

**Protein Flexibility in Structure-Based Drug Design: Method Development
and Novel Mechanisms for Inhibiting HIV-1 Protease**

by

Kelly Lynn Damm

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Medicinal Chemistry)
in The University of Michigan
2007

Doctoral Committee:

Associate Professor Heather A. Carlson, Chair
Professor Gordon M. Crippen
Professor Shaomeng Wang
Assistant Professor Hashim M. Al-Hashimi
Assistant Professor Jason E. Gestwicki

© Kelly Lynn Damm

All Rights Reserved

2007

ACKNOWLEDGMENTS

I am grateful for the support of many people, some that have been with me from the start and others that I have met along the way. First, I would like to thank my advisor Heather Carlson for her guidance. She has provided invaluable training that will serve me well in my professional career. I would like to thank the members of the Carlson Lab for insightful discussions and a stimulating work environment. In particular, I am indebted to Kristin Meagher and Michael Lerner for taking the time to train me and their assistance with the many computational programs used in the lab. I am also very appreciative of the guidance and suggestions from my dissertation committee.

Furthermore, I would like to thank the Medicinal Chemistry faculty and students. They have all played an integral part in my training at UofM. I am also grateful to Lynn Alexander; she is an invaluable asset to the graduate students and has gone out of her way multiple times to help me resolve a variety of concerns. I have been very fortunate to receive funding for my education through the Pharmacological Sciences Training Program (NIH funded), American Foundation for Pharmaceutical Education Predoctoral Fellowship, CARAT Learning Sciences Graduate Student Instructor Fellowship, Rackham Predoctoral Fellowship, and a Harold and Vivian Shapiro Award.

Finally, I would like to express gratitude to my friends and family for their love and support throughout the last four years. I am so fortunate to be surrounded by a group of such amazing people. Fernanda Burke, Angela Aldrete, and Lauren Wallner have proven to be dedicated and caring friends time and time again. In particular, Fernanda has endured the ups and downs of graduate school along side me, and I will always be

grateful for her advice and friendship. I would like to acknowledge my family, David and Sherri Kraft and Allan and Stephanie McLean, and my brother, Ryan Damm, as they have always offered unconditional support. My parents, Terry and Connie Damm, have consistently stood by my side providing support, encouragement, and love. Their devotion is unlimited, and they always make themselves available whenever I need them, whether it is just to listen or to fix the tire on my car. I am forever indebted to them. Along with my family, Alain Ganamet has given me more support and love than I could ever imagine possible. He is there every day by my side, celebrating, encouraging, and pushing me to succeed. Words can not express how grateful I am for his enduring loyalty.

TABLE OF CONTENTS

Acknowledgements.....	ii
List of Figures.....	vii
List of Tables.....	xv
List of Appendices.....	xvii
Abstract.....	xix
CHAPTER 1.	
Introduction.....	1
1.1 Background.....	1
Protein-Ligand Binding	
Protein Flexibility	
Structural Alignment of Flexible Proteins	
Ligands as Drugs	
1.2 Structure-Based Drug Design.....	6
Application to Pharmaceutical Research	
Cross-Docking Problem	
1.3 Accounting for Protein Flexibility in Structure-Based Drug Design....	7
Current Techniques	
The Multiple Protein Structures Method	
1.4 HIV-1 Protease as a Test Case.....	11
HIV/AIDS	
HIV-1p Dynamics	
Resistance	
Inhibition Mechanisms	
1.5 Theory.....	20
Molecular Mechanics	
Molecular Dynamics	
Langevin Dynamics	
Metropolis Monte Carlo	
Molecular Docking	
1.6 Specific Aims.....	28
1.7 References.....	29

CHAPTER 2.	
Gaussian-Weighted RMSD Superposition of Proteins: A Structural Comparison for Flexible Proteins.....	45
2.1 Introduction.....	45
2.2 Computational Methods.....	16
Protein Dataset	
Standard RMSD Fit	
Weighted RMSD Fit	
Alignment Method	
2.3 Results and Discussion.....	51
Gaussian-Weighted RMSD Alignment	
Gaussian Scaling Factor	
Identifying Domains and Hinge Regions	
2.4 Conclusions.....	62
2.5 References.....	64
CHAPTER 3.	
Application of the wRMSD Method to Predicted Protein Structures and Homologous Proteins.....	66
3.1 Introduction.....	66
3.2 Computational Methods.....	67
Protein Structure Prediction Dataset	
Homologous Protein Dataset	
3.3 Results and Discussion.....	69
Using wRMSD to Evaluate Protein Structure Predictions	
Homologs: Low Sequence Identity and Large Conformational Differences	
Overcoming Errors in the Initial Sequence Alignment	
Identifying Sequence Misassignments	
3.4 Conclusions.....	88
3.5 References.....	91
CHAPTER 4.	
Exploring Experimental Sources of Multiple Protein Conformations in Structure-Based Drug Design.....	95
4.1 Introduction.....	95
4.2 Computational Methods.....	98
Protein Preparation	
MUSIC Simulation	
Pharmacophore Elements	
Pharmacophore Model Evaluation	
4.3 Results and Discussion.....	102
Structural Comparison of the Protein Conformations	
Pharmacophore Model Comparison	
Evaluation of Pharmacophore Models	
Effect of the Structure Number in Ensemble	

4.4	Conclusions.....	119
4.5	References.....	121
CHAPTER 5.		
	Accounting for Multiple Protein Conformations in Ranking Ligand Databases.....	126
5.1	Introduction.....	126
5.2	Computational Methods.....	128
	Pharmacophore Model Generation	
	MPS-DOCK Orientation Spheres	
	Excluded Volume Representation	
	MPS-DOCK Ranking Function	
	Ligand Data Sets	
5.3	Results and Discussion.....	134
	Application to HIV-1p	
	MPS-DOCK Parameter Investigation	
	Effect of Protein Excluded Volumes	
	Extension to Additional HIV-1p Systems	
	Ranking Function Consistency: Application to DHFR	
5.4	Conclusions.....	152
5.5	References.....	153
CHAPTER 6.		
	Inhibition of HIV-1p By Modulating its Conformational Behavior of the Flap Region.....	156
6.1	Introduction.....	156
6.2	Methods.....	158
	Multiple Protein Structures Method	
	Virtual Screening	
	Dynamics Simulations	
	Ruling out the “Elbow Region”	
	HIV-1 Protease Activity Assay	
6.3	Results and Discussion.....	164
	Defining the Flap-Recognition Pocket	
	Virtual Screening using MPS Pharmacophore Model	
	Ligand Behavior in the Dynamics Simulations	
	Protein Behavior in the Dynamics Simulations	
	Ruling out the “Elbow Region”	
	Experimental Validation of Predicted Compounds	
6.4	Conclusions.....	179
6.5	References.....	182
APPENDICES.....		187

LIST OF FIGURES

- Figure 1.1.** Schematic diagram of a folding funnel illustrating the multiple states that a protein can assume.....2
- Figure 1.2.** Structures of the ten HIV-1p inhibitors currently on the market.....12
- Figure 1.3.** A cartoon representation of HIV1-p in the semi-open conformation; catalytic residues 25/25' are shown in stick representation. The location of key features discussed in Chapters 1-6 are illustrated by arrows to orient the reader.....13
- Figure 1.4.** Two conformational states assumed by HIV-1p. The flap tips change handedness upon flap closure. (A) Semi-open conformation illustrated using the crystal structure 1HHP. (B) Closed state demonstrated using the crystal structure 1PRO.....15
- Figure 1.5.** Structure 1TW7 (grey cartoon and surface representation). There are many contacts between the neighboring unit cells (not shown for clarity), but contact in the elbow regions is limited (yellow chain).....19
- Figure 1.6.** The β -sheet interface. Monomer A residues are shown in surface representation while monomer B residues are shown as ball and sticks. The N-terminal peptide (P1-W6) is at the top, and the C-terminal peptide (C95-F99) is at the bottom. Green residues are found to be important in designing mimics. (Figure courtesy of Jerome Quintero.).....20
- Figure 1.7.** Bonded and nonbonded components of a typical MM force field.....22
- Figure 2.1.** A series of iterations are needed to converge the wRMSD solution for overlaying two proteins. Four snapshots from the series of iterations are shown to demonstrate the process.....51
- Figure 2.2.** ER α . (A) The behavior of the wRMSD and %wSUM metrics as the weighted alignment is performed in an iterative manner using the entire protein sequence for the initial sRMSD fit. A scaling factor, c , of 2 \AA^2 is used. The vertical line indicates where convergence is reached. (B) sRMSD alignment of 3ERD (yellow) onto 3ERT (blue). (C) wRMSD alignment after convergence is reached. Arrows denote regions with improved fit.....53

Figure 2.3. The scaling factor, c , plotted against the sRMSD value for each weighted fit and the target coordinates. Open squares (\square) are for ER α , 3ERD fit onto 3ERT. The weighted fit is the same for c values from 0.3-20 \AA^2 . Filled triangles (\blacktriangle) are for PKA, 1JLU fit onto 1CMK. The weighted fit is the same for c values from 0.2-2 \AA^2 . The largest values of c simply reproduce the sRMSD solution for the PKA structures.....54

Figure 2.4. If the scaling factor is too small, the wRMSD fit fails to produce converged structures for GroEL. The behavior of the wRMSD metric versus iteration during the weighted fit, using the entire protein sequence for the initial RMSD fit and two values of c . (Left) wRMSD alignment of 1AON (yellow) onto 1OEL (blue) after 800 unconverged iterations of wRMSD fitting, $c = 1 \text{\AA}^2$. (Right) wRMSD alignment of 1AON (yellow) onto 1OEL (blue) after convergence is reached, $c = 5 \text{\AA}^2$55

Figure 2.5. If the scaling factor is too large, an wRMSD fit is the same as a sRMSD fit for EFG. (A) sRMSD alignment of 1FNM (yellow) onto 2EFG (blue). (B) wRMSD alignment of 1FNM (yellow) onto 2EFG (blue) after convergence is reached, $c = 100 \text{\AA}^2$. (C) wRMSD alignment of 1FNM (yellow) onto 2EFG (blue) after convergence is reached, $c = 2 \text{\AA}^2$56

Figure 2.6. Left: sRMSD alignment of two protein conformations. Right: wRMSD alignment of the same structures. (A) HIV-1p, 1KZK (yellow) onto 1HHP (blue), $c = 2 \text{\AA}^2$. (B) RAN, 1RRP (yellow) onto 1BYU (blue), $c = 5 \text{\AA}^2$. (C) RNA Pol, 1QLN (yellow) onto 1MSW (blue), $c = 5 \text{\AA}^2$57

Figure 2.7. EFG. The behavior of the %wSUM metric as the weighted alignment is performed in an iterative manner. Ten different subsets of 1FNM (yellow) were used for the initial standard alignment onto 2EFG (blue) and then the weighted iterations were performed using the entire sequence ($c = 2 \text{\AA}^2$). (Top) wRMSD alignment corresponding to the maximum %wSUM value. (Bottom) wRMSD alignment corresponding to the smaller %wSUM value.....59

Figure 2.8. RAN. (A) The behavior of %wSUM as the weighted alignment is performed in an iterative manner. Ten different subsets of 1RRP (yellow) were used for the initial standard alignment onto 1BYU (blue) and then the weighted iterations were performed using the entire sequence ($c = 2 \text{\AA}^2$). (B) wRMSD alignment corresponding to the maximum %wSUM value. (C) wRMSD alignment corresponding to the second largest %wSUM value.....60

Figure 2.9. DNA Pol. (A) The behavior of %wSUM as the weighted alignment is performed in an iterative manner. Ten different subsets of 1IH7 (yellow) were used for the initial standard alignment onto 1IG9 (blue) and then the weighted iterations were performed using the entire sequence ($c = 2 \text{\AA}^2$). The four distinct solutions are indicated on the right. (B) wRMSD alignment corresponding to the maximum %wSUM value. (C) wRMSD alignment corresponding to the second largest %wSUM value. (D) wRMSD alignment corresponding to the third largest %wSUM value. (E) wRMSD alignment

corresponding to the smallest %wSUM value. This overlay is oriented differently than in (B–D). Arrows in (B–E) highlight regions with good alignment.....61

Figure 3.1. The wRMSD alignments of (A) group 427’s and (B) group 32’s predictions (thick, colored lines) to Target 179 (thin, gray line). The wRMSD alignments of (C) group 400’s submission and (D) group 183’s submission are given as examples of the comparison of a fragment. The target has the same orientation in both alignments. (E) The scale at the bottom shows how smaller deviations (blue) are more heavily weighted in the wRMSD. Deviations over 3.9 Å have weights under 5% (red).73

Figure 3.2. The submission from group 517 to target 172 has two solutions (A) and (B) by wRMSD fitting. The second solution (B) is scored much lower because it is only a match of a small helix. The target (gray, thin line) is in the same orientation in both alignments. The color code of the weights is the same as in Figure 3.1E.....74

Figure 3.3. wRMSD fits for groups (A) 537 and (B) 417 to Target 172. The %wSUM_ALL values for the best wRMSD fit are given in parentheses. The color code of the weights is the same as in Figure 3.1E. The target (gray, thin line) is in the same orientation in both alignments.....75

Figure 3.4. The multiple wRMSD solutions for the top three structures chosen for Target 170 (thin, gray line). (A) The wRMSD alignments of team 517’s prediction (thick, colored line). (B) The wRMSD alignment of team 400’s fragment submission. (C) The solutions for team 51. The target has the same orientation in both alignments. (D) The scale shows the weights for these wRMSD fits based on $c = 12 \text{ \AA}^2$. Deviations over 6.0 Å have weights under 5% (red).....76

Figure 3.5. wRMSD fits for groups (A) 2, (B) 10, (C) 331, and (D) 437 (thick, colored lines) to Target 147 (gray, thin line). The %wSUM_ALL values for the best wRMSD fit are given in parentheses. The color code of the weights is the same as in Figure 3.4D. The target is in the same orientation in both alignments.....78

Figure 3.6. wRMSD fits for groups (A) 373, (B) 132, (C) 437, (D) 29, and (E) 2 to Target 162-3. The order A-E reflects a rank order based on the RMS/coverage graph, but the overlays and their weights are from a local wRMSD fit with $c = 12 \text{ \AA}^2$. Two significant solutions were obtained for each group’s entry but only the best is shown. The %wSUM_ALL values for the individual wRMSD solutions are given in parentheses. The color code of the weights is the same as in Figure 3.4D. The target (gray, thin line) is in the same orientation in both alignments.....79

Figure 3.7. Chaperonin family (20.8% ID). Most techniques would readily identify the similarity between the thermosome and GroEL in the similar bound conformation, but they may not identify its similarity with the apo conformation of GroEL. (A) wRMSD superposition of the bound conformation of GroEL (thick, colored lines) onto the homologous thermosome (thin, black lines). Light gray regions of GroEL indicate residues within gaps in the alignment. (B) wRMSD fit of the apo conformation of

GroEL³¹ (thick, colored lines) onto its homolog thermosome (thin, gray lines). The value of %wSUM gives the normalized weights of all residues, showing that the two bound conformations in A have greater similarity than the two conformations in B. The scale shows the weights for these wRMSD fits based on $c = 5 \text{ \AA}^2$81

Figure 3.8. DNA methylase family (23% ID). Weighted structural superpositions are nearly independent of the sequence alignment method, but standard superpositions are greatly effected. Six sequence-alignment codes were used to determine residue pairings. (A) Overlays of 1BOO (thin, colored lines) to 1EG2, (thick, black line) from standard superpositions based on six different sequence alignments. The average difference in the superpositions is 2.325 Å. (B) The six weighted superpositions of 1BOO to 1EG2, based on the same sequence alignments, are indistinguishable (average difference is 0.140 Å).....85

Figure 3.9. SpoU rRNA methylase family (26% ID). (A) BLAST sequence alignment of 1IPA and 1GZ0 using default parameters. Colons represent sequence identities, and gaps are shown with dashes. The underlined region notes domain 1, and the blue boxes represent misaligned residues corresponding to the labeled α -helix and β -sheet in (B). Atom pairs with a weighting of 40% or greater in the wRMSD calculation are noted with asterisks. Standard (B) and weighted (C) superpositions of 1IPA (thick, colored line) onto 1GZ0 (thin, gray line). In (C), the color coding by weight is the same as in Figure 3.7.....87

Figure 4.1. (A) Gaussian-weighted overlay of 28 models in NMR ensemble along with all cu ligands (front view). The corresponding cu ligands are also shown using a top view for clarity. The regions of the protein with high backbone deviations are highlighted with an arrow. (B) Gaussian-weighted overlay of 10 crystal structures bound to unique cu ligands (front view). A top view of the 10 ligands is also shown. (C) The scale shows how smaller deviations (blue) are more heavily weighted in the wRMSD fit, $c = 2 \text{ \AA}^2$. Deviations over 2.45 Å have weights under 5% (red).....103

Figure 4.2. (A) Pharmacophore model (radii of 1×RMSD) generated using 28 NMR structures. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; cyan, aromatic/hydrophobic. Top view of the protease backbone is shown in grey, as are the excluded volumes. (B) Pharmacophore model superimposed with 28 cu-ligands colored in grey. Both top and front views are shown.....105

Figure 4.3. The average NMR model is compared to a previously created a pharmacophore model from a static crystal structure. It is notable that the model from the average NMR structure, while having additional sites compared to the MPS NMR model, was still reasonable unlike the model from the static crystal structure. The static crystal structure model has many additional elements and is not appropriate for virtual screening applications. (A) Pharmacophore model (radii of 1×RMSD) generated using the average NMR structure. (B) Pharmacophore model (radii of 1×RMSD) generated using the static crystal structure 1HHP. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, aromatic/hydrophobic; green, aromatic. Top view of protease is shown; backbone is in grey.....106

Figure 4.4. (A) Pharmacophore model (radii of 1×RMSD) generated using 10 cu-crystal structures. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, aromatic/hydrophobic; green, aromatic. Top view of the protease backbone is shown in grey as are the excluded volumes. (B) Pharmacophore model superimposed with 10 unique cu-ligands colored in grey. Both top and front views are shown.....107

Figure 4.5. Comparison of known HIV-1p substrate recognition pockets with MPS pharmacophore models (radii of 1×RMSD): white, S1/S1' pocket; yellow, S2/S2' pocket; purple, S3/S3' pocket. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, aromatic/hydrophobic; green, aromatic. Flap residues 46/46' – 54/54' are removed for clarity. (A) NMR model. (B) cu-crystal structure model.....109

Figure 4.6. (A) Top view of an MPS pharmacophore model (radii of 1×RMSD) created using 11 structures generated from a 3-ns MD simulation of apo HIV-1p. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; cyan, aromatic/hydrophobic; green, aromatic. Excluded volumes are shown in grey. (B) Gaussian-weighted overlay of the 11 snapshots (front view). The color code of the weights is the same as in Figure 1C, and the view is comparable to Figure 1A and B...110

Figure 4.7. Receiver Operator Characteristic curves generated from screening a database of 89 known HIV-1p inhibitors against a set of 85 chemically similar known inactives and 2322 general decoy compounds. Each series represents a different stringency in the screen (i.e. 6 of 8 elements are required as a hit, 7 of 8 elements are required as a hit, etc.) Points in series are increasing radii values from 1× to 3×RMSD for the NMR model and 1× to 4× for the cu-crystal model. The radii are labeled on the 6 of 8 models based on NMR and the 9 of 11 models based on cu-crystals. The optimal pharmacophore models are highlighted by an arrow. (A) MPS NMR pharmacophore models, 89 known inhibitors vs. 85 decoy compounds (Optimal: 7/8, 2.0×RMSD). (B) MPS cu-crystal pharmacophore models, 89 known inhibitors vs. 85 decoy compounds (Optimal: 9/11, 3.0×RMSD). (C) MPS NMR pharmacophore models, 89 known inhibitors vs. 2322 general molecules (Optimal: 7/8, 2.0×RMSD). (D) MPS cu-crystal pharmacophore models, 89 known inhibitors vs. 2322 general molecules (Optimal: 9/11, 2.7×RMSD).....113

Figure 4.8. Gaussian-weighted overlay of 90 crystal structures from drug-susceptible strains of HIV-1p. The color code of the weights is the same as in Figure 4.1C, $c = 2 \text{ \AA}^2$. (A) Front View (A) Top View.116

Figure 4.9. Pharmacophore model (radii of 1×RMSD) generated using 90 crystal structures bound to a diverse set of ligands. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, aromatic/hydrophobic; green, aromatic. Top view of protease is shown; backbone is in grey.....118

Figure 5.1. Representation of the small molecule probes as atomic spheres.....129

Figure 5.2. Illustration of the aggregate pharmacophore element concept for a benzene consensus cluster. In our original model, the consensus cluster is represented as a single, spherical element (left). In our new model, each probe of the consensus cluster is represented by a set of atomistic spheres (right). The overlay of these sphere-sets generates a pharmacophore map whose size, shape, and density more accurately reflects the favorable interaction surface with the receptor.....130

Figure 5.3. Comparison of the MPS pharmacophore representations for 1HHP. Flap residues 46/46' – 54/54' have removed for clarity in (B) - (D). (A) Front view of HIV-1p with original pharmacophore model sitting in active site bottom to orient reader. (B) Close-up view of the original pharmacophore model derived for HIV-1p. (C) Close-up view of the aggregate sphere representation. (D) Close-up view of the clustered aggregate sphere representation. In all representations spheres are colored according to chemical functionality: red, hydrogen-bond donating; green, aromatic; cyan, hydrophobic. The clustered representations are also shaded by weight- the greater the number of spheres making up the cluster, the darker the color.....136

Figure 5.4. Comparison of excluded volume representations. (A) Minimal excluded volumes- 2 spheres centered on the 2 C γ positions of the catalytic aspartates (25, 25'). (B) RMSD cut-off of 0.50, 35 excluded volume spheres. (C) RMSD cut-off of 0.75, 351 excluded volume spheres. (D) RMSD cut-off of 1.00, 755 excluded volume spheres. (E) RMSD cut-off of 1.25, 1033 excluded volume spheres. (F) RMSD cut-off of 1.50, 1253 excluded volume spheres. Protein excluded volume spheres are colored gray and shown along with a surface representation. The atomic spheres are colored by chemical functionality (red, hydrogen-bond donating; green, aromatic; cyan, hydrophobic) and shaded by weight (the greater the number of spheres making up the cluster, the darker the color).....144

Figure 5.5. A representative ROC plot showing enrichments for discriminating a set of 89 known HIV-1p inhibitors from a set of 2324 general decoys. Compared are the enrichment profiles obtained using the four models- 1HHP, 3HVP, 3PHV, and CONS. The selectivity is plotted against 1 minus the specificity for each threshold value evaluated. TN - true negatives, FP - false positives, TP - true positives, FN - false negatives.....148

Figure 5.6. The clustered aggregate sphere representation of DHFR. Atomic spheres are colored according to chemical functionality (red, hydrogen-bond donating; blue, hydrogen-bond accepting; green, aromatic) and also shaded by weight- the greater the number of spheres making up the cluster, the darker the color. 818 excluded volume spheres are shown in grey along with a surface representation overlaid with a cartoon depiction of the entire protein. The cofactor NADPH is colored by atom type (carbon is shown in yellow).....150

Figure 5.7. A representative ROC plot showing enrichments obtained for the DHFR aggregate sphere model for discriminating a set of 50 high affinity ecDHFR inhibitors from a set of 2326 general decoys (blue) and a set of 541 general DHFR inhibitors from a

set of 2326 general decoys (red). The selectivity is plotted against 1 minus the specificity for each threshold value evaluated. TN - true negatives, FP - false positives, TP - true positives, FN - false negatives.....151

Figure 6.1. When a monomer closes, it places its flap tips against the “eyebrow” region of the other monomer. The right monomer is the apo, semi-open state and shown with a grey surface. The left monomer is in the bound, closed state and colored yellow. Ile 50 and Gly 51 are shown in stick representation in direct contact with the “eye”.....157

Figure 6.2. Compound 1 (2,2,4-trimethyl-1,2-dihydroquinolin-6-yl benzoate) was identified through a virtual screen and chosen for theoretical simulations.....160

Figure 6.3. The semi-open monomer is shown with the new site color-coded by atom type (green are carbons, red are oxygens, blue are nitrogens). (A) Front view. (B) 90 degree rotation. (C) The individual residues within the new site are each colored individually and labeled to show their placement within the cleft. G78 and V56 are not visible in this view.....164

Figure 6.4. Two representative pharmacophore models are shown. Both are derived from the same solvent-mapping data. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, hydrophobic; green, aromatic. (A) Isotropic MPS Pharmacophore model (radii of 1.3×RMSD) mapping the “eye” region of the semi-open conformation. (B) Close-up view of model and 90° rotation. Flap tip of the closed monomer is shown in yellow to demonstrate overlap with the pharmacophore model. (C) An atomistic representation of the solvent probes better shows the contour of the site. (D) Close-up view and 90° rotation.....167

Figure 6.5. Compound 1, identified through the virtual screen, is shown overlaid with MPS pharmacophore model (radii of 1.3×RMSD). The agreement between its chemical scaffold and the pharmacophore elements is demonstrated.....168

Figure 6.6. Ligand RMSD Plots. (A) Ligand RMSD values during the 10-ns MD simulation. Compound 1 is compared to its average position over the MD simulation. The average HIV-1p structure (green) and position of Compound 1 (pink) were calculated using data across the entire 10 ns. (B) Ligand RMSD values during the 5-ns LD simulations (5 random seeds). Compound 1 is compared to its minimized pose. In Run 2, Compound 1 starts in the flap-recognition pocket of Monomer A (red structure) but disassociates into the active center and binds in the opposite side pocket of Monomer B (purple structure). Several events are seen where the ligand dissociates and rebinds again in the same pocket (spikes up to 10/12 Å, which decrease again).....170

Figure 6.7. Overlay of snapshots across dynamics simulations. The conformations are colored in order of time reference across the simulations (MD: 0-10 ns, LD: 0-5ns). (A) MD simulation, snapshot taken every 1 ns, resulting in 11 overlaid conformations. (B-F)

LD Run 1-5, respectively. Snapshot taken every 0.5 ns, resulting in 11 overlaid conformations.....172

Figure 6.8. Distance calculated between the flap-tip residue I50 C α to the catalytic residue D25 C α throughout the MD trajectory. This metric quantifies the flap movement in the vertical direction.....174

Figure 6.9. Structures of HIV-1p. A front and top view is provided to demonstrate the conformation of the flap region and change in handedness of the flap tips that occurs between the different states. (A) Semi-open conformation (PDB ID: 1HHP). (B) Closed conformation (PDB ID: 1PRO) (C) Representative structure from the 10-ns MD in a closed-flap conformation but semi-open handedness of the flap tips.....175

Figure 6.10. Angle calculated between residues G48 C α - G49 C α - I50 C α to quantify the curling of the flap tips. An angle $\leq 115^\circ$ is defined as a curled state and $\geq 145^\circ$ as curled out. The blue curve displays the movement of Flap A and the yellow curve of Flap B..176

Figure 6.11. Distance is calculated between residues G51 C α and T80 C α to quantify the position of the flaps. The blue curve displays the movement of Flap A and the yellow curve of Flap B.....177

Figure 6.12. (A) HIV-1p shown in a surface representation. The predicted pose of Compound 1 (green, stick representation) by AutoDock 3 in the elbow region is highlighted by an arrow. (B) RMSD values of Compound 1 during MD simulation (5 random seeds) showing ligand disassociating from contact with HIV-1p. Compound 1 is compared to the starting pose given in (A).....178

Figure 6.13. (A) Para-methoxy analog: 2,2,4-trimethyl-1,2-dihydroquinolin-6-yl 4-methoxybenzoate. (B) The activity of HIV-1p was monitored using a fluorimetric assay; upon HIV-1p cleavage of the FRET peptide substrate, fluorescence is recovered. Inhibition is measured as a result of the time-dependent decrease of fluorescence intensity that is linearly related to substrate cleavage. Each data point represents an average of three experiment, and the error bars reflect the standard deviation of observed values. Pepstatin A is shown as a control.....179

LIST OF TABLES

Table 2.1. Test case proteins listed in order of small to large conformational changes.....	46
Table 2.2. Range of optimal scaling factors for each protein system, along with the calculated sRMSD of the wRMSD fit over the given range.....	55
Table 2.3. A comparison of the wRMSD fits using an initial global sRMSD alignment and the best result from initial local alignments. Two local wRMSD fits for DNA Pol are compared to two global wRMSD fits.....	62
Table 3.1. Summary of Targets used in CASP5 Evaluation.....	68
Table 3.2. Target 179, wRMSD rankings ($c = 5 \text{ \AA}^2$) compared to GDT_TS values.....	71
Table 3.3. Target 172, wRMSD rankings ($c = 5 \text{ \AA}^2$) compared to GDT_TS values.....	74
Table 3.4. Target 170, wRMSD rankings ($c = 12 \text{ \AA}^2$) compared to GDT_TS values.....	77
Table 3.5. Target 147, wRMSD rankings ($c = 12 \text{ \AA}^2$) compared to GDT_TS values.....	78
Table 3.6. Target 162-3, wRMSD rankings ($c = 12 \text{ \AA}^2$) compared to GDT_TS values...	80
Table 3.7. Differences in the structural superpositions for a diverse set of homologous proteins. A complete set of references for the crystal structures is provide in Appendix 2. Both standard and weighted superpositions were generated from a variety of sequence alignments. The sequence alignments were altered by varying the parameters within BLAST or varying the code used for the alignment. The differences across the superpositions were measured in RMSD (\AA) between the coordinates. Average differences are reported above, but all calculated RMSD are included in Appendix 2....	83
Table 5.1. Effect of varying the number of minimum required nodes (m), inter-node distance (d), the distance tolerance (t), and cluster size cut-off (c) on virtual screening performance using the 1HHP model. Minimal representation was used for the floor of the active site (2 excluded volume with a scoring penalty of 10) as to optimize favorable scoring elements prior to excluding compounds based on size and overlap with the pocket. The Cummings et al. data set was used, consisting of 1025 compounds seeded with 5 HIV-1p inhibitors. The number of spheres in the scoring set, the sum of the ranks of the 5 inhibitors and the number of compounds scored by MPS-DOCK (in parenthesis)	

is shown. A lower number for the sum of ranks indicates cases where the 5 known inhibitors are all ranked highly. A high number (close to 1025) in parenthesis is optimal as that means a large percentage of the database is being ranked. This proves a more appropriate test for our ranking function. The bolded column represents the optimal parameter set.....139

Table 5.2. Expansion of Table 1: Effect of varying cluster size cut-off (c) using the optimal MPS-DOCK parameters, $4m - 3d$ and $0.25t$. Data from Cummings et al. is also presented as a comparison. For a given fraction of the ranked database, the number of known HIV-1p inhibitors identified is shown along with the sum of the ranks of the 5 inhibitors and the number of compounds scored by MPS-DOCK. The bolded row represents the optimal sphere set.....141

Table 5.3. Virtual screening performance of the four optimal HIV-1p pharmacophore models (1HHP, 3HVP, 3PHV, and CONS) using the optimal dock parameters, $4m - 3d$ and $0.25t$ along with an excluded volume penalty of 10. The Cummings et al. data set was used consisting of 1025 compounds seeded with 5 HIV-1p inhibitors. For a given fraction of the ranked database, the number of known HIV-1p inhibitors identified is shown along with the sum of the ranks of the 5 inhibitors and the number of compounds scored by MPS-DOCK.....146

Table 6.1. List of residues defining the flap-recognition pocket. Those in bold can mutate to residues that contribute to drug resistance.....165

LIST OF APPENDICES

Appendix 1.1. Global wRMSD code.....	188
Appendix 1.2. Local wRMSD code.....	205
Appendix 2.1. RMS/coverage graphs.....	224
Appendix 2.2. PDB IDs and references for crystal structures obtained from the HOMSTRAD Database.....	230
Appendix 2.3. Homologous wRMSD code.....	234
Appendix 2.4. Raw Data from differences in superpositions.....	256
Appendix 2.5. An alternate version of Figure 3.4 with selenomethionine added to the sequence of 1GZ0.....	276
Appendix 3.1. Structures and inhibition constants of the ten unique cyclic urea ligands.....	278
Appendix 3.2. PDB IDs and references of 90 HIV-1 protease crystal structures.....	280
Appendix 3.3. MPS NMR and crystal structure pharmacophore models.....	288
Appendix 3.4. Raw pharmacophore screening data.....	290
Appendix 3.5. Identified false positives.....	294
Appendix 4.1. DOCK code modifications.....	297
Appendix 4.2. Modified DOCK parameter files.....	304
Appendix 4.3. Example INDOCK file.....	306
Appendix 4.4. Raw atomistic pharmacophore screening data.....	307
Appendix 5.1. MPS “eye” pharmacophore model.....	313

Appendix 5.2. 93 Identified Compounds from MPS pharmacophore screen grouped by 60% chemical similarity and overlap.....317

Appendix 5.3. Analysis of implicit-solvation Langevin Dynamics simulations.....329

ABSTRACT

Structure-based drug design (SBDD) has emerged as an important tool in drug discovery research. Traditionally, SBDD is based on a static crystal structure of the target protein. However, a protein in solution exists as an ensemble of energetically accessible conformations and is best described when all states are represented. Upon ligand binding, further conformational changes in the receptor can be induced. While ligand flexibility can be accurately reproduced, replicating the innumerable degrees of freedom of the protein is impractical due to limitations in computational power.

Previously, Carlson et al. developed a robust method to generate receptor-based pharmacophore models based on an ensemble of protein conformations. The use of multiple protein structures (MPS) allows a range of conformational space that can be assumed by the protein to be sampled and hence, simulates the inherent flexibility of a binding site in a computationally feasible manner. Small molecule probes are used to map energetically favorable regions of each protein active site, and the MPS are then overlaid to identify the most important, chemically relevant features conserved across the conformations.

Here, we have refined the MPS method by developing techniques to optimize different steps in the procedure. First, we outline tools to properly overlay flexible proteins based on the rigid regions of the structure by incorporating a Gaussian weight into a standard RMSD alignment. Atoms that barely move between the two conformations will have a greater weighting than those that have a large displacement. Using HIV-1 protease (HIV-1p) as a test case, we next examine the use of various

sources of MPS: snapshots of an apo structure across a molecular dynamics simulation, a bound NMR ensemble, and a collection of bound crystal structures. Finally, we implement a simple ranking metric into the MPS method to quantify ligand overlap with a contour-based representation of the pharmacophore model. Overlapping in a region of the active site dense with pharmacophore spheres results in a higher ranking of a ligand pose. The refined MPS method and other computational techniques are then applied to study HIV-1p and investigate a novel inhibition mechanism by modulating its conformational behavior.

CHAPTER 1

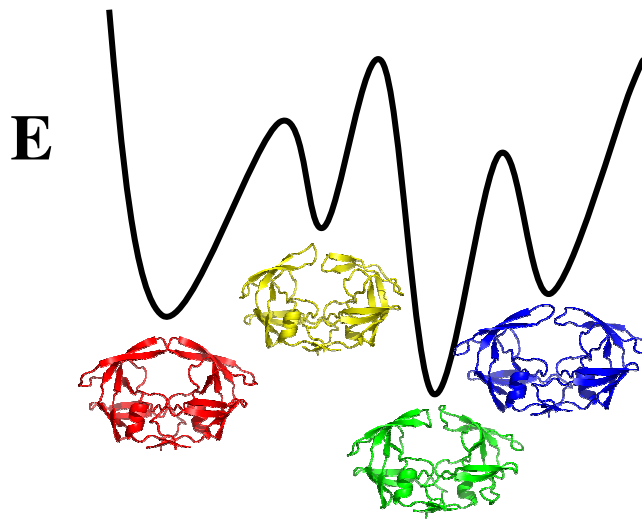
Introduction

1.1 Background

Protein-Ligand Binding

Proteins were once thought to assume a single static fold that could only interact with a single complementary ligand, the “lock-and-key” theory proposed by Fischer in 1890. This view then advanced to the “induced-fit” theory, suggesting that the active site of a protein could adjust to accommodate the ligand, as reviewed by Jorgensen.^{1,2} However, proteins under native conditions have been shown, through NMR and kinetic experiments, to pre-exist as a statistical ensemble of conformational states.³⁻⁶ This evidence sparked a new theory to describe protein-ligand interactions. The protein in solution exists as an equilibrium ensemble of pre-existing conformations from all energetically accessible states.⁷⁻¹² Each local minimum on the energy landscape corresponds to a different pre-existing structure with a discrete energy level, as demonstrated in Figure 1.1. Theoretically, a ligand could bind to any conformation in the ensemble, even those in less populated states. It is thought that ligand binding would shift the equilibrium of the system in its favor to further drive the binding reaction, resulting in a new ensemble distribution.¹³

Figure 1.1. Schematic diagram of a folding funnel illustrating the multiple states that a protein can assume.



Protein Flexibility

As a direct result of this new view on protein-ligand binding, Freire and co-workers realized that the Gibbs energy of stabilization of a protein is not equally distributed throughout its structure.^{6,11} Flexible regions allow ligands to enter and leave the binding site, and plasticity is required in any induced-fit binding, whether it is a simple side-chain reorientation or movement of a whole domain.^{12,14} They concluded that protein-binding sites are generally characterized by having concomitant regions of low and high structural stability. The regions of the protein with high structural stability, or “core regions,” remain relatively static between the multiple conformations, despite any movement of the flexible regions.

Protein flexibility is a common feature of many biological systems that can regulate ligand binding and also, a large variety of cellular processes. The conformational changes can give rise to motion in molecular motors, act as a switch to turn on or off the respective biological activity, or even allow the same protein to perform several different functions.^{9,12,15} Signaling proteins can communicate through the same interaction domain with many different effectors. This requires that the interaction domain be flexible

enough to accommodate structures of various sizes and chemical composition, yet the interactions must be specific and selective enough to continue the signal flow.

Databases that highlight structural variation through mobility or evolution are useful and growing resources. The Database of Simulated Molecular Motions provides computational data on protein motion and flexibility.¹⁶ The Structure Superposition Database was created to address the issue of properly aligning and understanding large sets of homologous protein structures.¹⁷ The Database of Macromolecular Movements presents a diverse set of proteins which display large conformational changes in different crystallographic structures.¹⁸⁻²⁰ A recent review discusses a range of motions observed in biopolymer synthesis and membrane transport seen in the Database of Macromolecular Movements.²¹ For instance, T7 RNA polymerase exhibits a large conformational change from the initiation to elongation phase, and a substantial motion is observed in Ca²⁺-ATPase as it converts between a calcium-bound and free state.

Structural Alignment of Flexible Proteins

The heart of comparing two conformations of a flexible protein is an appropriate overlay of the structures for visual inspection. Over a dozen different techniques have been proposed for comparing and overlaying flexible proteins.^{19,22-34} For almost 20 years, every technique has been based on two steps: first, identify related subsets of C α in the protein conformations and second, overlay that subset by a standard root mean square deviation (sRMSD) fit. Each technique differs in the way that it identifies the subsets, usually defining static, core regions of the protein. Some methods are quite elegant, even using weighted analytical techniques to define the subsets. The merits and caveats of each technique's definition of a subset are often debated, but when an alignment is made in the end, all of these techniques get simplified to each C α receiving a binary assignment of "in" or "out" of the subset. The C α that are in the subset get aligned with an sRMSD. Even if weights were used in the analysis, they are not used in the final overlay step.

Chapter 2 describes our contribution to such an approach; instead of identifying subsets for a sRMSD, we chose to change the RMSD fit process itself.

In order to perform a structural comparison, the corresponding residues (atom pairs) between the proteins must be determined. This task can be accomplished in a sequence-dependent manner using an initial sequence alignment or solely through structural information in a sequence-independent manner. Sequence-based techniques can miss similarity between homologous proteins with intermediate to low sequence identity (twilight zone). Fold-based methods can identify structural similarity, even between homologs with divergent sequences, but may be misled in the case of flexible proteins. A technique that combines the two approaches and overcomes limitations caused by the protein flexibility would be an ideal choice for superimposing homologs. We describe such a method in Chapter 3. An evaluation of six structural comparison techniques (SSAP³⁵, STRUCTAL^{18,36}, DALI³⁷, LSQMAN³⁸, CE³⁹, and SSM⁴⁰) demonstrates many benefits and limitations of current methods.⁴¹ Additional reviews of the field call for combining techniques and using consensus across several methods to best define a structural comparison.^{42,43}

Ligands as Drugs

Ligands that bind specifically to proteins known to be involved in disease pathways can be used as drugs. As such, the principles that govern protein-ligand binding apply to the design and discovery of new drug entities. The initial “hit” in the drug discovery process is termed a lead compound. A general distinction can be made between lead-like and drug-like compounds. Lead-like compounds are small molecules (200-400 Da) with micromolar affinity for the target compound.⁴⁴⁻⁴⁶ They may still possess undesirable properties such as insolubility, high toxicity, or metabolism problems. Drug-like compounds are generally larger (400-600 Da) and “have sufficiently acceptable ADME (absorption, distribution, metabolism, and excretion) properties and sufficiently

acceptable toxicity properties to survive through the completion of phase I clinical trials”.⁴⁷

Empirically derived rules are commonly used to describe “drug-likeness” as they offer general guidelines for describing the physical properties of orally available drugs. Pioneering work by Lipinski et al. led to Lipinski’s Rule of Five which states that, in general, an orally active drug possesses 1) ≤ 5 hydrogen-bond donors, 2) ≤ 10 hydrogen-bond acceptors, 3) a molecular weight < 500 Da, and 4) a partition coefficient ($\log P$) < 5 .⁴⁷ A variety of other drug-like guidelines have spawned from the Lipinski study and are reviewed by Walters et al.⁴⁸ Studies by Ajay et al. and Sadowski and Kubinyi have demonstrated the predictive power of empirical rules in distinguishing drugs from non-drugs using multiple datasets.^{49,50}

Oprea et al. noted that lead-like guidelines should be followed in the initial phases of drug discovery to filter compounds, not drug-like profiles.^{45,51} If drug-like rules were employed, the identified lead compounds may be difficult to optimize while remaining in “drug-like” space. Lead-like qualities include 1) relatively simple chemical features, amenable for combinatorial and medicinal chemistry optimization efforts, 2) membership to a well-established SAR (structure-activity relationship) series, wherein compounds with similar (sub) structures exhibit similar target binding affinity, 3) a favorable patent situation, and 4) good ADME and toxicity properties.⁴⁵ Moreover, following a physical properties analysis of compounds in lead-like datasets, empirical guidelines have been suggested: 1) a molecular weight < 450 Da, 2) $-3.5 < \text{LogP} < 4.5$, 3) ≤ 4 rings, 4) ≤ 10 nonterminal single bonds, 5) ≤ 5 hydrogen-bond donors, 6) ≤ 8 hydrogen-bond acceptors.⁴⁵

1.2 Structure-Based Drug Design

Application to Pharmaceutical Research

Structure-based drug design (SBDD) is a valuable technology that is seeing increased utilization to advance the process of drug discovery research.⁵²⁻⁵⁵ SBDD employs three-dimensional structures to design or predict ligands with high binding affinity and can generally be thought of as two varieties: *de novo* design and molecular docking. In *de novo* design a compound is designed to complement the inherent chemical characteristics of a binding site, while molecular docking is used to predict the binding mode of a known small molecule in a protein's active site. When a database of molecules is employed to identify a novel lead compound, the docking method is termed virtual screening. The three-dimensional structures may be experimentally determined by NMR spectroscopy or X-ray crystallography or by theoretical means such as Molecular Dynamics (MD) simulations or homology modeling tools.

The strength of SBDD lies in its potential ability to decrease the time and cost of bringing a drug to the market. To date, there has not been much success using SBDD to predict *de novo* compounds. However, molecular docking has played an important role in both lead discovery and optimization. By visualizing the interactions occurring between the ligand and protein, chemical modifications can be rationally directed to conserve those critical to binding. Furthermore, the protein-ligand complex can be evaluated to determine where chemical moieties can be eliminated and the ligand modified to improve its drug-like properties (e.g. solubility, oral bioavailability, selectivity, etc.). SBDD has become an integral part of the iterative drug design cycle⁴⁶, however due to limitations in computational power, the trade-off between speed and accuracy still exists in current techniques.

Cross-Docking Problem

Accounting for the conformational changes that can occur within the binding site of proteins has increased the difficulty of SBDD. As previously mentioned, the protein in solution exists as an ensemble of energetically accessible conformations and is best described when all states are represented.⁷⁻¹² Traditionally, SBDD is based on a static crystal structure of the target protein. However, often a single structure is insufficient because the conformation observed can be influenced by many factors including experimental conditions and the induced fit between ligand and protein.⁵⁶⁻⁵⁸ This pre-arrangement of the ligand binding site can lead to the cross-docking problem where the protein structure has adapted to bind a particular ligand or class of ligands but is unable to accommodate structurally diverse inhibitors. Incorporating protein flexibility has been recognized as a means to overcome the cross-docking problem and has become an important emphasis in improving SBDD techniques such as protein-ligand docking and protein-protein docking.

1.3 Accounting for Protein Flexibility in Structure-Based Drug Design

Current Techniques

Much progress has been made in developing clever, computationally feasible methods that simulate the inherent flexibility of a ligand-receptor system using both experimentally and theoretically determined structures. The original technique is termed “soft docking” and involves relaxing the criterion used to model steric fit, allowing for overlap of the protein and ligand surfaces.⁵⁹⁻⁶¹ A second method utilizes a single representative structure, the average of a collection of conformational states.⁶² A third way is to generate receptor conformations “on the fly” such that side chains are allowed to move to accommodate ligand binding using a pre-determined rotamer library to define

acceptable, alternative conformations.⁶³⁻⁶⁷ Research groups are now starting to account for protein backbone movements as well. The SLIDE method by Kuhn and coworkers^{68,69} as well as the GLIDE software from Schrodinger, Inc.⁷⁰ sample both side chain and backbone flexibility. FLIPDock is the first method that allows for fully flexible ligand and protein docking using a data structure termed the Flexibility Tree, which can even account for full, rigid, domain movements along with backbone and side chain motions.⁷¹ MD simulations are also employed to account for full protein flexibility.^{72,73} Accurate MD simulations can be computationally expensive but may be appropriate to dock a small number of compounds, as reviewed by Alonso et al.⁷⁴

A final approach, and the focus of this research project, is to use an ensemble of protein structures. Models can be generated by overlaying the different conformations in the ensemble or each structure can be considered separately. Pioneering work by Knegtel and coworkers employed NMR and crystal structures to calculate a single scoring grid using geometry-weighted and energy-weighted methods for the program DOCK^{75,76}.⁷⁷ Subsequently, Broughton used a similar weight-averaging method with FLOG to create composite grids derived from protein conformations taken from an MD simulation.⁷⁸ The AutoDock program has also been used to look at different ways to average grids and found that weight-averaged grids performed the best.^{79,80} Furthermore, the program FlexE⁸¹, an extension of FlexX utilized a “united protein” approach to represent an ensemble of superimposed conformations.^{82,83} The regions of the protein in good structural agreement are averaged whereas the orientations of the structurally dissimilar regions are discretely represented, similar to a rotamer library. Shoichet and co-workers recently described a method similar to FlexE that treats flexible regions of the protein as discrete conformations.⁸⁴ Their method employs collections of crystal structures and scales linearly with the number of flexible groups rather than exponentially. Multiple groups have explored docking ligands into each receptor conformation of an ensemble obtained by multiple crystal structures^{85,86}, NMR⁸⁷, normal mode analysis^{67,88}, or MD^{89,90}.

