueTorsionRestrains

M mini/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.cc  
M mini/src/protocols/ligand\_docking/UnconstrainedTorsionsMover.hh

**2008-09-19.**

Addition of operator==. Two ResidueTorsionRestrains are == if they share the same residue id.

M mini/src/protocols/ligand\_docking/ResidueTorsionRestrains.hh  
M mini/src/protocols/ligand\_docking/ResidueTorsionRestrains.cc

**2008-09-19.**

Addition of "rigid\_body" methods which act on a rigid body instead of just a residue

M mini/src/protocols/ligand\_docking/grid\_functions.hh  
M mini/src/protocols/ligand\_docking/grid\_functions.cc

**2008-09-19.**

Support for GCC compiler 4.3

M mini/tools/build/options.settings

**2008-08-05.**

I made methods out of several sections of the apply() method, increasing readability. Also I changed jump\_ids from ints to core::Size's

M mini/src/protocols/ligand\_docking/LigandBaseProtocol.hh  
M mini/src/protocols/ligand\_docking/LigandBaseProtocol.cc  
M mini/src/protocols/ligand\_docking/LigandDockProtocol.hh  
M mini/src/protocols/ligand\_docking/LigandDockProtocol.cc

**2008-08-05.**

Now instead of passing a "dummy" vector to centroids\_by\_jump, you can just call upstream\_centroid\_by\_jump or downstream\_centroid\_by\_jump, which return the centroid you want.

M mini/src/protocols/geometry/RB\_geometry.hh  
M mini/src/protocols/geometry/RB\_geometry.cc

**2008-08-01.**

I've added a "design" benchmark. The code does a complete redesign of HIV protease (198 AA), called "design\_in.pdb". It takes 1 min to run on my machine. I wanted to add the -ex1 flag, but I couldn't get that option to work (it is commented out).

M mini/src/apps/benchmark/benchmark.cc  
A mini/src/apps/benchmark/design\_in.pdb  
A mini/src/apps/benchmark/Design.bench.hh

**2008-06-19.**

how to move the ligand randomly or determine protonation state and pack a ligand

A mini/src/apps/pilot/lemmon/randomMove.cc  
A mini/src/apps/pilot/lemmon/protonatePack.cc

**2008-06-19.**

Now my mover translates and rotates

M mini/src/apps/pilot/lemmon/deepMove.cc

**2008-06-19.**

removing some junk

D mini/src/apps/pilot/lemmon/residueGluer.cc

**2008-06-19.**

With the addition of CountPairData\_1\_many, a single residue can be connected to more than 3 other residues.

M mini/src/core/scoring/hackelec/HackElecEnergy.cc  
M mini/src/core/scoring/trie/RotamerTrie.hh  
M mini/src/core/scoring/trie/trie\_vs\_trie.hh  
M mini/src/core/scoring/etable/etrie/TrieCountPairNone.cc  
M mini/src/core/scoring/hbonds/hbtrie/HBCountPairFunction.cc  
M mini/src/core/scoring/etable/etrie/TrieCountPair1BC3.hh  
A mini/src/core/scoring/etable/etrie/CountPairData\_1\_many.hh  
M mini/src/core/scoring/etable/etrie/TrieCountPair1BC4.hh  
M mini/src/core/scoring/etable/BaseEtableEnergy.tmpl.hh  
M mini/src/core/scoring/etable/etrie/TrieCountPairAll.cc  
M mini/src/core/scoring/trie/TrieCountPairBase.cc  
M mini/src/core/scoring/etable/etrie/TrieCountPairNone.hh  
M mini/src/core/scoring/hbonds/hbtrie/HBCountPairFunction.hh  
M mini/src/core/scoring/rna/HackElecEnergy.cc  
M mini/src/core.src.settings  
M mini/src/core/scoring/etable/etrie/TrieCountPairAll.hh  
A mini/src/core/scoring/etable/etrie/CountPairData\_1\_many.fwd.hh  
M mini/src/core/scoring/trie/TrieCountPairBase.hh  
M mini/src/core/scoring/etable/etrie/TrieCountPair1BC3.cc  
M mini/src/core/scoring/trie/RotamerTrieBase.hh  
M mini/src/core/scoring/etable/etrie/TrieCountPair1BC4.cc  
A mini/src/core/scoring/etable/etrie/CountPairData\_1\_many.cc

**2008-06-17.**

Now, use the option -in::file::extra\_res\_path to provide a directory where .params files are kept. Only files ending in .param or .params will be used.

M mini/src/utility/file/file\_sys\_util.hh  
M mini/src/core/options/keys/OptionKeys.cc.gen.hh  
M mini/src/utility/file/file\_sys\_util.cc  
M mini/src/core/chemical/ChemicalManager.cc  
M mini/src/core/options/option.cc.gen.hh  
M mini/src/core/options/keys/OptionKeys.hh.gen.hh  
M mini/src/core/options/options\_rosetta.py

**2008-06-05.**

doxygen

M mini/src/apps/pilot/lemmon/learnMini.dox

**2008-06-05.**

Files for learning simple MiniRosetta stuff

A mini/src/apps/pilot/lemmon/learnMini.dox  
A mini/src/apps/pilot/lemmon/pack.cc

**2008-06-05.**

Simple code to teach miniRosetta

A mini/src/apps/pilot/lemmon/deepMove.cc  
A mini/src/apps/pilot/lemmon/simpleScore.cc  
A mini/src/apps/pilot/lemmon/readAndWrite.cc

**2008-05-29.**

Read and write a PDB with minirosetta

M mini/src/apps/pilot/lemmon/residueGluer.cc

**2008-05-27. 2008-05-20.**

A mini/src/apps/pilot/lemmon/residueGluer.cc

```
M mini/src/core/kinematics/FoldTree.hh  
M mini/src/core/io/pdb/file_data.cc  
M mini/src/core/kinematics/FoldTree.cc  
A mini/src/apps/pilot/lemmon
```