However in this technique, the score for a ligand pose is typically calculated against a single static structure. Several reviews have been published and summarize the current state of the field.^{14,57,58,91,92}

The Multiple Protein Structures Method

Carlson et al. developed a robust receptor-based pharmacophore method based on solvent mapping of an ensemble of unbound Human Immunodeficiency Virus type 1 (HIV-1) integrase protein structures to account for protein flexibility.⁹³ The use of multiple protein structures (MPS), taken from an MD simulation, NMR ensemble, or collection of crystal structures, allows a range of conformational space that can be assumed by the protein to be sampled and hence, simulates the inherent flexibility of a binding site in a computationally feasible manner. Small molecule probes are subjected to a Monte Carlo energy minimization to map energetically favorable regions, or “hot spots”, of each protein active site. The MPS are overlaid to identify the most important, chemically relevant features conserved across the conformations, or “consensus sites”. The consensus sites are then reduced to a simple pharmacophore model that can be used in virtual screening to identify potential new ligands and scaffolds. This work was experimentally verified through biological testing and shown to have a high rate of success.⁹³

The MPS method has been further validated using HIV-1 protease (HIV-1p)⁹⁴⁻⁹⁶, dihydrofolate reductase (DHFR)^{97,98}, and MDM2⁹⁹. Using HIV-1p, several parameters for defining consensus and using the pharmacophore models in screening were examined: MD simulation length, pharmacophore element size, number of required elements, and alignment mechanism. The resulting pharmacophore models were highly selective for known HIV-1p inhibitors and effectively discriminated against the chemically similar non-inhibitors in database searches.⁹⁴ The use of MPS identified key features of known protease inhibitors from an apo protein structure, which is particularly dramatic as HIV-

1p undergoes a large rearrangement upon binding inhibitors and substrates. In a follow-up study, the MPS method was applied to two additional unbound HIV-1p structures.⁹⁵ Employing three similar but unique starting structures to obtain three independent MD trajectories of the unbound HIV-1p resulted in nearly identical pharmacophore models. This demonstrated that the MPS method is not overly dependent on a specific starting conformation or particular MD trajectory.

Crystal structures of DHFR from different species were used to examine whether MPS pharmacophore models could encode specificity between similar binding sites.⁹⁷ The models were shown to be capable of identifying high-affinity, species-specific DHFR inhibitors over weaker, general inhibitors, supporting the hypothesis that including protein flexibility improves SBDD. Furthermore, a second DHFR study demonstrated that longer MD simulations enhanced the performance of MPS models; again, this reveals that protein flexibility is needed to model protein-ligand binding interactions accurately.⁹⁸

Additionally, snapshots from MD simulations of unbound MDM2 and the p53-MDM2 complex were used to create MPS pharmacophore models of the binding cleft of MDM2. Of the 27 compounds identified using the MPS model, 23 were experimentally tested and five were shown to inhibit MDM2 (22% success rate).⁹⁹ An important feature of this study was that all five chemical scaffolds are unique and share no common substructures with reported inhibitors of MDM, as the MPS technique is intended to push the boundaries of chemical space and overcome the cross-docking problem. Chapters 4 and 5 discuss further refinement of the MPS method.

1.4 HIV-1 Protease as a Test Case

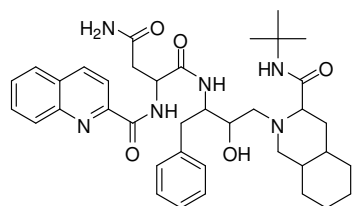
HIV/AIDS

HIV-1p is a viral enzyme critical to continuing the life cycle of HIV, the retrovirus responsible for Acquired Immunodeficiency Syndrome (AIDS).¹⁰⁰ Presently, HIV/AIDS affects approximately 40 million people worldwide and resulted in approximately 3 million deaths in 2006.¹⁰¹ Consequently, HIV-1p has been extensively studied over the years and is regarded as a key drug target. The first structure of HIV-1p was solved in 1989 by Navia et al.¹⁰² from Merck laboratories and a second shortly after by Kent and coworkers¹⁰³ at the NIH. A diverse set of experimentally and theoretically determined HIV-1p structures is now available, including over 300 crystal structures and 3 NMR structures, allowing for a thorough evaluation of new structure-based methods. Furthermore, hundreds of inhibitors have been reported throughout the literature.

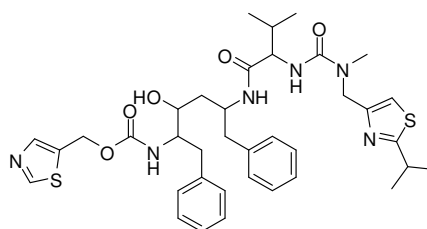
HIV-1p is responsible for the cleavage of the *gag* and *gag-pol* precursor polyproteins in at least nine different sites to form mature viral proteins.¹⁰⁴ It has been demonstrated that budded immature viral particles cannot undergo maturation to an infective form if HIV-1p is catalytically inactive.¹⁰⁵ Currently, there are eight peptidic drugs on the market that competitively inhibit HIV-1p by mimicking substrates and the transition state of cleavage: saquinavir, ritonavir, indinavir, nelfinavir, amprenavir, lopinavir, atazanavir, and fosamprenavir, and two non-peptidic competitive active-site inhibitors available: tipranavir and darunavir (Figure 1.2).^{106,107} The non-peptidic protease inhibitors (PIs) displace a conserved water molecule that coordinates substrates (and peptide PIs) to the protease flap tips and form direct hydrogen bonds to the flap region.¹⁰⁸ The structural water is a key difference between mammalian and HIV proteases, and this displacement may be one reason why non-peptidic PIs are very selective for HIV proteases.^{109,110} Saquinavir was the first protease inhibitor (PI) to be approved by the FDA and became available to the public in December of 1995. The

discovery of novel PIs is still a very active area of research due to the associated toxicity, poor pharmacokinetic properties, and resistance that has developed to the existing drugs.

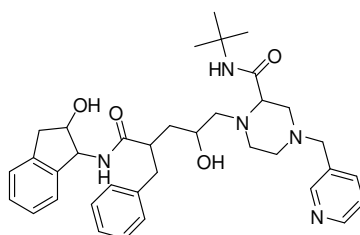
Figure 1.2. Structures of the ten HIV-1p inhibitors currently on the market.



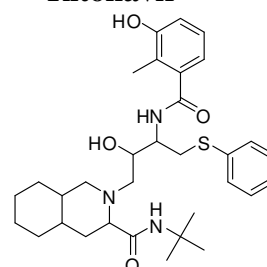
Saquinavir



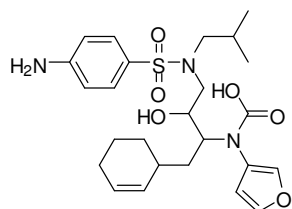
Ritonavir



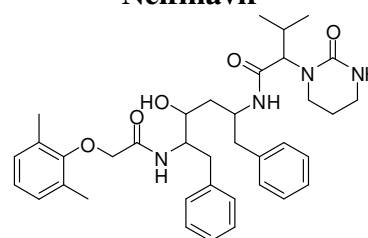
Indinavir



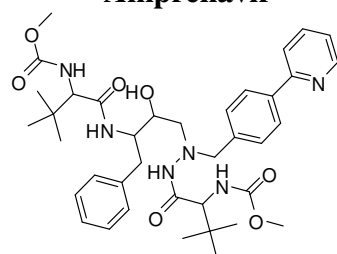
Nelfinavir



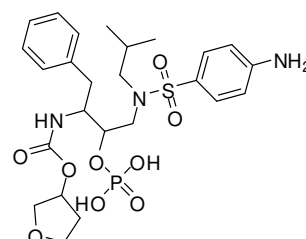
Amprenavir



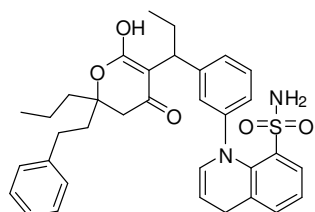
Lopinavir



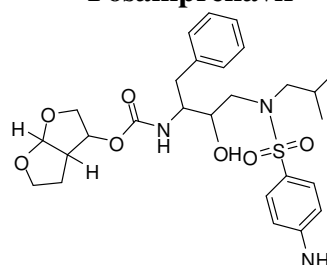
Atazanavir



Fosamprenavir



Tipranavir

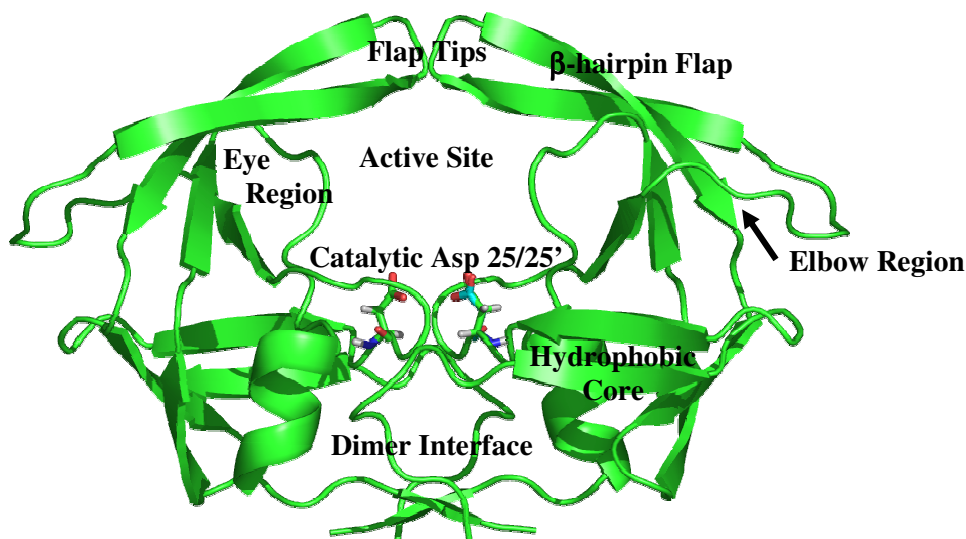


Darunavir

HIV-1p Dynamics

HIV-1p is found as a C_2 -symmetric dimer; each monomer is comprised of 99 residues. The active site is located at the dimer interface and contains two catalytic aspartate residues (25, 25'), which interact directly with inhibitors and substrates.^{104,111} This region is covered by two glycine-rich, anti-parallel β -hairpins, referred to as the “flaps”, consisting of residues 43-58 (Figure 1.3). The conformational behavior of the flap region of HIV-1p has been extensively studied in the last few years, as reviewed by Hornak and Simmerling.¹¹² It is thought that the largely populated states are closed, semi-open, and open, with the semi-open conformation being the most prevalent in the apo state. Recently, two groups have demonstrated through Langevin Dynamics (LD) simulations extensive sampling of the multiple flap conformations.^{113,114} Both groups also showed that upon introduction of a ligand into the active site of a semi-closed conformation, the flaps close down upon the ligand and replicate key hydrogen bonds seen in bound crystal structures.^{115,116}

Figure 1.3. A cartoon representation of HIV1-p in the semi-open conformation¹¹⁷; catalytic residues 25/25' are shown in stick representation. The location of key features discussed in Chapters 1-6 are indicated to orient the reader.

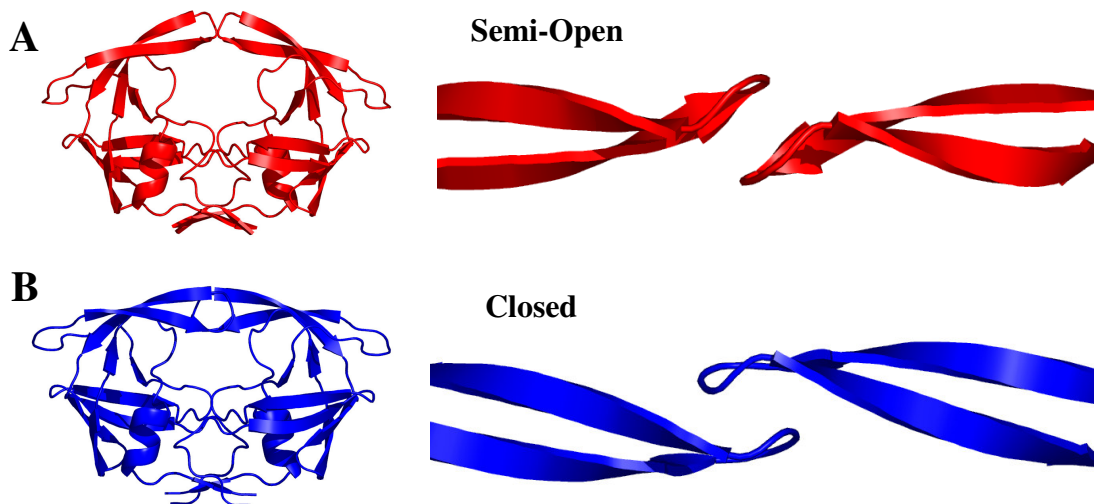


A ligand can only access the active site through the open conformation¹¹⁸, although the mechanism allowing such entry is still unknown. However, theoretical simulations have shown that there is a major rearrangement of the hydrophobic core region (residues 5, 11, 13, 15, 22, 24, 33, 36, 38, 62, 64, 75, 77, 85, 89, 90, 93, and 97) that helps facilitate the opening of the cavity.¹¹⁹ Additionally, Scott and Schiffer reported that curling of the flap tips created a “hydrophobic cluster” in the flap-recognition pocket (“eye” region of Figure 1.3, residues 79-81) of the same monomer and induced flap opening.¹²⁰ This curling mechanism, allowing for hydrophobic contacts between the flap tips and “eye” region residues, has been demonstrated to drive flap opening by additional groups.^{115,121-123} However, the open conformations sampled in the Hornak et al. simulation were not preceded by flap curling.¹¹³ In fact, the authors state that the flap tips actually moved away from the active site of the protease. NMR data has shown that the flap region of HIV-1p is in equilibrium among an ensemble of semi-open states and undergoes conformational changes to the open form on a microsecond time scale, while the flap tips appear to fluctuate on a subnanosecond time scale.^{124,125} The dynamics of the flaps and flap tips revealed by NMR is consistent with either observation. Two drug-susceptible apo crystal structures were recent reported in two distinctly different apo conformations that the authors term curled and open.¹²⁶ However, their curled structure is actually assuming the semi-open state, according to current literature definitions, and should not be confused with “curled” flap tips.

Upon ligand binding, multiple conformational changes occur in the protease.^{111,127} There is an inward rotation of each monomer, and the flaps assume a closed conformation over the active-site cavity (5-7 Å shift from apo form). In addition, the “handedness” of the flap tip (residues 49-52) orientation reverses upon closing.¹¹³⁻¹¹⁶ The conformational changes are demonstrated in Figure 1.4. The closing motion of the flaps has been correlated with the substrate movement towards the catalytic residues Asp

25/25', coordination of a water molecule, and positioning the substrate for optimal enzymatic activity.^{121,128,129}

Figure 1.4. Two conformational states assumed by HIV-1p. The flap tips change handedness upon flap closure. **(A)** Semi-open conformation illustrated using the crystal structure 1HHP¹¹⁷. **(B)** Closed state demonstrated using the crystal structure 1PRO¹³⁰.



Resistance

Drug resistant mutations pose a significant challenge in treating HIV/AIDS. In the presence of resistant strains, the potency of the current PIs against HIV-1p drastically decreases.¹³¹ The resistance is typically associated with specific mutations in the amino acid sequence that reduce the protease's affinity for each inhibitor. Perno et al. found that the active site residues D25-D29, tip of the flap residues G49-G52, and "turn" residues G78-P81 and G86-R87 are conserved (i.e. residues not associated with resistance in the presence of HIV-1p inhibitors).¹³² This suggests that these residues are essential for the activity and/or structural stabilization of HIV-1p.

Upon the introduction of drug therapies, the first mutations to appear on the protease are referred to as primary mutations. Typically such mutations are located in the active site and directly interfere with PI binding. However, primary mutations have been documented that are distal to the active site, and the authors hypothesize that the

mutations distort the geometry of the binding site.¹³³ In addition to reducing the potency of PIs, active site mutations can also detrimentally affect substrate binding and reduce the catalytic activity of the protease. In order for the protease to combat this issue, compensatory, or secondary mutations, typically emerge at non-active site residues. The secondary mutations appear to change the dynamics of flap opening and closing.^{121,123,134,135} Clemente et al. postulated that the transient binding of flexible substrates would not be affected by the altered dynamics as greatly as “rigid” inhibitor binding.¹³⁵ Additionally, compensatory mutations have been found that are located near the cleavage sites of the Gag substrates.¹³⁶⁻¹³⁸ These mutations appear to alter the conformation of the substrate to improve interactions with the mutant active site.

The first crystal structure was solved of an apo multi-drug resistant (MDR) HIV-1p strain at 1.8 Å and revealed an expanded active site cavity.¹³⁹ The authors determined that the 3.0 Å expansion of the site was primarily attributed to the shorter side chains from the V82A and I84V mutations and a larger distance between residues I50 and P81 resulting in a more open flap conformation. The expanded active site cavity results in decreased binding affinity of the PIs due to a loss of van der Waals contacts and hydrogen bonds. Ohtaka and Freire propose that more flexible HIV-1p inhibitors, or “adaptive inhibitors”, are able to move to accommodate the structural changes associated with the resistance mutations.¹³¹ Adaptive inhibitors show less of a potency loss than conformationally constrained inhibitors.

Inhibition Mechanisms

All current PIs on the market bind in the active site of HIV-1p and are competitive inhibitors of the natural substrates. Hence, HIV-1p cannot cleave the substrates into functional proteins, preventing further maturation and proliferation of HIV.¹¹¹ Novel inhibition mechanisms are needed to overcome the resistance associated with existing PIs. Different mechanisms have been proposed in the literature and are

summarized below. Our contribution to the discovery of PIs through a novel mechanism of action is presented in Chapter 6.

The substrate envelope theory stems from the observation that HIV-1p is a promiscuous enzyme that does not recognize specific amino acid sequences for substrate cleavage.¹⁴⁰⁻¹⁴³ Instead, the substrates assume a conserved shape upon interacting with HIV-1p known as the “substrate envelope”; the amino acids upstream of the cleavage site form a toroidal shape while the downstream residues adopt an extended conformation providing space for essential interactions with water molecules.^{140,144,145} Current PIs are shorter than the natural substrates and contain hydrophobic moieties that protrude outside of this “envelope”, interacting with the side chains of the active site residues. As such, a mutation in these residues (e.g. Val 82 or Ile 84) can result in decreased van der Waals interactions and diminish the inhibitor binding affinity.¹³⁹ It is hypothesized that designing inhibitors that assume a similar shape rather than chemical functionality to the natural substrate may be one mode for overcoming resistance. In fact, amprenavir predominately fits the substrate envelope and displays a different mutation profile than other PIs.¹⁴¹

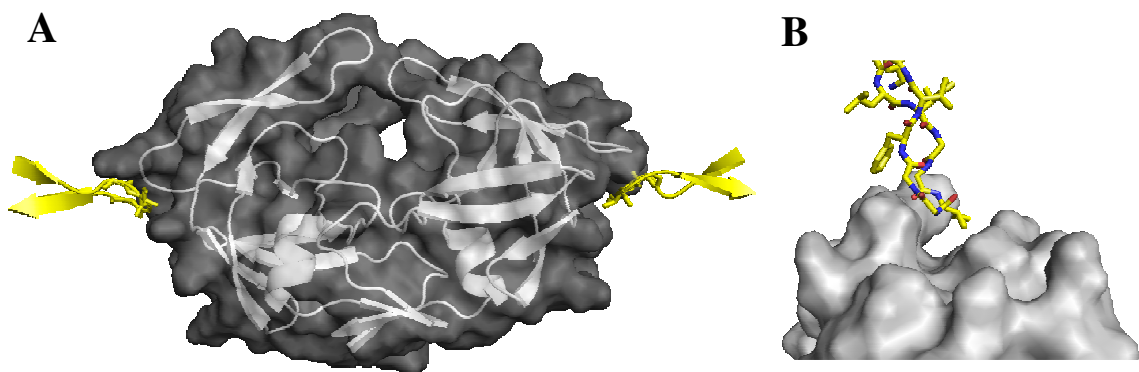
An additional element that differs between substrate and PI binding is the degree of hydrogen bonding with backbone atoms, the “backbone binding” concept. There are a significant amount of backbone hydrogen bonds formed between the substrate and HIV-1p that are not present in PI binding.¹⁴⁵ Mutations in backbone atoms cannot occur; hence, these interactions will likely be maintained and could potentially evade drug resistance. The “backbone binding” concept was validated by the discovery and development of the recently approved nonpeptidic PI, darunavir, as reviewed by Ghosh et al.¹⁴⁶ A co-crystal structure of darunavir bound to HIV-1p demonstrated hydrogen-bonding between the bis-tetrahydrofuran oxygens and Asp 29 and Asp 30 backbone amides and between the aniline moiety and the carbonyl oxygen of Asp 30'.^{146,147} Experimental studies showed exceptional broad-spectrum activity against a large panel of

MDR HIV-1 strains¹⁴⁷, and darunavir was approved by the FDA in June 2006 as the first treatment for drug-resistant HIV.

A third technique is targeting an allosteric site on HIV-1p. Various groups have identified anti-correlated motion between the flap and elbow (residues 35/35'-42/42') regions through normal mode analysis and MD simulations.^{122,128,148,149} The closed and semi-open conformations are distinctly different in this region. A follow-up study demonstrated that restricting movement of the elbow region resulted in immobilization of the flaps.¹⁵⁰ Although harmonic restraints were used to control the HIV-1p flap region, such results establish a theory of allosteric control, which may be manipulated in drug design.

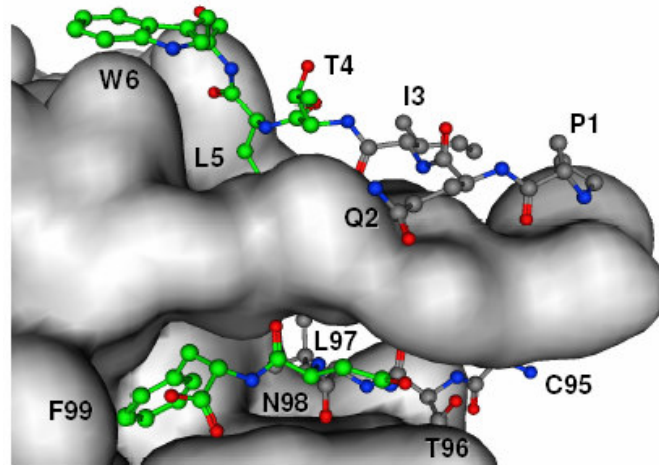
A recent crystal structure of the apo form of a multidrug-resistant HIV-1p (PDB ID: 1TW7) has the flaps displaced wider and more open than other semi-open structures.¹⁵¹ Crystal packing creates contacts between the flap tips in the neighboring unit cell and the elbow region of HIV-1p, as shown in Figure 1.5, and it has been proposed that this might be experimental corroboration of allosteric control. Recent LD simulations by Simmerling and coworkers examined this structure and showed that in the absence of the crystal contacts, 1TW7 takes the typical semi-open conformation.¹⁴⁹ Furthermore, they replicated all the packing neighbors (e.g. residues) within 15 Å of the central dimer and were able to demonstrate that the central HIV-1p remained in the open conformation. The crystal packing contacts suggest the possibility allosteric control; however, to date an “elbow” inhibitor has never been experimentally verified in the literature.

Figure 1.5. Structure 1TW7 (grey cartoon and surface representation). There are many contacts between the neighboring unit cells (not shown for clarity), but contact in the elbow regions is limited (yellow chain). (A) Front view. (B) View looking down elbow cavity.



A final mode of action currently being pursued is to disrupt HIV-1p dimerization by targeting the interdigitating, highly conserved β -sheet region of the C- and N- termini and the bulk contact region in the core of the protein.¹⁵² Todd et al. showed through calorimetry measurements that approximately 75% of the free energy of dimerization comes from the β -sheet region⁹³, shown in Figure 1.6. The HIV-1p dimer is in equilibrium with the monomeric form in the cell; multiple groups have experimentally verified that HIV-1p can be inhibited by blocking the dimerization event.¹⁵³⁻¹⁵⁷ Several groups are employing a peptidomimetic approach to design inhibitors based on residues 1-6 and 95-99 and tethering the peptides together using flexible and rigid linkers.^{154,156-161} The Chmielewski group has been the most successful and has shown through cross-competitive inhibition assays that the compounds are binding at an allosteric site.¹⁶²⁻¹⁶⁴ However, none of the compounds showing inhibition activity have been structurally verified to demonstrate that the binding is actually occurring at the dimer interface.

Figure 1.6. The β -sheet interface. Monomer A residues are shown in surface representation while monomer B residues are shown as ball and sticks. The N-terminal peptide (P1-W6) is at the top, and the C-terminal peptide (C95-F99) is at the bottom. Green residues are found to be important in designing mimics. (Figure courtesy of Jerome Quintero.)



1.5 Theory

Molecular Mechanics

Numerous computational techniques are utilized to study the behavior of biomolecular systems. The most accurate representation of a molecule would employ a quantum mechanical treatment of every atom; however, such a high level of accuracy is not computationally feasible for large biomolecular systems. A simpler yet reasonably accurate model of a molecule can be obtained using molecular mechanics (MM). MM provides an energetic description of a set of atoms treating the atoms as points without explicit electrons. The set of equations that relates the potential energy to the atomic positions is called a force field. A crucial component of the force field is a set of parameters that are plugged into the force field equation to describe the system as accurately as possible.¹⁶⁵

MM force fields have been derived to model the behavior of the biomolecules and enable the potential energy of the system to be calculated fairly accurately. A variety of

MM force fields have been developed; some of the most commonly employed include AMBER (Assisted Model Building with Energy Refinement)¹⁶⁶, CHARMM (Chemistry at Harvard Molecular Mechanics)¹⁶⁷, GROMOS (Groningen Molecular Simulation)¹⁶⁸, and OPLS (Optimized Potential for Liquid Simulations)¹⁶⁹. Each atom is represented as a sphere and assigned a van der Waals radius and typically, a constant net charge. To reduce the complexity of a system, a “united-atom” approach may be used where groups of atoms are represented as a single particle. Different hybridization states and element types may be modeled by using a discrete set of atom types. The potential energy of the system is described as the sum of the bonded (e.g. covalent) and nonbonded (e.g. electrostatic, dipole, and van der Waals) interactions. For large biomolecular systems, the equations describing such interactions include numerous approximations to allow for a rapid calculation of the total energy; hence, there is often a trade-off between speed and accuracy.¹⁷⁰

In the Carlson Lab, the MM simulation suite AMBER is used, along with the AMBER force field described by the potential energy function provided below.¹⁶⁶

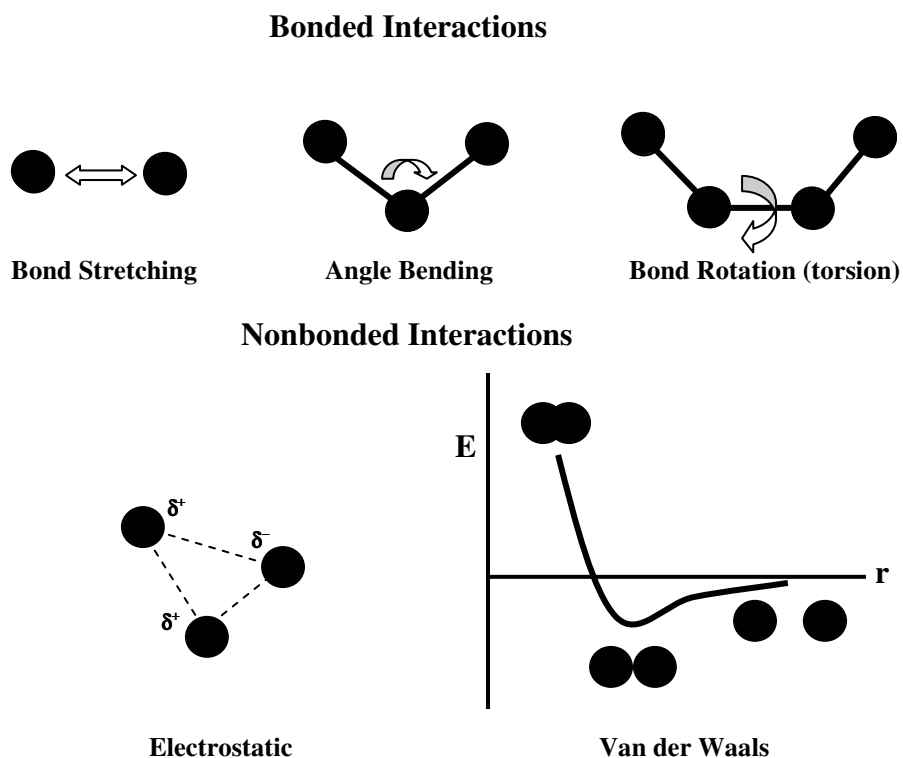
$$E = \sum_{bonds} K_b (b - b_{eq})^2 + \sum_{angles} K_\theta (\theta - \theta_{eq})^2 + \sum_{dihedrals} \frac{V_n}{2} [1 + \cos(n\phi - \gamma)] + \sum_{i < j} \left[\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} + \frac{q_i q_j}{\epsilon r_{ij}} \right] \quad (1.1)$$

where b and θ are the bond length and bond angle, respectively, and K_b and K_θ are the force constants for the bond and bond angles. ϕ is the dihedral angle and V_n is the corresponding force constant (commonly referred to as the torsional barrier height); the phase angle γ values are either 0° or 180° . The nonbonded part of the potential is represented by van der Waals (A_{ij}) and London dispersion terms (B_{ij}), partial atomic charges (q_i and q_j), and the dielectric constant ϵ .

The first three terms of the equation represent the bonded interactions. The potential energy of both the bond stretching and bending terms is modeled as a simple harmonic oscillator while the torsional component is approximated as a periodic function.

The nonbonded term is modeled using Coulomb's Law for electrostatics and the Lennard-Jones Potential for van der Waals interactions. Hydrogen bonds are not explicitly represented, rather they are accounted for through appropriate parameterization of Lennard-Jones and Coulomb Interactions.¹⁷⁰ Figure 1.7 schematically illustrates the key components of the MM force field. The constants in the potential energy terms such as spring stiffnesses, equilibrium distances, torsional barriers and periodicities, partial charges, and Lennard-Jones coefficients have been parameterized using experimental data such as crystallographic bond lengths, vibrational frequencies, and solvation free energies or by high-level quantum mechanical calculations. One limitation to the current force fields is the neglect of polarizable effects from the environment and is an area of ongoing research.¹⁷¹

Figure 1.7. Bonded and nonbonded components of a typical MM force field.



Molecular Dynamics

MD simulations are commonly used to study the dynamics of biomolecules and aid in the refinement of X-ray crystallography and nuclear magnetic resonance (NMR) structures. The first MD simulation of a biomolecule, the bovine pancreatic trypsin inhibitor, has carried out in 1977 by McCammon et al.¹⁷² paving the way for what has now become a routine tool for computational chemists. MD is a deterministic method that calculates the time-dependent behavior from an initial set of velocity distributions and atomic coordinates through the integration of Newton's equations of motion. Newton's second law of motion (Eq. 1.2) relates the force along a molecular trajectory to the mass and acceleration; hence, for a given, initial starting point, the system's future position and momenta can be calculated over time.¹⁷⁰

$$F_i = m_i a_i = -\nabla_i V \longrightarrow -\frac{dV}{dr_i} = m_i \frac{d^2 r_i}{dt^2} \quad (1.2)$$

where F_i is the force exerted on a particle i , m_i is the mass of the particle i , a_i is the acceleration of particle i , and V is the potential energy of the system, usually calculated by an MM force field.

For all but the simplest systems there is not an analytical solution to the equations of motion, as the potential energy is a function of the 3N coordinates of all the atoms in the system. Multiple numerical algorithms are available for integrating the equations of motion and include the Verlet, Velocity Verlet, Leap-frog, and Beeman's algorithms.¹⁷⁰ A proper time step for the integration is very important; it must be an order of magnitude smaller than the vibration of bonds that involve hydrogen atoms, the fastest motions of the system (approximately 10 fs), for the integration algorithm to be stable. Alternatively, if the time step is too small, the amount of phase space sampled is affected. Typically, 1 fs is employed, although longer time steps may be used if lengths of bonds involving hydrogens are constrained. Biochemical simulations are often performed using a single

molecule isolated in a box and solvated with explicit water; the assumption is made that the number of particles in the box remains constant. The current water models include ST2, SPC, TIP3P, TIP4P, and TIP5P, and all treat water as a rigid molecule.¹⁷³ Periodic boundary conditions are applied to approximate an infinitely tiled system. The results obtained from an MD simulation are dependent on the MM force field and approximations employed; hence, proper care must be taken when setting up a simulation.

Langevin Dynamics

When solvent molecules are not explicitly represented, an alternative approach to MD is the use of LD. The Langevin equation is a stochastic differential equation, which incorporates two force terms in Newton's second law to approximate the loss of the biomolecule's interaction with solvent. The first is a frictional term and is related to the collision frequency while the second is a *random* force. The solvent effects are approximated using a frictional drag as well as random jolts associated with the thermal motions of the solvent molecules. Langevin's equation for the motion of particle i is provided below.¹⁷⁰

$$m_i \frac{d^2 r_i}{dt^2} = F_i \{r_i(t)\} - \xi \frac{dr_i(t)}{dt} + R_i(t) \quad (1.3)$$

where R is the random force and ξ is the frictional coefficient.

Even though the overall integrity of biomolecules are better maintained with MD than with LD, LD has a significant speed advantage over MD and allows for longer simulations. This is because water molecules are implicitly represented, hence there are fewer atoms in the system. Furthermore, LD can sample a greater amount of conformational space than MD in a shorter time scale due to the decrease of friction.

Metropolis Monte Carlo

Monte Carlo (MC) simulations are stochastic methods based on the use of random numbers to sample conformational space rather than deterministic algorithms as in MD simulations. As such, the system literally jumps from conformation to conformation, overcoming large energy barriers, which allows for greater sampling and the potential to find the global minimum and not simply a local minimum. However, because MC does not sample a realistic dynamics trajectory, it cannot provide time-dependent properties. Each conformation depends only on the previous conformation. Typically, MC simulations are performed under conditions of the canonical ensemble (constant number of particles, volume, and temperature) unlike MD, which traditionally uses microcanonical ensemble conditions (constant number of particles, volume, and energy).¹⁷⁰

The potential energy associated with the initial conformation is calculated (E_1), and the system is perturbed to a new conformation and the energy recalculated (E_2). If $E_2 < E_1$, the new conformation is retained, but if $E_2 > E_1$, the Boltzmann factor of the energy difference is determined. The Boltzmann factor is compared to a random number between zero and one, and if the random number is higher, the conformation is discarded. However, if the random number is lower, the conformation is kept. Therefore, the smaller the uphill move and hence, the Boltzmann factor, the greater the probability the conformation will be accepted. This process continues for a set number of steps or until energy convergence is reached.¹⁷⁰ Low-temperature MC minimization (MCM) techniques are widely used to search for low-energy local minima. One such example is the use of MCM to minimize small molecule probes (e.g. benzene) to map the most energetically favorable interactions with a protein-binding site.^{93,174}

Molecular Docking

Molecular docking is a method used to predict the binding mode between a ligand and biomolecule (typically a protein). It was introduced in 1982 by Kuntz et al.⁷⁵ and has now become an integral part of many drug discovery programs. Docking can reduce the time and cost required to attain a clinical candidate by allowing for rapid screening of databases containing millions of compounds in the lead discovery phase. Furthermore, rationally evaluating where and how to modify the lead compound can minimize the number of compounds that need to be synthesized in the optimization phase. However, a drawback of this technique is that structural information is required. Frequently used docking programs include DOCK⁷⁵, AutoDock¹⁷⁵, FlexX⁸², GOLD¹⁷⁶, ICM¹⁷⁷, and GLIDE^{178,179}.

A typical docking protocol is comprised of two components: the search algorithm and scoring function. The role of the search algorithm is to generate low-energy ligand conformations and predict the orientation of each ligand conformation within the receptor's active site. The predicted ligand conformation and orientation is termed the "pose". A variety of search algorithms is available and consists of random or stochastic techniques such as Monte Carlo, genetic algorithms, and tabu search methods and simulation methods including molecular dynamics and pure energy minimizations.^{180,181} A rigorous search algorithm would account for all possible binding poses by allowing both the protein and ligand to be flexible; however, this is not computationally feasible due to the enormity of the conformational space that must be searched. A common approximation in the early search algorithms was to keep both the protein and ligand rigid and later advanced to using pre-generating ligand conformations to model ligand flexibility. However, neither approach accounts for the flexibility of the protein or the induced fit that may occur upon ligand binding. Developing novel and computationally feasible techniques to model the conformational changes that can occur in both the protein and ligand is an active area of research, as previously discussed.

The role of the scoring function is to evaluate each pose by predicting the free energy of binding (binding affinity) of a ligand as a function of its position in a protein binding site.¹⁸² The scoring function employed is generally either a molecular mechanics force field, empirical free energy scoring function, or knowledge based function. However, several groups are now using “consensus” scoring techniques that combine information from the different algorithms and have demonstrated success over using the algorithms individually.¹⁸³⁻¹⁸⁵ As many simplifications and assumptions must be made to increase the speed of the scoring function resulting in a loss of accuracy, to date no method is able to predict the free energy of binding correctly. However, the relative ranking between compounds in a single screen can provide useful information, even though the absolute numbers are not meaningful.

Scoring functions can be utilized using different techniques. The first is to use a rigorous scoring function to both direct the search algorithm and rank the resulting poses. The second method uses a “reduced” scoring function initially for rapid evaluation during the search step and a rigorous scoring function to rank the resulting poses.¹⁸¹ Furthermore, the top scoring ligand poses may also be re-ranked using a more sophisticated, but slower, scoring function.

As with any technology, there are identified limitations associated with molecular docking. First, the employed structures are not exact (e.g. may contain experimental errors), and crystal structures represent an average structure. The conformational space that must be sampled is enormous, and the molecules usually undergo conformational changes upon association. The scoring functions contain many approximations, as accurate calculations are too slow for most applications, and solvent-related terms are typically ignored. A review by Taylor et al. suggests that the best docking techniques are hybrid methods, which incorporate multiple search and scoring algorithms.¹⁸¹ The field of molecular docking is a very active area of research, and there is a plethora of reviews

available on the status of protein-ligand docking algorithms and the utilized scoring functions.^{180,181,186-188}

1.6 Specific Aims

The goal of this project was to optimize the MPS method and apply it, and other computational techniques, to study HIV-1p and discover novel inhibitors to overcome existing resistance. My specific aims included:

1. To develop a tool to properly overlay different conformations of the same protein based on the rigid regions of the structure (key to identifying “regions of consensus” for MPS models).
2. To probe the source of MPS using conformational snapshots of an apo structure across an MD simulation, a bound NMR ensemble, and a collection of bound crystal structures.
3. To quantify ligand overlap with MPS pharmacophore models by incorporating a ranking function using DOCK.
4. To apply the MPS method and other computational techniques to study HIV-1 protease and investigate a novel inhibition mechanism by modulating its conformational behavior.

1.7 References

1. Koshland, D. E., Jr.; Ray, W. J., Jr.; Erwin, M. J. Protein Structure and Enzyme Action. *Fed Proc* **1958**, *17*, 1145-1150.
2. Jorgensen, W. L. Rusting of the Lock and Key Model for Protein-Ligand Binding. *Science* **1991**, *254*, 954-955.
3. Radford, S. E.; Buck, M.; Topping, K. D.; Dobson, C. M.; Evans, P. A. Hydrogen Exchange in Native and Denatured States of Hen Egg-White Lysozyme. *Proteins* **1992**, *14*, 237-248.
4. Bai, Y.; Sosnick, T. R.; Mayne, L.; Englander, S. W. Protein Folding Intermediates: Native-State Hydrogen Exchange. *Science* **1995**, *269*, 192-197.
5. Chamberlain, A. K.; Handel, T. M.; Marqusee, S. Detection of Rare Partially Folded Molecules in Equilibrium with the Native Conformation of Rnaseh. *Nat Struct Biol* **1996**, *3*, 782-787.
6. Hilser, V. J.; Dowdy, D.; Oas, T. G.; Freire, E. The Structural Distribution of Cooperative Interactions in Proteins: Analysis of the Native State Ensemble. *Proc Natl Acad Sci U S A* **1998**, *95*, 9903-9908.
7. Miller, D. W.; Dill, K. A. Ligand Binding to Proteins: The Binding Landscape Model. *Protein Sci* **1997**, *6*, 2166-2179.
8. Ma, B.; Kumar, S.; Tsai, C. J.; Nussinov, R. Folding Funnels and Binding Mechanisms. *Protein Eng* **1999**, *12*, 713-720.
9. Tsai, C. J.; Kumar, S.; Ma, B.; Nussinov, R. Folding Funnels, Binding Funnels, and Protein Function. *Protein Sci* **1999**, *8*, 1181-1190.
10. Carlson, H. A.; McCammon, J. A. Accommodating Protein Flexibility in Computational Drug Design. *Mol Pharmacol* **2000**, *57*, 213-218.
11. Luque, I.; Freire, E. Structural Stability of Binding Sites: Consequences for Binding Affinity and Allosteric Effects. *Proteins* **2000**, *Suppl 4*, 63-71.
12. Ma, B.; Shatsky, M.; Wolfson, H. J.; Nussinov, R. Multiple Diverse Ligands Binding at a Single Protein Site: A Matter of Pre-Existing Populations. *Protein Sci* **2002**, *11*, 184-197.
13. Volkman, B. F.; Lipson, D.; Wemmer, D. E.; Kern, D. Two-State Allosteric Behavior in a Single-Domain Signaling Protein. *Science* **2001**, *291*, 2429-2433.

14. Carlson, H. A. Protein Flexibility Is an Important Component of Structure-Based Drug Discovery. *Curr Pharm Des* **2002**, *8*, 1571-1578.
15. Jeffery, C. J. Molecular Mechanisms for Multitasking: Recent Crystal Structures of Moonlighting Proteins. *Curr Opin Struct Biol* **2004**, *14*, 663-668.
16. Finocchiaro, G.; Wang, T.; Hoffmann, R.; Gonzalez, A.; Wade, R. C. DSMM: A Database of Simulated Molecular Motions. *Nucleic Acids Res* **2003**, *31*, 456-457.
17. Chiang, R. A.; Meng, E. C.; Huang, C. C.; Ferrin, T. E.; Babbitt, P. C. The Structure Superposition Database. *Nucleic Acids Res* **2003**, *31*, 505-510.
18. Gerstein, M.; Krebs, W. A Database of Macromolecular Motions. *Nucleic Acids Res* **1998**, *26*, 4280-4290.
19. Krebs, W. G.; Gerstein, M. The Morph Server: A Standardized System for Analyzing and Visualizing Macromolecular Motions in a Database Framework. *Nucleic Acids Res* **2000**, *28*, 1665-1675.
20. Echols, N.; Milburn, D.; Gerstein, M. Molmovdb: Analysis and Visualization of Conformational Change and Structural Flexibility. *Nucleic Acids Res* **2003**, *31*, 478-482.
21. Gerstein, M.; Echols, N. Exploring the Range of Protein Flexibility, from a Structural Proteomics Perspective. *Curr Opin Chem Biol* **2004**, *8*, 14-19.
22. Freer, S. T.; Kraut, J.; Robertus, J. D.; Wright, H. T.; Xuong, N. H. Chymotrypsinogen: 2.5-Angstrom Crystal Structure, Comparison with Alpha-Chymotrypsin, and Implications for Zymogen Activation. *Biochemistry* **1970**, *9*, 1997-2009.
23. Gerstein, M.; Altman, R. B. Average Core Structures and Variability Measures for Protein Families: Application to the Immunoglobulins. *J Mol Biol* **1995**, *251*, 161-175.
24. Irving, J. A.; Whisstock, J. C.; Lesk, A. M. Protein Structural Alignments and Functional Genomics. *Proteins* **2001**, *42*, 378-382.
25. Wriggers, W.; Schulten, K. Protein Domain Movements: Detection of Rigid Domains and Visualization of Hinges in Comparisons of Atomic Coordinates. *Proteins* **1997**, *29*, 1-14.
26. Shatsky, M.; Nussinov, R.; Wolfson, H. J. Flexible Protein Alignment and Hinge Detection. *Proteins* **2002**, *48*, 242-256.

27. Kotlovyyi, V.; Nichols, W. L.; Ten Eyck, L. F. Protein Structural Alignment for Detection of Maximally Conserved Regions. *Biophys Chem* **2003**, *105*, 595-608.
28. Jewett, A. I.; Huang, C. C.; Ferrin, T. E. Minrms: An Efficient Algorithm for Determining Protein Structure Similarity Using Root-Mean-Squared-Distance. *Bioinformatics* **2003**, *19*, 625-634.
29. Alexandrov, V.; Gerstein, M. Using 3d Hidden Markov Models That Explicitly Represent Spatial Coordinates to Model and Compare Protein Structures. *BMC Bioinformatics* **2004**, *5*, 2.
30. Ye, Y.; Godzik, A. Database Searching by Flexible Protein Structure Alignment. *Protein Sci* **2004**, *13*, 1841-1850.
31. Schneider, T. R. A Genetic Algorithm for the Identification of Conformationally Invariant Regions in Protein Molecules. *Acta Crystallogr D Biol Crystallogr* **2002**, *58*, 195-208.
32. Schneider, T. R. Domain Identification by Iterative Analysis of Error-Scaled Difference Distance Matrices. *Acta Crystallogr D Biol Crystallogr* **2004**, *60*, 2269-2275.
33. Nichols, W. L.; Rose, G. D.; Ten Eyck, L. F.; Zimm, B. H. Rigid Domains in Proteins: An Algorithmic Approach to Their Identification. *Proteins* **1995**, *23*, 38-48.
34. Nichols, W. L.; Zimm, B. H.; Ten Eyck, L. F. Conformation-Invariant Structures of the Alpha1beta1 Human Hemoglobin Dimer. *J Mol Biol* **1997**, *270*, 598-615.
35. Taylor, W. R.; Orengo, C. A. Protein Structure Alignment. *J Mol Biol* **1989**, *208*, 1-22.
36. Subbiah, S.; Laurents, D. V.; Levitt, M. Structural Similarity of DNA-Binding Domains of Bacteriophage Repressors and the Globin Core. *Curr Biol* **1993**, *3*, 141-148.
37. Holm, L.; Sander, C. Protein Structure Comparison by Alignment of Distance Matrices. *J Mol Biol* **1993**, *233*, 123-138.
38. Kleywegt, G. J. Use of Non-Crystallographic Symmetry in Protein Structure Refinement. *Acta Crystallogr D Biol Crystallogr* **1996**, *52*, 842-857.
39. Shindyalov, I. N.; Bourne, P. E. Protein Structure Alignment by Incremental Combinatorial Extension (Ce) of the Optimal Path. *Protein Eng* **1998**, *11*, 739-747.

40. Krissinel, E.; Henrick, K. Protein structure comparison in 3D based on secondary structure matching (SSM) followed by C-alpha alignment, scored by a new structural similarity function. 2003, In Proceedings of the Fifth international Conference on Molecular Structural Biology, Vienna, September 3-7 (Kungl, A.J. and Kungl, P.J., eds).
41. Kolodny, R.; Koehl, P.; Levitt, M. Comprehensive Evaluation of Protein Structure Alignment Methods: Scoring by Geometric Measures. *J Mol Biol* **2005**, *346*, 1173-1188.
42. Watson, J. D.; Laskowski, R. A.; Thornton, J. M. Predicting Protein Function from Sequence and Structural Data. *Curr Opin Struct Biol* **2005**, *15*, 275-284.
43. Dunbrack, R. L., Jr. Sequence Comparison and Protein Structure Prediction. *Curr Opin Struct Biol* **2006**, *16*, 374-384.
44. Hann, M. M.; Leach, A. R.; Harper, G. Molecular Complexity and Its Impact on the Probability of Finding Leads for Drug Discovery. *J Chem Inf Comput Sci* **2001**, *41*, 856-864.
45. Oprea, T. I.; Davis, A. M.; Teague, S. J.; Leeson, P. D. Is There a Difference between Leads and Drugs? A Historical Perspective. *J Chem Inf Comput Sci* **2001**, *41*, 1308-1315.
46. Scapin, G. Structural Biology and Drug Discovery. *Curr Pharm Des* **2006**, *12*, 2087-2097.
47. Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery, *Adv Drug Deliv Rev* **1997**, *23*, 3-25.
48. Walters, W. P.; Ajay; Murcko, M. A. Recognizing Molecules with Drug-Like Properties. *Curr Opin Chem Biol* **1999**, *3*, 384-387.
49. Ajay, A.; Walters, W. P.; Murcko, M. A. Can We Learn to Distinguish between "Drug-Like" and "Nondrug-Like" Molecules? *J Med Chem* **1998**, *41*, 3314-3324.
50. Sadowski, J.; Kubinyi, H. A Scoring Scheme for Discriminating between Drugs and Nondrugs. *J Med Chem* **1998**, *41*, 3325-3329.
51. Oprea, T. I.; Allu, T. K.; Fara, D. C.; Rad, R. F.; Ostopovici, L.; Bologa, C. G. Lead-Like, Drug-Like or "Pub-Like": How Different Are They? *J Comput Aided Mol Des* **2007**, *21*, 113-119.
52. Lyne, P. D. Structure-Based Virtual Screening: An Overview. *Drug Discov Today* **2002**, *7*, 1047-1055.

53. Alvarez, J. C. High-Throughput Docking as a Source of Novel Drug Leads. *Curr Opin Chem Biol* **2004**, *8*, 365-370.
54. Jorgensen, W. L. The Many Roles of Computation in Drug Discovery. *Science* **2004**, *303*, 1813-1818.
55. Lengauer, T.; Lemmen, C.; Rarey, M.; Zimmermann, M. Novel Technologies for Virtual Screening. *Drug Discov Today* **2004**, *9*, 27-34.
56. McCammon, J. A. Target Flexibility in Molecular Recognition. *Biochim Biophys Acta* **2005**, *1754*, 221-224.
57. Teodoro, M. L.; Kavraki, L. E. Conformational Flexibility Models for the Receptor in Structure Based Drug Design. *Curr Pharm Des* **2003**, *9*, 1635-1648.
58. May, A.; Zacharias, M. Accounting for Global Protein Deformability During Protein-Protein and Protein-Ligand Docking. *Biochim Biophys Acta* **2005**, *1754*, 225-231.
59. Jiang, F.; Kim, S. H. "Soft Docking": Matching of Molecular Surface Cubes. *J Mol Biol* **1991**, *219*, 79-102.
60. Abagyan, R.; Totrov, M. High-Throughput Docking for Lead Generation. *Curr Opin Chem Biol* **2001**, *5*, 375-382.
61. Ferrari, A. M.; Wei, B. Q.; Costantino, L.; Shoichet, B. K. Soft Docking and Multiple Receptor Conformations in Virtual Screening. *J Med Chem* **2004**, *47*, 5076-5084.
62. Erickson, J. A.; Jalaie, M.; Robertson, D. H.; Lewis, R. A.; Vieth, M. Lessons in Molecular Recognition: The Effects of Ligand and Protein Flexibility on Molecular Docking Accuracy. *J Med Chem* **2004**, *47*, 45-55.
63. Leach, A. R. Ligand Docking to Proteins with Discrete Side-Chain Flexibility. *J Mol Biol* **1994**, *235*, 345-356.
64. Kairys, V.; Gilson, M. K. Enhanced Docking with the Mining Minima Optimizer: Acceleration and Side-Chain Flexibility. *J Comput Chem* **2002**, *23*, 1656-1670.
65. Frimurer, T. M.; Peters, G. H.; Iversen, L. F.; Andersen, H. S.; Moller, N. P.; Olsen, O. H. Ligand-Induced Conformational Changes: Improved Predictions of Ligand Binding Conformations and Affinities. *Biophys J* **2003**, *84*, 2273-2281.
66. Kallblad, P.; Dean, P. M. Efficient Conformational Sampling of Local Side-Chain Flexibility. *J Mol Biol* **2003**, *326*, 1651-1665.

67. Cavasotto, C. N.; Abagyan, R. A. Protein Flexibility in Ligand Docking and Virtual Screening to Protein Kinases. *J Mol Biol* **2004**, *337*, 209-225.
68. Jacobs, D. J.; Rader, A. J.; Kuhn, L. A.; Thorpe, M. F. Protein Flexibility Predictions Using Graph Theory. *Proteins* **2001**, *44*, 150-165.
69. Zavodszky, M. I.; Lei, M.; Thorpe, M. F.; Day, A. R.; Kuhn, L. A. Modeling Correlated Main-Chain Motions in Proteins for Flexible Molecular Recognition. *Proteins* **2004**, *57*, 243-261.
70. Sherman, W.; Beard, H. S.; Farid, R. Use of an Induced Fit Receptor Structure in Virtual Screening. *Chem Biol Drug Des* **2006**, *67*, 83-84.
71. Zhao, Y.; Sanner, M. F. Flipdock: Docking Flexible Ligands into Flexible Receptors. *Proteins* **2007**, *68*, 726-737.
72. Mangoni, M.; Roccatano, D.; Di Nola, A. Docking of Flexible Ligands to Flexible Receptors in Solution by Molecular Dynamics Simulation. *Proteins* **1999**, *35*, 153-162.
73. Nakajima, N.; Higo, J.; Kidera, A.; Nakamura, H. Flexible Docking of a Ligand Peptide to a Receptor Protein by Multicanonical Molecular Dynamics Simulation. *Chem Phys Lett* **1997**, *278*, 297-301.
74. Alonso, H.; Bliznyuk, A. A.; Gready, J. E. Combining Docking and Molecular Dynamic Simulations in Drug Design. *Med Res Rev* **2006**, *26*, 531-568.
75. Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. A Geometric Approach to Macromolecule-Ligand Interactions. *J Mol Biol* **1982**, *161*, 269-288.
76. Shoichet, B. K.; Kuntz, I. D. Matching Chemistry and Shape in Molecular Docking. *Protein Eng* **1993**, *6*, 723-732.
77. Knegtel, R. M.; Kuntz, I. D.; Oshiro, C. M. Molecular Docking to Ensembles of Protein Structures. *J Mol Biol* **1997**, *266*, 424-440.
78. Broughton, H. B. A Method for Including Protein Flexibility in Protein-Ligand Docking: Improving Tools for Database Mining and Virtual Screening. *J Mol Graph Model* **2000**, *18*, 247-257, 302-244.
79. Osterberg, F.; Morris, G. M.; Sanner, M. F.; Olson, A. J.; Goodsell, D. S. Automated Docking to Multiple Target Structures: Incorporation of Protein Mobility and Structural Water Heterogeneity in Autodock. *Proteins* **2002**, *46*, 34-40.

80. Zentgraf, M.; Fokkens, J.; Sotriffer, C. A. Addressing Protein Flexibility and Ligand Selectivity by "in Situ Cross-Docking". *ChemMedChem* **2006**, *1*, 1355-1359.
81. Claussen, H.; Buning, C.; Rarey, M.; Lengauer, T. Flexe: Efficient Molecular Docking Considering Protein Structure Variations. *J Mol Biol* **2001**, *308*, 377-395.
82. Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. A Fast Flexible Docking Method Using an Incremental Construction Algorithm. *J Mol Biol* **1996**, *261*, 470-489.
83. Kramer, B.; Rarey, M.; Lengauer, T. Evaluation of the Flexx Incremental Construction Algorithm for Protein-Ligand Docking. *Proteins* **1999**, *37*, 228-241.
84. Wei, B. Q.; Weaver, L. H.; Ferrari, A. M.; Matthews, B. W.; Shoichet, B. K. Testing a Flexible-Receptor Docking Algorithm in a Model Binding Site. *J Mol Biol* **2004**, *337*, 1161-1182.
85. Barril, X.; Morley, S. D. Unveiling the Full Potential of Flexible Receptor Docking Using Multiple Crystallographic Structures. *J Med Chem* **2005**, *48*, 4432-4443.
86. Huang, S. Y.; Zou, X. Ensemble Docking of Multiple Protein Structures: Considering Protein Structural Variations in Molecular Docking. *Proteins* **2007**, *66*, 399-421.
87. Huang, S. Y.; Zou, X. Efficient Molecular Docking of NMR Structures: Application to HIV-1 Protease. *Protein Sci* **2007**, *16*, 43-51.
88. Cavasotto, C. N.; Kovacs, J. A.; Abagyan, R. A. Representing Receptor Flexibility in Ligand Docking through Relevant Normal Modes. *J Am Chem Soc* **2005**, *127*, 9632-9640.
89. Lin, J. H.; Perryman, A. L.; Schames, J. R.; McCammon, J. A. The Relaxed Complex Method: Accommodating Receptor Flexibility for Drug Design with an Improved Scoring Scheme. *Biopolymers* **2003**, *68*, 47-62.
90. Yoon, S.; Welsh, W. J. Identification of a Minimal Subset of Receptor Conformations for Improved Multiple Conformation Docking and Two-Step Scoring. *J Chem Inf Comput Sci* **2004**, *44*, 88-96.
91. Teague, S. J. Implications of Protein Flexibility for Drug Discovery. *Nat Rev Drug Discov* **2003**, *2*, 527-541.
92. Wong, C. F.; McCammon, J. A. Protein Flexibility and Computer-Aided Drug Design. *Annu Rev Pharmacol Toxicol* **2003**, *43*, 31-45.

93. Carlson, H. A.; Masukawa, K. M.; Rubins, K.; Bushman, F. D.; Jorgensen, W. L.; Lins, R. D.; Briggs, J. M.; McCammon, J. A. Developing a Dynamic Pharmacophore Model for HIV-1 Integrase. *J Med Chem* **2000**, *43*, 2100-2114.
94. Meagher, K. L.; Carlson, H. A. Incorporating Protein Flexibility in Structure-Based Drug Discovery: Using HIV-1 Protease as a Test Case. *J Am Chem Soc* **2004**, *126*, 13276-13281.
95. Meagher, K. L.; Lerner, M. G.; Carlson, H. A. Refining the Multiple Protein Structure Pharmacophore Method: Consistency across Three Independent HIV-1 Protease Models. *J Med Chem* **2006**, *49*, 3478-3484.
96. Damm, K. L.; Carlson, H. A. Exploring Experimental Sources of Multiple Protein Conformations in Structure-Based Drug Design. *J Am Chem Soc* **2007**, *129*, 8225-8235.
97. Bowman, A. L.; Lerner, M. G.; Carlson, H. A. Protein Flexibility and Species Specificity in Structure-Based Drug Discovery: Dihydrofolate Reductase as a Test System. *J Am Chem Soc* **2007**, *129*, 3634-3640.
98. Lerner, M. G.; Bowman, A. L.; Carlson, H. A. Incorporating dynamics in E. coli dihydrofolate reductase enhances structure-based drug discovery. *J. Chem. Info. Model.*, *submitted*.
99. Anna L. Bowman, Zaneta Nikolovska-Coleska, Haizhen Zhong, Shaomeng Wang, and Heather A. Carlson. Small molecule inhibitors of the MDM2-p53 interaction discovered by ensemble-based receptor models. *Manuscript in preparation*.
100. Babine, R. E.; Bender, S. L. Molecular Recognition of Proteinminus Signligand Complexes: Applications to Drug Design. *Chem Rev* **1997**, *97*, 1359-1472.
101. 2006 Report on the global AIDS epidemic: Executive summary. http://data.unaids.org/pub/GlobalReport/2006/2006_GR-ExecutiveSummary_en.pdf.
102. Navia, M. A.; Fitzgerald, P. M.; McKeever, B. M.; Leu, C. T.; Heimbach, J. C.; Herber, W. K.; Sigal, I. S.; Darke, P. L.; Springer, J. P. Three-Dimensional Structure of Aspartyl Protease from Human Immunodeficiency Virus HIV-1. *Nature* **1989**, *337*, 615-620.
103. Wlodawer, A.; Miller, M.; Jaskolski, M.; Sathyanarayana, B. K.; Baldwin, E.; Weber, I. T.; Selk, L. M.; Clawson, L.; Schneider, J.; Kent, S. B. Conserved Folding in Retroviral Proteases: Crystal Structure of a Synthetic HIV-1 Protease. *Science* **1989**, *245*, 616-621.

104. Seelmeier, S.; Schmidt, H.; Turk, V.; von der Helm, K. Human Immunodeficiency Virus has an Aspartic-Type Protease That Can Be Inhibited by Pepstatin A. *Proc Natl Acad Sci U S A* **1988**, *85*, 6612-6616.
105. Kohl, N. E.; Emini, E. A.; Schleif, W. A.; Davis, L. J.; Heimbach, J. C.; Dixon, R. A.; Scolnick, E. M.; Sigal, I. S. Active Human Immunodeficiency Virus Protease Is Required for Viral Infectivity. *Proc Natl Acad Sci U S A* **1988**, *85*, 4686-4690.
106. Flexner, C. Update from the 6th International Workshop on the Clinical Pharmacology of HIV Therapy: New Drugs, New Formulations and Drug Interactions. *Hopkins HIV Rep* **2005**, *17*, 1-3, 10-11.
107. Eder, J.; Hommel, U.; Cumin, F.; Martoglio, B.; Gerhartz, B. Aspartic Proteases in Drug Discovery. *Curr Pharm Des* **2007**, *13*, 271-285.
108. Flexner, C.; Bate, G.; Kirkpatrick, P. Tipranavir. *Nat Rev Drug Discov* **2005**, *4*, 955-956.
109. Jadhav, P. K.; Ala, P.; Woerner, F. J.; Chang, C. H.; Garber, S. S.; Anton, E. D.; Bacheler, L. T. Cyclic Urea Amides: HIV-1 Protease Inhibitors with Low Nanomolar Potency against Both Wild Type and Protease Inhibitor Resistant Mutants of HIV. *J Med Chem* **1997**, *40*, 181-191.
110. Wlodawer, A.; Erickson, J. W. Structure-Based Inhibitors of HIV-1 Protease. *Annu Rev Biochem* **1993**, *62*, 543-585.
111. Wlodawer, A.; Gustchina, A. Structural and Biochemical Studies of Retroviral Proteases. *Biochim Biophys Acta* **2000**, *1477*, 16-34.
112. Hornak, V.; Simmerling, C. Targeting Structural Flexibility in HIV-1 Protease Inhibitor Binding. *Drug Discov Today* **2007**, *12*, 132-138.
113. Hornak, V.; Okur, A.; Rizzo, R. C.; Simmerling, C. HIV-1 Protease Flaps Spontaneously Open and Reclose in Molecular Dynamics Simulations. *Proc Natl Acad Sci U S A* **2006**, *103*, 915-920.
114. Toth, G.; Borics, A. Flap Opening Mechanism of HIV-1 Protease. *J Mol Graph Model* **2006**, *24*, 465-474.
115. Toth, G.; Borics, A. Closing of the Flaps of HIV-1 Protease Induced by Substrate Binding: A Model of a Flap Closing Mechanism in Retroviral Aspartic Proteases. *Biochemistry* **2006**, *45*, 6606-6614.
116. Hornak, V.; Okur, A.; Rizzo, R. C.; Simmerling, C. HIV-1 Protease Flaps Spontaneously Close to the Correct Structure in Simulations Following Manual

- Placement of an Inhibitor into the Open State. *J Am Chem Soc* **2006**, *128*, 2812-2813.
117. Spinelli, S.; Liu, Q. Z.; Alzari, P. M.; Hirel, P. H.; Poljak, R. J. The Three-Dimensional Structure of the Aspartyl Protease from the HIV-1 Isolate Bru. *Biochimie* **1991**, *73*, 1391-1396.
118. Rick, S. W.; Erickson, J. W.; Burt, S. K. Reaction Path and Free Energy Calculations of the Transition between Alternate Conformations of HIV-1 Protease. *Proteins* **1998**, *32*, 7-16.
119. Foulkes-Murzycki, J. E.; Scott, W. R.; Schiffer, C. A. Hydrophobic Sliding: A Possible Mechanism for Drug Resistance in Human Immunodeficiency Virus Type 1 Protease. *Structure* **2007**, *15*, 225-233.
120. Scott, W. R.; Schiffer, C. A. Curling of Flap Tips in HIV-1 Protease as a Mechanism for Substrate Entry and Tolerance of Drug Resistance. *Structure* **2000**, *8*, 1259-1265.
121. Piana, S.; Carloni, P.; Parrinello, M. Role of Conformational Fluctuations in the Enzymatic Reaction of HIV-1 Protease. *J Mol Biol* **2002**, *319*, 567-583.
122. Perryman, A. L.; Lin, J. H.; McCammon, J. A. HIV-1 Protease Molecular Dynamics of a Wild-Type and of the V82f/I84v Mutant: Possible Contributions to Drug Resistance and a Potential New Target Site for Drugs. *Protein Sci* **2004**, *13*, 1108-1123.
123. Seibold, S. A.; Cukier, R. I. A Molecular Dynamics Study Comparing a Wild-Type with a Multiple Drug Resistant HIV Protease: Differences in Flap and Aspartate 25 Cavity Dimensions. *Proteins* **2007**.
124. Yamazaki, T.; Hinck, A. P.; Wang, Y. X.; Nicholson, L. K.; Torchia, D. A.; Wingfield, P.; Stahl, S. J.; Kaufman, J. D.; Chang, C. H.; Dommelle, P. J.; Lam, P. Y. Three-Dimensional Solution Structure of the HIV-1 Protease Complexed with Dmp323, a Novel Cyclic Urea-Type Inhibitor, Determined by Nuclear Magnetic Resonance Spectroscopy. *Protein Sci* **1996**, *5*, 495-506.
125. Freedberg, D. I.; Ishima, R.; Jacob, J.; Wang, Y. X.; Kustanovich, I.; Louis, J. M.; Torchia, D. A. Rapid Structural Fluctuations of the Free HIV Protease Flaps in Solution: Relationship to Crystal Structures and Comparison with Predictions of Dynamics Calculations. *Protein Sci* **2002**, *11*, 221-232.
126. Heaslet, H.; Rosenfeld, R.; Giffin, M.; Lin, Y. C.; Tam, K.; Torbett, B. E.; Elder, J. H.; McRee, D. E.; Stout, C. D. Conformational Flexibility in the Flap Domains of Ligand-Free HIV Protease. *Acta Crystallogr D Biol Crystallogr* **2007**, *63*, 866-875.

127. Ala, P. J.; Huston, E. E.; Klabe, R. M.; Jadhav, P. K.; Lam, P. Y.; Chang, C. H. Counteracting HIV-1 Protease Drug Resistance: Structural Analysis of Mutant Proteases Complexed with Xv638 and Sd146, Cyclic Urea Amides with Broad Specificities. *Biochemistry* **1998**, *37*, 15042-15049.
128. Piana, S.; Carloni, P.; Rothlisberger, U. Drug Resistance in HIV-1 Protease: Flexibility-Assisted Mechanism of Compensatory Mutations. *Protein Sci* **2002**, *11*, 2393-2402.
129. Baca, M.; Kent, S. B. Catalytic Contribution of Flap-Substrate Hydrogen Bonds in "HIV-1 Protease" Explored by Chemical Synthesis. *Proc Natl Acad Sci U S A* **1993**, *90*, 11638-11642.
130. Sham, H. L.; Zhao, C.; Stewart, K. D.; Betebenner, D. A.; Lin, S.; Park, C. H.; Kong, X. P.; Rosenbrook, W., Jr.; Herrin, T.; Madigan, D.; Vasavanonda, S.; Lyons, N.; Molla, A.; Saldivar, A.; Marsh, K. C.; McDonald, E.; Wideburg, N. E.; Denissen, J. F.; Robins, T.; Kempf, D. J.; Plattner, J. J.; Norbeck, D. W. A Novel, Picomolar Inhibitor of Human Immunodeficiency Virus Type 1 Protease. *J Med Chem* **1996**, *39*, 392-397.
131. Ohtaka, H.; Freire, E. Adaptive Inhibitors of the HIV-1 Protease. *Prog Biophys Mol Biol* **2005**, *88*, 193-208.
132. Ceccherini-Silberstein, F.; Erba, F.; Gago, F.; Bertoli, A.; Forbici, F.; Bellocchi, M. C.; Gori, C.; D'Arrigo, R.; Marcon, L.; Balotta, C.; Antinori, A.; Monforte, A. D.; Perno, C. F. Identification of the Minimal Conserved Structure of HIV-1 Protease in the Presence and Absence of Drug Pressure. *Aids* **2004**, *18*, F11-19.
133. Muzammil, S.; Ross, P.; Freire, E. A Major Role for a Set of Non-Active Site Mutations in the Development of HIV-1 Protease Drug Resistance. *Biochemistry* **2003**, *42*, 631-638.
134. Sinha, N.; Nussinov, R. Point Mutations and Sequence Variability in Proteins: Redistributions of Preexisting Populations. *Proc Natl Acad Sci U S A* **2001**, *98*, 3139-3144.
135. Clemente, J. C.; Moose, R. E.; Hemrajani, R.; Whitford, L. R.; Govindasamy, L.; Reutzel, R.; McKenna, R.; Agbandje-McKenna, M.; Goodenow, M. M.; Dunn, B. M. Comparing the Accumulation of Active- and Nonactive-Site Mutations in the HIV-1 Protease. *Biochemistry* **2004**, *43*, 12141-12151.
136. Nijhuis, M.; van Maarseveen, N. M.; Lastere, S.; Schipper, P.; Coakley, E.; Glass, B.; Rovenska, M.; de Jong, D.; Chappey, C.; Goedegebuure, I. W.; Heilek-Snyder, G.; Dulude, D.; Cammack, N.; Brakier-Gingras, L.; Konvalinka, J.; Parkin, N.; Krausslich, H. G.; Brun-Vezinet, F.; Boucher, C. A. A Novel

Substrate-Based HIV-1 Protease Inhibitor Drug Resistance Mechanism. *PLoS Med* **2007**, *4*, e36.

137. Crawford, H.; Prado, J. G.; Leslie, A.; Hue, S.; Honeyborne, I.; Reddy, S.; van der Stok, M.; Mncube, Z.; Brander, C.; Rousseau, C.; Mullins, J. I.; Kaslow, R.; Goepfert, P.; Allen, S.; Hunter, E.; Mulenga, J.; Kiepiela, P.; Walker, B. D.; Goulder, P. J. Compensatory Mutation Partially Restores Fitness and Delays Reversion of Escape Mutation within the Immunodominant Hla-B*5703-Restricted Gag Epitope in Chronic Human Immunodeficiency Virus Type 1 Infection. *J Virol* **2007**, *81*, 8346-8351.
138. Gatanaga, H.; Suzuki, Y.; Tsang, H.; Yoshimura, K.; Kavlick, M. F.; Nagashima, K.; Gorelick, R. J.; Mardy, S.; Tang, C.; Summers, M. F.; Mitsuya, H. Amino Acid Substitutions in Gag Protein at Non-Cleavage Sites Are Indispensable for the Development of a High Multitude of HIV-1 Resistance against Protease Inhibitors. *J Biol Chem* **2002**, *277*, 5952-5961.
139. Logsdon, B. C.; Vickrey, J. F.; Martin, P.; Proteasa, G.; Koepke, J. I.; Terlecky, S. R.; Wawrzak, Z.; Winters, M. A.; Merigan, T. C.; Kovari, L. C. Crystal Structures of a Multidrug-Resistant Human Immunodeficiency Virus Type 1 Protease Reveal an Expanded Active-Site Cavity. *J Virol* **2004**, *78*, 3123-3132.
140. Prabu-Jeyabalan, M.; Nalivaika, E.; Schiffer, C. A. Substrate Shape Determines Specificity of Recognition for HIV-1 Protease: Analysis of Crystal Structures of Six Substrate Complexes. *Structure* **2002**, *10*, 369-381.
141. King, N. M.; Prabu-Jeyabalan, M.; Nalivaika, E. A.; Schiffer, C. A. Combating Susceptibility to Drug Resistance: Lessons from HIV-1 Protease. *Chem Biol* **2004**, *11*, 1333-1338.
142. Prabu-Jeyabalan, M.; King, N. M.; Nalivaika, E. A.; Heilek-Snyder, G.; Cammack, N.; Schiffer, C. A. Substrate Envelope and Drug Resistance: Crystal Structure of Ro1 in Complex with Wild-Type Human Immunodeficiency Virus Type 1 Protease. *Antimicrob Agents Chemother* **2006**, *50*, 1518-1521.
143. Chellappan, S.; Kairys, V.; Fernandes, M. X.; Schiffer, C.; Gilson, M. K. Evaluation of the Substrate Envelope Hypothesis for Inhibitors of HIV-1 Protease. *Proteins* **2007**, *68*, 561-567.
144. Prabu-Jeyabalan, M.; Nalivaika, E.; Schiffer, C. A. How Does a Symmetric Dimer Recognize an Asymmetric Substrate? A Substrate Complex of HIV-1 Protease. *J Mol Biol* **2000**, *301*, 1207-1220.
145. Yin, P. D.; Das, D.; Mitsuya, H. Overcoming HIV Drug Resistance through Rational Drug Design Based on Molecular, Biochemical, and Structural Profiles of HIV Resistance. *Cell Mol Life Sci* **2006**, *63*, 1706-1724.

146. Ghosh, A. K.; Ramu Sridhar, P.; Kumaragurubaran, N.; Koh, Y.; Weber, I. T.; Mitsuya, H. Bis-Tetrahydrofuran: A Privileged Ligand for Darunavir and a New Generation of HIV Protease Inhibitors That Combat Drug Resistance. *ChemMedChem* **2006**, *1*, 939-950.
147. Ghosh, A. K.; Sridhar, P. R.; Leshchenko, S.; Hussain, A. K.; Li, J.; Kovalevsky, A. Y.; Walters, D. E.; Wedekind, J. E.; Grum-Tokars, V.; Das, D.; Koh, Y.; Maeda, K.; Gatanaga, H.; Weber, I. T.; Mitsuya, H. Structure-Based Design of Novel HIV-1 Protease Inhibitors to Combat Drug Resistance. *J Med Chem* **2006**, *49*, 5252-5261.
148. Zoete, V.; Michielin, O.; Karplus, M. Relation between Sequence and Structure of HIV-1 Protease Inhibitor Complexes: A Model System for the Analysis of Protein Flexibility. *J Mol Biol* **2002**, *315*, 21-52.
149. Layten, M.; Hornak, V.; Simmerling, C. The Open Structure of a Multi-Drug-Resistant HIV-1 Protease Is Stabilized by Crystal Packing Contacts. *Journal of the American Chemical Society* **2006**, *128*, 13360-13361.
150. Perryman, A. L.; Lin, J. H.; McCammon, J. A. Restrained Molecular Dynamics Simulations of HIV-1 Protease: The First Step in Validating a New Target for Drug Design. *Biopolymers* **2006**, *82*, 272-284.
151. Martin, P.; Vickrey, J. F.; Proteasa, G.; Jimenez, Y. L.; Wawrzak, Z.; Winters, M. A.; Merigan, T. C.; Kovari, L. C. "Wide-Open" 1.3 a Structure of a Multidrug-Resistant HIV-1 Protease as a Drug Target. *Structure* **2005**, *13*, 1887-1895.
152. Bannwarth, L.; Reboud-Ravaux, M. An Alternative Strategy for Inhibiting Multidrug-Resistant Mutants of the Dimeric HIV-1 Protease by Targeting the Subunit Interface. *Biochem Soc Trans* **2007**, *35*, 551-554.
153. Schramm, H. J.; Breipohl, G.; Hansen, J.; Henke, S.; Jaeger, E.; Meichsner, C.; Riess, G.; Ruppert, D.; Rucknagel, K. P.; Schafer, W.; et al. Inhibition of HIV-1 Protease by Short Peptides Derived from the Terminal Segments of the Protease. *Biochem Biophys Res Commun* **1992**, *184*, 980-985.
154. Bouras, A.; Boggetto, N.; Benatalah, Z.; de Rosny, E.; Sicsic, S.; Reboud-Ravaux, M. Design, Synthesis, and Evaluation of Conformationally Constrained Tongues, New Inhibitors of HIV-1 Protease Dimerization. *J Med Chem* **1999**, *42*, 957-962.
155. Shultz, M. D.; Bowman, M. J.; Ham, Y. W.; Zhao, X.; Tora, G.; Chmielewski, J. Small-Molecule Inhibitors of HIV-1 Protease Dimerization Derived from Cross-Linked Interfacial Peptides. *Angew Chem Int Ed Engl* **2000**, *39*, 2710-2713.

156. Song, M.; Rajesh, S.; Hayashi, Y.; Kiso, Y. Design and Synthesis of New Inhibitors of HIV-1 Protease Dimerization with Conformationally Constrained Templates. *Bioorg Med Chem Lett* **2001**, *11*, 2465-2468.
157. Merabet, N.; Dumond, J.; Collinet, B.; Van Baelinghem, L.; Boggetto, N.; Ongeri, S.; Ressad, F.; Reboud-Ravaux, M.; Sicsic, S. New Constrained "Molecular Tongs" Designed to Dissociate HIV-1 Protease Dimer. *J Med Chem* **2004**, *47*, 6392-6400.
158. Babe, L. M.; Rose, J.; Craik, C. S. Synthetic "Interface" Peptides Alter Dimeric Assembly of the HIV 1 and 2 Proteases. *Protein Sci* **1992**, *1*, 1244-1253.
159. Zutshi, R.; Brickner, M.; Chmielewski, J. Inhibiting the Assembly of Protein-Protein Interfaces. *Curr Opin Chem Biol* **1998**, *2*, 62-66.
160. Ulysse, L. G.; Chmielewski, J. Restricting the Flexibility of Crosslinked, Interfacial Peptide Inhibitors of HIV-1 Protease. *Bioorg Med Chem Lett* **1998**, *8*, 3281-3286.
161. Boggetto, N.; Reboud-Ravaux, M. Dimerization Inhibitors of HIV-1 Protease. *Biol Chem* **2002**, *383*, 1321-1324.
162. Lee, S. G.; Chmielewski, J. Rapid Synthesis and in Situ Screening of Potent HIV-1 Protease Dimerization Inhibitors. *Chem Biol* **2006**, *13*, 421-426.
163. Hwang, Y. S.; Chmielewski, J. Development of Low Molecular Weight HIV-1 Protease Dimerization Inhibitors. *J Med Chem* **2005**, *48*, 2239-2242.
164. Shultz, M. D.; Ham, Y. W.; Lee, S. G.; Davis, D. A.; Brown, C.; Chmielewski, J. Small-Molecule Dimerization Inhibitors of Wild-Type and Mutant HIV Protease: A Focused Library Approach. *J Am Chem Soc* **2004**, *126*, 9886-9887.
165. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M. Jr.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. *J Am Chem Soc* **1995**, *117*, 5179-5197.
166. Case, D. A. P., D. A.; Caldwell, J. W.; Cheatham III, T. E.; Ross, W. S.; Simmerling, C. L.; Darden, T. A.; Merz, K. M.; Stanton, R. V.; Cheng, A. L.; Vincent, J. J.; Crowley, M.; Tsui, V.; Radmer, R. J.; Duan, Y.; Pitera, J.; Massova, I.; Seibel, G. L.; Singh, U. C.; Weiner, P. K.; Kollman, P. A.; University of California San Francisco: San Francisco, CA, 1996.
167. Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. Charmm - a Program for Macromolecular Energy, Minimization, and Dynamics Calculations. *Journal of Computational Chemistry* **1983**, *4*, 187-217.

168. Hermans, J.; Berendsen, H. J. C.; van Gunsteren, W. F., Postma, J. P. M. A Consistent Empirical Potential for Water-Protein Interactions. *Biopolymers* **1984**, *23*, 1513-1518.
169. Jorgensen, W. L.; Maxwell, D. S.; TiradoRives, J. Development and Testing of the Opls All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *Journal of the American Chemical Society* **1996**, *118*, 11225-11236.
170. Leach, A. *Molecular Modelling: Principles and Applications*; 2nd Ed.; Prentice Hall; 2001.
171. Halgren, T. A.; Damm, W. Polarizable Force Fields. *Curr Opin Struct Biol* **2001**, *11*, 236-242.
172. McCammon, J. A.; Karplus, M. Internal Motions of Antibody Molecules. *Nature* **1977**, *268*, 765-766.
173. Scheraga, H. A.; Khalili, M.; Liwo, A. Protein-Folding Dynamics: Overview of Molecular Simulation Techniques. *Annu Rev Phys Chem* **2007**, *58*, 57-83.
174. Miranker, A.; Karplus, M. Functionality Maps of Binding Sites: A Multiple Copy Simultaneous Search Method. *Proteins* **1991**, *11*, 29-34.
175. Morris, G. M.; Goodsell, D. S.; Halliday, R.S.; Huey, R.; Hart, W. E.; Belew, R. K.; Olson, A. J. Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function. *J Comp Chem* **1998**, *19*, 1639-1662.
176. Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and Validation of a Genetic Algorithm for Flexible Docking. *J Mol Biol* **1997**, *267*, 727-748.
177. Abagyan, R. A.; Mazur, A. K. New Methodology for Computer-Aided Modelling of Biomolecular Structure and Dynamics. 2. Local Deformations and Cycles. *J Biomol Struct Dyn* **1989**, *6*, 833-845.
178. Halgren, T. A.; Murphy, R. B.; Friesner, R. A.; Beard, H. S.; Frye, L. L.; Pollard, W. T.; Banks, J. L. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 2. Enrichment Factors in Database Screening. *J Med Chem* **2004**, *47*, 1750-1759.
179. Friesner, R. A.; Banks, J. L.; Murphy, R. B.; Halgren, T. A.; Klicic, J. J.; Mainz, D. T.; Repasky, M. P.; Knoll, E. H.; Shelley, M.; Perry, J. K.; Shaw, D. E.; Francis, P.; Shenkin, P. S. Glide: A New Approach for Rapid, Accurate Docking

- and Scoring. 1. Method and Assessment of Docking Accuracy. *J Med Chem* **2004**, *47*, 1739-1749.
180. Sousa, S. F.; Fernandes, P. A.; Ramos, M. J. Protein-Ligand Docking: Current Status and Future Challenges. *Proteins* **2006**, *65*, 15-26.
181. Taylor, R. D.; Jewsbury, P. J.; Essex, J. W. A Review of Protein-Small Molecule Docking Methods. *J Comput Aided Mol Des* **2002**, *16*, 151-166.
182. Muryshev, A. E.; Tarasov, D. N.; Butygin, A. V.; Butygina, O. Y.; Aleksandrov, A. B.; Nikitin, S. M. A Novel Scoring Function for Molecular Docking. *J Comput Aided Mol Des* **2003**, *17*, 597-605.
183. Charifson, P. S.; Corkery, J. J.; Murcko, M. A.; Walters, W. P. Consensus Scoring: A Method for Obtaining Improved Hit Rates from Docking Databases of Three-Dimensional Structures into Proteins. *J Med Chem* **1999**, *42*, 5100-5109.
184. Bissantz, C.; Folkers, G.; Rognan, D. Protein-Based Virtual Screening of Chemical Databases. 1. Evaluation of Different Docking/Scoring Combinations. *J Med Chem* **2000**, *43*, 4759-4767.
185. Terp, G. E.; Johansen, B. N.; Christensen, I. T.; Jorgensen, F. S. A New Concept for Multidimensional Selection of Ligand Conformations (Multiselect) and Multidimensional Scoring (Multiscore) of Protein-Ligand Binding Affinities. *J Med Chem* **2001**, *44*, 2333-2343.
186. Mohan, V.; Gibbs, A. C.; Cummings, M. D.; Jaeger, E. P.; DesJarlais, R. L. Docking: Successes and Challenges. *Curr Pharm Des* **2005**, *11*, 323-333.
187. Leach, A. R.; Shoichet, B. K.; Peishoff, C. E. Prediction of Protein-Ligand Interactions. Docking and Scoring: Successes and Gaps. *J Med Chem* **2006**, *49*, 5851-5855.
188. Warren, G. L.; Andrews, C. W.; Capelli, A. M.; Clarke, B.; LaLonde, J.; Lambert, M. H.; Lindvall, M.; Nevins, N.; Semus, S. F.; Senger, S.; Tedesco, G.; Wall, I. D.; Woolven, J. M.; Peishoff, C. E.; Head, M. S. A Critical Assessment of Docking Programs and Scoring Functions. *J Med Chem* **2006**, *49*, 5912-5931.

CHAPTER 2

Gaussian-Weighted RMSD Superposition of Proteins: A Structural Comparison for Flexible Proteins

2.1 Introduction

An appropriate structural superposition provides a means to compare the similarity or dissimilarity between protein conformations. Kabsch previously described the algorithm that optimally overlays two molecules by minimizing the deviation between their atomic coordinates.¹ This algorithm is the basis of most alignment methods that overlay molecules using an sRMSD fit. The Kabsch method notes a means to incorporate weighting or biasing into the sRMSD fit, but this is not regularly used.

Our technique incorporates a Gaussian-weighting term and minimizes the weighted deviation to overlay two structures. The individual weights are directly based on the distance between each atom pair; consequently, atoms with little movement will have a greater weighting in the least squares fit than those that are further apart. Our use of a Gaussian-weighting term inherently selects out atom pairs with similar relative positions between the two structures, while discounting loops and other flexible regions. This method removes the subjective nature of selecting out and overlaying a subset of atoms and does not require any prior knowledge of the protein structure or its dynamics. The weighted RMSD (wRMSD) fit is heavily biased by the coordinates found in the similar regions of the two conformations, highlighting the static regions and the dynamic movement of the protein. Hence, this technique can be a useful way to identify domains and hinge regions within a protein structure.

Here, we show how weights can be used during the fit to produce an wRMSD alignment. Furthermore, we found that predefining the domains is not needed with wRMSD fits when using our implementation of weighting. All the C α coordinates are used in the wRMSD alignment, and the resulting weights and alignments can identify the domains. This technique is the reverse of other methods in the literature. The overlay defines the domains, rather than the domains defining the overlay.

2.2 Computational Methods

Protein Dataset

We have chosen to test this method on eight representative proteins found in the Database of Macromolecular Movements²⁻⁴, Table 2.1. The proteins were chosen based on their interest to the community, variation in size, and range of conformational changes. Investigating protein systems which undergo small and large conformational changes will allow us to create a robust procedure, appropriate for a full range of applications. PyMOL was used for various visualization purposes and the creation of figures.⁵

Table 2.1. Test case proteins listed in order of small to large conformational changes.

Protein System	Conformation 1 PDB ⁶ Code	Conformation 2 PDB Code	Standard RMSD*	Number of Residues
HIV-1p	1KZK ⁷	1HHP ⁸	1.2	94
cAMP-Dependent PK (PKA)	1JLU ⁹	1CMK ¹⁰	1.9	337
Elongation Factor G (EFG)	1FNM ¹¹	2EFG ¹²	2.3	580
Estrogen Receptor α (ER α)	3ERD ¹³	3ERT ¹³	4.9	238
Rb69 Phage DNA Polymerase (DNA Pol)	1IH7 ¹⁴	1IG9 ¹⁴	6.5	895
GroEL	1AON ¹⁵	1OEL ¹⁶	12.4	524
RAN	1RRP ¹⁷	1BYU ¹⁸	14.4	200
T7 Phage RNA Polymerase (RNA Pol)	1QLN ¹⁹	1MSW ²⁰	18.3	843

* Standard RMSD parallels the degree of conformational change

Standard RMSD Fit

A widely used algorithm to calculate the least-squares solution was previously described by Kabsch¹. Flower has presented a thorough discussion of various mathematical approaches to the superposition problem²¹, and he notes that Diamond²² has proposed a more accurate and sophisticated mathematical approach. We have chosen to work with Kabsch's technique because it is more widely used than Diamond's. This will allow our modifications to be easily incorporated into more existing programs and applications.

Following Kabsch's nomenclature¹, let us assume that we have two proteins **X** and **Y**, both having n atoms. The centers of mass of both proteins are at the origin (it is trivial to translate any set of protein coordinates to accomplish this). If we wish to rotate protein **X** to best match the coordinates of protein **Y**, we start by calculating a 3x3 covariance matrix (R) between the two set of points X and Y where i and j denote the 3D components of each atom n and

$$R = Y^T X \quad \text{or} \quad r_{ij} = \sum_n y_{ni} \cdot x_{nj} \quad (2.1)$$

The square of the covariance matrix (R^2) is calculated as

$$R^2 = R^T R \quad (2.2)$$

The eigenvectors (A) and eigenvalues of R^2 are determined and sorted in decreasing order of eigenvalues. The normalized product of ($R \times A$) is denoted as matrix

B . Matrices A and B are used to calculate the rotation matrix (U) where

$$U = B^T A \quad \text{or} \quad u_{ij} = \sum_k b_{ki} \cdot a_{kj} \quad (2.3)$$

All coordinates of protein **X** are rotated to produce coordinates X' .

$$X'^T = U X^T \quad \text{or} \quad x'_{ni} = \sum_k u_{ik} \cdot x_{nk} \quad (2.4)$$

These new coordinates X' are compared back to coordinates Y of protein **Y**. The sRMSD is calculated as follows

$$\text{sRMSD} = \left(\frac{1}{n} \sum_n d_n^2 \right)^{1/2} \quad (2.5)$$

$$\text{where } d_n = \left((y_{nx} - x'_{nx})^2 + (y_{ny} - x'_{ny})^2 + (y_{nz} - x'_{nz})^2 \right)^{1/2} \quad (2.6)$$

Weighted RMSD Fit

We use a Gaussian-weighting factor in the wRMSD procedure. The weight is given by

$$w_n = e^{-(d_n)^2/c} \quad (2.7)$$

where c is an arbitrary scaling factor and d_n is determined with Eq. 2.6. It should be noted that d is the distance between atom n in each protein conformation (\mathbf{X} and \mathbf{Y}). The distance is not between two atoms (n and m) in the same protein nor is it a comparison of the n - m distance in conformations \mathbf{X} and \mathbf{Y} .

The weighted term is incorporated into the calculation of a weighted center of mass (Eq. 2.8), and this term is used to orient the weighted center of mass of each protein at the origin.

$$wCM_x = \sum_n w_n m_n x_n / \sum_n m_n \quad \text{and} \quad wCM_y = \sum_n w_n m_n y_n / \sum_n m_n \quad (2.8)$$

An sRMSD fit minimizes the sum of d_n^2 , but a wRMSD fit minimizes the sum of $w_n d_n^2$. Kabsch noted that weighting terms can be used in the RMSD fit by simply incorporating them to the covariance matrix.

$$r_{ij} = \sum_n w_n y_{ni} x_{nj} \quad (2.9)$$

At this point, the procedure is the same. The eigenvectors of R^2 are found and used with R to produce the rotation matrix U (Eq. 2.2-2.4). The sRMSD from Eq. 2.5 is rewritten as a weighted RMSD.

$$\text{wRMSD} = \left(\frac{1}{n} \sum_n w_n d_n^2 \right)^{1/2} \quad (2.10)$$

A second metric can be created from a sum of all weights. The maximum value occurs when all weights are 1.0 and the sum is n (all atom pairs are perfectly overlaid).

The number of atoms will vary for each protein system, so a normalized measure is more appropriate. We write the sum of all weights (%wSUM) as

$$\%wSUM = \frac{1}{n} \sum_n w_n \quad (2.11)$$

Technically, it may be more appropriate to calculate wRMSD as the square root of the sum of $w_n d_n^2$ divided by the sum of w_n . However, we found that Eq. 2.10 better reflects the agreement in the overlay. If the user desires to calculate wRMSD in the alternate fashion, it is simply wRMSD from Eq. 2.10 divided by the square root of %wSUM in Eq. 2.11.

We verified that our code produced proper sRMSD fits before incorporating the weighted terms. We also confirmed that when the scaling factor c is set to a very high number (10^4 or higher), the weights become approximately one for all atom pairs, and a sRMSD fit is produced.

It should be noted that Diamond also outlined how weights could be included in his alignment process.²² Our Gaussian weighting idea could be added to any code based on Diamond's approach by following the discussions in that work. Neither Kabsch nor Diamond ever define how weights should be calculated, and to the best of our knowledge, no one has published a weighted alignment using either Kabsch's or Diamond's methods. Even Diamond's proposal for overlaying ensembles of NMR structures²³ does not weight the contributions of different atom pairs. In that application, subsets of $C\alpha$ are simply described as "in" or "out" of the overlay process. However, that application does show how alignments can be extended to ensembles of structures, through iteratively fitting N structures in $N(N-1)/2$ pairs until convergence is achieved. For simplicity, our code (provided in Appendix 1) and our examples in this paper use only two conformations of each protein, but this code could be inserted into any program that iteratively aligns ensembles of structures.

Another issue that deserves discussion is the importance of coordinate accuracy. Schneider^{24,25} developed a method for aligning two protein conformations which analyzes the interatomic distances within each independent protein structure to determine subsets of atoms to use in an sRMSD alignment. A unique caveat is his use of a weighting term, biased by coordinate accuracy, to define the subset for the alignment. (Though weights are used to define the subset, the weights are not part of the sRMSD.) Our implementation of wRMSD indirectly accounts for coordinate accuracy. The coordinate uncertainty is highest in the flexible regions of the protein, and the flexible regions of the protein are inherently underweighted in our implementation. According to Diamond²², the errors would have to be on the order of the coordinate measurement itself to be significant. In our implementation, the errors would have to be on the order of Ångstroms (similar to the scaling factor c) which only happens in poorly resolved loop regions.

Alignment Method

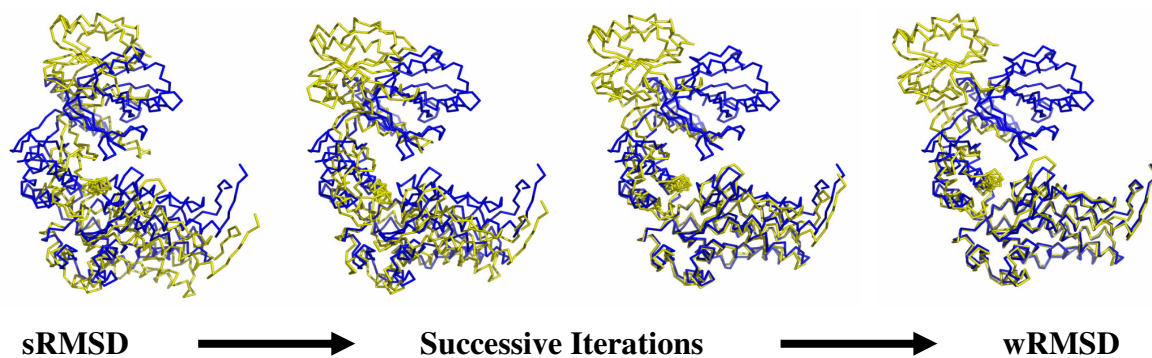
Our code currently implements the wRMSD method using C α coordinates of two protein conformations (it is straightforward to use all atoms, only backbone atoms, etc.). The procedure requires three steps: first, create a list of corresponding atom pairs; second, perform an initial sRMSD alignment to bring the two proteins into proximity; third, conduct iterative wRMSD fitting until convergence is reached. Our method can be used to align two conformations of the same protein, but aligning two homologs could be accomplished by incorporating some initial sequence or structural comparison to create the corresponding atom pairs.

The first step in our alignment method is to compare the residues of proteins **X** and **Y**. This is done to ensure that each residue is present in both structures and can be included in the alignment. A residue ID list is parsed for both proteins from its respective PDB file. A residue ID is included only if the residue has C α atomic coordinates in both structures. Next, we remove any inappropriate residues from the residue ID lists, which

include duplicate residues, disordered residues, or heterogroups. Duplicate and disordered residues are typically the result of alternative conformations revealed by the electron density maps. As our method inherently underweights flexible regions, it is justified to remove these residues from the alignment. The C α coordinates that correspond to the residues remaining in the residue ID lists are parsed from their respective PDB files and used for the initial sRMSD alignment.

An sRMSD alignment (non-weighted) is performed first to bring the structures into close proximity to calculate an appropriate weighted alignment. Consequently, an atom's initial Gaussian-weight is based on the distance between its positions in protein **X** and protein **Y**, calculated after the sRMSD fit. The Gaussian-weighted alignment is then performed in an iterative manner until convergence is reached, Figure 2.1. Each iteration recalculates an appropriate weighted center of mass and a new rotation matrix.

Figure 2.1. A series of iterations are needed to converge the wRMSD solution for overlaying two proteins. Four snapshots from the series of iterations are shown to demonstrate the process.



2.3 Results and Discussion

Gaussian-Weighted RMSD Alignment

A weighted alignment is not as straightforward as a standard alignment. The structures must be nearly aligned to calculate appropriate weights, hence our use of an

initial sRMSD alignment. The wRMSD procedure requires successive iterations until convergence is achieved because every wRMSD fit changes the distances, which changes the weights, which changes the wRMSD fit (Figure 2.1). In order to evaluate convergence, a metric is needed to describe optimal partial alignment. Proper metrics are even more important with wRMSD because a weighted alignment does not have a unique solution like an sRMSD fit. If we align the same protein on itself, there are two minima where the sum of $w_n d_n^2$ is zero. The first occurs when the difference between all atom pairs is zero, and the protein is perfectly overlaid (all $d_n^2 = 0$); the second minimum happens at infinite separation when all weights go to zero (all $w_n = 0$).

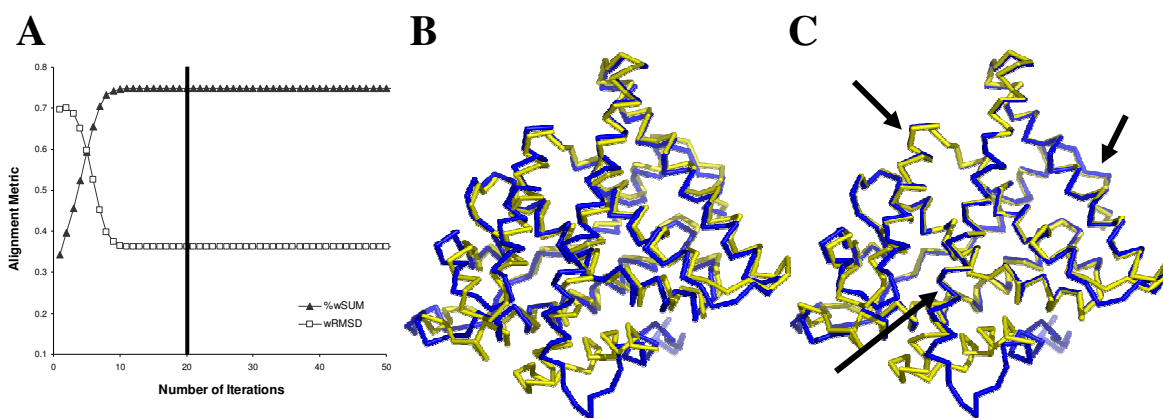
It was previously shown that there is generally not a unique solution when calculating a global alignment of dynamic proteins.²⁶ By determining a metric to identify an optimal partial alignment, we can fully automate our method and remove the subjectivity of evaluating the RMSD fit by visual inspection. We have chosen to explore two different metrics in detail: the wRMSD of Eq. 2.10 and the %wSUM given in Eq. 2.11. The wRMSD decreases to a stable minimum, while the %wSUM increases to a stable maximum. The optimal solution should occur when the maximum number of atoms makes a significant contribution. Hence, in our example of the wRMSD alignment of a protein upon itself, %wSUM identifies the perfectly overlaid minimum to be more significant because more atoms are contributing significantly to the weights (%wSUM = 1). The infinitely separated minimum has a %wSUM = 0.

Gaussian Scaling Factor

We started by investigating the most appropriate way to weight the RMSD fit. The Gaussian scaling factor c in Eq. 2.7 controls the weight given to a pair of C α atoms. For instance, a C α pair that is 1 Å apart will have a weight of 0.368 with $c = 1 \text{ \AA}^2$. If $c = 5 \text{ \AA}^2$, the weight is 0.819. Smaller values of c result in tighter, more restrictive coupling that forces only very similar atoms to have significant weights during the wRMSD fit.

We found that performing the weighted alignment in an iterative manner with a constant scaling factor exhibits converging behavior as demonstrated in Figure 2.2. We defined convergence by $\Delta wRMSD < 1 \times 10^{-6} \text{ \AA}$. After 19 iterations, both the wRMSD and %wSUM metrics converge to stable values of 0.36 \AA and 74.8% , respectively (scaling factor c set to 2 \AA^2). In the final alignment, 182 of the 238 C α common to both structures are within 1 \AA , and the average distance between all 238 C α pairs is 2.0 \AA . The same converging behavior was also observed for all other test cases.

Figure 2.2. ER α .¹³ (A) The behavior of the wRMSD and %wSUM metrics as the weighted alignment is performed in an iterative manner using the entire protein sequence for the initial sRMSD fit. A scaling factor, c , of 2 \AA^2 is used. The vertical line indicates where convergence is reached. (B) sRMSD alignment of 3ERD (yellow) onto 3ERT (blue). (C) wRMSD alignment after convergence is reached. Arrows denote regions with improved fit.

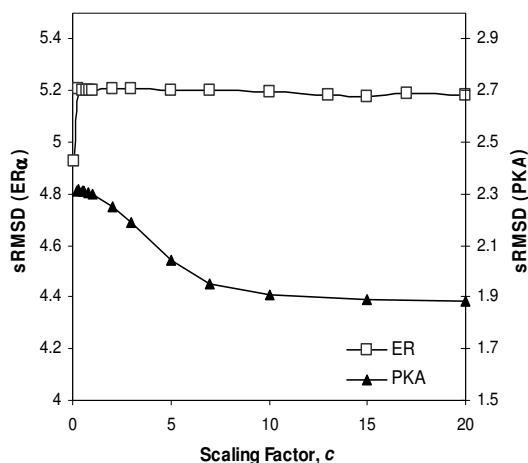


We varied the scaling factor, using c values from 0.10 to 20 \AA^2 , to determine its effect on the weighted alignment. We found that upon convergence a range of c values produced nearly identical alignments. This was determined by calculating the sRMSD of each solution. The sRMSD remained relatively constant for c in the range of 0.3 to 20 \AA^2 for the ER α structures¹³ 3ERD and 3ERT. The constant regions were defined as the change in sRMSD ($\Delta sRMSD$) of less than 0.1 \AA from the maximum to the minimum value in the range. The reader will notice that the sRMSD of 5.2 \AA is higher than the 4.9 \AA listed for the ER α structures (Table 2.1). This is appropriate; an sRMSD measurement

from a wRMSD fit should be higher because some fit of flexible regions is sacrificed to better align the rigid core.

For the PKA structures 1JLU⁹ and 1CMK¹⁰, the sRMSD stayed constant for a much smaller range of c values, 0.2 to 2 Å², shown in Figure 2.3. In the case of PKA, high values of c (≥ 10 Å²) simply produce the sRMSD solution (the later values of sRMSD in Figure 2.3 are 1.9 Å, the same as the value in Table 2.1).

Figure 2.3. The scaling factor, c , plotted against the sRMSD value for each weighted fit and the target coordinates. Open squares (\square) are for ER α ¹³, 3ERD fit onto 3ERT. The weighted fit is the same for c values from 0.3-20 Å². Filled triangles (\blacktriangle) are for PKA^{9,10}, 1JLU fit onto 1CMK. The weighted fit is the same for c values from 0.2-2 Å². The largest values of c simply reproduce the sRMSD solution for the PKA structures.



A range of values for c works well for each protein system, as provided in Table 2.2; however, as c is decreased, more iterations are needed to reach convergence. We found a correlation between the sRMSD and the optimal scaling factor for wRMSD. When the structures are very similar (characterized by a small sRMSD), a smaller scaling factor is required to obtain a “tighter fit” of the rigid core. A scaling factor equal to 2 Å² performs well for all systems except RNA Pol, corresponding to the largest sRMSD. We suggest that when the sRMSD is less than 5 Å, a scaling factor of 2 Å² should be used, and sRMSD above 5 Å requires a scaling factor of 5 Å².

Table 2.2. Range of optimal scaling factors for each protein system, along with the calculated sRMSD of the wRMSD fit over the given range.

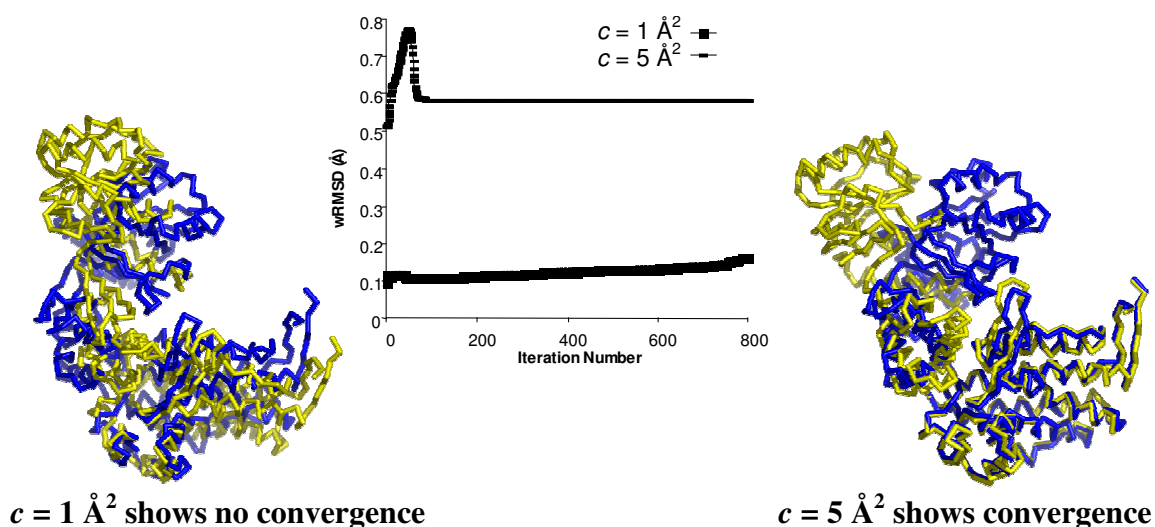
Protein System	sRMSD from the wRMSD fit (\AA)	Range of c (\AA^2) that produce the same sRMSD*
HIV-1p	1.4	2 – 5
PKA	2.3	0.2 – 2
EFG	3.6	0.2 – 4
ER α	5.2	0.3 – 20
DNA Pol [†]	7.2, 7.6	2 – 3, 4 – 6
GroEL	15.9	2 – 16
RAN	16.8	0.5 – 20
RNA Pol	20.6	3 – 20

* The values for sRMSD changed less than 0.1 \AA over the noted range.

[†] The DNA Pol system converged to two different solutions when c was changed. Both were stably converged over the noted ranges.

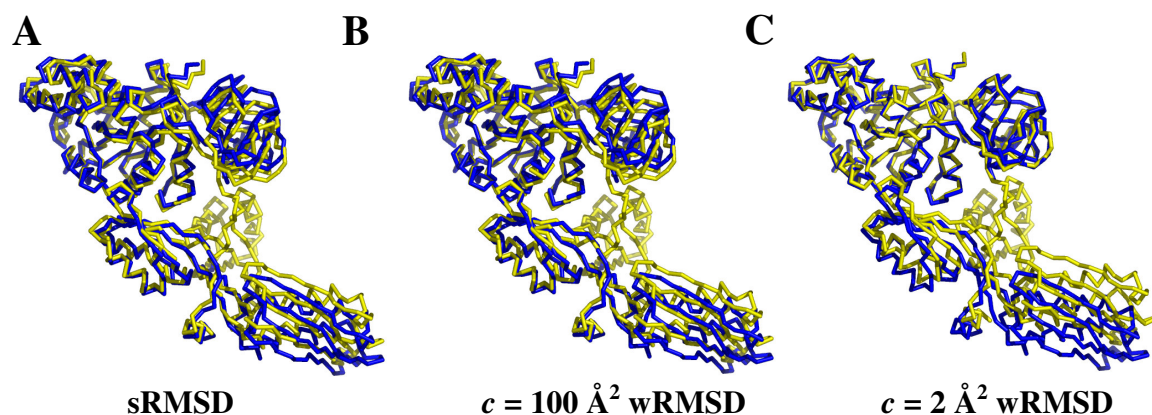
If the scaling factor used is too small for the particular system, we do not see converging behavior, and an optimal solution is never reached. As demonstrated in Figure 2.4 using the GroEL system^{15,16}, when c is equal to 1 \AA^2 we do not see converging behavior. This unconverged alignment is very similar to the sRMSD alignment used to start the wRMSD fit. However, when a larger scaling factor is used ($c = 5 \text{\AA}^2$), we observe convergence after 93 iterations.

Figure 2.4. If the scaling factor is too small, the wRMSD fit fails to produce converged structures for GroEL^{15,16}. The behavior of the wRMSD metric versus iteration during the weighted fit, using the entire protein sequence for the initial RMSD fit and two values of c . **(Left)** wRMSD alignment of 1AON (yellow) onto 1OEL (blue) after 800 unconverged iterations of wRMSD fitting, $c = 1 \text{\AA}^2$. **(Right)** wRMSD alignment of 1AON (yellow) onto 1OEL (blue) after convergence is reached, $c = 5 \text{\AA}^2$.



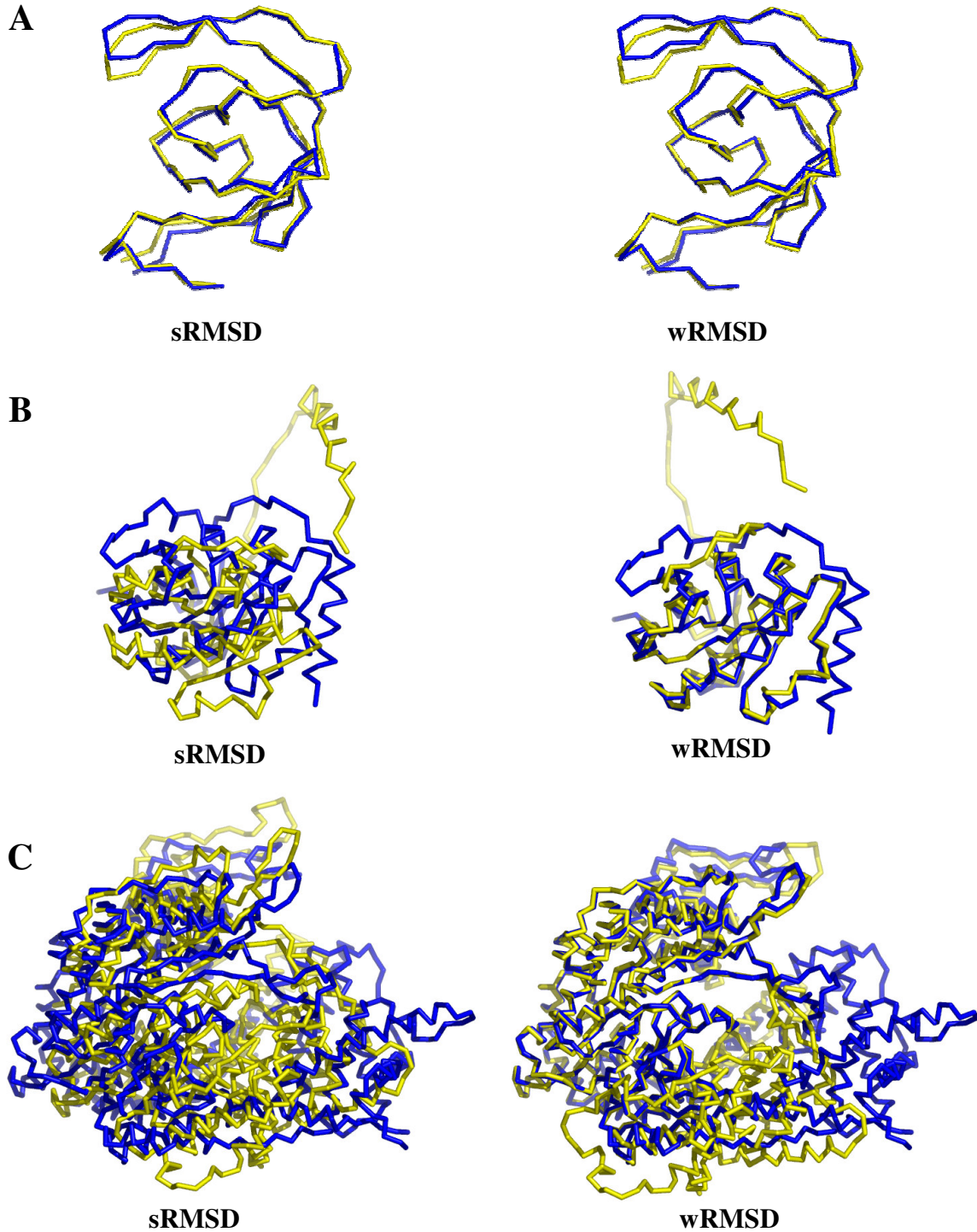
We use EFG^{11,12} to show the problem that occurs when c is too large, Figure 2.5. Large scaling factors (i.e. 100 \AA^2) produce a superposition similar to the standard alignment. However conducting an wRMSD fit using an appropriate scaling factor of 2 \AA^2 is able to highlight the similarity of the rigid core region.

Figure 2.5. If the scaling factor is too large, an wRMSD fit is the same as a sRMSD fit for EFG^{11,12}. **(A)** sRMSD alignment of 1FNM (yellow) onto 2EFG (blue). **(B)** wRMSD alignment of 1FNM (yellow) onto 2EFG (blue) after convergence is reached, $c = 100 \text{ \AA}^2$. **(C)** wRMSD alignment of 1FNM (yellow) onto 2EFG (blue) after convergence is reached, $c = 2 \text{ \AA}^2$.



In all cases, the weighted alignment resulted in an improved fit over the standard alignment. However, the improvement is minimal when the two conformations of a protein are very similar (e.g. HIV-1p^{7,8}). When the sRMSD is small, the conformational change is only slight as shown in Figure 2.6A. This means that most of the calculated weights are approximately equal to one unless an incredibly small value is used for c . The wRMSD still biases the rigid core (most noticeable for the C-terminus at the bottom of the structure), but the overall effect on the system is slight. Representative sRMSD and wRMSD alignments for RAN^{17,18} and RNA Pol^{19,20} are also provided in Figure 2.6; a scaling factor of 2 \AA^2 was used for all systems with a sRMSD less than 5 \AA , and $c = 5 \text{ \AA}^2$ was used for systems with a sRMSD greater than 5 \AA .

Figure 2.6. Left: sRMSD alignment of two protein conformations. Right: wRMSD alignment of the same structures. (A) HIV-1p^{7,8} 1KZK (yellow) onto 1HHP (blue), $c = 2 \text{ \AA}^2$. (B) RAN^{17,18} 1RRP (yellow) onto 1BYU (blue), $c = 5 \text{ \AA}^2$. (C) RNA Pol^{19,20} 1QLN (yellow) onto 1MSW (blue), $c = 5 \text{ \AA}^2$.



Identifying Domains and Hinge Regions

Inspection of the wRMSD alignment of EFG^{11,12} clearly shows that two possible solutions should exist: one where the upper domain is aligned and one where the lower domain is aligned. This inspired us to modify the technique in an effort to identify domains and hinge regions. This is possible if we change the initial sRMSD alignment.

As previously mentioned, the Gaussian weights are a direct result of the difference between the transformed atom pairs, calculated from the initial sRMSD fit. If the sRMSD alignment is performed using a select subset of the protein, this changes the weights and biases the wRMSD fit. In a way, we are taking advantage of the fact that a wRMSD fit has more than one minimum. Diamond suggested using multiple starting orientations to search for alternate solutions to an overlay²³, and we chose to align different sections of the protein as starting points to provide multiple solutions that can be ranked by the metrics previously discussed. This method will allow us to align different regions of the protein and identify common domains and linker regions.

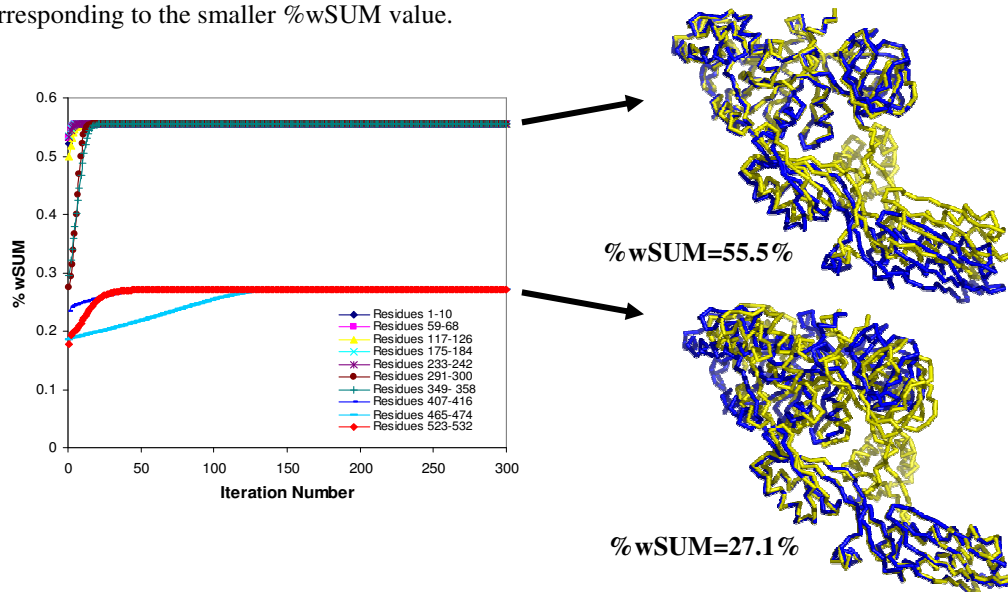
We chose to generate 10 initial sRMSD alignments based on local regions of the proteins. The initial standard alignment used 10 residues, chosen evenly spaced through the sequence. When larger sections are used (i.e. 20 residues), we found that the initial alignment could be based on two different mobile regions simultaneously. In such a case, the weighted alignment would not converge to a successful solution. We also found that evenly spacing our 10 local regions (i.e., 10 residues from every 10% section of the sequence) appears to adequately sample the entire protein structure (at least for the diverse test set used here). Making more than 10 initial alignments through choosing more frequent sections of the sequence yielded the same optimal alignments (data not shown).

After the initial local alignments, the 10 starting structures were refined with iterative wRMSD calculations in our regular way using the entire protein chain. The Gaussian scaling factor was set to a small value to maintain the local bias, $c = 2 \text{ \AA}^2$. This

resulted in 10 final, weighted alignments. The %wSUM was plotted against the iteration number for each test case. As previously mentioned, the %wSUM should increase to a stable maximum value corresponding to an optimal solution where the maximum number of atoms makes the most significant contribution. When starting from different subsets of the protein sequence, the alignment of the largest domain corresponded to the solution with the largest %wSUM value. Aligning the second largest domain lead to the second largest %wSUM value, and so on. This behavior was expected, and it was observed for all test cases. Below we demonstrate the technique on EFG, RAN, and DNA Pol.

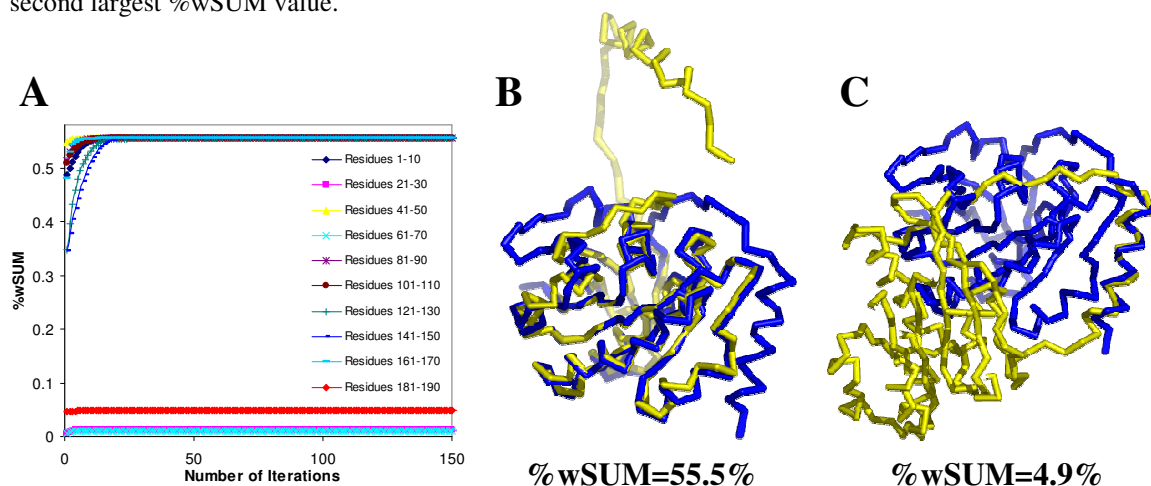
A plot of the %wSUM versus iteration number for the protein system EFG^{11,12} shows how using small subsets of the protein in the initial sRMSD leads to the identification of the two different domains, as shown in Figure 2.7. The weighted alignment based on the largest domain of EFG converged to the maximum %wSUM value (55.5%). Seven of the 10 local alignments (residues taken from 1-358) converged to this same solution. A second solution, based on the smaller domain of EFG, has a smaller %wSUM (27.1%) as expected. Three of the 10 local alignments converged to this second solution. The later solutions were initially aligned by sRMSD of residues 407-416, 465-474, and 523-532.

Figure 2.7. EFG.^{11,12} The behavior of the %wSUM metric as the weighted alignment is performed in an iterative manner. Ten different subsets of 1FNM (yellow) were used for the initial standard alignment onto 2EFG (blue) and then the weighted iterations were performed using the entire sequence ($c = 2 \text{ \AA}^2$). **(Top)** wRMSD alignment corresponding to the maximum %wSUM value. **(Bottom)** wRMSD alignment corresponding to the smaller %wSUM value.



In the case of RAN^{17,18}, the final weighted alignments from 7 of the 10 local alignments (residues 1-10, 41-50, 81-90, 101-110, 121-130, 141-150, and 161-170) converged to the maximum %wSUM value (55.5%), Figure 2.8. The corresponding weighted alignment shows how the largest domain of the protein is superimposed between the conformations. Our technique is even capable of the difficult task of finding an alignment based on RAN's N-terminal helix. This alignment corresponds to residues 181-190 in Figure 2.8A and has a much smaller %wSUM (4.9%) than the first 7 weighted alignments, as expected. The local alignments using residues 21-30 and 61-70 were to less structured regions of the protein, and the weighted alignments essentially converged to solutions with %wSUM near zero. Poor convergence and near-zero values indicate loop or hinge regions of a protein.

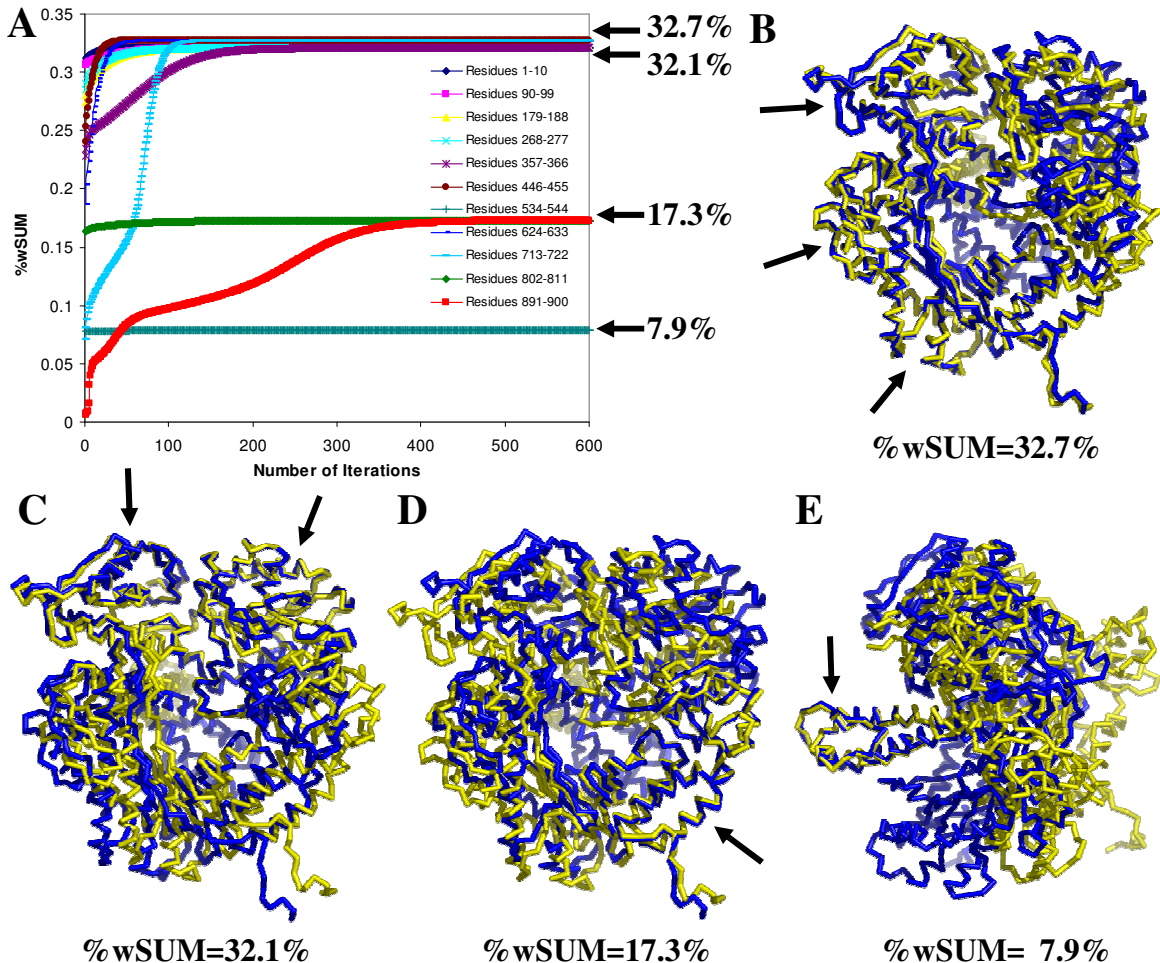
Figure 2.8. RAN.^{17,18} (A) The behavior of %wSUM as the weighted alignment is performed in an iterative manner. Ten different subsets of 1RRP (yellow) were used for the initial standard alignment onto 1BYU (blue) and then the weighted iterations were performed using the entire sequence ($c = 2 \text{ \AA}^2$). (B) wRMSD alignment corresponding to the maximum %wSUM value. (C) wRMSD alignment corresponding to the second largest %wSUM value.



DNA Pol is a very large protein with almost 900 residues and multiple domains.¹⁴ Using subsets of the protein for the initial sRMSD, we were able to find 4 distinct alignments based on different regions of the protein, provided in Figure 2.9. The weighted alignments from 3 of the 10 local alignments (residues 446-455, 624-633, 713-722) converged to the maximum %wSUM value (32.7%). The resulting weighted

alignment is based on the largest region of DNA Pol. The alignment from the second solution (%wSUM = 32.1%) corresponds to the weighted alignment based on 5 of the 10 local alignments (residues 1-10, 90-99, 179-188, 268-277, and 357-366). The alignments from the first two solutions are not the same; however, they share a common domain that is superimposed in both overlays. One of the 10 local alignments (residues 802-811) converged to a third solution (17.3%), based on a different region of DNA Pol than the first two solutions. The weighted alignment for the fourth solution initially aligned by sRMSD of residues 534-544 is based on a small region of secondary structure, and it has the lowest %wSUM (7.9%).

Figure 2.9. DNA Pol.¹⁴ (A) The behavior of %wSUM as the weighted alignment is performed in an iterative manner. Ten different subsets of 1IH7 (yellow) were used for the initial standard alignment onto 1IG9 (blue) and then the weighted iterations were performed using the entire sequence ($c = 2 \text{ \AA}^2$). The four distinct solutions are indicated on the right. (B) wRMSD alignment corresponding to the maximum %wSUM value. (C) wRMSD alignment corresponding to the second largest %wSUM value. (D) wRMSD alignment corresponding to the third largest %wSUM value. (E) wRMSD alignment corresponding to the smallest %wSUM value. This overlay is oriented differently than in (B–D). Arrows in (B–E) highlight regions with good alignment.



The optimal local alignments are basically identical to the global wRMSD alignments initiated from the sRMSD fit of the entire structure. This trend was seen for all of the protein systems as demonstrated by the comparison of the global wRMSD fit and the best wRMSD fit from local alignments, defined by highest %wSUM in Table 2.3. In the case of DNA Pol, two solutions had been found when examining the appropriate values for c to report. The alignment corresponding to the largest %wSUM value (32.7%) was found to be identical to the global fit with $c = 2 \text{ \AA}^2$. However, the second largest %wSUM value (32.1%) matched the global wRMSD fit with $c = 5 \text{ \AA}^2$.

Table 2.3. A comparison of the wRMSD fits using an initial global sRMSD alignment and the best result from initial local alignments. Two local wRMSD fits for DNA Pol are compared to two global wRMSD fits.

Protein System	Difference in sRMSD (\AA) between Global and Local wRMSD fits	Global Scaling Factor
HIV-1p	0	$c = 2 \text{ \AA}^2$
PKA	0	$c = 2 \text{ \AA}^2$
EFG	0	$c = 2 \text{ \AA}^2$
ER α	0	$c = 2 \text{ \AA}^2$
DNA Pol	0 (Fig. 9b), 5.90 (Fig. 9c)	$c = 2 \text{ \AA}^2$
DNA Pol	5.71 (Fig. 9b), 0.23 (Fig. 9c)	$c = 5 \text{ \AA}^2$
GroEL	0	$c = 5 \text{ \AA}^2$
RAN	0	$c = 5 \text{ \AA}^2$
RNA Pol	0.25	$c = 5 \text{ \AA}^2$

2.4 Conclusions

Our Gaussian-weighted alignment tool has been successfully applied to many dynamic proteins with two known conformations. We have also shown that an sRMSD alignment for these proteins is usually inappropriate. Our method is capable of selecting out the static core regions of flexible proteins and returning an alignment heavily weighted by those coordinates.

We have developed two techniques to utilize our Gaussian-weighted method. The first, a global wRMSD fit, uses the entire protein sequence for an initial sRMSD

alignment and performs iterative wRMSD fits of the entire structure with $c = 2$ or 5 \AA^2 . When protein conformations are similar ($\text{sRMSD} < 5 \text{ \AA}$), $c = 2 \text{ \AA}^2$ is suggested. For larger conformational changes ($\text{sRMSD} \geq 5 \text{ \AA}$), the larger scaling factor is recommended. These values work well, allowing the wRMSD fit to converge to an appropriate solution.

Our second technique, a local wRMSD fit, uses subsets of the protein sequence for an initial, local sRMSD alignment and then performs a wRMSD fit of the entire protein. The Gaussian scaling factor is kept set to 2 \AA^2 to maintain the local bias in the fit. The optimal solution is identified by the largest %wSUM. Using this second method, we were able to achieve multiple alignments based on different domains of the protein, and the solutions could be ranked by %wSUM.

Although a variety of alignment methods have previously been described to account for protein flexibility, we have developed a new method that is both general and robust. This method does not require any prior knowledge of the protein structure and removes the subjective nature of overlaying user-defined core regions of flexible proteins. Our novel technique can easily be incorporated into many RMSD overlay calculations.

This work has been published as:

Damm, K.L. and Carlson, H.A. Gaussian-Weighted RMSD Superposition of Proteins: A Structural Comparison for Flexible Proteins and Predicted Protein Structures. *Biophys. J.* **2006**, *90*, 4558-4573.

2.5 References

1. Kabsch, W. A Solution for the Best Rotation to Relate Two Sets of Vectors. *Acta Cryst* **1976**, A32, 922-923.
2. DeLano, W. L. The PyMOL Molecular Graphics System. 2002. DeLano Scientific LLC, San Carlos, CA, USA. <http://www.pymol.org>.
3. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res* **2000**, 28, 235-42.
4. Reiling, K. K.; Endres, N. F.; Dauber, D. S.; Craik, C. S.; Stroud, R. M. Anisotropic Dynamics of the Je-2147-HIV Protease Complex: Drug Resistance and Thermodynamic Binding Mode Examined in a 1.09 Å Structure. *Biochemistry* **2002**, 41, 4582-94.
5. Spinelli, S.; Liu, Q. Z.; Alzari, P. M.; Hirel, P. H.; Poljak, R. J. The Three-Dimensional Structure of the Aspartyl Protease from the HIV-1 Isolate Bru. *Biochimie*. **1991**, 73, 1391-6.
6. Madhusudan; Trafny, E. A.; Xuong, N. H.; Adams, J. A.; Ten Eyck, L. F.; Taylor, S. S.; Sowadski, J. M. Camp-Dependent Protein Kinase: Crystallographic Insights into Substrate Recognition and Phosphotransfer. *Protein Sci* **1994**, 3, 176-87.
7. Zheng, J.; Knighton, D. R.; Xuong, N. H.; Taylor, S. S.; Sowadski, J. M.; Ten Eyck, L. F. Crystal Structures of the Myristylated Catalytic Subunit of Camp-Dependent Protein Kinase Reveal Open and Closed Conformations. *Protein Sci* **1993**, 2, 1559-73.
8. Laurberg, M.; Kristensen, O.; Martemyanov, K.; Gudkov, A. T.; Nagaev, I.; Hughes, D.; Liljas, A. Structure of a Mutant Ef-G Reveals Domain Iii and Possibly the Fusidic Acid Binding Site. *J Mol Biol* **2000**, 303, 593-603.36.
9. Czworkowski, J.; Wang, J.; Steitz, T. A.; Moore, P. B. The Crystal Structure of Elongation Factor G Complexed with Gdp, at 2.7 a Resolution. *Embo J* **1994**, 13, 3661-8.37.
10. Shiau, A. K.; Barstad, D.; Loria, P. M.; Cheng, L.; Kushner, P. J.; Agard, D. A.; Greene, G. L. The Structural Basis of Estrogen Receptor/Coactivator Recognition and the Antagonism of This Interaction by Tamoxifen. *Cell* **1998**, 95, 927-37.
11. Franklin, M. C.; Wang, J.; Steitz, T. A. Structure of the Replicating Complex of a Pol Alpha Family DNA Polymerase. *Cell* **2001**, 105, 657-67.

12. Xu, Z.; Horwich, A. L.; Sigler, P. B. The Crystal Structure of the Asymmetric GroEL-GroES-(Adp)₇ Chaperonin Complex. *Nature* **1997**, *388*, 741-50.
13. Braig, K.; Adams, P. D.; Brunger, A. T. Conformational Variability in the Refined Structure of the Chaperonin GroEL at 2.8 Å Resolution. *Nat Struct Biol* **1995**, *2*, 1083-94.
14. Vetter, I. R.; Nowak, C.; Nishimoto, T.; Kuhlmann, J.; Wittinghofer, A. Structure of a RAN-Binding Domain Complexed with RAN Bound to a GTP Analogue: Implications for Nuclear Transport. *Nature* **1999**, *398*, 39-46.
15. Stewart, M.; Kent, H. M.; McCoy, A. J. The Structure of the Q69I Mutant of GDP-RAN Shows a Major Conformational Change in the Switch II Loop That Accounts for Its Failure to Bind Nuclear Transport Factor 2 (Ntf2). *J Mol Biol* **1998**, *284*, 1517-27.
16. Cheetham, G. M.; Steitz, T. A. Structure of a Transcribing T7 RNA Polymerase Initiation Complex. *Science* **1999**, *286*, 2305-9.
17. Yin, Y. W.; Steitz, T. A. Structural Basis for the Transition from Initiation to Elongation Transcription in T7 RNA Polymerase. *Science* **2002**, *298*, 1387-95.
18. Flower, D. R. Rotational Superposition: A Review of Methods. *J Mol Graph Model* **1999**, *17*, 238-44.
19. Diamond, R. A Note on the Rotational Superposition Problem. *Acta Cryst.* **1988**, *A44*, 211-216.
20. Diamond, R. On the Multiple Simultaneous Superposition of Molecular Structures by Rigid Body Transformations. *Protein Sci* **1992**, *1*, 1279-87.
21. Schneider, T. R. A Genetic Algorithm for the Identification of Conformationally Invariant Regions in Protein Molecules. *Acta Crystallogr D Biol Crystallogr* **2002**, *58*, 195-208.
22. Schneider, T. R. Domain Identification by Iterative Analysis of Error-Scaled Difference Distance Matrices. *Acta Crystallogr D Biol Crystallogr* **2004**, *60*, 2269-75.
23. Godzik, A. The Structural Alignment between Two Proteins: Is There a Unique Answer? *Protein Sci* **1996**, *5*, 1325-38.

CHAPTER 3

Application of the wRMSD Method to Predicted Protein Structures and Homologous Proteins

3.1 Introduction

In the previous chapter, the development of a Gaussian-weighted superposition method was described.¹ Given the prior success with this technique, we have now applied it to evaluate predicted protein structures. Comparing a predicted fold to its experimentally determined target structure is another case of comparing two protein conformations of the same sequence. The quality of the fit directly measures the accuracy of the prediction. The nature of our weighted RMSD (wRMSD) implementation also notes if substructures are correctly predicted but misoriented relative to one another. Furthermore, it is possible to create a version of RMS/coverage graphs² by varying the weighting term. These features make wRMSD fits a complementary method for evaluating protein structure predictions.

In addition, we have now coupled our wRMSD tool with a BLAST pair-wise sequence alignment as a method to overlay homologous proteins. Evolutionarily related proteins generally retain a similar tertiary fold that is more conserved than the amino acid sequence.³⁻⁵ Structure typically is related to function; hence, they may also share a common biological activity.⁶ As a result, the identification of a homolog is a very useful means to infer the function and/or predict the structure of an uncharacterized protein. Many databases exist that classify proteins into families by their structures, including but not limited to SCOP⁷, CATH⁸, FSSP⁹, CAMPASS¹⁰, Entrez3D¹¹, ASTRAL¹²,

HOMSTRAD¹³, ALBASE¹⁴, and LPFC¹⁵. A review from Orengo and Thornton provides a very thorough discussion of protein evolution from a structural standpoint.¹⁶

The wRMSD method can overcome errors in the initial sequence alignment because misaligned atom pairs will not be in close spatial proximity and consequently will make little contribution to the wRMSD calculation. We are able to show that to a minimal sequence identity we can recover the same weighted superposition whereas the standard RMSD (sRMSD) fit gives varying results. Hence, our weighted superposition technique overcomes the dependency of structural overlays on the initial atom pairing and removes the need to determine the best sequence alignment method and parameters to use for a particular system. We can also use the wRMSD method to determine potential mispaired residues because the calculated weights correlate back to the initial sequence alignment. Furthermore, our technique can be used to align homologs with low sequence identity and large conformational differences, an area where both sequence and structural-based methods may fail.

3.2 Computational Methods

Protein Structure Prediction Dataset

To show the method's applicability to evaluate protein structure predictions, we explored several targets from the Critical Assessment of Techniques for Protein Structure Prediction (CASP) 5 competition¹⁷. Five targets were chosen based on their category and difficulty: Target 147 Ycdx¹⁸, Target 162-3 Actin Filament Capping Protein CapZ¹⁹, Target 170 model 1 for the FF domain of HYPA/FBP11²⁰, Target 172 S-Adenosylmethionine-Dependent Methyltransferase²¹, and Target 179 Spermidine Synthase²². The corresponding experimental structures were downloaded from the Protein Data Bank (PDB)²³, and the first chain of each structure was used as the reference

structure for the wRMSD alignments. We chose several predicted structures that ranged from high to low GDT-TS (Global Distance Test Total Score). Using the CASP5 website, we obtained the “Model 1” coordinates from the groups listed in Table 2, except as noted for Target 162. Table 3.1 is a summary of the targets, their category, their entry in the PDB, and the groups that generated the predictions used in this study.

Table 3.1. Summary of Targets used in CASP5 Evaluation.

Target	Category*	PDB ID	Groups
147	FR	1M65	2, 29, 10, 331, 437, 52, 246, 64, 25
162-3	NF	1IZN	132, 373, 29_3, 531, 52, 25_2, 169, 368, 105
170	FR/NF	1UZC	517, 51, 294, 373, 45, 28, 80, 61, 314
172	CM/FR	1M6Y	517, 373, 417, 537, 40, 56, 513, 282, 180, 397
179	CM	1IY9	427, 246, 471, 270, 16, 529, 291, 183, 400, 32, 531, 139

* FR is Fold Recognition, NF is New Fold, and CM is Comparative Modeling.

Homologous Protein Dataset

Homologous protein pairs were obtained from the Aug. 2005 release of HOMSTRAD¹³ and used as a benchmark to evaluate the wRMSD method. The HOMSTRAD database consists of 3454 proteins clustered into 1032 homologous families, ranging from 8-94% sequence identity (%ID). The protein coordinates were downloaded from the PDB²³, and according to the information provided on the HOMSTRAD website, residues from the specified chain were extracted to use in the sequence and structural alignments. Aligning proteins with high sequence identity is straightforward, so for this study we chose to focus on the more difficult cases of homologous proteins in the low to intermediate range (17-39% ID).

Gaussian-Weighted Superposition

The predicted protein structures were aligned using the Gaussian-weighted superposition technique described in Chapter 2.¹ A scaling factor of $c = 5 \text{ \AA}^2$ was employed for easy targets with small deviations (Targets 179 and 172), and $c = 12 \text{ \AA}^2$ was used for hard targets with greater differences (Targets 170, 147, and 162-3).

To superimpose homologous proteins, the BLAST-based tool ‘BLAST 2 Sequences’^{24,25} was incorporated into the wRMSD method described in Chapter 2. Default BLAST parameters are used with the exception that the low complexity filter parameter is turned off. The technique is performed using C α coordinates, but it could easily be extended to any atom subset. The procedure requires 4 steps to align homologs: (1) parse residue sequence from PDB files to generate FASTA files for BLAST alignment, (2) run BLAST to determine an appropriate list of atom pairs, (3) calculate an initial sRMSD alignment (non-weighted) to bring the two proteins into proximity, and (4) conduct iterative wRMSD fitting until convergence is reached. Complete mathematical details of the wRMSD procedure can be found in Chapter 2.

3.3 Results and Discussion

Using wRMSD to Evaluate Protein Structure Predictions

The act of evaluating a predicted protein structure against its experimentally determined target is another example of comparing two conformations of the same protein sequence. To show how wRMSD can be used to evaluate a predicted structure, we examined five systems used in the CASP5 competition¹⁷. The targets were chosen based on increasing difficulty: Target 179 (Comparative Modeling), Target 172 (Comparative Modeling/Fold Recognition), Target 170 (Fold Recognition/New Fold), Target 147 (Fold Recognition), and Target 162-3 (New Fold). These specific targets were discussed in several papers that assessed the community’s performance as a whole.²⁶⁻²⁸ Each of these assessment papers relied heavily on the GDT-TS metric in their ranking of submitted predictions. The GDT-TS values discussed here were obtained from the CASP5 website (<http://predictioncenter.org/casp5/Casp5.html>).

Like other techniques in the literature, the GDT procedure evaluates two structures based on an sRMSD fit of a subset of atoms²⁹, but what makes GDT unique is that it is implemented to provide a type of weighted evaluation in its final GDT-TS value. GDT is an iterative method that determines the maximum number of residues that can be sRMSD fit within a given distance (i.e., performs an sRMSD overlay of all atoms in the structure that can be simultaneously superimposed within 0.5 Å, 1 Å, 1.5 Å, 2 Å... up to 10 Å). GDT uses many starting alignments and an iterative procedure to identify the optimal sRMSD alignment of the largest subset possible. The GDT-TS score is based on the percent of atoms that can contribute to a particular sRMSD alignment: $\text{GDT-TS} = (P_1 + P_2 + P_4 + P_8)/4$ where P_m is the percent of atoms that sRMSD fit within m Å. In the GDT-TS value, the atoms within 1 Å agreement have a weight of 100% in the GDT-TS; atoms within 2 Å, 4 Å, and 8 Å have weights of 75%, 50%, and 25%, respectively. The GDT technique can be used to create RMS/coverage graphs² by plotting the percentage of atoms (P_m) versus the cutoff m . As the cutoff m increases, P_m also increases.

Comparing a predicted structure to its target involves more structural variation than the comparison of two related crystal structures. As one might expect, we found that larger scaling factors were necessary to provide accurate comparisons. Paralleling our study of flexible proteins, we again found that a smaller scaling factor ($c = 5 \text{ \AA}^2$) was necessary for easy targets with small deviations and larger values ($c = 12 \text{ \AA}^2$) were needed for hard targets with greater differences. The figures below provide a scale to show how the distances (d_n) compare with their corresponding weights for $c = 5$ or 12 \AA^2 . This allows the reader to compare the wRMSD weights in the figures to those of GDT-TS noted above. The wRMSD technique can also be used to create RMS/coverage graphs by plotting the %wSUM versus c . As the scaling factor c increases, %wSUM also increases in a manner similar to RMS/coverage graphs from GDT (provided in Appendix 2).

Overall, the GDT-TS metric is the most representative measure of a prediction, and it is the most widely accepted evaluation tool^{17,26-28}. However, its rankings do not always match manual/visual rankings of challenging targets like new folds and difficult fold recognition cases.²⁷ In particular, Aloy et al.²⁸ found that GDT-TS over-ranked “fragment” submissions that provided coordinates for only a portion of the sequence. To prevent a similar bias in our use of wRMSD, we provide a %wSUM score based on the fit of the coordinates in the prediction (n in Eq. 2.11 equals the number of atoms in the prediction) and a %wSUM-ALL which corrects for any omitted coordinates (n in Eq. 2.11 equals the number of atoms in the target). If a prediction provides all C α coordinates, %wSUM and %wSUM-ALL are equal. If some are omitted, %wSUM-ALL will be proportionally less than %wSUM. (For a more accurate comparison, %wSUM-ALL is used in our RMS/coverage graphs in Appendix 2).

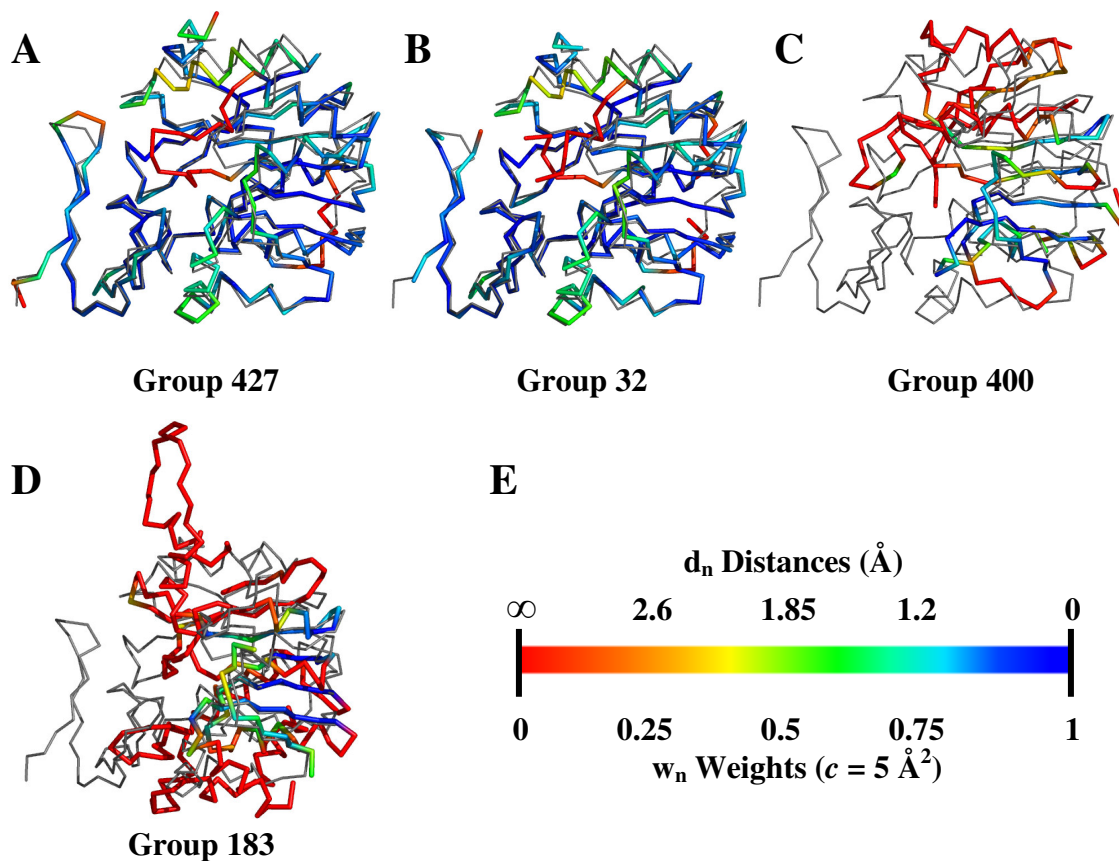
Target 179 is the “easiest” target included in our study. Many of the teams provided submissions that closely resembled the target. We randomly chose five exceptional submissions, two good/moderate submissions, and five poor submissions. Table 3.2 shows that the ranking provided by %wSUM-ALL matches that of GDT-TS with the exception of groups 32 and 400.

Table 3.2. Target 179, wRMSD rankings ($c = 5 \text{ \AA}^2$) compared to GDT-TS values.

Group	%wSUM-ALL	%wSUM	GDT-TS
427	76.6	76.6	86.95
32	76.5	77.0	28.65
246	76.3	76.3	86.68
471	75.8	75.8	85.77
270	74.6	74.6	84.40
16	64.0	64.0	77.47
529	63.8	75.1	72.08
291	24.0	37.4	34.12
400	18.9	32.6	29.11
183	16.3	19.1	29.29
531	5.6	5.6	11.13

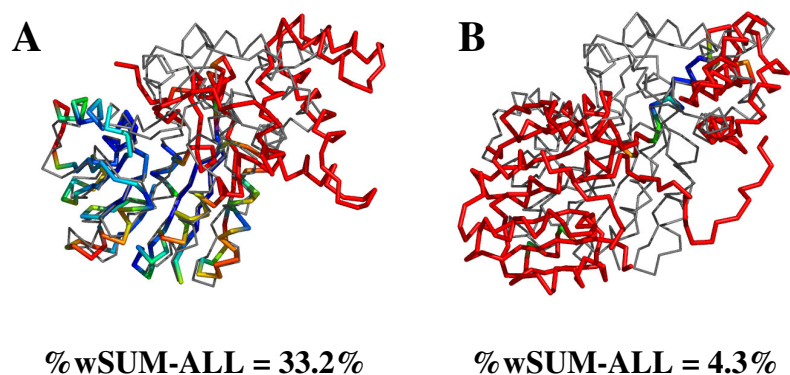
Figure 3.1 shows the wRMSD alignment of teams 427's and 32's top-ranked predictions. The regions in blue and green have high weights and are in excellent agreement with the target structure. The cause for 32's poor GDT-TS rank is unknown. The CASP5 website gives the low rank and also provides a weak RMS/coverage plot (see Appendix 2), but the values for P_1 , P_2 , P_4 , and P_6 listed from the GDT-TS analysis do not match the plot and indicate that the GDT-TS score should be greater than 85. It appears that there may have been a simple typographical or data processing error. The other good predictions look very similar to the alignments in Figure 3.1; the differences are minor and are localized in the two red, low-weight regions. Teams 529, 291, 400, and 183 provided significantly fragmented submissions, and the %wSUM-ALL does not match %wSUM in those cases. Without the correction of %wSUM-ALL, team 529 would have been ranked too high. Figure 3.1C,D shows that 400 should be higher ranked than 183 because the lowest part of the β -sheet region has better agreement and higher weights.

Figure 3.1. The wRMSD alignments of (A) group 427's and (B) group 32's predictions (thick, colored lines) to Target 179 (thin, gray line). The wRMSD alignments of (C) group 400's submission and (D) group 183's submission are given as examples of the comparison of a fragment. The target has the same orientation in both alignments. (E) The scale at the bottom shows how smaller deviations (blue) are more heavily weighted in the wRMSD. Deviations over 3.9 Å have weights under 5% (red).



Target 172 has a central domain that a few teams predicted well; of those teams, we examined the submissions of groups 517 and 373. We also randomly chose four moderate submissions and four poor submissions. An interesting feature of the wRMSD local alignments is that alternate, lower-ranked overlays are also provided. Figure 3.2 shows that the submission from group 517 has two solutions, one for the agreement in the central domain and a second solution showing a properly predicted helix in the more difficult domain, respectively. Two independent wRMSD solutions show that the two regions were properly solved but not oriented in the correct relative positions. This example is simply provided to demonstrate a feature of the method.

Figure 3.2. The submission from group 517 to target 172 has two solutions (A) and (B) by wRMSD fitting. The second solution (B) is scored much lower because it is only a match of a small helix. The target (gray, thin line) is in the same orientation in both alignments. The color code of the weights is the same as in Figure 3.1E.

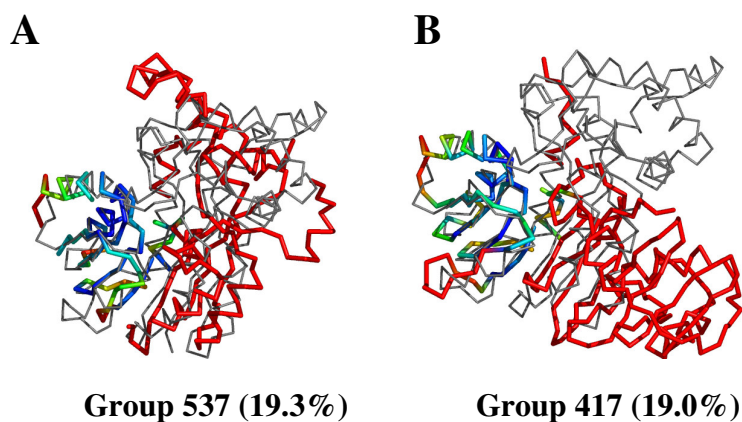


The second solution has a very low %wSUM-ALL, and its consideration is not necessary to properly rank the predictions of Target 172. Table 3.3 shows that the rankings from wRMSD match those of GDT-TS with the exception of the moderate submissions of groups 537 and 417. Figure 3.3 shows that the difference in the rankings is due to small improvements in the weighted core and possibly the cumulative contributions of small weights in the large red region. However, the difference in ranks is small, and both groups 537 and 417 can be maximally aligned to highlight the good agreement in the same core region.

Table 3.3. Target 172, wRMSD rankings ($c = 5 \text{ \AA}^2$) compared to GDT-TS values.

Group	%wSUM-ALL	%wSUM	GDT-TS
517	33.2	33.2	46.50
373	22.0	22.0	31.83
537	19.3	22.9	25.85
417	19.0	20.6	26.20
40	18.8	30.7	25.00
56	15.5	36.4	22.27
513	7.5	8.4	17.32
282	4.8	5.3	10.50
180	3.7	3.7	8.53
397	1.4	17.6	2.99

Figure 3.3. wRMSD fits for groups (A) 537 and (B) 417 to Target 172. The %wSUM-ALL values for the best wRMSD fit are given in parentheses. The color code of the weights is the same as in Figure 3.1E. The target (gray, thin line) is in the same orientation in both alignments.



Target 170 is a “new fold” target. It is considered a relatively straightforward example of the most difficult category.^{27,28} Predictions for these more challenging targets tend to have larger deviations, and a scaling factor of 12 was necessary. We found that alternate solutions became more common and more significant as the difficulty of the target increased. The submissions from the top chosen groups provided secondary alignments showing that more than one region of the structure was solved properly, but the regions were not correctly oriented relative to one another. This feature of the local wRMSD fitting is an advantage over using GDT, which does not provide alternate, lower-ranked solutions.

Figure 3.4. The multiple wRMSD solutions for the top three structures chosen for Target 170 (thin, gray line). **(A)** The wRMSD alignments of team 517's prediction (thick, colored line). **(B)** The wRMSD alignment of team 400's fragment submission. **(C)** The solutions for team 51. The target has the same orientation in both alignments. **(D)** The scale shows the weights for these wRMSD fits based on $c = 12 \text{ \AA}^2$. Deviations over 6.0 \AA have weights under 5% (red).

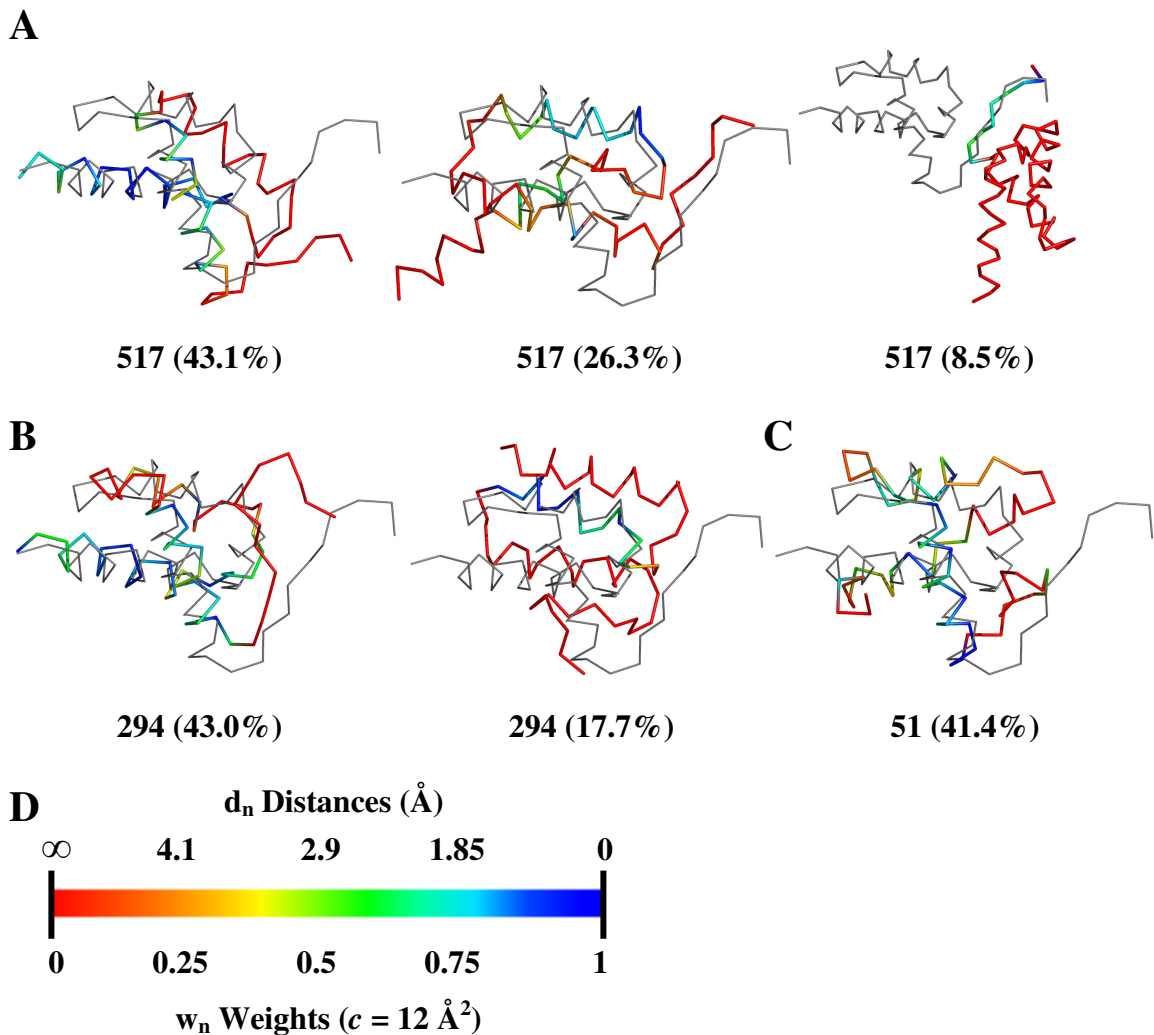


Figure 3.4 compares the multiple wRMSD solutions for the first three groups. wRMSD and GDT-TS have similar rankings for the best alignments (Table 3.4), except that the good submissions of groups 294 and 51 are switched. The best alignments of all groups match the central helix down the center of the structure, but groups 517 and 294 also provide a second helix in the correct relative positions. Group 51 does provide additional helical structure, but the orientation is not quite as good, and the weights are correspondingly lower (with scaling factors $> 20 \text{ \AA}^2$ the weights become more significant

and group 51's submission is ranked highest, see Appendix 2). The second solutions for groups 517 and 294 show that the third helix is properly predicted but misoriented relative to the first two helices. The third solution for group 517 shows additional agreement in the sheet region. This third solution has a low %wSUM-ALL and is an example of the border line for a significant solution.

Table 3.4. Target 170, wRMSD rankings ($c = 12 \text{ \AA}^2$) compared to GDT-TS values.

Group	%wSUM-ALL	%wSUM	GDT-TS
517	43.1	43.1	53.26
294	43.0	43.0	51.45
51	41.4	41.4	51.81
373	31.9	31.9	40.94
45	28.3	28.3	39.86
28	25.2	25.2	36.96
80	25.1	25.1	35.51
61	13.2	20.2	26.81
314	10.7	10.7	19.56

Target 147 is a challenging case because of its classification and its size. Table 3.5 shows that the wRMSD alignment ranks entries 2, 29, 10, and 437 in agreement with the GDT-TS metric. All of the alignments have good %wSUM-ALL scores because of good to moderate agreement throughout much of the structure. Figure 3.5A,B,D shows the similar fits of the submissions for groups 2, 10, and 437 to the target, and Figure 3.5A,C shows the fit of the submission from group 331. We were surprised to see the structure from group 331 ranked so much higher with the %wSUM-ALL metric as compared to the GDT-TS metric. The 331 entry is pulled up in rank by wRMSD because it has excellent placement of three adjacent secondary structures (significantly blue regions in Figure 3.5C). With $c = 12 \text{ \AA}^2$, there is still the intended bias of the method to identify local regions with exceptional agreement over a larger collection of residues with modest agreement. When the scaling factor is larger than 20 \AA^2 , the bias shifts toward matching more of the global structure, and 437 is ranked significantly higher than 331 (see Appendix 2). The disagreement in the rank of entry 246 is not significant because of its low rank by both wRMSD and GDT-TS.

Figure 3.5. wRMSD fits for groups (A) 2, (B) 10, (C) 331, and (D) 437 (thick, colored lines) to Target 147 (gray, thin line). The %wSUM-ALL values for the best wRMSD fit are given in parentheses. The color code of the weights is the same as in Figure 3.4D. The target is in the same orientation in both alignments.

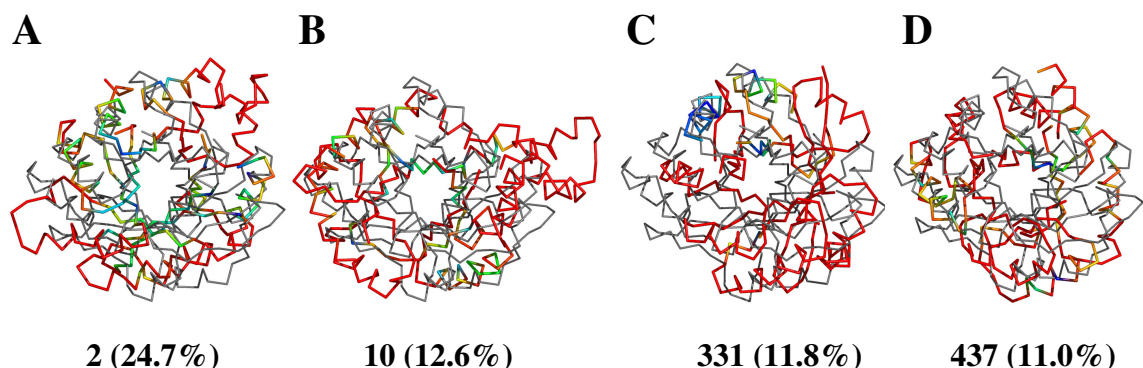


Table 3.5. Target 147, wRMSD rankings ($c = 12 \text{ \AA}^2$) compared to GDT-TS values.

Group	%wSUM-ALL	%wSUM	GDT-TS
2	24.7	24.7	33.44
29	19.2	19.2	27.57
10	12.6	12.6	24.36
331	11.8	12.9	16.66
437	11.0	13.0	21.80
52	5.9	8.7	9.62
246	5.3	5.3	12.07
64	4.1	7.4	7.16
25	3.6	20.7	4.28

The most difficult target we investigated was 162. The third domain was classified as a new fold, and we focused our analyses on these residues in the submitted predictions. Table 3.6 shows that the best submissions are ranked highest but are in mixed order between wRMSD and GDT-TS. The rank order when $c > 50 \text{ \AA}^2$ appeared to be a good metric of a more global score. The groups rank $373 > 132 > 437 > 29 > 2$ with this high scaling factor (Appendix 2). This is in agreement with Aloy et al.²⁸ who ranked group 373 highest based on visual inspection, followed by group 132; groups 2 and 29 scored significantly below 373 and 132. It is encouraging that wRMSD with a larger scaling factor matches the rankings provided by visual inspection. Furthermore, the GDT-TS rank order is $132 > 373 > 29 > 437 > 2$, indicating that our larger- c calculation is not a simple reproduction of GDT-TS. Figure 3.6 provides the best wRMSD solution for

each of the top five submissions evaluated in this study. The entries are ordered by the “global” group rank noted above, but the alignments and weights are from an wRMSD fit with $c = 12 \text{ \AA}^2$. This allows the reader to compare the structures for global and local characteristics. The best solutions have several pieces of secondary structure in proper relative locations. wRMSD fits have a short coming that is also seen in GDT-TS: matching a single long helix provides a relatively good score. The regular structure of a helix is simply easy to superimpose with good agreement (easier than superimposing a loop, turn, or twisted β -sheet that has more structural variation). The high score for helices simply reflects that they are the easiest substructure to properly predict.

Figure 3.6. wRMSD fits for groups (A) 373, (B) 132, (C) 437, (D) 29, and (E) 2 to Target 162-3. The order A-E reflects a rank order based on the RMS/coverage graph, but the overlays and their weights are from a local wRMSD fit with $c = 12 \text{ \AA}^2$. Two significant solutions were obtained for each group’s entry but only the best is shown. The %wSUM-ALL values for the individual wRMSD solutions are given in parentheses. The color code of the weights is the same as in Figure 3.4D. The target (gray, thin line) is in the same orientation in both alignments.

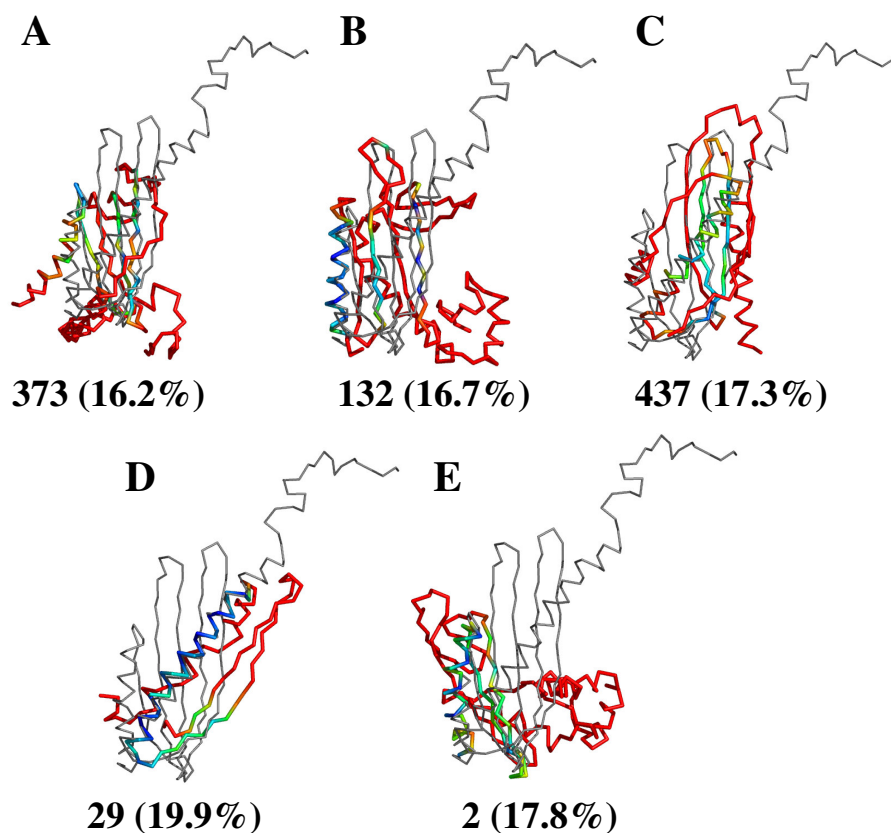


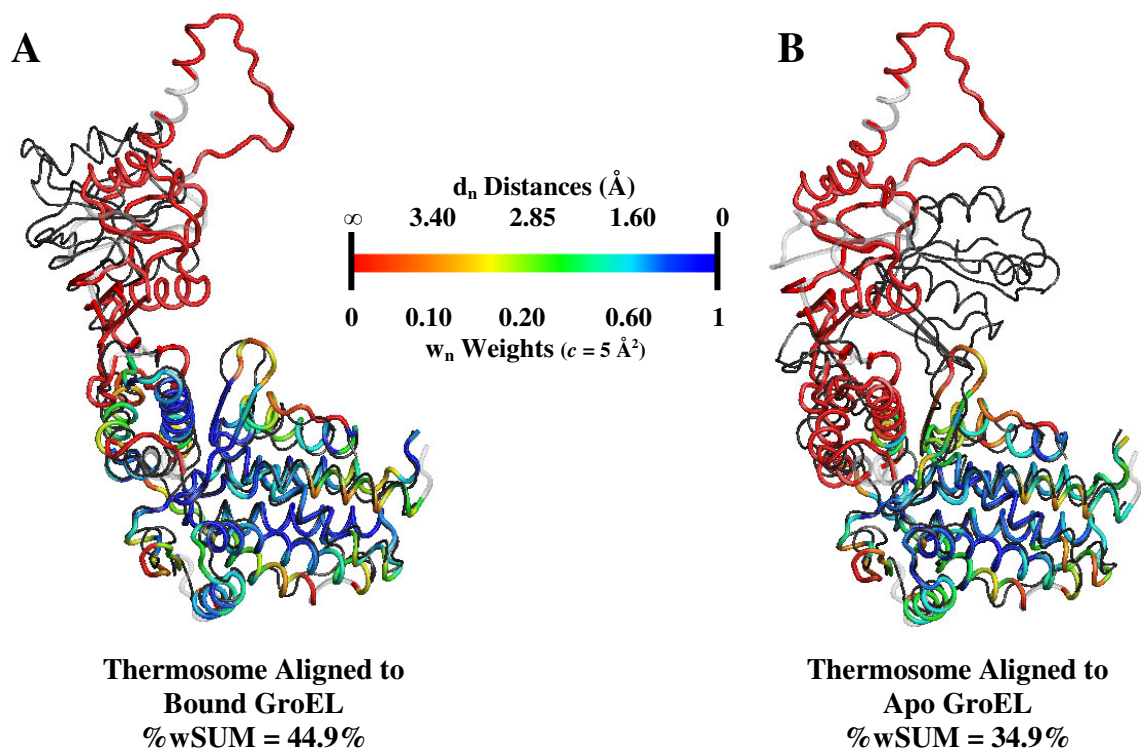
Table 3.6. Target 162-3, wRMSD rankings ($c = 12 \text{ \AA}^2$) compared to GDT-TS values.

Group	%wSUM-ALL	%wSUM	GDT-TS
29_3	19.9	34.4	23.512
2	17.8	17.8	20.238
437	17.3	19.1	22.173
132	16.7	16.7	24.405
373	16.2	16.2	24.107
397	13.8	34.5	18.452
282	10.8	12.6	15.923
227	10.4	25.1	14.435
180	7.3	7.8	12.649
291	6.8	9.1	10.268
196	6.0	20.1	7.589

Homologs: Low Sequence Identity and Large Conformational Differences

In Chapter 2, we were able to show an improved superposition of two conformations of the chaperonin protein GroEL, which undergoes a large conformational change between the bound (PDB ID: 1AON³⁰) and apo (PDB ID: 1OEL³¹) forms. In Figure 3.7, we use this system again to demonstrate the potential difficulties of fitting homologous, flexible proteins. With our technique, either conformation of GroEL can be appropriately superimposed to the bound form of its archaeal homolog, the thermosome (PDB ID 1A6E³²). The easier case of fitting the two bound conformations is shown in Figure 3.7A, and Figure 3.7B shows the more difficult comparison of the bound form of the thermosome to the apo form of GroEL. Using sequence alignments alone may prove difficult in some regions because the sequence identity between the homologs is low (20.8%). Fold-based techniques can identify the homolog from the similar, bound conformation and provide an appropriate standard fit. In cases where the structures of homologs are only available in alternate conformations, those same techniques have difficulty. Our method is able to overcome errors from the initial atom pairing due to low sequence identity and large conformational differences by only weighting regions of the protein in good structural agreement.

Figure 3.7. Chaperonin family (20.8% ID). Most techniques would readily identify the similarity between the thermosome and GroEL in the similar bound conformation, but they may not identify its similarity with the apo conformation of GroEL. **(A)** wRMSD superposition of the bound conformation of GroEL³⁰ (thick, colored lines) onto the homologous thermosome³² (thin, black lines). Light gray regions of GroEL indicate residues within gaps in the alignment. **(B)** wRMSD fit of the apo conformation of GroEL³¹ (thick, colored lines) onto its homolog thermosome³² (thin, gray lines). The value of %wSUM gives the normalized weights of all residues, showing that the two bound conformations in A have greater similarity than the two conformations in B. The scale shows the weights for these wRMSD fits based on $c = 5 \text{ \AA}^2$.



Overcoming Errors in the Initial Sequence Alignment

When the sequence identity between proteins is high ($\geq 40\%$ ID), most pair-wise, sequence-alignment methods perform equally well and generate appropriate correspondence between residues.³³⁻³⁶ However, pairing residues is a difficult task when homologous proteins have sequences with intermediate to low sequence identity. Sequence alignments may be considerably different depending on the program that is used and the parameters employed, such as the scoring matrix, gap opening penalty, and gap extension penalty.^{37,38} In turn, this will affect a standard superposition because of the dependency on the initial pairing. Conversely, weighted fits create superpositions that are largely independent of the method for sequence alignment. The wRMSD technique

overcomes errors in the alignment because mispairings are not in close spatial proximity and so have low weights and little contribution to the fitting.

For each pair of homologs in the low to intermediate range (17-39% ID), a variety of sequence alignments were obtained. First, BLAST parameters were varied (scoring matrix, gap opening penalty, and gap extension penalty) and second, default parameters were used with different sequence-alignment programs: BLAST^{24,25}, SIM³⁹, FASTA⁴⁰, ALIGN⁴¹, CLUSTALW³⁷, and TCOFFEE⁴². (It should be noted that the low-complexity filter was turned off in all applications of BLAST.) Each alignment was used to generate standard and weighted structural superpositions. For each protein, the variation across the structural superpositions was measured as RMSD between the final coordinates (note: this use of RMSD is simply a measure of the difference in two sets of coordinates, not a fitting procedure). The raw data is provide in Appendix 2. For each test case, the difference across the standard superpositions is appreciably larger, and the use of weighted superpositions overcomes the differences in the sequence alignments to give a more consistent structural comparison.

Five variations of BLAST were used which altered the gap penalty, extension penalty, and scoring matrix. Each of the resulting superpositions was compared to one another; the ten unique comparisons across each of the five results were averaged for Table 3.7. Table 3.7 shows that when varying the BLAST parameters the standard superpositions gave very different results (average differences ranged from 0.494 – 4.866 Å), but the weighted fits showed little difference (averages only ranged 0.062 – 0.742 Å). When the five sRMSD superpositions are very similar (low average difference), it indicates that the initial sequence alignments are also very similar. The weighted superpositions show consistency whether or not the sequence alignments agree.

Table 3.7. Differences in the structural superpositions for a diverse set of homologous proteins.^a A complete set of references for the crystal structures is provide in Appendix 2. Both standard and weighted superpositions were generated from a variety of sequence alignments. The sequence alignments were altered by varying the parameters within BLAST or varying the code used for the alignment. The differences across the superpositions were measured in RMSD (Å) between the coordinates. Average differences are reported above, but all calculated RMSD are included in Appendix 2.

Homologs (% ID)¹³ PDB Codes	BLAST Parameters		Seq Alignment Codes	
	Standard Fit	Weighted Fit	Standard Fit	Weighted Fit
Serine/Threonine Phosphatase (39%) 1FJM & 1TCO	0.494	0.062	1.185	0.021
Glutathione Synthase (37%) 1M9W & 2HGS	0.600	0.080	0.162	0.049
Interferon (35%) 1AU1 & 1ITF	1.224	0.301	1.188	0.339
Adenosylmethionine Decarboxylase (33%) 1I7B & 1MHM	0.655	0.135	0.716	0.102
Clostridial Neurotoxin Zinc Protease (31%) 1EPW & 3BTA	0.719	0.098	0.632	0.091
Sulfatase (29%) 1AUK & 1FSU	1.523	0.352	1.273	0.598
Protocatechuate-3,4-Dioxygenase (28%) 3PCG (chain A) & 3PCG (chain M)	1.160	0.111	3.358	0.132
Aminotransferase (27%) 1A3G & 5DAA	1.181	0.329	1.719	0.137
SpoU rRNA Methylase (26%) 1IPA & 1GZ0	4.866	0.644	3.676	0.347
FMN Oxidoreductase (25%) 1OYC & 2TMD	1.736	0.536	3.298	0.363
Queuine tRNA-Ribosyltransferase (25%) 1IQ8 & 1K4G	0.819	0.072	0.505	0.170
tRNA Synthetase (24%) 1GLN & 1QTQ	2.407	0.459	1.821	0.536
DNA Methylase (23%) 1BOO & 1EG2	1.325	0.119	2.325	0.140
DNA Topoisomerase (22%) 1AB4 & 1BJT	2.742	0.742	1.445	0.983
Pyridoxal-Phosphate Enzymes (21%) 1TDJ & 2TYS	1.022	0.331	3.681	0.649
Iron/Ascorbate Oxidoreductase (20%) 1BK0 & 1DCS	2.455	0.556	5.069	1.185
Molybdopterin Dehydrogenase (19%) 1FFV & 1FO4	0.531	0.098	1.238	0.555
Splicesomal Protein, Internalin B (19%) 1A9N & 1D0B	2.126	0.542	2.890	0.965

Table 3.7 cont.

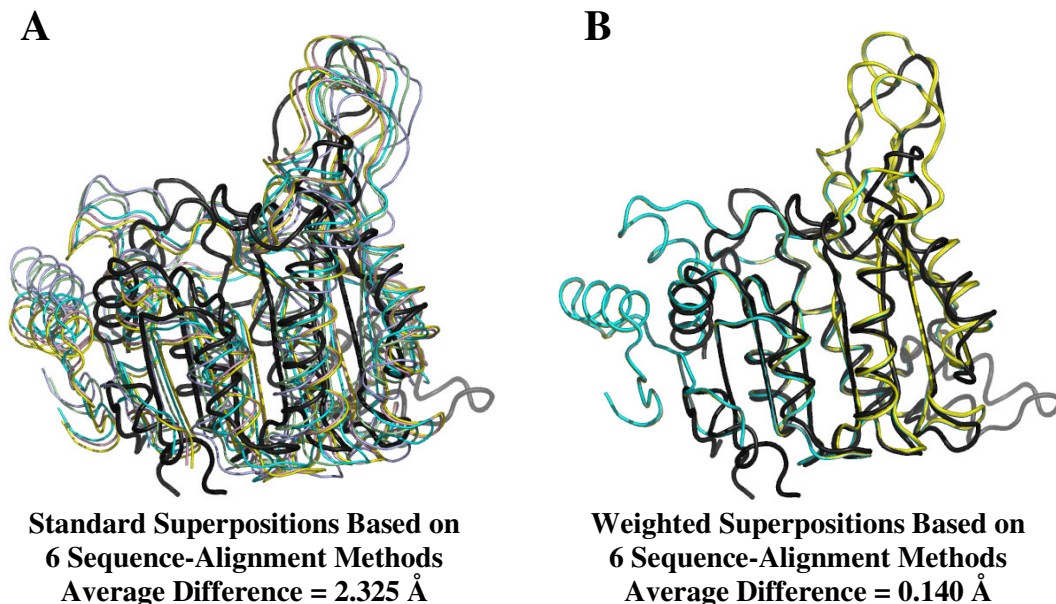
Asp/Glu/Hydantoin Racemase (18%) 1B74 & 1JFL	1.513	0.264	4.961	0.414
Polysaccharide Lyase (18%) 1CB8 & 1EGU	0.792	0.081	2.089	0.436
PHBH-like Proteins (17%) 1FOH & 1PB3	2.287	0.437	4.744	1.231

^a Smaller values note a greater agreement between the superpositions.

For the comparison of alignment programs, standard and weighted superpositions were created using sequence alignments from BLAST, SIM, FASTA, ALIGN, CLUSTALW, and TCOFFEE. The fifteen unique comparisons across each of the six resulting superpositions were averaged for Table 3.7. When changing the programs used for the initial sequence alignment, the difference across the standard superpositions ranged 0.162 – 5.069 Å, but the difference across the weighted superpositions were only 0.021 – 1.231 Å.

The largest differences in structural superpositions occurred from varying the programs used for the initial sequence alignment. Figure 3.8 uses DNA methylase homologs^{43,44} (23% ID) to show how the standard superpositions are noticeably different when varying the sequence comparison method (Figure 3.8A). For this example, the average difference across the six standard superpositions is 2.325 Å, and the variation between each standard fit is visibly large. Conversely, the weighted superpositions are indistinguishable by eye (Figure 3.8B); the average difference of the weighted superpositions is only 0.140 Å. Most importantly, the weighted superpositions resulted in an improved fit over the standard superpositions, particularly in the core region which is structurally conserved between the homologous proteins. After all, a consistent superposition is only useful if it is also an improved superposition!

Figure 3.8. DNA methylase family (23% ID). Weighted structural superpositions are nearly independent of the sequence alignment method, but standard superpositions are greatly effected. Six sequence-alignment codes were used to determine residue pairings. **(A)** Overlays of 1BOO⁴³ (thin, colored lines) to 1EG2⁴⁴, (thick, black line) from standard superpositions based on six different sequence alignments. The average difference in the superpositions is 2.325 Å. **(B)** The six weighted superpositions of 1BOO⁴³ to 1EG2⁴⁴, based on the same sequence alignments, are indistinguishable (average difference is 0.140 Å).



The weighted technique performs well for the homolog pairs tested. In fact, only two cases resulted in a set of weighted overlays that had an average difference over 1 Å: Iron/Ascorbate Oxidoreductase was 1.185 Å and PHBH-like proteins was 1.231 Å. However, the average differences for the same systems were much larger with the standard overlays, 5.069 Å and 4.744 Å, respectively.

Of course, there may be situations where it is difficult to obtain an appropriate superposition with the weighted fitting. One instance may occur when a protein is large and has multiple domains. If the initial sequence alignment from each sequence comparison program focuses the best agreement on different domains rather than the entire protein structure, then the weighted superpositions may not converge to the same solution. Another case is when there is too little sequence or structural similarity, but this is when most comparison methods breakdown. For the test cases employed in this study, the sequence alignment tools broke down at ~16% ID, returning sporadic aligned segments that were too short and too infrequent. Homologs with so little sequence

similarity are notoriously difficult, but it may be possible in some cases to compare them using methods based on structural information such as geometric comparisons of folds.³⁴ These techniques would be most successful when there is little structural variation or flexibility.

Identifying Sequence Misassignments

The residue pairings in regions with good structural agreement will be heavily weighted in the wRMSD calculation. As a result, we can use poor weights to determine potential misassignments in the sequence alignment. These are “potential misassignments” because the sequence pair may not be misaligned but could be located in a flexible region of the protein. Regions that have been brought into close spatial proximity, but have a low weighting, indicate potentially incorrect pairings of residues in the sequences. This concept is demonstrated in Figure 3.9 using two homologs from the SpoU rRNA methylase family^{45,46} with 26% ID.

Figure 3.9. SpoU rRNA methylase family (26% ID). (A) BLAST sequence alignment of 1IPA⁴⁵ and 1GZ0⁴⁶ using default parameters. Colons represent sequence identities, and gaps are shown with dashes. The underlined region notes domain 1, and the blue boxes represent misaligned residues corresponding to the labeled α -helix and β -sheet in B. Atom pairs with a weighting of 40% or greater in the wRMSD calculation are noted with asterisks. Standard (B) and weighted (C) superpositions of 1IPA⁴⁵ (thick, colored line) onto 1GZ0⁴⁶ (thin, gray line). In C, the color coding by weight is the same as in Figure 3.7.

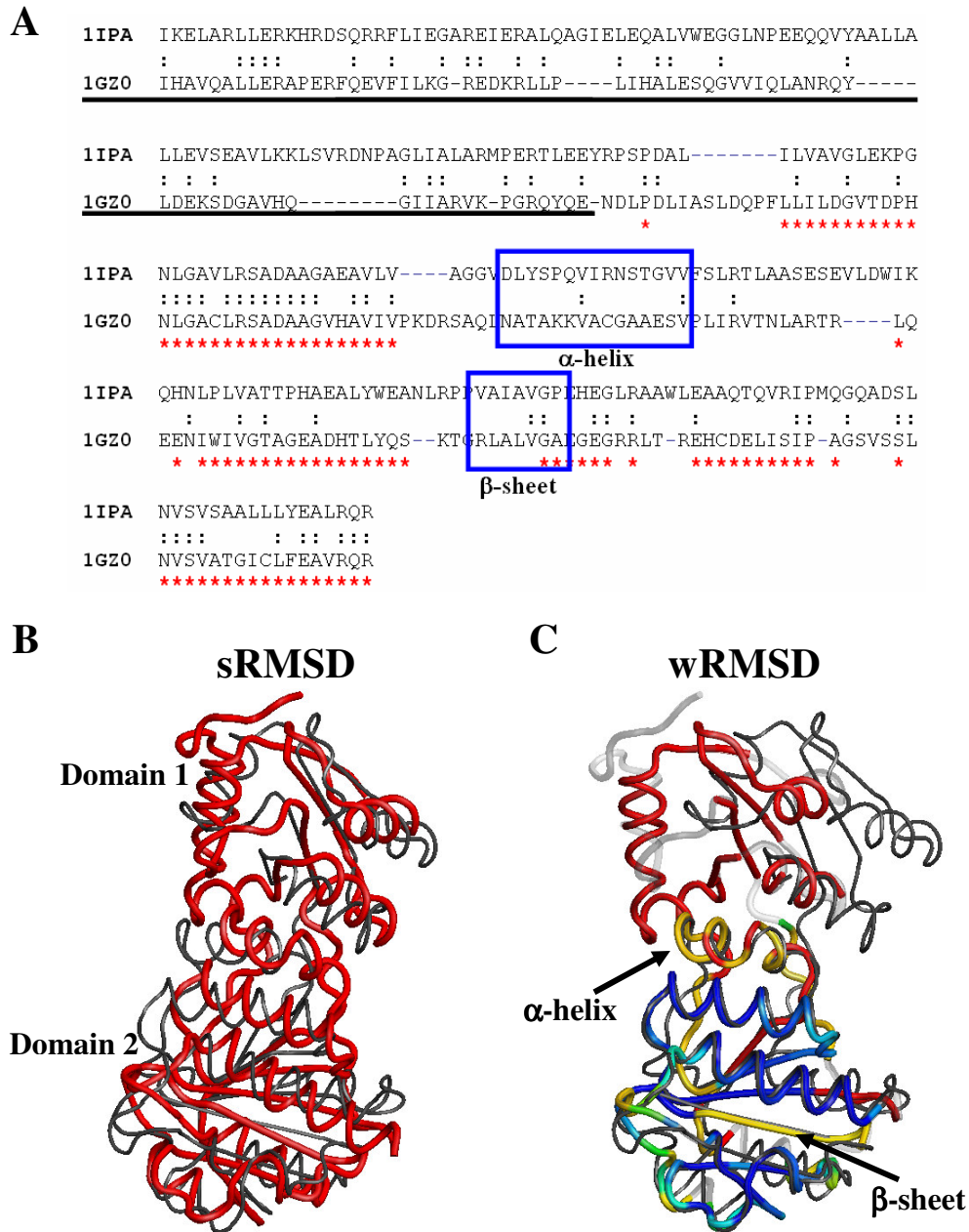


Figure 3.9A shows the initial BLAST sequence alignment using default parameters, and the resulting standard and weighted superpositions are provided in Figure 4B,C. Any sequence-aligned residues that received a weight of 40% or greater from the

wRMSD calculation is noted with a red asterisk. The underlined region of the sequence alignment in Figure 3.9A corresponds to domain 1, a flexible region between the proteins; as would be expected, none of those residues were significantly weighted to contribute to the superposition.

The blue boxes in Figure 3.9A indicate regions of incorrect atom pairing. The first is due to an erroneous gap placement (before and after the blue box) and corresponds to the residues of the denoted α -helix in Figure 3.9C. In the standard superposition, the residues of the α -helix were not aligned properly, and the appropriate atoms were not paired together. However, after the weighted overlay, they are brought into close spatial proximity. This structural information can then be used to correct the initial atom pairing. It is interesting to note that when we created an appropriate gap placement to correct the atom pairing, a large percentage of the residues within and adjacent to the α -helix have a weight of 40% or greater in the weighted superposition (data not shown). The β -sheet noted in Figure 3.9C is also a misalignment in the sequences that is overcome by the wRMSD superposition. This instance is caused by a feature in the Biopython parser⁴⁷ used to pull information from the PDB files. Parsers ignore non-standard amino acids, and in the 1GZ0 structure, the methionines have been replaced with selenomethionine to aid in obtaining the structure. This results in missing residues in the sequence which are difficult to overcome in the sequence alignment because of gap penalties. If we introduce selenomethionine into the sequence, the superposition remains the same, but the weights properly reflect the agreement in the structures (see Appendix 2).

3.4 Conclusions

We have shown how the local wRMSD technique from Chapter 2 can be used to evaluate protein structure predictions through an overlay with the experimentally

determined target. The agreement with the standard GDT-TS metric is very good for most targets, with more variability in the rankings as the target becomes more difficult. The overlays provided by wRMSD are compelling for Comparative Modeling and Fold Recognition targets. Comparing predictions to New Fold targets and more difficult Fold Recognition targets can provide more than one solution, highlighting cases where local, secondary, and tertiary structure is properly assigned but misoriented relative to one another. The %wSUM-ALL metric appears to be a good measure of global accuracy of a difficult target when the scaling factor is larger ($\sim 50 \text{ \AA}^2$), and it is not a simple reproduction of the GDT-TS metric. By varying the scaling factor and examining the multiple solutions, the user can evaluate predictions for both local and global accuracy.

Furthermore, we have now coupled our wRMSD method with a BLAST sequence alignment. Our method is capable of preferentially selecting out the regions with the best structural agreement between homologous proteins and generating a superposition that can identify significant similarities and differences. This technique can be used to superimpose homologs with low sequence identity and large conformational differences, an area where both sequence-based and structure-based methods may fail.

Based on homologs in the range of intermediate to low sequence identity, we have shown that applying a weighting term can overcome the dependence of a structural superposition on the initial sequence alignment used to determine the appropriate C α pairs. The wRMSD superpositions are not significantly affected by the choice of the sequence alignment method or the employed parameters, but the standard RMSD fits are highly dependent on both. The conserved regions of the structures are heavily weighted, thus errors made in the initial sequence alignment are relatively discounted. Moreover, the calculated weights can be used to determine potential misassignments in the initial sequence alignments. The wRMSD technique does not require prior knowledge of any protein system, and it removes the need to determine the best alignment method or

parameters for each application. However, we must note that our tool, like any other, will breakdown when sequence or structural similarity is too low.

This work has been published as:

Damm, K.L. and Carlson, H.A. Gaussian-Weighted RMSD Superposition of Proteins: A Structural Comparison for Flexible Proteins and Predicted Protein Structures. *Biophys. J.* **2006**, *90*, 4558-4573.

Damm, K.L. and Carlson, H.A. Overcoming Sequence Misalignments with Weighted Structural Superpositions. *Submitted to Bioinformatics*.

3.5 References

1. Damm, K. L.; Carlson, H. A. Gaussian-weighted RMSD superposition of proteins: a structural comparison for flexible proteins and predicted protein structures. *Biophys J* **2006**, *90*, 4558-4573.
2. Hubbard, T. J. RMS/coverage graphs: a qualitative method for comparing three-dimensional protein structure predictions. *Proteins* **1999**, *Suppl 3*, 15-21.
3. Chothia, C.; Lesk, A. M. The relation between the divergence of sequence and structure in proteins. *Embo J* **1986**, *5*, 823-826.
4. Holm, L.; Sander, C. Mapping the protein universe. *Science* **1996**, *273*, 595-603.
5. Watson, J. D.; Laskowski, R. A.; Thornton, J. M. Predicting protein function from sequence and structural data. *Curr Opin Struct Biol* **2005**, *15*, 275-284.
6. Marsden, R. L.; Ranea, J. A.; Sillero, A.; Redfern, O.; Yeats, C. et al. Exploiting protein structure data to explore the evolution of protein function and biological complexity. *Philos Trans R Soc Lond B Biol Sci* **2006**, *361*, 425-440.
7. Murzin, A. G.; Brenner, S. E.; Hubbard, T.; Chothia, C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* **1995**, *247*, 536-540.
8. Orengo, C. A.; Michie, A. D.; Jones, S.; Jones, D. T.; Swindells, M. B. et al. CATH - a hierarchic classification of protein domain structures. *Structure* **1997**, *5*, 1093-1108.
9. Holm, L.; Sander, C. The FSSP database of structurally aligned protein fold families. *Nucleic Acids Res* **1994**, *22*, 3600-3609.
10. Sowdhamini, R.; Rufino, S. D.; Blundell, T. L. A database of globular protein structural domains: clustering of representative family members into similar folds. *Fold Des* **1996**, *1*, 209-220.
11. Hogue, C. W.; Ohkawa, H.; Bryant, S. H. A dynamic look at structures: WWW-Entrez and the Molecular Modeling Database. *Trends Biochem Sci* **1996**, *21*, 226-229.
12. Brenner, S. E.; Koehl, P.; Levitt, M. The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Res* **2000**, *28*, 254-256.
13. Mizuguchi, K.; Deane, C. M.; Blundell, T. L.; Overington, J. P. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci* **1998**, *7*, 2469-2471.

14. Sali, A.; Overington, J. P. Derivation of rules for comparative protein modeling from a database of protein structure alignments. *Protein Sci* **1994**, *3*, 1582-1596.
15. Schmidt, R.; Gerstein, M.; Altman, R. B. LPFC: an Internet library of protein family core structures. *Protein Sci* **1997**, *6*, 246-248.
16. Orengo, C. A.; Thornton, J. M. Protein families and their evolution-a structural perspective. *Annu Rev Biochem* **2005**, *74*, 867-900.
17. Moult, J.; Fidelis, K.; Zemla, A.; Hubbard, T. Critical assessment of methods of protein structure prediction (CASP)-round V. *Proteins* **2003**, *53 Suppl 6*, 334-339.
18. Teplyakov, A.; Obmolova, G.; Khil, P. P.; Howard, A. J.; Camerini-Otero, R. D. et al. Crystal structure of the Escherichia coli YcdX protein reveals a trinuclear zinc active site. *Proteins* **2003**, *51*, 315-318.
19. Yamashita, A.; Maeda, K.; Maeda, Y. Crystal structure of CapZ: structural basis for actin filament barbed end capping. *Embo J* **2003**, *22*, 1529-1538.
20. Allen, M.; Friedler, A.; Schon, O.; Bycroft, M. The structure of an FF domain from human HYPA/FBP11. *J Mol Biol* **2002**, *323*, 411-416.
21. Miller, D. J.; Ouellette, N.; Evdokimova, E.; Savchenko, A.; Edwards, A. et al. Crystal complexes of a predicted S-adenosylmethionine-dependent methyltransferase reveal a typical AdoMet binding domain and a substrate recognition domain. *Protein Sci* **2003**, *12*, 1432-1442.
22. Tan, A.Y., P. C. Smith, J. Shen, R. Xiao, T. Acton, B. Rost, G. Montelione, and J. F. Hunt. Crystal structure of spermidine synthase. To be published.
23. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N. et al. The Protein Data Bank. *Nucleic Acids Res* **2000**, *28*, 235-242.
24. Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. Basic local alignment search tool. In *J Mol Biol*, 1990; pp 403-410.
25. Tatusova, T. A.; Madden, T. L. BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences. *FEMS Microbiol Lett* **1999**, *174*, 247-250.
26. Tramontano, A.; Morea, V. Assessment of homology-based predictions in CASP5. *Proteins* **2003**, *53 Suppl 6*, 352-368.
27. Kinch, L. N.; Wrabl, J. O.; Krishna, S. S.; Majumdar, I.; Sadreyev, R. I. et al. CASP5 assessment of fold recognition target predictions. *Proteins* **2003**, *53 Suppl 6*, 395-409.

28. Aloy, P.; Stark, A.; Hadley, C.; Russell, R. B. Predictions without templates: new folds, secondary structure, and contacts in CASP5. *Proteins* **2003**, *53 Suppl 6*, 436-456.
29. Zemla, A. LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Res* **2003**, *31*, 3370-3374.
30. Xu, Z.; Horwich, A. L.; Sigler, P. B. The crystal structure of the asymmetric GroEL-GroES-(ADP)₇ chaperonin complex. *Nature* **1997**, *388*, 741-750.
31. Braig, K.; Adams, P. D.; Brunger, A. T. Conformational variability in the refined structure of the chaperonin GroEL at 2.8 Å resolution. *Nat Struct Biol* **1995**, *2*, 1083-1094.
32. Ditzel, L.; Lowe, J.; Stock, D.; Stetter, K. O.; Huber, H. et al. Crystal structure of the thermosome, the archaeal chaperonin and homolog of CCT. *Cell* **1998**, *93*, 125-138.
33. Rost, B. Twilight zone of protein sequence alignments. *Protein Eng.* **1999**, *12*, 85-94.
34. Elofsson, A. A study on protein sequence alignment quality. *Proteins* **2002**, *46*, 330-339.
35. Van Walle, I.; Lasters, I.; Wyns, L. SABmark--a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics* **2005**, *21*, 1267-1268.
36. Wallner, B.; Elofsson, A. All are not equal: a benchmark of different homology modeling programs. *Protein Sci* **2005**, *14*, 1315-1327.
37. Thompson, J. D.; Higgins, D. G.; Gibson, T. J. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* **1994**, *22*, 4673-4680.
38. Sauder, J. M.; Arthur, J. W.; Dunbrack, R. L., Jr. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins* **2000**, *40*, 6-22.
39. Huang, X.; Miller, W. A time-efficient, linear-space local similarity algorithm. *Adv Appl Math* **1991**, *12*, 337-357.
40. Pearson, W. R.; Lipman, D. J. Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A* **1988**, *85*, 2444-2448.
41. Cohen, G. H. ALIGN: A program to superimpose protein coordinates, accounting for insertions and deletions. *J Appl Cryst* **1997**, *30*, 1160-1161.

42. Notredame, C.; Higgins, D. G.; Heringa, J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol* **2000**, *302*, 205-217.
43. Gong, W.; O'Gara, M.; Blumenthal, R. M.; Cheng, X. Structure of pvu II DNA-(cytosine N4) methyltransferase, an example of domain permutation and protein fold assignment. *Nucleic Acids Res* **1997**, *25*, 2702-2715.
44. Scavetta, R. D.; Thomas, C. B.; Walsh, M. A.; Szegedi, S.; Joachimiak, A. et al. Structure of RsrI methyltransferase, a member of the N6-adenine beta class of DNA methyltransferases. *Nucleic Acids Res* **2000**, *28*, 3950-3961.
45. Nureki, O.; Shirouzu, M.; Hashimoto, K.; Ishitani, R.; Terada, T. et al. An enzyme with a deep trefoil knot for the active-site architecture. *Acta Crystallogr D Biol Crystallogr* **2002**, *58*, 1129-1137.
46. Michel, G.; Sauve, V.; Larocque, R.; Li, Y.; Matte, A. et al. The structure of the RlmB 23S rRNA methyltransferase reveals a new methyltransferase fold with a unique knot. *Structure (Camb)* **2002**, *10*, 1303-1315.
47. Biopython 1.42 parser. (2006) <http://biopython.org/>.

CHAPTER 4

Exploring Experimental Sources of Multiple Protein Conformations in Structure-Based Drug Design

4.1 Introduction

A difficulty that arises when working with multiple structures in SBDD is deciding which receptor conformations are the most appropriate to use. A further issue is the source of the structures; are structures generated from MD simulations or solved using NMR or x-ray crystallography more suitable? Previously, NMR structures have been shown to sample more conformational space than MD simulations and account for additional protein flexibility.¹ However, multiple groups have demonstrated that dynamics simulations provide complete sampling of the multiple flap conformations of HIV-1p.^{2,3}

Crystal structures are thought to provide a more accurate depiction of a protein despite the fact that NMR structures are solved in a more biologically relevant environment.^{4,5,6} This may be due to the fact that X-ray crystallography generally provides a greater amount of high quality experimental data than NMR spectroscopy, which can be assessed using standard quality control measurements. Good agreement is usually seen in the protein backbones of crystal structures versus NMR structures, while the conformational sampling is focused on loop regions and side chains.^{4,7} Two independent groups found that crystal and NMR structures often provide complementary structural information and should be used in conjunction with one another as methods to solve protein structures.^{8,9} It is also known that the choice of protein structure

can heavily influence the outcome of a simulation; different conformations do perform better than others in virtual screening applications.^{10,11} A recent review summarizes the use of crystal structures in SBDD and discusses the associated limitations.¹²

Crystal structures provide only a static snapshot of the dynamic structure of a protein, and bound structures can lead to the “cross-docking” problem. The binding site is already predefined for the co-crystallized ligand and may not fit other conformationally diverse structures. Nonetheless, using a collection of crystal structures bound to a variety of ligand classes offers an ensemble of conformations and can elucidate structural changes that occur upon ligand binding.¹³ Limitations exist such that most systems rarely have a large number of crystal structures solved in complex with many diverse ligands. Also, crystal structure conformations can be influenced by crystallization conditions such as crystal packing effects, pH, buffers, and temperature and may not be a fully correct representation of the structure in solution. Finally, flexible regions may be ill-defined due to a lack of electron density.

Conversely, using NMR spectroscopy as a method of three-dimensional structure determination provides an ensemble of conformations found in solution. The ensemble is comprised of low energy structures that satisfy acceptance criteria based on the experimental data. Each conformation alone can be thought of as a static snapshot; however, they provide a dynamic representation of the protein when used as a collection. As with crystallography, experimental conditions may influence the determined conformations. Also, the structural variability may not be a result of true motion in the protein but rather due to insufficient experimental data.¹⁴

In the literature, almost all studies use crystal structures in structure-based drug design, both collections and single, static conformations. There are a few occurrences where NMR ensembles are also employed. For example, Knegt et al. used NMR ensembles to examine ras p21 and uteroglobin.¹⁵ Additionally, Huang and Zou found that ensemble docking to NMR structures of HIV-1p resulted in the identification of more

known inhibitors than docking to single, static crystal structures (91% versus 66%, respectively).¹⁶ Furthermore, it is common to utilize information from NMR such as NOE-derived distance constraints and torsion angle constraints to aid in both protein-protein and ligand-protein docking.^{17,18,19} Fragment-based screening through NMR or “SAR by NMR” has also been widely used in drug discovery for the past 10 years.^{20,21,22} However, no one has compared the use of NMR structures to collections of crystal structures.

We are now interested in expanding our MPS technique to incorporate experimental structures from either an NMR ensemble or a collection of crystal structures. There are few examples where a diverse set of experimentally determined structures is available, but one such case is HIV-1p; structures are available from both NMR and X-ray crystallography.

We focus on comparing the use of two protein collections in our MPS method, an NMR ensemble of HIV-1p with a bound cyclic urea (cu) inhibitor and multiple unique crystal structures with cu inhibitors. The location and chemical characteristics of the pharmacophore elements are consistent between the models; however, additional elements exist in the cu-crystal model. Interestingly, even when the protease is in a bound conformation, the features of our previous model generated from apo HIV-1p²³ are still reproduced. In an effort to incorporate the most structural data, we also create a model from 90 crystal structures of susceptible HIV-1p. We show that the structural variation between the collection is very small, resulting in a similar model to the cu-crystal model. We are also able to show that models generated from protein ensembles are more successful at discriminating between known HIV-1p inhibitors and inactive drug-like molecules than are models from a single “average” structure. Erickson et al. have shown that “average” structures of HIV-1p also perform poorly when docking a ligand into its binding site with a successful docking rate of only 32.5%.²⁴ To our knowledge, this is the

first time a direct comparison of NMR ensembles and crystal collections was made using the same protein in structure-based drug design.

4.2 Computational Methods

Protein Preparation

A cu-bound NMR structure (PDB ID: 1BVE)²⁵ comprised of 28 distinct models was downloaded from the PDB²⁶ along with the restrained minimized average NMR structure (PDB ID: 1BVG)²⁵. The Binding MOAD database²⁷ was used to obtain 174 bound crystal structures, all having a resolution of ≤ 2.5 Å. Any structure with a mutation known to confer resistance or known to alter the biological activity of the protein (i.e. A25N) was discarded, resulting in a collection of only drug-susceptible, active strains. Because of the ambiguity in the data, any structure with residues in multiple orientations in the active site region (defined as any residue within 10 Å of the active site center) was removed. Structure 1AID was also discarded from this study as an outlier due to the unusual conformation of the flap region²⁸, resulting in a final set of 90 structures. Of the 90 structures, 10 are drug-susceptible, active strains bound to unique cu ligands and were used as a collection to provide a direct comparison to the NMR ensemble: 1AJX²⁹, 1DMP³⁰, 1HVR³¹, 1HWR³², 1PRO³³, 1QBR³⁴, 1QBS³⁵, 1QBT³⁴, 1QBU³⁴, and 1T7K³⁶. The structures of the ten cu ligands and inhibition constants are provided in Appendix 3 along with the PDB IDs and corresponding references of the entire crystal collection.

All NMR and crystal structures were first prepared by using MolProbity³⁷ to check the side-chain orientations, and histidine tautomers were checked by hand. Next, ligands and solvent ions were removed from each structure. Any hydrogen atom was stripped from the crystal structures then added with xleap in the AMBER6³⁸ suite and minimized to convergence with 10,000 steps of conjugate gradient energy minimization using Sander Classic. This ensured uniform setup of all structures, whether from NMR or crystallographic sources.

MUSIC Simulation

The active site of each NMR and crystal structure was flooded with 500 small molecule probes using a 12-Å radius sphere to define the initial placement. Benzene, ethane, and methanol were utilized as the probes. The sphere was centered at the midpoint of the active site to ensure complete random sampling throughout the entire binding cavity. Each structure was then used in a multi-unit search for interacting conformers (MUSIC) simulation with the BOSS (Biochemical and Organic Simulation System) program³⁹, using the OPLS force field⁴⁰ and holding the protein atoms fixed. The small molecule probes were minimized via a low-temperature Monte Carlo sampling, revealing energetically favorable regions of the active site surface for each chemical functionality. Benzene probes elucidate aromatic and hydrophobic interactions, ethane probes clarify general hydrophobic interactions from aromatic, and methanol probes demonstrate hydrogen-bond donating and accepting sites. Probes do not interact with other probes, but the full interaction energy is calculated with the protein atoms. Further details describing the MUSIC simulation have been previously published.⁴¹

Pharmacophore Elements

Each structure was then examined to determine clusters, regions where multiple probes had minimized to the same location on the protein surface. This was done both manually and using an auto-clustering method based on our in-house Jarvis-Patrick codes. Any cluster within 9.5 Å of the catalytic aspartic acid residues 25 and 25' was investigated, and if 8 probes were present, the cluster was represented by its "parent", the lowest-energy probe calculated in the MUSIC simulation.

An average structure was calculated for each protein set: the NMR ensemble, all-crystal collection (90 structures), and cu-crystal collection (10 structures). Each set of structures was superimposed to a reference protein, the structure in the ensemble with the smallest RMSD to the calculated average structure, using a Gaussian-weighted RMSD

(wRMSD) alignment⁴², setting the scaling factor equal to 2 \AA^2 . The overlaid sets were then used to determine “cluster of clusters” or consensus clusters of the probe molecules. A consensus cluster is defined by having parent probes from $\geq 50\%$ of the protein conformations. For example, the NMR ensemble contains 28 structures; hence, there must be 14 parents in close proximity to be a consensus cluster. Only probes found in energetically favorable regions, conserved throughout the ensemble, will remain as a consensus cluster. Thus, protein flexibility is implicitly accounted for by focusing chemical requirements on the rigid, unforgiving regions of the binding site and allowing chemical and steric flexibility in the mobile regions.

The consensus clusters were then represented as spherical pharmacophore elements. The center of each pharmacophore element was defined by the average position of the benzene centroid, the midpoint of the carbon-carbon bond for ethane, and the oxygen atom of the methanol probe, while the radius was based on the RMSD of the probe positions. Overlapping benzene and ethane clusters were combined and termed aromatic/hydrophobic elements. Individual benzene elements were labeled aromatic whereas extraneous ethane clusters were removed. Methanol elements were classified as a hydrogen-bond donor, acceptor, or doneptor (donor and acceptor). Two excluded volumes were defined by the average position of the C γ of each catalytic aspartic acid residue and used to represent the bottom of the active site. The radii of the excluded volumes were set to 1.5 \AA , the approximate length of a C γ – O δ bond. A more detailed description of the MPS method can be found elsewhere.⁴³

Pharmacophore Model Evaluation

The resulting pharmacophore models were screened against databases of compounds with pre-generated multiple conformers (maximum number of conformations was 300) using the search option within the Pharmacophore Query Editor of Molecular Operating Environment (MOE)⁴⁴. This is simply a fit/no-fit comparison based on the

geometry of each conformer's chemical features and the physical arrangement of the pharmacophore elements. It is not a docking calculation based on scoring functions.

Three previously created databases of compounds were used. The first database consists of 89 diverse known HIV-1p inhibitors taken from the PDB and the literature while two databases of non-inhibitors from the Comprehensive Medicinal Chemistry Index^{45,46} were used as decoys. The first non-inhibitor database is comprised of 85 ligands⁴³ identified by filtering based on size and chemistry comparable to that of known protease inhibitors, whereas the second is more general and contains 2322 drug-like ligands of very diverse sizes and chemical characteristics.⁶⁷ The full set created for use in our previous work contained 2324 compounds, but for this work it was appropriate to remove the two known HIV-1p inhibitors. The preparation and composition of these data sets has been described previously.^{43,47} The stringency of the pharmacophore model was examined by varying the required number of pharmacophore elements that must be matched by enabling the partial match option in the Pharmacophore Query Editor of MOE and also by varying the radii of the elements.

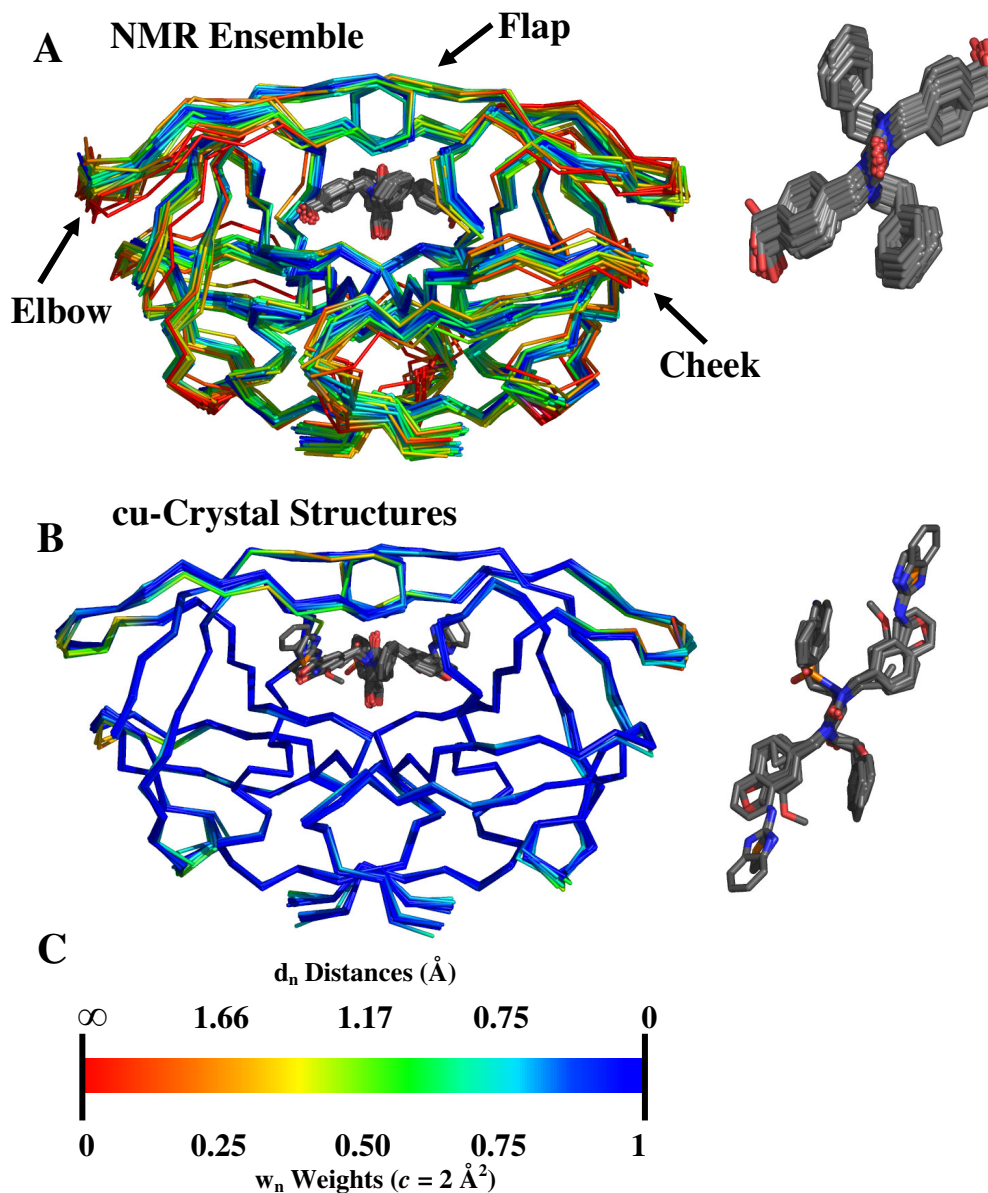
The performance of the models was evaluated by comparing the percentage of identified known inhibitors (true positives) versus the percentage of drug-like non-inhibitors identified (false positives). The database screening results are presented as receiver operator characteristic (ROC) curves, where the optimal model would lie at the (0,100) point predicting 100% of true positives and 0% of false positives. The models were also qualitatively compared back to the cu ligands.

4.3 Results and Discussion

Structural Comparison of the Protein Conformations

A sequence comparison was made of the 10 cu-crystal structure sequences and that of the NMR ensemble. The sequences differ at three amino acid positions: 3, 37, and 95. However, none of the mutations confer resistance or alter the biological activity of HIV-1p. The 28 NMR models and 10 cu-crystal structures were compared by aligning their C α coordinates to their respective average structure using a Gaussian-weighted alignment⁴². The superposition of the 28 NMR models is provided in Figure 4.1 along with the bound cu ligands, and the 10 crystal structures and their unique cu ligands. The majority of the variation between the NMR backbones is in the “elbows” of the flaps and in the “cheek” region, while the active site appears quite rigid. A detailed analysis of the NMR ensemble is provided by Yamazaki et al.²⁵ The backbones of the cu-crystal structures show much less deviation.

Figure 4.1. (A) Gaussian-weighted overlay of 28 models in NMR ensemble along with all cu ligands (front view). The corresponding cu ligands are also shown using a top view for clarity. The regions of the protein with high backbone deviations are highlighted with an arrow. (B) Gaussian-weighted overlay of 10 crystal structures bound to unique cu ligands (front view). A top view of the 10 ligands is also shown. (C) The scale shows how smaller deviations (blue) are more heavily weighted in the wRMSD fit, $c = 2 \text{ \AA}^2$. Deviations over 2.45 \AA have weights under 5% (red).



The RMSD was calculated between each wRMSD aligned structure and its reference structure. For the NMR ensemble, the $C\alpha$ RMSD ranges from $0.65 - 1.71 \text{ \AA}$ with the average being 0.92 \AA . The cu-crystal collection had much lower RMSD values and also a smaller range, $0.26 - 0.80 \text{ \AA}$ and an average of 0.43 \AA . (RMSD values

calculated from a wRMSD alignment are higher than those calculated from a standard RMSD alignment.⁴² This is because a wRMSD alignment sacrifices the fit in the flexible regions to better align the rigid core.) The point here is not to compare the literal RMSD values per se but rather to evaluate the range of the values illustrating the conformation variation between the NMR ensemble and cu-crystal collection. This analysis demonstrates that there is a greater variation between the C α coordinates of the NMR ensemble than the cu-crystal collection. Other groups have also found that the variation between the active sites of different crystal structures is usually small, 0.3 - 0.8 Å RMSD.^{28,48}

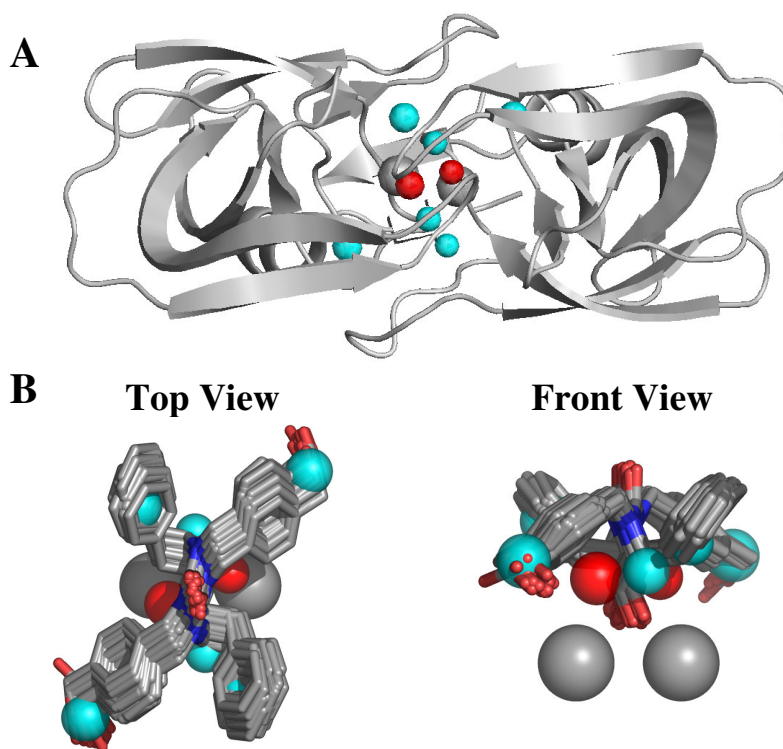
Across all NMR and crystal conformations, the cu ligands are in relatively the same conformation and position in the active site of the protease, the urea oxygen accepting a hydrogen-bond from the protease flaps and the diols off the 7-membered ring donating hydrogen bonds to the 25/25' aspartic acids. The position of the urea oxygen shows more spread across the 28 NMR structures than within the crystal structures. The side chains of the cu inhibitors occupy their complementary S1/S1' and S2/S2' substrate recognition sites. The cu ligands bound in crystal structures 1QBR³⁴, 1QBT³⁴, and 1QBU³⁴ have larger side chains and also hydrogen bond with the flap residue Gly 48/48'.

Pharmacophore Model Comparison

The NMR pharmacophore model maintains the C₂ symmetry of the protease and has 8 sites: 2 hydrogen-bond donor elements near the catalytic aspartic acid residues 25/25', 2 aromatic/hydrophobic elements that anchor the hydrophobic regions near the active site center, and 4 aromatic/hydrophobic elements that occupy the S1/S1' and S2/S2' pockets of the active-site. The chemical characteristics of the NMR pharmacophore elements differ slightly from the chemical features of the bound cu ligand. The hydrogen-bond donor elements are slightly displaced from the location of the hydroxyl groups that extend below the seven-membered ring. The two interior

aromatic/hydrophobic elements are located between the cu ligand side chains and represent the hydrophobic features of the central scaffold, while the four exterior aromatic/hydrophobic elements complement the chemical features of the ligand side chains. The MPS model based on the NMR structures is shown in Figure 4.2 in relation to the HIV-1p structure and also superimposed with the 28 cu ligands.

Figure 4.2. (A) Pharmacophore model (radii of $1 \times \text{RMSD}$) generated using 28 NMR structures. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; cyan, aromatic/hydrophobic. Top view of the protease backbone is shown in grey, as are the excluded volumes. (B) Pharmacophore model superimposed with 28 cu-ligands colored in grey. Both top and front views are shown.

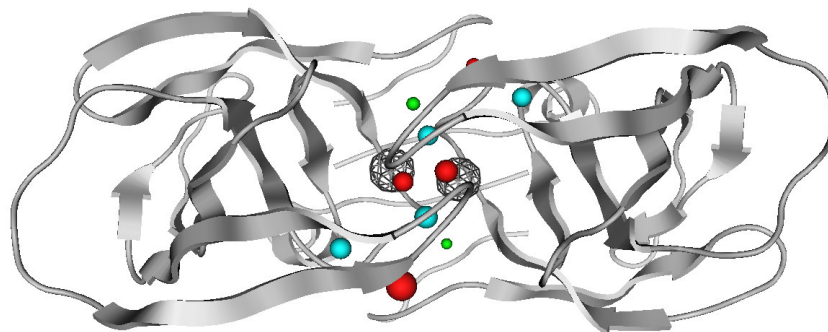


In order to assess how the models are improved through the use of MPS, we also generated a “static” model from a single structure. The pharmacophore element centers were defined in the same manner as previously described; however, the static model is based on the probes docked into one structure, rather than the parent probes across MPS. The pharmacophore model generated from the average NMR structure maintained the features of the model created from the NMR ensemble, but the radii are much smaller for

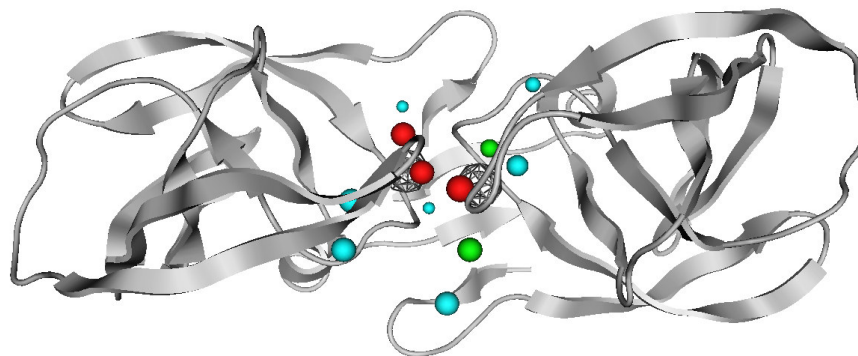
most elements, as demonstrated in Figure 4.3. This is most likely because without the consensus clustering step, the positions of the probes comes from one structure only, even if it is an average. The spread between the probes docked within one structure is usually much smaller than the spread between the parent probes across many conformations. There are also two additional donor sites occupying the S3/S3' subsites. Though the model is inferior to MPS models, it does show improvement over our earlier static model based on an apo crystal structure⁴³.

Figure 4.3. The average NMR model is compared to a previously created a pharmacophore model from a static crystal structure⁴³. It is notable that the model from the average NMR structure, while having additional sites compared to the MPS NMR model, was still reasonable unlike the model from the static crystal structure. The static crystal structure model has many additional elements and is not appropriate for virtual screening applications. **(A)** Pharmacophore model (radii of $1 \times \text{RMSD}$) generated using the average NMR structure. **(B)** Pharmacophore model (radii of $1 \times \text{RMSD}$) generated using the static crystal structure 1HHP. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, aromatic/hydrophobic; green, aromatic. Top view of protease is shown; backbone is in grey.

A **Static Model from Average NMR Structure**

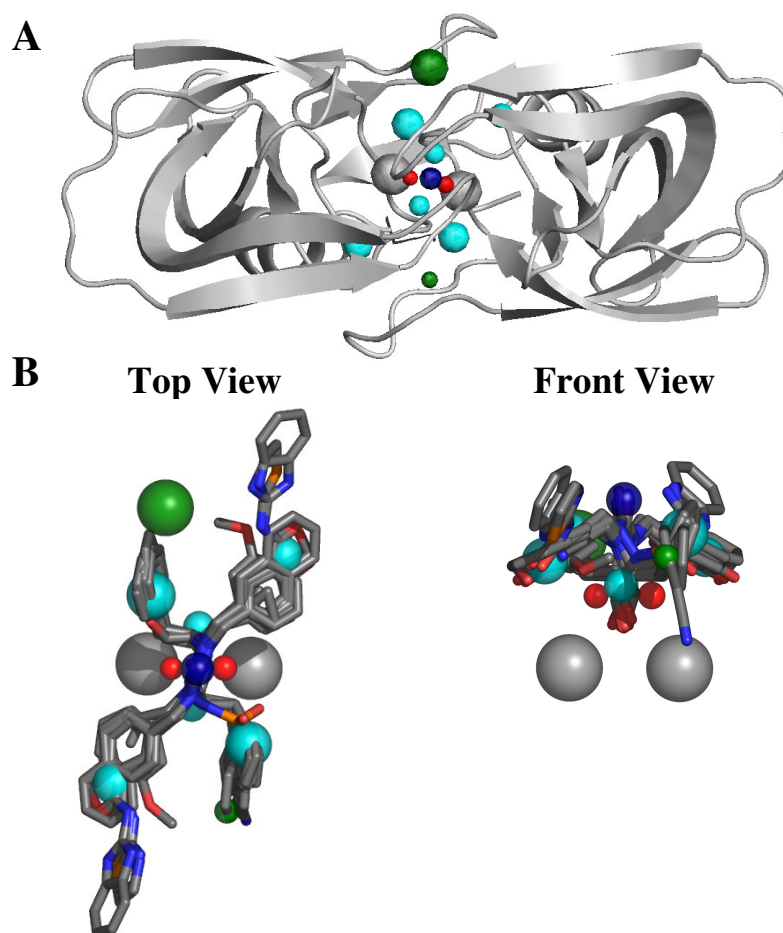


B **Static Model from 1HHP**



The cu-crystal pharmacophore model also maintains the C_2 symmetry of the protease, as shown in Figure 4.4. However, it contains 11 elements: a hydrogen-bond acceptor element near the tips of the protease flaps, 2 hydrogen-bond donor elements near the catalytic aspartic acid residues 25/25', 2 aromatic/hydrophobic elements near the core of the cu ligands, 4 aromatic/hydrophobic elements that occupy the S1/S1' and S2/S2' pockets of the active site, and 2 aromatic sites in the S3/S3' pockets. The crystal model overlaid with the 10 unique cu ligands is also provided (Figure 4.4).

Figure 4.4. (A) Pharmacophore model (radii of $1 \times \text{RMSD}$) generated using 10 cu-crystal structures. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, aromatic/hydrophobic; green, aromatic. Top view of the protease backbone is shown in grey as are the excluded volumes. (B) Pharmacophore model superimposed with 10 unique cu-ligands colored in grey. Both top and front views are shown.



The most interesting feature of this model is the hydrogen-bond acceptor element that perfectly overlays with urea oxygen of the 10 Cu ligands. The urea oxygen is known to displace a structural water molecule that coordinates substrates/inhibitors to the tips of the protease flaps. The structural water is a key difference between mammalian and HIV proteases, and this displacement may be one reason why Cu ligands are very selective for HIV proteases.^{34,49} Similar to the NMR models, the hydrogen-bond donor elements at the bottom of the pocket are slightly higher than the ligand diols. The 4 aromatic/hydrophobic elements complement the chemical features of some Cu ligands but do not agree with others. The additional 2 aromatic sites at the S3/S3' subsites fall at the edge of the aromatic rings in the Cu ligands.

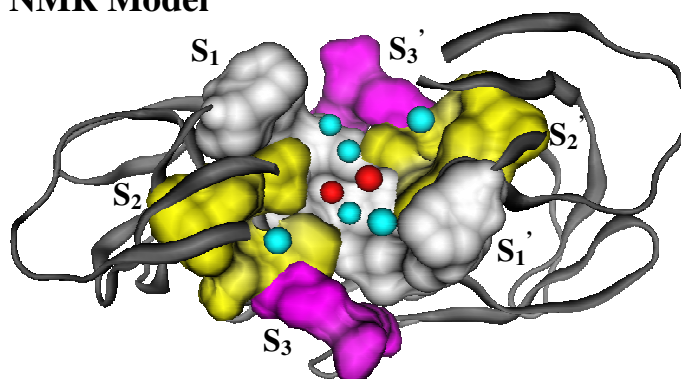
The most significant difference between the NMR and Cu-crystal models is the hydrogen-bond acceptor element in the crystal model. This site was not occupied by probes in any of the 28 NMR structures or the average structure. Yamazaki et al. state that the flap tips (residues 48-51) are dynamic in solution and exhibit motion on a nanosecond time scale whereas in crystal structures the flap tips are well-ordered.²⁵ The conformational variation across an NMR ensemble can be due to two things: protein dynamics or an under-resolved structure from lack of experimental data. The HIV-1p NMR structure solved by Yamazaki et al. is regarded as a high quality ensemble, and hence, the variation is thought to be from the dynamics of the structure.

There are also two additional aromatic sites in the Cu-crystal model that are not found in the NMR model. These elements are located in the S3/S3' subsites found at the solvent interface, a pocket known to accommodate broad substrate specificity.⁴⁹ The NMR and Cu-crystal models are shown compared to the substrate recognition motifs of the HIV-1p active site in Figure 4.5. In the NMR structures, the arginine 8/8' side chains are pushed out from the active site in variable locations. For this reason, there was more spread in the probes across the multiple conformations. The high flexibility of the arginine 8/8' side chains that is seen in the NMR structures was also observed in the

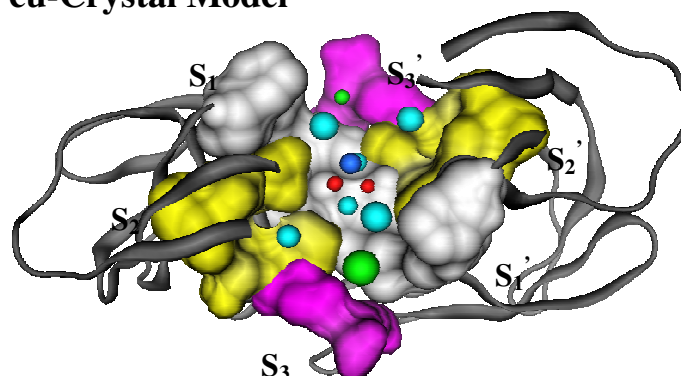
conformations sampled by MD simulations used to create previous pharmacophore models^{23,43}. In both the NMR and cu-crystal structures, additional hydrogen-bond doneptor (donor and acceptor) probes were observed between residues arginine 8/8' and aspartic acid 29/29' in the S3/S3' pockets. However, these doneptor sites fall outside the 9.5 Å cut-off; consequently they were not included in the pharmacophore models. A few of the larger inhibitors seen in the collection of 90 crystal structures have features that hydrogen bond to arginine 8/8' or aspartic acid 29/29'. Nonetheless, there are many smaller ligands that maintain an extremely high potency (nM-pM) without complementing this region, so it is appropriate that these sites were not included as an essential feature. Accordingly, ligands with the hydrogen bonding feature will be accepted by the model, but it will not be required for identification as a potential inhibitor of HIV-1p.

Figure 4.5. Comparison of known HIV-1p substrate recognition pockets with MPS pharmacophore models (radii of 1×RMSD): white, S1/S1' pocket; yellow, S2/S2' pocket; purple, S3/S3' pocket. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, aromatic/hydrophobic; green, aromatic. Flap residues 46/46' – 54/54' are removed for clarity. (A) NMR model. (B) cu-crystal structure model.

A NMR Model

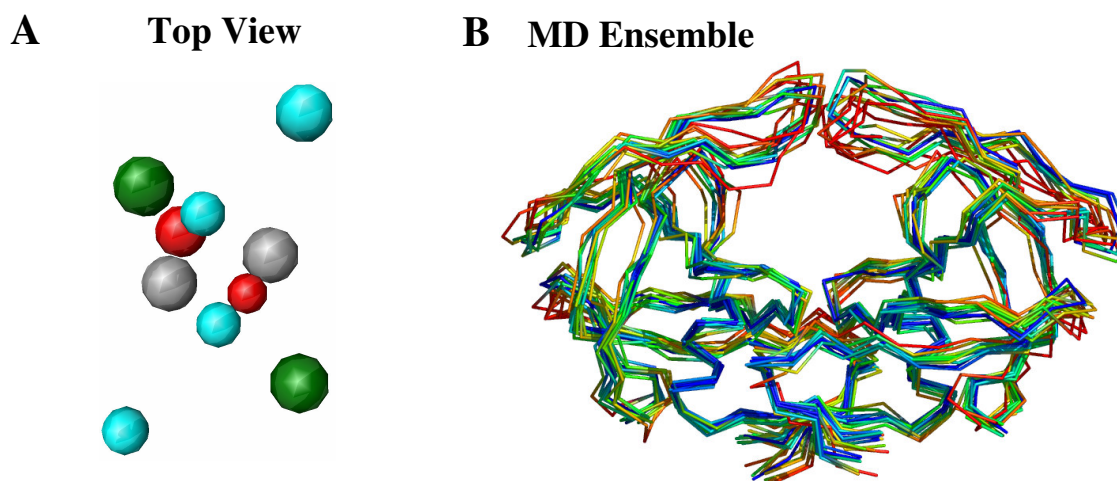


B cu-Crystal Model



Eight elements of the cu-crystal model were common to the NMR model: 2 hydrogen-bond donor elements and 6 aromatic/hydrophobic sites. The position and radii of these 8 elements are very similar with the exception that the radii of the hydrogen-bond donor elements are slightly smaller for the cu-crystal model. The location and chemical character of the 8-site NMR model is highly consistent with pharmacophore models generated from MD simulations of apo HIV-1p (apo-MD model). However, in the apo-MD model, several pharmacophore elements were aromatic, but all of the similar elements in the NMR model are aromatic/hydrophobic. The elements of the apo-MD model are also spread further apart due to the larger active-site cavity in the semi-open conformation than the bound form. The cu-crystal model clearly differs from the apo model by the additional hydrogen-bond acceptor and two aromatic sites. A representative apo-MD model based on data from our previous work²³ is provided in Figure 4.6.

Figure 4.6. (A) Top view of an MPS pharmacophore model (radii of $1 \times \text{RMSD}$) created using 11 structures generated from a 3-ns MD simulation of apo HIV-1p.²³ Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; cyan, aromatic/hydrophobic; green, aromatic. Excluded volumes are shown in grey. (B) Gaussian-weighted overlay of the 11 snapshots (front view). The color code of the weights is the same as in Figure 1C, and the view is comparable to Figure 1A and B.



Additionally, we observed that the range of C α RMSD for the apo-MD ensemble (11 structures) is similar to that of the NMR (28 structures): 0.94 – 1.50 Å versus 0.65 – 1.71 Å, respectively. The Gaussian-weighted superposition of the 11 structures from the

apo-MD ensemble in Figure 4.6 demonstrates the conformation variation observed across the 3-ns MD trajectory. The overlay of the MD ensemble clearly displays more movement in the bottom of the active site and, as one would expect due to the apo conformation, in the flap region than the NMR ensemble. It is interesting to find better agreement between bound HIV-1p conformations from NMR and the apo HIV-1p conformations from MD, rather than agreement to other bound conformations from X-ray crystallography.

Evaluation of Pharmacophore Models

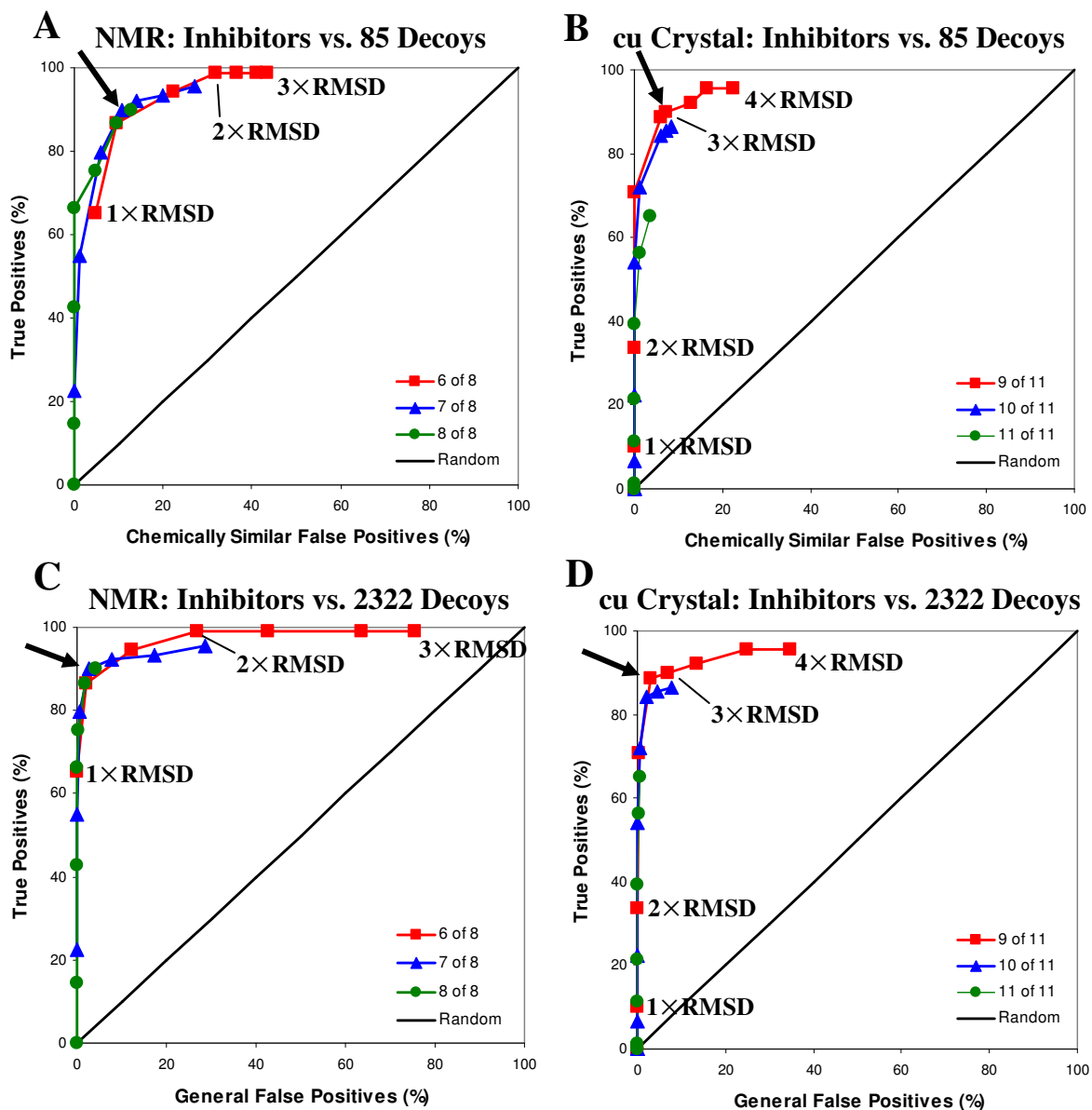
The NMR and cu-crystal models were screened against a database of known HIV-1p inhibitors and two decoy datasets using the search option within the Pharmacophore Query Editor in MOE⁴⁴, while varying the stringency of the search (i.e., enabling a partial match in the pharmacophore search and increasing radii size). Each ligand is described as a set of “annotation points” based on its chemistry and position in space. The ligand annotation points are then mapped to the pharmacophore elements and identified as a “hit” only if each of the required elements is satisfied. Hence, this is a binary fit or no-fit method, where all identified ligands are considered compatible with the pharmacophore model. The resulting data is presented as ROC curves. The best models identify the greatest number of true positives and the least number of false positives; consequently the optimal pharmacophore model is defined by having the smallest distance from (0,100). The raw data used to generate the ROC curves of the NMR and cu-crystal pharmacophore screens is available in Appendix 3.

The MPS models from NMR and cu-crystals were both very successful at discriminating known inhibitors versus a database of non-inhibitors with similar size and chemistry, as demonstrated in Figure 4.7. The optimal NMR model (7/8 sites, 2×RMSD) identifies 89.9% of the true positives and only 10.6% of the false positives. The optimal

cu-crystal model (9/11 sites, 3×RMSD) also identifies the same number of true positives, 89.9%, but hits less false positive than the NMR model, 7.1%.

However, these results may be misleading. The 11-site cu-crystal model shows the best performance when 9 of the 11 sites are required. This demonstrates that the 11 sites are too specific; less essential features of the active-site were selected out from using multiple cu-crystal structures, unlike with the NMR ensemble. If the three “extra” sites unique to the crystal model are dropped, the performance is nearly identical to the optimal 11-site model. The best, “core 8-site” cu-crystal model (8/8 sites, 3×RMSD) identifies 88.8% of the true positives and only 10.6% of the false positives. The extra sites do not significantly improve the performance of the model, which indicates that the hits from the cu-crystal model are really using the elements in common with the NMR model and apo-MD model. Extraneous sites which do not improve the performance of the models are problematic for database screening and undesirable for the MPS technique.

Figure 4.7. Receiver Operator Characteristic curves generated from screening a database of 89 known HIV-1p inhibitors against a set of 85 chemically similar known inactives and 2322 general decoy compounds. Each series represents a different stringency in the screen (i.e. 6 of 8 elements are required as a hit, 7 of 8 elements are required as a hit, etc.) Points in series are increasing radii values from 1× to 3×RMSD for the NMR model and 1× to 4× for the cu-crystal model. The radii are labeled on the 6 of 8 models based on NMR and the 9 of 11 models based on cu-crystals. The optimal pharmacophore models are highlighted by an arrow. (A) MPS NMR pharmacophore models, 89 known inhibitors vs. 85 decoy compounds (Optimal: 7/8, 2.0×RMSD). (B) MPS cu-crystal pharmacophore models, 89 known inhibitors vs. 85 decoy compounds (Optimal: 9/11, 3.0×RMSD). (C) MPS NMR pharmacophore models, 89 known inhibitors vs. 2322 general molecules (Optimal: 7/8, 2.0×RMSD). (D) MPS cu-crystal pharmacophore models, 89 known inhibitors vs. 2322 general molecules (Optimal: 9/11, 2.7×RMSD).



The optimal NMR performance is comparable to the optimal 11-site cu-crystal model but with fewer sites. Additionally the 8/8 site, 3×RMSD, NMR model performed

quite similarly to the optimal NMR model (7/8 sites, 2×RMSD); the number of true positives identified remained the same while only identifying 2 additional false compounds. Therefore, all of the sites in the NMR model appear to encode useful information. The 11/11 cu-crystal models demonstrate mediocre performance; the best model (4×RMSD) has a larger false positive hit rate and identifies only 65.2% of the true positives. The reduced amount of conformational sampling of the protein had to be overcome by significantly increasing the scaling factor for radii.

We use multiple structures to determine the most essential features that are conserved across different receptor conformations. Overall, the NMR model is more general, and the features do not simply reproduce the chemical characteristics of the bound cu ligand. In a recent study by our group using a different protein target, dihydrofolate reductase (DHFR), crystal structures were also employed as MPS and very minor conformational changes were observed between the collections.⁴⁷ The minimal conformational variation between the structures resulted in relatively small radii of the elements. Hence, the radii of the pharmacophore models had to be multiplied by 4× or 5×RMSD for optimal performance.

We anticipated that the use of a more general model will be beneficial when searching large databases for novel compounds from new chemical space. We compared the performance of the pharmacophore models at discriminating known HIV-1p inhibitors from a large, general dataset of 2322 decoy compounds. Again, both the NMR and crystal models display excellent performance at selecting out the known inhibitors. The NMR model again performs very well when 7/8 or 8/8 sites are required. Both 7/8, 2×RMSD and 8/8, 3×RMSD identified 89.9% of the true positives and only 2.8% and 4.1% of the false positives, respectively. Once more, the optimal cu-crystal model required 9/11 sites (2.7×RMSD) to perform similarly to the optimal NMR model, identifying 88.8% of the true positives and 3% of the false positives. However, only the NMR model was able to identify almost 100% of the true positives. The presence of

extraneous sites may explain why the cu-crystal models miss identifying several of the known inhibitors even with the most generous criteria.

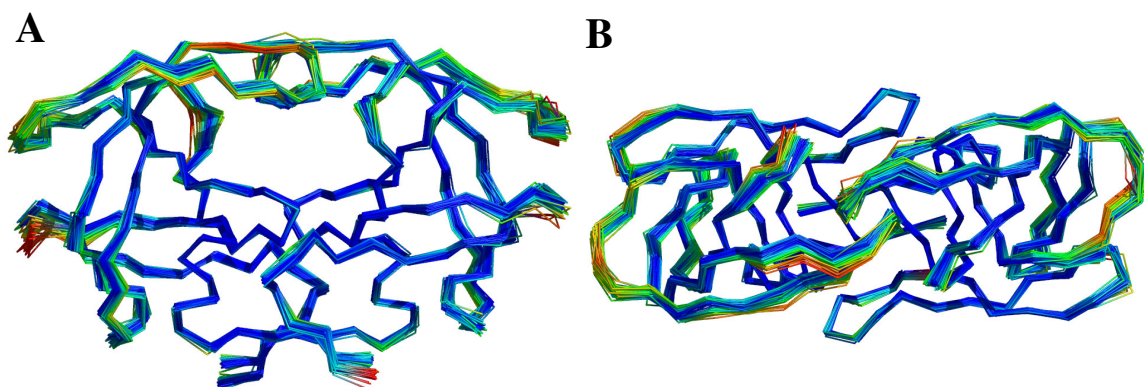
Several aspects of the model's performance confirm patterns we observed with the apo-MD models in our previous studies.^{23,43} First, all sets of ROC curves are very steep at the beginning indicating the potential for models with smaller radii to be used in virtual screening applications. When screening large databases of compounds, the number of true positives can be sacrificed to reduce the amount of false positives. Second, larger radii are needed when more elements are required. Third, among the false positives identified by the pharmacophore models are renin inhibitors, transition-state mimics of peptide cleavage, and small hydrophobic signaling peptides. This is not surprising since renin is a homologous aspartic protease, the function of HIV-1p is to cleave peptides, and the substrates are hydrophobic in nature. Similar classes were identified from both decoy databases, but as one would expect due to the size of the general database (2322), additional classes were also seen. This list includes macrocyclics (another HIV-1p inhibitor class), beta-lactams, tetracenes, and other polycyclic systems. However, for brevity we are providing only the structures of the identified false positives from the database of 85 chemically similar compounds in Appendix 3.

Effect of the Structure Number in Ensemble

Only 10 structures were used to generate the cu-crystal model, but the NMR ensemble contains 28 conformations. We were concerned that the larger number of structures in the NMR ensemble may bias the model for better performance. To ensure a fair comparison between the NMR and crystal structures, we also generated an additional MPS pharmacophore model from 90 crystal structures (all-crystal model). The 90 structures are bound to a variety of ligand classes. Once again, there is little backbone variation between the 90 structures; the C α RMSD values range from 0.12 – 0.71 Å. An

Gaussian-weighted overlay of the structures using the C α coordinates is provided in Figure 4.8. Zoete and coworkers also found minimal variation between 73 HIV-1p backbones bound to different ligands.⁵⁰ Moreover, we observed that adding 35 structures with resistant mutations did not provide any additional conformational variation (125 structures total, data not shown).

Figure 4.8. Gaussian-weighted overlay of 90 crystal structures from drug-susceptible strains of HIV-1p. The color code of the weights is the same as in Figure 4.1C, $c = 2 \text{ \AA}^2$. (A) Front View (A) Top View.



We also calculated the RMSD for each heavy-atom of the protein active site (defined as any atom within 10 \AA of the active site center) using 1PRO as the reference structure. This was chosen because 1PRO is the representative structure for the HIV-1p family in the Binding MOAD database²⁷ as it has the tightest bound inhibitor. Moreover, it is also bound to a cu ligand, making it appropriate choice for comparing the cu-crystal structures to the larger set of 90 structures. The RMSD values ranged from $0.16 - 1.80 \text{ \AA}$, with an average of $0.51 \pm 0.37 \text{ \AA}$.

The small conformational variation between the crystal structures does not appear to be an effect of crystal structure refinement. It is common practice to use a previously solved crystal structure when determining the coordinates of another. However, our inspection of the electron density maps showed the structures to be of high quality with well-resolved density defining the coordinates. Crystal packing effects are known to be

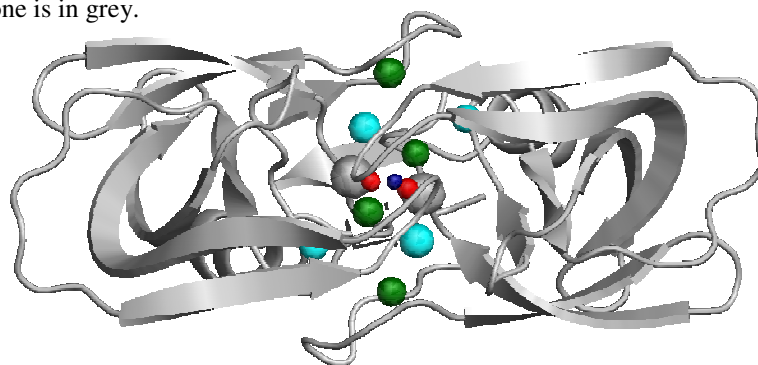
important for conformations of HIV-1p, but there is no evidence to suggest that is the cause of the limited sampling. Most likely, the resulting conformations are influenced by a variety of factors including the conditions used in X-ray crystallography such as temperature and pH. For example, low temperatures are typical for growing crystals and may not provide enough thermal energy for a protein to overcome the barrier to sample conformations outside of a particular local minimum. In the case of HIV-1p, apo crystal structures are found in the semi-open conformation while bound structures exist in the closed state. A 1.3-Å apo crystal structure of a highly mutated HIV-1p strain was recently solved in a novel open conformation (PDB ID: 1TW7)⁵¹, but the open state was later shown to be caused by crystal packing effects⁵².

Cross-docking studies in the literature demonstrate how different HIV-1p structures perform poorly when trying to dock ligands taken from other crystal structures.^{53,54} However, there are also cross-docking examples where HIV-1p performs quite well.⁵⁵ Furthermore, there are examples in the literature where HIV-1p is able to reproduce docking poses of its own ligands (i.e. co-crystal structure) successfully⁵³ and also unsuccessfully⁵⁴. We propose that the difficulties in those studies arise from the ligands of HIV-1p, not the structures of the proteins. The majority of the bound ligands are large, flexible peptides. It is well known that many of the docking programs have difficulty with ligands that have many rotatable bonds. The different studies in the literature used different routines for sampling ligands, and this could be the real source of poor cross-docking results. This argument supports our structural analysis of HIV-1p; there is very little variation between the crystal structures.

The resulting all-crystal pharmacophore model, shown in Figure 4.9, is very similar to the cu-crystal model. The only exception is two elements that are aromatic in the all-crystal model, rather than aromatic/hydrophobic. The inclusion of more structures appears to cause the two elements to become less general. The sphere centers and radii are nearly identical between the cu-crystal and all-crystal models, apart from the aromatic

sites flanking the solvent exposed region of the binding site. In the all-crystal model, they better replicate the C_2 symmetry of the protein. The model performance does change slightly; the optimal model now requires even more elements to be dropped: 8 out of 11 elements (8/11 sites, $2.7\times$ RMSD). Furthermore, it identifies less of the true positives (86.5% compared to 89.9%) and more of the false positives (14.1% compared to 7.1%). The raw data from the pharmacophore screen is available in Appendix 3. Again, as in the case of the average NMR model, the loss of the hydrophobic character of the aromatic elements in the S1/S1' pocket does seem to negatively affect the performance of the model; it appears that an aromatic/hydrophobic element truly provides a more accurate representation of the active-site pockets.

Figure 4.9. Pharmacophore model (radii of $1\times$ RMSD) generated using 90 crystal structures bound to a diverse set of ligands. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, aromatic/hydrophobic; green, aromatic. Top view of protease is shown; backbone is in grey.



As previously mentioned, the range of $C\alpha$ RMSD values for the apo-MD ensemble (11 structures) is comparable to that of the NMR (28 structures), 0.94 – 1.50 Å versus 0.65 – 1.71 Å respectively, and that the MPS models are nearly identical. However, the range of $C\alpha$ RMSD values for both the cu-crystal (10 structures) and all-crystal (90 structures) collections is much smaller, 0.26 – 0.80 Å and 0.12 – 0.71 Å, respectively. Both models based on crystal structures have 3 additional sites. We strive to generate pharmacophore models from an ensemble that represents an appropriate sampling of conformational space. It appears that both NMR and MD ensembles can

account for more accessible conformations than bound protein-ligand crystals structures, even those bound to a set of diverse ligands. We stress that crystal structures are very useful in many other SBDD applications, but we believe that bound HIV-1p crystal structures do not provide a complete sampling of receptor conformations, and NMR models can have definite advantages when trying to represent the protein's flexibility.

4.4 Conclusions

Incorporating protein flexibility into structure-based drug design is necessary to simulate a more accurate representation of a protein in solution. By looking for favorable interaction regions across multiple conformations of a protein, we can determine the most essential and conserved features of the active site. We are able to show that the MPS method can be extended to include the use of experimental structures as a source of multiple conformations. The use of experimentally determined structures is attractive over generating conformations from an MD simulation in order to reduce the amount of time required to develop an MPS model. Additionally, to our knowledge this is the first direct comparison of NMR ensembles and crystal collections for incorporating receptor flexibility in structure-based drug design.

The MPS pharmacophore models generated from an NMR ensemble and collections of crystal structures were able to discriminate known HIV-1p inhibitors from drug-like decoys and showed better performance than a model previously created using apo HIV-1p structures. They also showed superior performance over a model created from the average NMR structure. The average NMR model contained additional elements and lost important chemical characteristics that appeared to diminish the performance of the model, while the use of MPS identified the most important, chemically relevant features. The use of an average structure from multiple receptor conformations is an

alternate method that has been proposed for incorporating protein flexibility in structure-based drug design, but we find that ensembles of structures is a superior approach.

The present results are strong support for the use of NMR ensembles in structure-based drug design. The NMR model revealed only the most essential features of the binding site. Instead, the collection of crystal structures identified three additional, and less essential, elements. These were highly related to chemical features specific to the class of cu ligands. In order to achieve a reasonable performance, additional elements had to be dropped or the radii had to be multiplied by large scaling factors. The NMR model did not simply reproduce its bound ligand. It could be used in its entirety (8/8 sites, 3×RMSD) with exceptional performance for discriminating true inhibitors from decoy molecules. The performance improved slightly with 7/8 sites, 2×RMSD models, which is in good agreement with the parameters previously suggested for MPS based on MD (generally, $n-1$ of n features and radii of $\sim 2\times$ RMSD). Furthermore, the NMR ensemble samples a greater amount of conformational space than the crystal collection and is comparable to the amount of sampling seen in a 3-ns MD simulation of apo HIV-1p.²³

Overall, we recommend NMR structures over crystal structures for incorporating protein flexibility into SBDD studies of HIV-1p. By no means are the crystal structures inaccurate; instead, there is simply too little variation between the different structures, even when bound to a variety of ligand classes. In fact, this analysis strongly suggests that the difficulties seen in cross-docking studies of HIV-1p do not arise from the protein structures themselves. Most likely, the difficulty comes from the ligands which inhibit HIV-1p. Many routines employed to generate ligand conformations have difficulty with large, flexible compounds, and this could be the cause of the inconsistencies in the cross-docking results.^{53,54,55}

This work has been published as:

Damm, K.L. and Carlson, H.A. Exploring Sources of Multiple Protein Conformations in Structure-Based Drug Design. *J. Am. Chem. Soc.* **2007**, *129*, 8225-8235.

4.5 References

1. Philippopoulos, M.; Lim, C. Exploring the Dynamic Information Content of a Protein NMR Structure: Comparison of a Molecular Dynamics Simulation with the NMR and X-Ray Structures of Escherichia Coli Ribonuclease Hi. *Proteins* **1999**, *36*, 87-110.
2. Hornak, V.; Okur, A.; Rizzo, R. C.; Simmerling, C. Hiv-1 Protease Flaps Spontaneously Open and Reclose in Molecular Dynamics Simulations. *Proc Natl Acad Sci U S A* **2006**, *103*, 915-920.
3. Toth, G.; Borics, A. Flap Opening Mechanism of Hiv-1 Protease. *J Mol Graph Model* **2006**, *24*, 465-474.
4. Garbuzynskiy, S. O.; Melnik, B. S.; Lobanov, M. Y.; Finkelstein, A. V.; Galzitskaya, O. V. Comparison of X-Ray and NMR Structures: Is There a Systematic Difference in Residue Contacts between X-Ray- and NMR-Resolved Protein Structures? *Proteins* **2005**, *60*, 139-147.
5. Lee, M. R.; Kollman, P. A. Free-Energy Calculations Highlight Differences in Accuracy between X-Ray and NMR Structures and Add Value to Protein Structure Prediction. *Structure* **2001**, *9*, 905-916.
6. Spronk, C. A. E. M.; Linge, J. P.; Hilbers, C. W.; Vuister, G. W. Improving the Quality of Protein Structures Derived by NMR Spectroscopy. *J Biomol NMR* **2002**, *22*, 281-289.
7. Billeter, M. Q. Comparison of Protein Structures Determined by NMR in Solution and by X-Ray Diffraction in Single Crystals. *Rev Biophys* **1992**, *25*, 325-377.
8. Snyder, D. A.; Chen, Y.; Denissova, N. G.; Acton, T.; Aramini, J. M.; Ciano, M.; Karlin, R.; Liu, J.; Manor, P.; Rajan, P. A.; Rossi, P.; Swapna, G. V.; Xiao, R.; Rost, B.; Hunt, J.; Montelione, G. T. Comparisons of NMR Spectral Quality and Success in Crystallization Demonstrate That NMR and X-Ray Crystallography Are Complementary Methods for Small Protein Structure Determination. *J Am Chem Soc* **2005**, *127*, 16505-16511.
9. Yee, A. A.; Savchenko, A.; Ignachenko, A.; Lukin, J.; Xu, X.; Skarina, T.; Evdokimova, E.; Liu, C. S.; Semesi, A.; Guido, V.; Edwards, A. M.; Arrowsmith, C. H. NMR and X-Ray Crystallography, Complementary Tools in Structural Proteomics of Small Proteins. *J Am Chem Soc* **2005**, *127*, 16512-16517.
10. Thomas, M. P.; McInnes, C.; Fischer, P. M. Protein Structures in Virtual Screening: A Case Study with Cdk2. *J Med Chem* **2006**, *49*, 92-104.

11. Subramanian, J.; Sharma, S.; B-Rao, C. A Novel Computational Analysis of Ligand-Induced Conformational Changes in the Atp Binding Sites of Cyclin Dependent Kinases. *J Med Chem* **2006**, *49*, 5434-5441.
12. Davis, A. M.; Teague, S. J.; Kleywegt, G. J. Application and Limitations of X-Ray Crystallographic Data in Structure-Based Ligand and Drug Design. *Angew Chem Int Ed Engl* **2003**, *42*, 2718-2736.
13. Barril, X.; Morley, S. D. Unveiling the Full Potential of Flexible Receptor Docking Using Multiple Crystallographic Structures. *J Med Chem* **2005**, *48*, 4432-4443.
14. Spronk, C. A. E. M.; Nabuurs, S. B.; Bonvin, A. M.; Krieger, E.; Vuister, G. W.; Vriend, G. The precision of NMR structure ensembles revisited. *J Biomol NMR* **2003**, *25*, 225-234.
15. Knegt, R. M.; Kuntz, I. D.; Oshiro, C. M. Molecular Docking to Ensembles of Protein Structures. *J Mol Biol* **1997**, *266*, 424-440.
16. Huang, S-Y.; Zou, X. Efficient Molecular Docking of NMR Structures: Application to Hiv-1 Protease. *Protein Sci* **2007**, *16*, 43-51.
17. Fesik, S.W. NMR Structure-Based Drug Design. *J Biomol NMR* **1993**, *3*, 261-269.
18. McCoy, M. A.; Wyss, D. F. Structures of Protein-Protein Complexes Are Docked Using Only NMR Restraints from Residual Dipolar Coupling and Chemical Shift Perturbations. *J Am Chem Soc* **2002**, *124*, 2104-2105.
19. Zabell, A. P. R.; Post, C. B. Docking Multiple Conformations of a Flexible Ligand into a Protein Binding Site Using NMR Restraints. *Proteins* **2002**, *46*, 295-307.
20. Hicks, R. P. Recent Advances in NMR: Expanding Its Role in Rational Drug Design. *Curr Med Chem* **2001**, *8*, 627-640.
21. Hajduk P. J. Sar by NMR: Putting the Pieces Together *Mol Interv* **2006**, *6*, 266-272.
22. Zartler, E. R.; Shapiro, M. J. Protein NMR-Based Screening in Drug Discovery. *Curr Pharm Des* **2006**, *12*, 3963-72.
23. Meagher, K. L.; Lerner M. G.; Carlson, H. A. Refining the Multiple Protein Structure Pharmacophore Method: Consistency across Three Independent Hiv-1 Protease Models. *J Med Chem* **2006**, *49*, 3478-3484.
24. Erickson, J. A.; Jalaie, M.; Robertson, D. H.; Lewis, R. A.; Vieth, M. Lessons in Molecular Recognition: The Effects of Ligand and Protein Flexibility on Molecular Docking Accuracy. *J Med Chem* **2004**, *47*, 45-55.

25. Yamazaki, T.; Hinck, A. P.; Wang, Y. X.; Nicholson, L. K.; Torchia, D. A.; Wingfield, P.; Stahl, S. J.; Kaufman, J. D.; Chang, C. H.; Domaille, P. J.; Lam, P. Y. Three-Dimensional Solution Structure of the Hiv-1 Protease Complexed with Dmp323, a Novel Cyclic Urea-Type Inhibitor, Determined by Nuclear Magnetic Resonance Spectroscopy. *Protein Sci* **1996**, *5*, 495-506.
26. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res* **2000**, *28*, 235-242.
27. Hu, L.; Benson, M. L.; Smith, R. D.; Lerner, M. G.; Carlson, H. A. Binding MOAD (Mother Of All Databases). *Proteins* **2005**, *60*, 333-340.
28. Rutenber, E.; Fauman, E. B.; Keenan, R. J.; Fong, S.; Furth, P. S.; Ortiz de Montellano, P. R.; Meng, E.; Kuntz, I. D.; DeCamp, D. L.; Salto, R.; Rose, J. R.; Craik, C. S.; Stroud, R. M. Structure of a Non-Peptide Inhibitor Complexed with Hiv-1 Protease. Developing a Cycle of Structure-Based Drug Design. *J Biol Chem* **1993**, *268*, 15343-15346.
29. Backbro, K.; Lowgren, S.; Osterlund, K.; Atepo, J.; Unge, T.; Hulten, J.; Bonham, N. M.; Schaal, W.; Karlen, A.; Hallberg, A. Unexpected Binding Mode of a Cyclic Sulfamide Hiv-1 Protease Inhibitor. *J Med Chem* **1997**, *40*, 898-902.
30. Hodge, C. N.; Aldrich, P. E.; Bacheler, L. T.; Chang, C. H.; Eyermann, C. J.; Garber, S.; Grubb, M.; Jackson, D. A.; Jadhav, P. K.; Korant, B.; Lam, P. Y.; Maurin, M. B.; Meek, J. L.; Otto, M. J.; Rayner, M. M.; Reid, C.; Sharpe, T. R.; Shum, L.; Winslow, D. L.; Erickson-Viitanen, S. Improved Cyclic Urea Inhibitors of the Hiv-1 Protease: Synthesis, Potency, Resistance Profile, Human Pharmacokinetics and X-Ray Crystal Structure of Dmp 450. *Chem Biol* **1996**, *3*, 301-314.
31. Lam, P. Y.; Jadhav, P. K.; Eyermann, C. J.; Hodge, C. N.; Ru, Y.; Bacheler, L. T.; Meek, J. L.; Otto, M. J.; Rayner, M. M.; Wong, Y. N.; Chang, C-H.; Weber, P. C.; Jackson, D. A.; Sharpe, T. R.; Erickson-Viitanen, S. Rational Design of Potent, Bioavailable, Nonpeptide Cyclic Ureas as Hiv Protease Inhibitors. *Science* **1994**, *263*, 380-384.
32. Ala, P. J.; Huston, E. E.; Klabe, R. M.; Jadhav, P. K.; Lam, P. Y.; Chang, C. H. Counteracting Hiv-1 Protease Drug Resistance: Structural Analysis of Mutant Proteases Complexed with Xv638 and Sd146, Cyclic Urea Amides with Broad Specificities. *Biochemistry* **1998**, *37*, 15042-15049.
33. Sham, H. L.; Zhao, C.; Stewart, K. D.; Betebenner, D. A.; Lin, S.; Park, C. H.; Kong, X. P.; Rosenbrook, W., Herrin Jr., T.; Madigan, D.; Vasavanonda, S.; Lyons, N.; Molla, A.; Saldivar, A.; Marsh, K. C.; McDonald, E.; Wideburg, N. E.; Denissen, J. F.; Robins, T.; Kempf, D. J.; Plattner, J. J.; Norbeck, D. W. A Novel,

- Picomolar Inhibitor of Human Immunodeficiency Virus Type 1 Protease. *J Med Chem* **1996**, *39*, 392-397.
34. Jadhav, P. K.; Ala, P.; Woerner, F. J.; Chang, C. H.; Garber, S. S.; Anton, E. D.; Bacheler, L. T. Cyclic Urea Amides: Hiv-1 Protease Inhibitors with Low Nanomolar Potency against Both Wild Type and Protease Inhibitor Resistant Mutants of HIV *J Med Chem* **1997**, *40*, 181-191.
35. Lam, P. Y.; Ru, Y.; Jadhav, P. K.; Aldrich, P. E.; DeLucca, G. V.; Eyermann, C. J.; Chang, C. H.; Emmett, G.; Holler, E. R.; Daneker, W. F.; Li, L.; Confalone, P. N.; McHugh, R. J.; Han, Q.; Li, R.; Markwalder, J. A.; Seitz, S. P.; Sharpe, T. R.; Bacheler, L. T.; Rayner, M. M.; Klabe, R. M.; Shum, L.; Winslow, D. L.; Kornhauser, D. M.; Jackson, D. A.; Erickson-Viitanen, S.; Hodge, C. N. Cyclic Hiv Protease Inhibitors: Synthesis, Conformational Analysis, P2/P2' Structure-Activity Relationship, and Molecular Recognition of Cyclic Ureas. *J Med Chem* **1996**, *39*, 3514-3525.
36. Huang, P. P.; Randolph, J. T.; Klein, L. L.; Vasavanonda, S.; Dekhtyar, T.; Stoll, V. S.; Kempf, D. J. Synthesis and Antiviral Activity of P1' Arylsulfonamide Azacyclic Urea Hiv Protease Inhibitors. *Bioorg Med Chem Lett* **2004**, *14*, 4075-4078.
37. Lovell, S. C.; Davis, I. W.; Arendall, W. B., 3rd; de Bakker, P. I.; Word, J. M.; Prisant, M. G.; Richardson, J. S.; Richardson, D. C. Structure Validation by Calpha Geometry: Phi, Psi and Cbeta Deviation. *Proteins* **2003**, *50*, 437-450.
38. Case, D. A.; Pearlman, D. A.; Caldwell, J. W.; Cheatham, T. E., III; Ross, W. S.; Simmerling, C. L.; Darden, T. A.; Merz, K. M.; Stanton, R. V.; Cheng, A. L.; Vincent, J. J.; Crowley, M.; Tsui, V.; Radmer, R. J.; Duan, Y.; Pitera, J.; Massova, I.; Seibel, G. L.; Singh, U. C.; Weiner, P. K.; Kollman, P. A. *AMBER 6*; University of California San Francisco: San Francisco, CA, 1996..
39. Jorgensen, W. L. *BOSS*, Version 4.2; Yale University: New Haven, CT, 2000.
40. Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J., Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J Am Chem Soc* **1996**, *118*, 11225-11236.
41. Carlson, H. A.; Masukawa, K. M.; Rubins, K.; Bushman, F. D.; Jorgensen, W. L.; Lins, R. D.; Briggs, J. M.; McCammon, J. A. Developing a Dynamic Pharmacophore Model for Hiv-1 Integrase. *J Med Chem* **2000**, *43*, 2100-2114.
42. Damm K. L.; Carlson, H. A. Gaussian-Weighted Rmsd Superposition of Proteins: A Structural Comparison for Flexible Proteins and Predicted Protein Structures. *Biophys J* **2006**, *90*, 4558-4573.

43. Meagher, K. L.; Carlson, H. A. Incorporating Protein Flexibility in Structure-Based Drug Discovery: Using Hiv-1 Protease as a Test Case. *J Am Chem Soc* **2004**, *126*, 13276-13281.
44. *Molecular Operating Environment*; Chemical Computing Group Inc.: Montreal, Canada, 2001.
45. *Comprehensive Medicinal Chemistry*; Hansch, C., Sammes, P. G., Taylor, J. B., Eds.; Pergamon Press: Oxford, 1990; Vol. 1-6.
46. *Comprehensive Medicinal Chemistry Database*; MDL Information Systems, Inc.: San Leandro, CA, 2003.
47. Bowman, A. L.; Lerner, M. G.; Carlson, H. A. Protein Flexibility and Species Specificity in Structure-Based Drug Discovery: Dihydrofolate Reductase as a Test System. *J Am Chem Soc* **2007**, *129*, 3634-3640.
48. Erickson, J. W. Design and structure of symmetry-based inhibitors of HIV-1 protease. *Perspect. Drug Discov. Des.* **1993**, *1*, 109-128.
49. Wlodawer, A.; Erickson, J. W. Structure-Based Inhibitors of Hiv-1 Protease. *Annu. Rev. Biochem.* **1993**, *62*, 543-585.
50. Zoete, V.; Michielin, O.; Karplus, M. Relation between Sequence and Structure of Hiv-1 Protease Inhibitor Complexes: A Model System for the Analysis of Protein Flexibility. *J Mol Biol* **2002**, *315*, 21-52.
51. Martin, P.; Vickrey, J. F.; Proteasa, G.; Jimenez, Y. L.; Wawrzak, Z.; Winters, M. A.; Merigan, T. C.; Kovari, L. C. "Wide-Open" 1.3 a Structure of a Multidrug-Resistant Hiv-1 Protease as a Drug Target. *Structure* **2005**, *13*, 1887-1895.
52. Layten, M.; Hornak, V.; Simmerling, C. The Open Structure of a Multi-Drug-Resistant Hiv-1 Protease Is Stabilized by Crystal Packing Contacts. *J Am Chem Soc* **2006**, *128*, 13360-13361.
53. Ferrara, P.; Gohlke, H.; Price, D. J.; Klebe, G.; Brooks III, C. L. Assessing Scoring Functions for Protein-Ligand Interactions. *J Med Chem* **2004**, *47*, 3032-3047.
54. Perola, E.; Walters, W. P.; Charifson, P. S. A Detailed Comparison of Current Docking and Scoring Methods on Systems of Pharmaceutical Relevance. *Proteins* **2004**, *56*, 235-249.
55. Joseph-McCarthy, D.; Thomas, B. E. t.; Belmarsh, M.; Moustakas, D.; Alvarez, J. C. Pharmacophore-Based Molecular Docking to Account for Ligand Flexibility. *Proteins* **2003**, *51*, 189-202.

CHAPTER 5

Accounting for Multiple Protein Conformations in Ranking Ligand Databases

5.1 Introduction

A difficulty that occurs when employing ensembles of protein structures is developing a ranking function that accounts for all protein conformations. As discussed in Chapter 1, various groups have utilized different approaches; however, many are ultimately ranking the ligand pose against a single, static protein structure. Our current implementation of the MPS pharmacophore models is geared towards speed – allowing rapid filtering of databases to identify subsets that possess the correct features and geometry. No attempt to rank the molecules is made, as the discriminatory criteria is simply whether at least one molecular conformation could satisfy the pharmacophore constraints. Rather than averaging multiple grids, or taking into account the combinatorial possibilities of different conformations, we asked whether quantifying ligand overlap with a set pharmacophore spheres generated from multiple protein conformations could provide a simpler approach for ranking ligands with our pharmacophore-based method.

To probe this question, we have developed an atomistic pharmacophore representation that more specifically maps the contours of the interaction surface between each individual probe and protein. As the location and chemical characteristics of each probe are merged from MPS, the model reveals only the most energetically favorable interactions conserved across the protein conformations. Consequently, the density of the

pharmacophore model can be used to quantify, and provide a ranking metric, for how well a candidate ligand fits the pharmacophore. Our implementation does not determine specific interactions with a static protein conformation; instead it measures the amount of overlap between ligand atoms and the atomistic MPS pharmacophore elements.

We have implemented our atomistic MPS pharmacophore models in DOCK 4.0.1 (DOCK)^{1,2} as sphere sets used to guide the ligand orientation algorithm (MPS-DOCK). DOCK orients ligands within a binding pocket by matching ligand heavy atoms to a set of spheres or site-points which map the negative volume of the cavity. It is common practice to encode pharmacophoric information in DOCK by chemically labeling spheres and require ligand poses to complement the labeling during orienting.²⁻⁶ Our method is different in that we are generating receptor-based pharmacophore models based on ensembles of structures.

Furthermore, we have developed a simple ranking metric to measure ligand overlap with the pharmacophore model. Each probe in the consensus cluster is represented as a set of atomic, site-point spheres, and the number of site-points matched is used as a metric for ranking ligand orientations. This ranking function is similar to the knowledge-based scoring implemented in SLIDE.⁷ SLIDE counts hydrogen bonds and hydrophobic complementarities between the protein and a particular ligand orientation. By quantifying the overlap between ligand atoms and the atomistic pharmacophore spheres, we are able to identify known HIV-1p inhibitors in the top fractions of ranked data set. This ranking method also shows enrichment in discriminating small data sets of known HIV-1p inhibitors from both chemically similar decoys and general drug-like compounds. Additionally we are able to demonstrate the robustness of our method through the study of an additional protein system, *E. coli* DHFR (ecDHFR).

5.2 Computational Methods

Multiple Protein Structures Pharmacophore Model Generation

Multiple protein conformations taken from 3-ns MD simulations of three unbound structures of HIV-1p (PDB⁸ ID: 1HHP⁹, 3HVP¹⁰, and 3PHV¹¹) were used to generate MPS pharmacophore models.¹² An additional pharmacophore model was created using conformations from a 4 ns MD simulation of ecDHFR.¹³ The starting conformation for the ecDHFR MD simulation was a closed M20 loop conformation wild-type *E. coli* DHFR-NADPH co-crystal structure (PDB ID: 1RX1¹⁴). Each protein structure was solvated with explicit water and equilibrated using the AMBER94 force field¹⁵ and the AMBER6¹⁶ suite of programs. Counter ions were added to neutralize the systems. Protein conformations were taken after equilibration and every 300 ps along the 3 ns HIV-1p MD trajectory and every 200ps along the 4 ns DHFR MD trajectory.

Receptor-based pharmacophore models were then created using the 11 conformations from each individual HIV-1p MD trajectory (1HHP, 3HVP, and 3PHV) and the 20 conformations from the ecDHFR MD trajectory. The active site of each structure was alternately flooded with 500 (1000 for ecDHFR) small molecule probes (benzene, ethane, and methanol) which were minimized using MUSIC¹⁷ and the OPLS¹⁸ force field in the program BOSS¹⁹ as described previously¹². This resulted in clusters of small molecule probes at favorable interaction regions within the active site of each snapshot. If 8 probes were present in the cluster, it was represented by a single “parent” probe, defined as the probe within each cluster possessing the most favorable interaction energy with the protein.

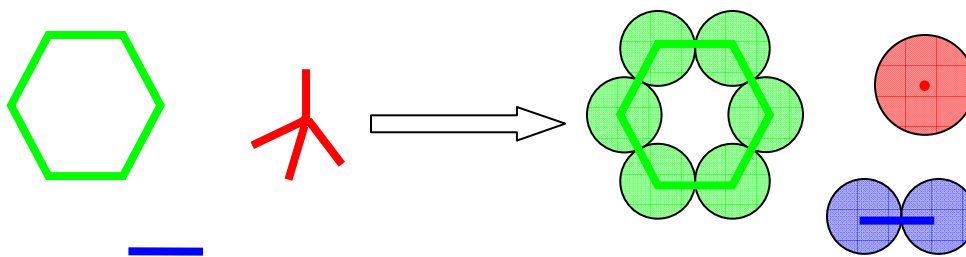
Consensus pharmacophore elements were determined by aligning the protein snapshots with each respective starting structure²⁰ and looking for interactions (positions of parent probes) that were common over multiple protein conformations. This procedure resulted in 8-site pharmacophore models for 1HHP and 3PHV, a 9-site model for 3HVP,

and a 5-site model for ecDHFR. The three individual HIV-1p models were overlaid, and the common features were averaged to generate an 8-site consensus pharmacophore model (CONS) which encodes conformational information from 3 independent MD simulations and 3 independent starting points. The detailed creation of these models has been previously described.^{12,13}

MPS-DOCK Orientation Spheres

The identified consensus clusters were used to create a contour-based pharmacophore representation to be employed as the MPS-DOCK sphere set for ligand orientation matching, the “orientation spheres”. Heavy atoms of each probe making up the consensus cluster were represented by an individual sphere as shown in Figure 5.1. Benzene probes are represented by 6 spheres centered on the aromatic carbons, ethane probes are mapped by 2 spheres centered on the aliphatic carbons, and methanol probes are described by a single sphere centered at the hydroxyl oxygen position. The radii of these atomic spheres was set to 0.75 Å for benzene and ethane which is approximately half a carbon-carbon bond length, and 1 Å for methanol which is approximately the length of an oxygen-hydrogen bond.

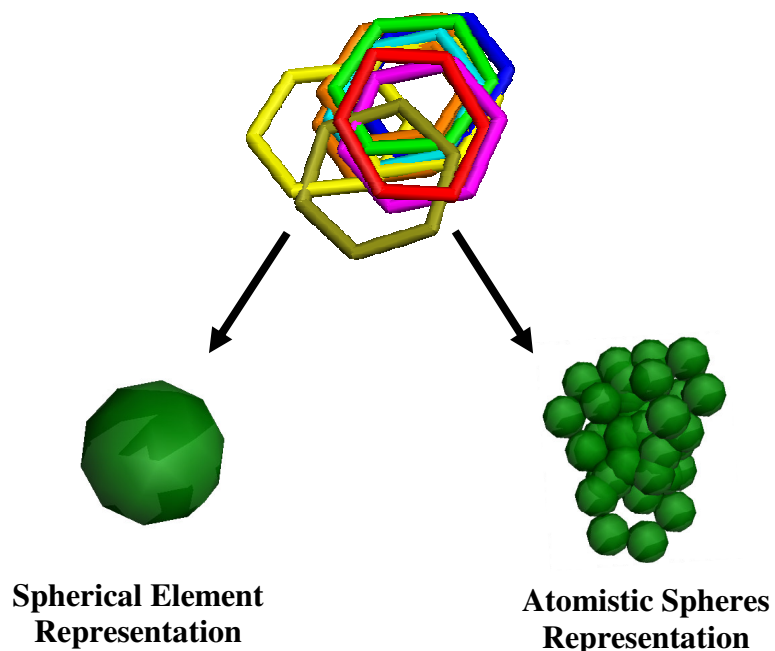
Figure 5.1. Representation of the small molecule probes as atomic spheres.



Each sphere was also labeled by its chemical functionality: aromatic, hydrophobic and hydrogen-bond donating or accepting for benzene, ethane and methanol, respectively. These functionality definitions are user specified and based on Tripos atom types (provided in Appendix 4). The use of chemical labeling allows the DOCK chemical

match feature to enrich the ligand orientations sampled with those that best match not only the position but also the chemistry of the spheres. The 3-ns pharmacophore models derived from the 1HHP, 3HVP and 3PHV simulations were represented as 397, 478 and 365 spheres respectively. The CONS pharmacophore elements common to the 3 individual models were represented as 1178 atomic spheres and the ecDHFR model as 266 spheres. Figure 5.2 shows a comparison of our original spherical pharmacophore element representation of a consensus cluster to our new atomistic, contoured representation.

Figure 5.2. Illustration of the aggregate pharmacophore element concept for a benzene consensus cluster. In our original model, the consensus cluster is represented as a single, spherical element (left). In our new model, each probe of the consensus cluster is represented by a set of atomistic spheres (right). The overlay of these sphere-sets generates a pharmacophore map whose size, shape, and density more accurately reflects the favorable interaction surface with the receptor.



A caveat of using our aggregate sphere representation is the large number of spheres in each set. The DOCK manual suggests using sets of approximately 50 spheres for orientation matching. To alleviate this potential issue, the sphere sets were broken up by chemical functionality, and each group was clustered using an iterative distance-based

approach. Using this approach, there are three necessary parameters: tolerance (the distance cut-off that a sphere must be within to be included in a cluster), step size, and maximum cluster size. We chose to set the tolerance to 2.0 Å, the step size to 0.005 Å, and vary the allowed size for each cluster, c . For example, if the maximum cluster size is set to 10, an initial round of clustering is performed using a tolerance of 2.0 Å. For any cluster that contains more than 10 spheres, it is re-clustered using a new tolerance of 2.0 Å minus the step size 0.005 Å. This procedure is iteratively continued until all clusters are within the maximum cluster size. Each cluster was then represented by a single sphere centered at the average position of the spheres in the cluster and given the same radius as discussed above. The values for c were varied from 1- 15, resulting in orientation sphere sets that ranged from 98 – 397 for 1HHP, 90 - 365 for 3PHV, 94 - 478 for 3HPV, 183-615 for CONS, and 91-266 for ecDHFR.

Excluded Volume Representation

Protein excluded volume spheres were determined by averaging the 2 C- γ positions of the catalytic aspartates (25, 25') (11 snapshots from 1HHP) and assigned a fixed radius of 1.5 Å. A more detailed representation of the protein volume was also investigated for both HIV-1p and ecDHFR by including all protein atoms within an RMSD cut-off. A heavy atom RMSD was calculated, and all atoms with a RMSD less than 0.5 Å, 0.75 Å, 1.00 Å, 1.25 Å, and 1.50 Å were defined as excluded volumes. The size of the excluded volumes was investigated using a fixed value of 0.5, 1, and 1.5 Å.

MPS-DOCK Ranking Function

An empirical ranking function was incorporated into MPS-DOCK to quantify how well a ligand overlaps with the orientation sphere set (i.e. aggregate pharmacophore spheres). The set of “scoring spheres” was defined as the combination of the orientation sphere set and protein excluded volume spheres. The density of the orientation spheres

reflects the relative importance of a particular ligand atom position within the interaction while the protein excluded volume spheres account for steric interactions with the protein.

To rank a pose, each ligand heavy atom position is sequentially compared to the coordinates of each orientation sphere set for all ligand conformations. If the distance between the atom and sphere center is less than the pre-defined sphere radius, the chemistry is evaluated. The ligand atom type is compared to the chemical definition of the sphere according to the matching rules provided in Appendix 4, and if the chemistry agrees, the metric is incremented using a weighted scoring system. The value incremented for each sphere is scaled by $1/n$, where n is the number of atoms in the probe. For example, as benzene has 6 spheres in the aggregate representation, a sphere representing a benzene atom would be incremented $1/6$ or 0.17 whereas a methanol sphere would be incremented $1/1$ or 1 because of the single oxygen atom in the methanol probe. This ensures that an aromatic interaction is weighted the same as a hydrogen bond interaction. Also, in order to maintain the “density” of the aggregate representation within the clustered representation, each sphere value is multiplied by the number of spheres making up its representative cluster. For example if a particular sphere represented a cluster of 15 benzene spheres, the value incremented would be 15 multiplied by 0.17 or 2.55. If the ligand heavy atom is not found within the volume of any atomic sphere or the chemistry with a sphere does not match, no penalty is applied. However, the metric is incremented for each sphere the atom is within so that matching to a region that is dense with spheres leads to a more favorable rank.

The ligand orientation algorithm in MPS-DOCK matches ligands solely to the orientation sphere set (i.e. aggregate pharmacophore sphere set). In order to account for the volume occupied by the protein, a steric penalty was incorporated into the ranking function. If a ligand heavy atom is within the radius of an excluded volume sphere, the score is given a deduction of 10 points. This straightforward ranking function orders

screened ligands based on their ability to complement the aggregate sphere pharmacophore model, yet does not severely penalize molecules for placing groups into flexible regions of the binding site.

Our ranking function was implemented into DOCK by altering the chemical scoring routine to sum ligand heavy atom interactions with the set of scoring spheres rather than computing the van der Waals interactions between ligand and protein. Additionally, the scoring sum was computed a single time rather than iterating over the number of receptor atoms. This approach was chosen to minimize effects on the program in general, specifically the orientation routines of DOCK. During the screening method, each ligand conformation was rigidly oriented varying the number of nodes, inter-node distance, and distance tolerance. A maximum of 5000 orientations were generated for each conformation. The highest valued orientations for each ligand were ranked, and the score of the top pose was compared to that of the other ligands in a second round of ranking. Receptor site chemical matching was enabled, but no grid scoring or minimization of the pose was performed.

Ligand Data Sets

Three data sets of pre-generated multiple conformers were used to evaluate the enrichment capabilities of our DOCK ranking function for the HIV-1p systems investigated. The first was a set of 89 structurally diverse known HIV-1p inhibitors taken from the PDB and the literature and was used to assess selectivity. The second was a database of 2324 drug-like non-inhibitors (2 known HIV-1p inhibitors were removed) from the Comprehensive Medicinal Chemistry Index^{21,22} and was used to evaluate specificity. In the original work, 23 folate-like inhibitors (structurally similar to known DHFR inhibitors) were removed; however, they were kept in this study to ensure a fair comparison across the protein systems as peptide-like compounds (structurally similar to known HIV-1p inhibitors) are present in the database. The preparation and composition

of these data sets has been described previously.^{23,24} Furthermore, a non-proprietary data set used by Cummings and co-workers was employed to replicate a virtual screening application.²⁵ Their data set consists of 1025 compounds seeded with 5 known HIV-1p inhibitors. It was prepared by assigning PEOE partial charges and using default OMEGA²⁶ parameters to generate a maximum of 400 ligand conformations per structure.

Two additional databases were also used to evaluate the ecDHFR model: a set of 50 high affinity *E. coli* inhibitors, where high affinity is defined by IC₅₀ values ranging from 2 – 28 nM, and another containing 541 general DHFR inhibitors. Further details of their creation can be found in a previous study.^{13,24} Rule-based torsion driving in OMEGA²⁶ was used to produce multiple conformations of each molecule in the DHFR databases, using an energy cutoff of 14 kcal/mol and heavy-atom RMSD criterion of 1 Å.

5.3 Results and Discussion

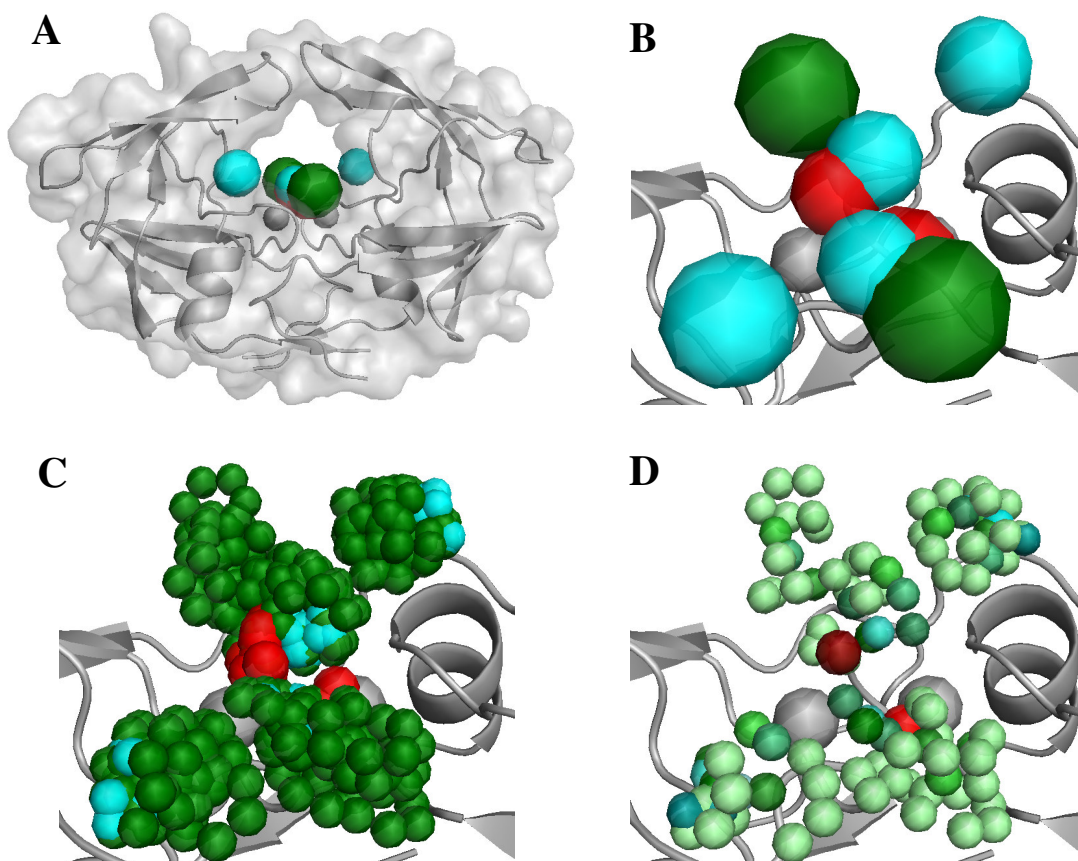
Application to HIV-1p

The MPS pharmacophore method has consistently discriminated known HIV-1p inhibitors from structurally similar non-inhibitors. However, our current pharmacophore implementation, while very fast at screening databases of compounds, does not provide a means to rank or prioritize compounds; it simply provides a “fit or no fit” evaluation. To address this aspect of the method, we sought to incorporate a ranking function into our MPS method. We chose to use a straightforward counting function that maximizes overlap with a contour representation of the pharmacophore model.

Here we have employed HIV-1p as a test case, using conformational snapshots of an apo structure (1HHP⁹) across a MD simulation to generate a receptor-based pharmacophore model. The resulting MPS model generally had 8 sites: 2 hydrogen-bond donating, 2 aromatic, and 4 aromatic/hydrophobic.¹² As an example, the pharmacophore

model for 1HHP is shown in Figure 5.3A positioned in the bottom of the active site while Figure 5.3B provides a close-up view. This pharmacophore model can then be represented as a set of 397 atomic spheres corresponding to each probe in the consensus clusters identified using the MPS protocol, Figure 5.3C. The density of the sphere distribution reflects the favorability for placing a ligand heavy atom at a given position. With this representation, we are including specific atomic interactions rather than looking at averaged molecular positions. This atomistic representation was condensed by clustering the spheres using an iterative distance-based method. Figure 5.3D shows a representative “clustered” set of Figure 5.3C. Each sphere represents a cluster ranging from 1-18 spheres; the shade of the chemical functionality color corresponds to the number of spheres in the cluster (the shading decreases with decreasing spheres).

Figure 5.3. Comparison of the MPS pharmacophore representations for 1HHP. Flap residues 46/46' – 54/54' have removed for clarity in (B) - (D). (A) Front view of HIV-1p with original pharmacophore model sitting in active site bottom to orient reader. (B) Close-up view of the original pharmacophore model derived for HIV-1p. (C) Close-up view of the aggregate sphere representation. (D) Close-up view of the clustered aggregate sphere representation. In all representations spheres are colored according to chemical functionality: red, hydrogen-bond donating; green, aromatic; cyan, hydrophobic. The clustered representations are also shaded by weight- the greater the number of spheres making up the cluster, the darker the color.



MPS-DOCK Parameter Investigation

To evaluate the use of quantifying overlap with a contour-based representation of the MPS pharmacophore models, we have employed a virtual screening approach. In virtual screening, large databases of mostly inactive compounds are screened with the hope of selecting the few inhibitors in the top fractions of the database. Recently, Cummings and coworkers have made available the non-proprietary portions of a data set they used to compare different docking programs in a virtual screening application.²⁵ In

their dataset, 5 known HIV-1p inhibitors were seeded among 1025 compounds selected to represent a typical screening collection and whose properties generally reflected those of the seeded inhibitors. Cummings et al. found HIV-1p to be a particularly challenging case for the docking programs investigated.²⁵ Of the programs they investigated, only Glide was able to dock 4 out of the 5 seeded inhibitors in the top 50% of the database screened. Other programs failed to identify any known inhibitors in the top 20% of the ranked database and only selected 2 inhibitors in the top 50%. We have adopted the Cummings et al. data set to examine how our scoring method performed in a virtual screening situation and to provide a frame of reference to compare screening using the MPS pharmacophore methods with traditional docking approaches. It should be noted that Cummings et al. used both flexible ligand docking as well as a single bound structure (PDB ID: 1HVR²⁷) for their docking trials. Our method uses pre-generated rigid ligand conformers and a minimal protein representation based on the apo structure of HIV-1p. Thus our study addresses different aspects of virtual screening, but the use of a common data set provides context to evaluate the results.

The aggregate sphere sets both define the binding pocket and restrain the orientation space. In the MPS-DOCK orientation search routine, each sphere is considered a node with respect to ligand orienting and chemical matching. Only ligand orientations that match a minimum number of nodes, m , and whose matching nodes are a specified distance apart, d , are accepted during the search and subsequently scored. Varying m and d is analogous to varying the pharmacophore query stringency and radii size in our previous work.²³ The minimum number of nodes required was varied from 2 to 5, and the distance required between nodes varied from 2 to 4 Å. We also investigated the affect of varying the distance tolerance, t , using values of 0.25 and 0.30. Two excluded volumes were used to define the floor of the HIV-1p active site in a manner analogous to our original pharmacophore approach, and a penalty of 10 was applied if a ligand heavy atom was within the radius of an excluded volume sphere. The value

incremented for each sphere was scaled by $1/n$, where n is the number of atoms in the probe, and multiplied by the number of spheres making up its representative cluster, where c is the maximum allowed size for a cluster. We were interested in probing the effect of varying the maximum cluster size, c , while changing the number of minimum required nodes, m , and their distance apart, d .

Results of varying the MPS-DOCK parameters for the 1HHP aggregate sphere model are presented in Table 5.1 for each maximum cluster size, c . The number of spheres in each clustered set is also shown and ranged from 110-399. We found that below $3m - 4d$ (3 nodes matched with a distance of 4 Å between them), the query was too lenient; the ligands were “clumping” to small areas of the sphere set, and the poses did not look realistic. The query was too stringent above $4m - 3d$, and only a small percentage of the database was able to meet these criteria and be subsequently ranked. This stringent performance is ideal for screening against HIV-1p but may be target specific. Given the large size of the HIV-1p active site, this query is likely selecting molecules based partly on size and is therefore not an appropriate test of the ranking function effectiveness. To balance performance and ranking an appropriate amount of the database, we examined requiring $3m - 4d$, $4m - 2d$, and $4m - 3d$ more thoroughly.

Table 5.1. Effect of varying the number of minimum required nodes (m), inter-node distance (d), the distance tolerance (t), and cluster size cut-off (c) on virtual screening performance using the 1HHP model. Minimal representation was used for the floor of the active site (2 excluded volume with a scoring penalty of 10) as to optimize favorable scoring elements prior to excluding compounds based on size and overlap with the pocket. The Cummings et al. data set²⁵ was used, consisting of 1025 compounds seeded with 5 HIV-1p inhibitors. The number of spheres in the scoring set, the sum of the ranks of the 5 inhibitors and the number of compounds scored by MPS-DOCK (in parenthesis) is shown. A lower number for the sum of ranks indicates cases where the 5 known inhibitors are all ranked highly. A high number (close to 1025) in parenthesis is optimal as that means a large percentage of the database is being ranked. This proves a more appropriate test for our ranking function. The bolded column represents the optimal parameter set.

Sum of Ranks (# of Ranked Compounds)

# c , spheres		$m = 3, d = 4$		$m = 4, d = 2$		$m = 4, d = 3$	
		$t = 0.25$	$t = 0.30$	$t = 0.25$	$t = 0.30$	$t = 0.25$	$t = 0.30$
15	110	1048 (963)	1059 (968)	913 (924)	1059 (968)	563 (722)	633 (754)
14	114	1163 (965)	1192 (969)	1069 (932)	1192 (969)	593 (730)	628 (763)
13	117	1028 (967)	1150 (971)	1143 (943)	1044 (957)	596 (737)	723 (769)
12	124	829 (970)	843 (972)	903 (966)	840 (981)	459 (751)	578 (781)
11	131	861 (982)	946 (985)	1100 (971)	983 (981)	717 (792)	707 (828)
10	140	829 (982)	863 (986)	1282 (978)	1182 (986)	674 (799)	616 (838)
9	148	744 (981)	763 (986)	799 (980)	686 (990)	806 (807)	995 (846)
8	157	476 (988)	560 (991)	898 (991)	989 (1004)	625 (820)	716 (853)
7	170	705 (991)	729 (994)	864 (996)	715 (1007)	676 (841)	516 (870)
6	185	645 (994)	722 (996)	774 (1003)	552 (1008)	676 (862)	352 (890)
5	205	689 (993)	599 (994)	517 (1009)	314 (1016)	835 (878)	635 (897)
4	237	768 (998)	706 (999)	792 (1011)	694 (1018)	914 (907)	891 (922)
3	291	800 (999)	937 (1000)	884 (1020)	950 (1022)	784 (929)	568 (942)
1	399	960 (1000)	1165 (1002)	595 (1023)	902 (1024)	517 (937)	529 (948)

The Sum of Ranks for the 5 known inhibitors is presented in Table 5.1 as a means of comparing parameters, along with the number of ranked compounds in parenthesis.

The lower the Sum of Ranks, the better the enrichment capabilities of the ranking function. The first trend that emerged was the superior performance of the distance tolerance set to 0.25 over 0.30 as a whole. The second trend was the overall enhanced enrichment when 4 nodes with at least 3 Å separation was required. Additionally for $4m - 3d$, it appeared that increasing the maximum allowed cluster size and hence, decreasing the number of spheres in the set, also improved performance; however, the majority of the sphere sets resulted in reasonable sums. As the maximum allowed cluster size decreased, more ligands were ranked. We chose to focus on $4m - 3d$ and $0.25t$ for our further investigations.

In Table 5.2, the number of known inhibitors returned in the top fractions of the ranked Cummings et al. dataset is shown for the optimal parameter set, $4m - 3d$ and $0.25t$, along with the Sum of Ranks, Number of Ranked Compounds, and the actual rank of each known inhibitor (expansion of Table 5.1). The raw data for $3m - 4d$; $0.25t$, $3m - 4d$; $0.30t$, $4m - 2d$; $0.25t$, $4m - 2d$; $0.30t$, and $4m - 3d$; $0.30t$ is provided in Appendix 4. Each clustered 1HHP sphere set was able to identify all 5 known inhibitors in the top 50%, and the majority was also able to distinguish 3 or 4 in the top 20% of the database. Only 3 of the 15 spheres sets did not identify at least 1 known inhibitor in the top 1%. The sphere set consisting of 124 spheres performed the best with a value of 459 for the Sum of Ranks, identifying 2 known inhibitors in the top 2% of the database. The number of ranked compounds is also presented. As the clustered sphere size was increased, more compounds were able to be ranked. However, any compound not ranked was essentially filtered out based on inappropriate size and shape complementarily to the sphere set.

Table 5.2. Expansion of Table 1: Effect of varying cluster size cut-off (c) using the optimal MPS-DOCK parameters, $4m - 3d$ and $0.25t$. Data from Cummings et al.²⁵ is also presented as a comparison. For a given fraction of the ranked database, the number of known HIV-1p inhibitors identified is shown along with the sum of the ranks of the 5 inhibitors and the number of compounds scored by MPS-DOCK. The bolded row represents the optimal sphere set.

c (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
15 (110)	1	2	4	4	5	563	722	12, 31, 74, 85, 361
14 (114)	1	2	3	4	5	593	730	4, 34, 85, 112, 358
13 (117)	1	2	3	4	5	596	737	15, 31, 84, 167, 299
12 (124)	2	3	4	4	5	459	751	10, 19, 45, 99, 286
11 (131)	2	2	2	3	5	717	792	15, 17, 162, 245, 278
10 (140)	0	2	2	4	5	674	799	38, 47, 167, 228, 199
9 (148)	0	2	2	3	5	806	807	21, 40, 193, 237, 315
8 (157)	1	2	3	4	5	625	820	10, 39, 90, 176, 310
7 (170)	0	2	3	3	5	676	841	21, 27, 70, 250, 308
6 (185)	1	2	2	3	5	676	862	3, 21, 106, 214, 332
5 (205)	1	2	2	3	5	835	878	3, 32, 203, 271, 326
4 (237)	1	2	2	2	5	914	907	4, 31, 240, 269, 370
3 (291)	2	2	2	3	5	784	929	2, 11, 128, 285, 358
1 (399)	1	3	3	4	5	517	937	2, 21, 29, 129, 336
DOCK ²⁵	0	0	0	0	0	4686	N/A ^a	N/A ^a
DOCKVISION ²⁵	0	0	0	0	2	3352	N/A ^a	N/A ^a
GLIDE ²⁵	1	1	3	4	5	1267	N/A ^a	N/A ^a
GOLD ²⁵	0	0	0	0	1	3654	N/A ^a	N/A ^a

^aData not reported in Cumming et al. study.²⁵

The Sum of Ranks metric provides a means of comparison to the work of Cummings et al.²⁵; their results using four docking programs with the same data set are presented in Table 5.2 as well. Our goal is to incorporate a ranking function into the MPS pharmacophore method, not to develop a novel docking algorithm or scoring function. Therefore, it is not appropriate to explicitly compare the results of ranking using our pharmacophore models to the docking studies, but the published work does provide context for how other programs perform given the same data set. The best performing

docking program investigated by Cummings et al., Glide, had a sum of ranks equal to 1267, while other methods had values of 3000 to 5000.²⁵

The improved virtual screening performance using the MPS pharmacophore ranking method highlights the importance of including protein flexibility in docking and screening. In contrast, Cummings et al. used a single bound structure for their virtual screening. The cross-docking approach used by Cummings and coworkers could be one reason why the various docking programs they used failed to identify the diverse inhibitors in the top fractions of the ranked database. Furthermore, Cummings et al. employed flexible ligand docking whereas our study utilized pre-generated poses.

Effect of Protein Excluded Volumes

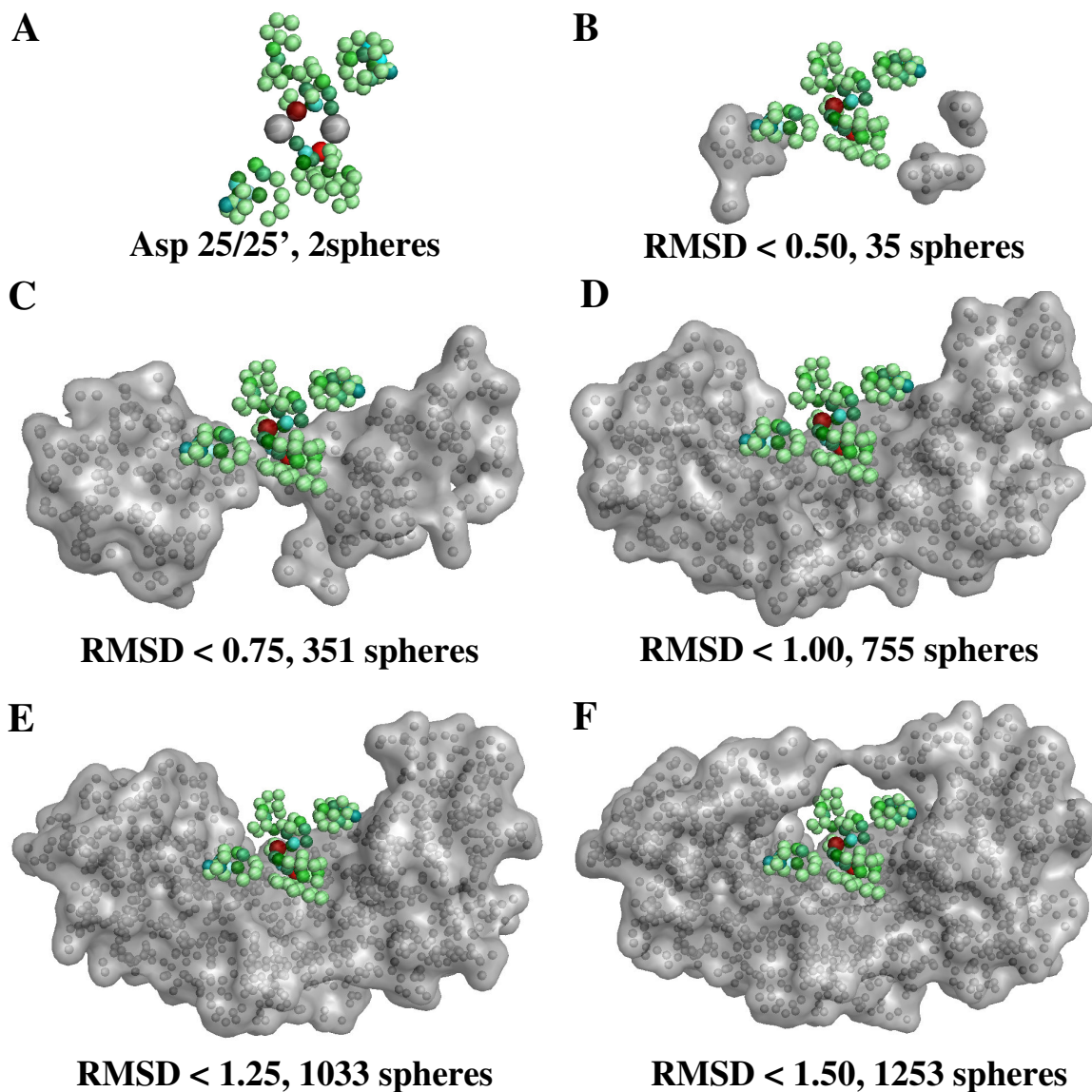
The number of excluded volumes used to represent the protein was also examined. To be consistent with our previous pharmacophore approach and bias the conformational sampling as little as possible, we first chose to use a minimum number of excluded volumes (2 volumes positioned at the average C γ positions of Asp 25 and 25'). This directs the ligand orientation to the top face of the model ensuring that it occupied the active site cavity but provides no penalty for placing ligand atoms into regions possibly occupied by the protein. However, as we are now using a more detailed representation of the pharmacophore model, we were also interested in investigating the effect of using a more detailed representation of the protein. The protein heavy atoms were averaged among the ensemble of structures, and 0.50, 0.75, 1.00, 1.25, and 1.5 Å RMSD cutoffs were used to determine the rigid atoms represented with excluded volumes. The average C α fluctuations calculated from the MD simulation was approximately 1.5 Å so this value was chosen as the upper bound.²³ The 0.50 Å minimum cutoff was the same as the cutoff used to determine variable regions in the approach by Knegtel et al.²⁸

Figure 5.4 shows the six different excluded volume representations (radii equal to 0.50 Å) along with the 110 sphere pharmacophore model ($c = 15$). Figures 5.4A and 5.4B demonstrate the minimalist representations of the receptor, 2 excluded volumes and a RMSD cut-off of 0.50 Å (35 spheres). The bottom of the active site becomes well defined at the RMSD cut-off of 1.00 Å (755 excluded volume spheres), Figure 5.4D, while the flap region begins to emerge at cut-offs of 1.25 Å (1033 excluded volume spheres) and 1.50 Å (1253 excluded volume spheres), given in Figure 5.4E and 5.4F.

Employing the optimal ligand orientations parameters ($4m - 3d$ and $0.25t$) and the Cummings et al. data set²⁵, we investigated the affect of the number of excluded volumes using a representative clustered sphere set, $c = 12$ (124 spheres). Using each excluded volume representation, fixed radii sizes of 0.5, 1, and 1.5 Å were first explored. For all of the excluded volume representations, the Sum of Ranks was quite high compared to our previous results when fixed radii sizes were equal to 1 and 1.5 Å (data not shown). Interestingly, using a fixed radius of 0.50 Å resulted in an almost identical Sum of Ranks for all excluded volume representations (0.50, 0.75, 1.00, 1.25, and 1.5 Å RMSD cutoffs); the values were 462 ± 3 . It appears that the excluded volume representation does not significantly affect the MPS-DOCK outcome.

We chose to continue with an RMSD cut-off of 1.00 Å and a fixed radius of 0.50 Å for the 1HHP excluded volumes. When a minimalist representation is used, the ligand is not penalized for protruding into areas occupied by receptor atoms that remain relatively static across the multiple conformations. Conversely, using a RMSD cut-off of 1.25 and 1.50 appears to be too restrictive as the flap region, which is known to be very flexible, is also now defined. The point of using MPS to generate the pharmacophore model is to overcome the cross-docking problem. By applying a “softer” representation of the receptor, we hope to probe novel conformational space while still representing the important chemical features of the binding site.

Figure 5.4. Comparison of excluded volume representations. **(A)** Minimal excluded volumes- 2 spheres centered on the 2 C γ positions of the catalytic aspartates (25, 25'). **(B)** RMSD cut-off of 0.50, 35 excluded volume spheres. **(C)** RMSD cut-off of 0.75, 351 excluded volume spheres. **(D)** RMSD cut-off of 1.00, 755 excluded volume spheres. **(E)** RMSD cut-off of 1.25, 1033 excluded volume spheres. **(F)** RMSD cut-off of 1.50, 1253 excluded volume spheres. Excluded volume spheres are colored gray and shown along with a surface representation. The atomic spheres are colored by chemical functionality (red, hydrogen-bond donating; green, aromatic; cyan, hydrophobic) and shaded by weight (the greater the number of spheres making up the cluster, the darker the color).



Extension to Additional HIV-1p Systems

Having demonstrated the success of the atomistic sphere overlap ranking function for the 1HHP model of HIV-1p, we applied the same technique to the additional HIV-1p models 3HVP and 3PHV, as well as the CONS model. The original 3PHV and CONS MPS pharmacophore models are nearly identical to the 1HHP model, resulting in very similar aggregate sphere representations. However, the original 3HVP model has 9 elements – an extra aromatic/hydrophobic element was conserved across the protein conformations.

For 1HHP we found that the optimal scoring set included 755 excluded volume spheres (RMSD cut-off of 1.00 Å). We anticipated that a similar number of scoring spheres should also perform well for the additional three HIV-1p systems. 3PHV followed the same trend as 1HHP and using a RMSD cut-off of 1.00 Å resulted in 750 excluded volumes. However, 3HVP required a RMSD cut-off of 0.75 Å to obtain a similar number of excluded volumes (768), and a RMSD cut-off of 1.25 Å was needed to produce 807 excluded volumes for the CONS model. These results are not surprising as a larger RMSD cut-off is required for less flexible systems, such as 3HVP, while a smaller cut-off is needed when the system is quite dynamic, as in the CONS conformational ensemble (33 snapshots from 3 different MD trajectories are employed). As previously noted, 3HVP also has an extra element in the original pharmacophore model; additional sites are characteristic of ensembles that do not display large amounts conformational variation.²⁹

Shown in Table 5.3 are the results for the best performing 1HHP, 3HVP, 3PHV and CONS models using the optimal MPS-DOCK parameters: $4m - 3d$, $0.25t$, and an excluded volume penalty of 10. The performance for all variations of each model's sphere sets (i.e. a range of maximum cluster size cut-offs) is provided in Appendix 4. We were pleased to see that all models show comparable ability to select known inhibitors in the top fractions of the ranked data set. Some variation is observed, for example, the

1HHP, 3HPV, and CONS models identified 2 known inhibitors in the top 2% screened as opposed to 1 inhibitor for 3PHV. However, the 3PHV model produced the lowest rank sum of the seeded HIV-1p inhibitors, 385, and predicted all 5 inhibitors within the top 20% screened. Additionally, all models were able to identify each of the known inhibitors in the top 50%, and the 1HHP model was even able to rank 4 inhibitors in the top 10%!

Table 5.3. Virtual screening performance of the four optimal HIV-1p pharmacophore models (1HHP, 3HVP, 3PHV, and CONS) using the optimal MPS-DOCK parameters, $4m - 3d$ and $0.25t$ along with an excluded volume penalty of 10. The number of orientation spheres and excluded volumes is provided. The Cummings et al. data set was used consisting of 1025 compounds seeded with 5 HIV-1p inhibitors. For a given fraction of the ranked database, the number of known HIV-1p inhibitors identified is shown along with the sum of the ranks of the 5 inhibitors and the number of compounds scored by MPS-DOCK.

	Sph Exvol		Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
			2%	5%	10%	20%	50%			
1HHP	122	755	2	3	4	4	5	459	751	5, 19, 45, 98, 292
3HPV	128	750	2	2	2	5	5	490	856	3, 14, 109, 170, 194
3PHV	97	768	1	2	3	5	5	385	806	3, 22, 75, 134, 151
CONS	513	807	2	2	3	4	5	581	981	6, 20, 102, 132, 321

It is interesting to note that the model with the lowest rank sum also has the smallest sphere set (3PHV), while the model with the highest rank sum is comprised of the largest sphere set (CONS). We hypothesize that a reason for the variation in performance may be due to the orientation sampling. It is difficult for the orientation routine in MPS-DOCK to sample appropriate conformations when a large number of spheres is required to be matched. In fact, the DOCK 4.0 manual suggests using a set of 50 spheres or less for optimal performance. Additionally, it took significantly longer to screen the databases of compounds when larger sphere sets were used.

An additional explanation for the performance variation may be due to the maximum allowed cluster size cut-off, c . When the sphere sets are sizeable, larger values of c are needed to generate small sphere sets. For example with the CONS model, a

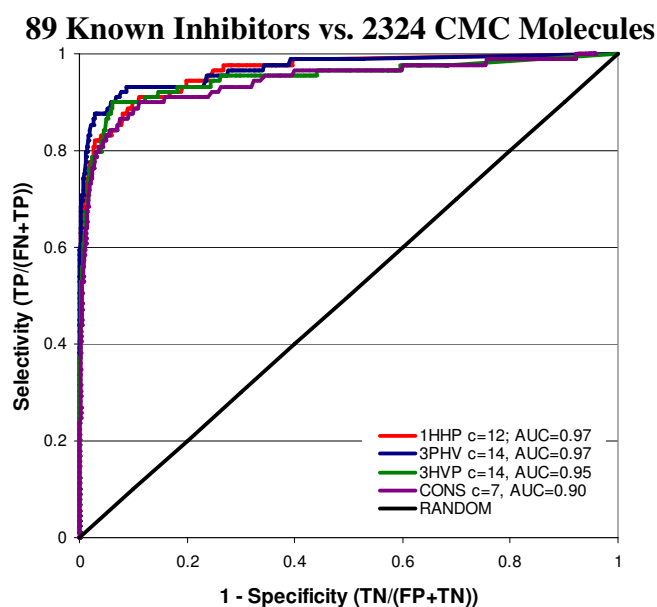
maximum cluster size cut-off of 65 is needed to produce a sphere set of 204. A large weight is then given if that particular weighted sphere is hit – even though the rest of the pose may not be ideal – causing a bias in the calculated score. Furthermore, the three smaller models (1HHP, 3HVP, and 3PHV) showed optimal performance with sphere sets ranging in size from 97-128. Unfortunately when working with the large CONS model, we were not able to even obtain sphere sets in that range; a maximum cluster size of 100 still produced a sphere set of 183. This is similar to what we observed in creating the original pharmacophore models. An optimal number of snapshots were necessary to create pharmacophore models with meaningful elements. Including too few snapshots did not provide sufficient consensus, and when employing too many snapshots, the information became too cluttered to interpret. To overcome this potential sphere size limitation, we suggest that the optimal aggregate sphere sets should contain approximately 100 spheres.

Receiver operator characteristic (ROC) curves^{30,31} based on the rank score were used to quantify the performance of the optimal aggregate sphere sets for all HIV-1p systems using a database of 89 known HIV-1p inhibitors versus a decoy dataset containing 2324 drug-like ligands. Given two populations - known HIV-1p inhibitors and known false positives - and their ranks according to our computational model, we can examine how well our ranking the model is able to discriminate between them. The selectivity and specificity of the model can be calculated from the number of true and false positives and negatives. These metrics are able to assess how well our ranking function is identifying known inhibitors as high rankers and how well it is able to discriminate them from inactive ligands – plotting true positives against false positives. The perfect computational ranking method would give complete separation between the active and inactive compounds; however, in practice these populations often overlap. Thus, a perfect model would lie at to point (0, 1) while an indiscriminate model would lie along the line with slope equal to 1.

For every possible threshold, the selectivity is plotted against 1 minus the specificity to give a ROC plot comparing the effect of the scoring weights on discriminatory ability. The area under the curve (AUC) is an unbiased way to compare the performance of the different models using the same test data. If the AUC was equal to 0.50, it would be equivalent to a known inhibitor being ranked higher than a decoy molecule 8 out of 10 times.³¹

Excellent discrimination was found for all HIV-1p models, as demonstrated by the ROC plot shown in Figure 5.5. In this study, all HIV-1p models produced an AUC value of 0.90 or greater. In fact for three of the four models, the AUC value was ≥ 0.95 . Discriminating between two similar populations is very challenging, and we were pleased to see this enrichment. It is also of particular interest that the ROC curves are very steep at the beginning, indicating the potential for virtual screening applications. When screening large databases of compounds, the number of true positives can be sacrificed to reduce the amount of false positives.

Figure 5.5. A representative ROC plot showing enrichments for discriminating a set of 89 known HIV-1p inhibitors from a set of 2324 general decoys. Compared are the enrichment profiles obtained using the four models- 1HHP, 3HVP, 3PHV, and CONS. The selectivity is plotted against 1 minus the specificity for each threshold value evaluated. TN - true negatives, FP - false positives, TP - true positives, FN - false negatives.

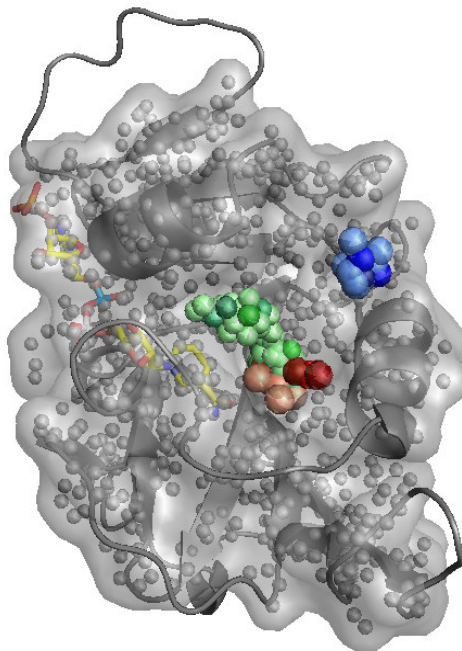


Nonetheless, we see excellent discrimination between the dataset of known inhibitors and the more general decoy dataset, as demonstrated by the ROC plot shown in Figure 5.5B. In this study, all HIV-1p models have an AUC value of 0.90 or greater. It is also of particular interest that the ROC curves are very steep at the beginning, indicating the potential for virtual screening applications. When screening large databases of compounds, the number of true positives can be sacrificed to reduce the amount of false positives.

Ranking Function Consistency: Application to DHFR

To fully demonstrate the robustness of the MPS-DOCK method, we have applied it to an additional well-studied system, ecDHFR. The active site of ecDHFR is much smaller than that of HIV-1p, and consequently, the original ecDHFR MPS pharmacophore model had fewer elements: 2 aromatic sites, 2 hydrogen-bond donor sites, and 1 hydrogen-bond acceptor site. As we found that HIV-1p sphere sets in the range of 97-128 demonstrated optimal performance, we chose to probe clustered aggregate sphere sets ranging from 91 – 127 spheres (c equal to 12 - 6) for ecDHFR. Figure 5.6 shows the sphere set containing 91 spheres in the active site of ecDHFR (1RX1¹⁴) defined by 818 excluded volumes (RMSD cut-off = 0.75 Å). The cofactor, nicotinamide adenine dinucleotide phosphate (NADPH), is also provided to orient the reader. Again, the sphere set is colored by chemical functionality and shaded according to weight (i.e. number of spheres making up the cluster).

Figure 5.6. The clustered aggregate sphere representation of DHFR. Atomic spheres are colored according to chemical functionality (red, hydrogen-bond donating; blue, hydrogen-bond accepting; green, aromatic) and also shaded by weight- the greater the number of spheres making up the cluster, the darker the color. 818 excluded volume spheres are shown in grey along with a surface representation overlaid with a cartoon depiction of the entire protein. The cofactor, NADPH, is colored by atom type (carbon is shown in yellow).

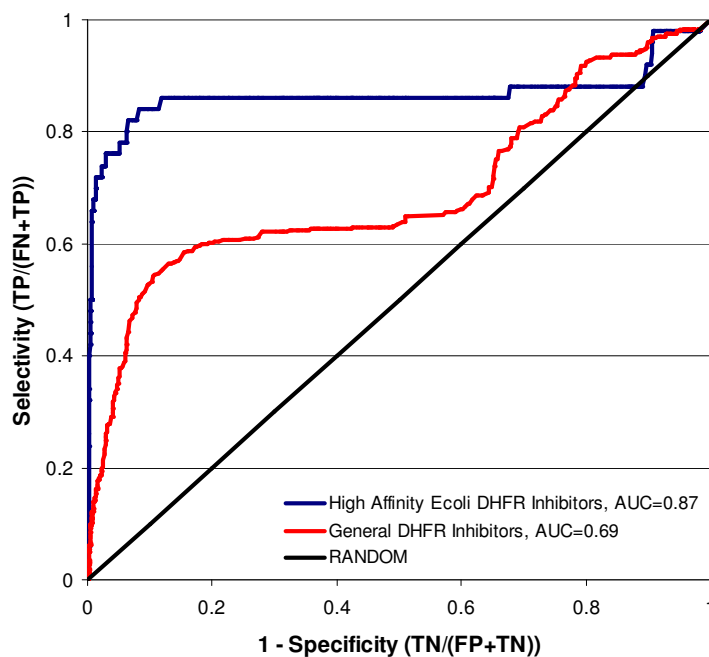


To investigate the performance of the ecDHFR aggregate sphere sets, three databases were screened: one consisting of 50 high affinity ecDHFR inhibitors, another of 541 inhibitors from various DHFR species, and a final dataset of 2326 general drug-like decoys^{13,24}. The optimal MPS-DOCK parameters for the HIV-1p test case were used, $4m - 3d$, $0.25t$ and an excluded volume penalty of 10. We found that $4m - 3d$ was too stringent; only a small percentage of the compounds in the DHFR databases were able to meet this criteria. However when we employed a more lenient parameter set, $4m - 2d$ (4 nodes matched with a distance of 2 Å between them rather than 3 Å), the compounds were subsequently ranked. This is not surprising because the DHFR active site is much smaller than that of HIV-1p, and consequently, the DHFR inhibitors have much fewer atoms. HIV-1p inhibitors are typically large, flexible peptide-like molecules.

We found that the aggregate sphere sets ranging from 91 – 127 spheres performed quite similarly when ranking the two known inhibitor databases versus the non-inhibitor

dataset. However, the clustered sphere sets with less than 100 spheres did show modest improvement, and a representative ROC plot is provided in Figure 5.7 ($c = 12$, 91 spheres). We were pleased to see the excellent discrimination between the dataset of high affinity ecDHFR inhibitor and the more general decoy dataset, as demonstrated by the AUC value of 0.87. While moderate enrichment emerged from our ecDHFR model when screening the general DHFR dataset (AUC = 0.69), it demonstrates superior performance at discriminating the high affinity ecDHFR inhibitors from the decoy ligands (AUC = 0.87). This supports a recent study by our laboratory that showed the ability of the MPS models to successfully differentiate species-specific inhibitors of DHFR²⁴, and we were very excited to see the same trend using our new pharmacophore representation and ranking method. The AUC provides a metric to compare the performance of the high affinity ecDHFR dataset versus the general DHFR database, and hence the specifies differentiating capabilities of our ranking function.

Figure 5.7. A representative ROC plot showing enrichments obtained for the DHFR aggregate sphere model for discriminating a set of 50 high affinity ecDHFR inhibitors from a set of 2326 general decoys (blue) and a set of 541 general DHFR inhibitors from a set of 2326 general decoys (red). The selectivity is plotted against 1 minus the specificity for each threshold value evaluated. TN - true negatives, FP - false positives, TP - true positives, FN - false negatives.



5.4 Conclusions

The MPS pharmacophore models have been represented as a set of atomic spheres which provides a contour-based representation of the pharmacophore models and also allows ligand orientations to be ranked based on the number of aggregate spheres complemented. We have demonstrated that this approach is successful at identifying known HIV-1p inhibitors seeded in large data set, simulating the situation commonly encountered in virtual screening. Our results compare favorably to those reported by Cummings et al. with the same data set – highlighting the importance of incorporating protein flexibility into SBDD. Additionally, we are able to demonstrate the robustness of our method through an application to ecDHFR. Our approach is both successful at discriminating known inhibitors from general drug-like ligands and discriminating between species-specific inhibitors. The MPS-DOCK technique allows for ranking of ligand poses while still accounting for protein flexibility. This ranking function is complementary to our previous use of the MPS pharmacophore models which enables fast database searching and facile expansion into new chemical space.

We are continuing to develop and improve the MPS pharmacophore method and demonstrate the utility of the simple ranking metric. In the future, we would like to investigate the addition of penalties for a ligand heavy atom not overlapping with a scoring sphere or possessing a dissimilar chemistry. An additional DOCK feature that could be utilized is the critical points filter where a requirement is made that a certain sphere be used in the ligand orientation. A similar technique would be to give extra weight to certain spheres in locations that are known to be critical in ligand binding.

This work has been published as:

Damm, K.L., Meagher, K.L., and Carlson, H.A. Accounting for Multiple Protein Conformations in Ranking Ligand Databases. *Manuscript in preparation.*

5.5 References

1. Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. A Geometric Approach to Macromolecule-Ligand Interactions. *J Mol Biol* **1982**, *161*, 269-288.
2. Shoichet, B. K.; Kuntz, I. D. Matching Chemistry and Shape in Molecular Docking. *Protein Eng* **1993**, *6*, 723-732.
3. Moitessier, N.; Henry, C.; Maigret, B.; Chapleur, Y. Combining Pharmacophore Search, Automated Docking, and Molecular Dynamics Simulations as a Novel Strategy for Flexible Docking. Proof of Concept: Docking of Arginine-Glycine-Aspartic Acid-Like Compounds into the Alphavbeta3 Binding Site. *J Med Chem* **2004**, *47*, 4178-4187.
4. Joseph-McCarthy, D.; Alvarez, J. C. Automated Generation of Mcss-Derived Pharmacophoric Dock Site Points for Searching Multiconformation Databases. *Proteins* **2003**, *51*, 189-202.
5. Joseph-McCarthy, D.; Thomas, B. E. t.; Belmarsh, M.; Moustakas, D.; Alvarez, J. C. Pharmacophore-Based Molecular Docking to Account for Ligand Flexibility. *Proteins* **2003**, *51*, 172-188.
6. Wang, K.; Murcia, M.; Constans, P.; Perez, C.; Ortiz, A. R. Gaussian Mapping of Chemical Fragments in Ligand Binding Sites. *J Comput Aided Mol Des* **2004**, *18*, 101-118.
7. Schnecke, V.; Kuhn, L. A., Virtual Screening with Solvation and Ligand-Induced Complementarity. *Perspectives in Drug Discovery and Design* 2000, *20*, 171-190.
8. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res* **2000**, *28*, 235-242.
9. Spinelli, S.; Liu, Q. Z.; Alzari, P. M.; Hirel, P. H.; Poljak, R. J. The Three-Dimensional Structure of the Aspartyl Protease from the Hiv-1 Isolate Bru. *Biochimie* **1991**, *73*, 1391-1396.
10. Wlodawer, A.; Miller, M.; Jaskolski, M.; Sathyanarayana, B. K.; Baldwin, E.; Weber, I. T.; Selk, L. M.; Clawson, L.; Schneider, J.; Kent, S. B. Conserved Folding in Retroviral Proteases: Crystal Structure of a Synthetic Hiv-1 Protease. *Science* **1989**, *245*, 616-621.
11. Lapatto, R.; Blundell, T.; Hemmings, A.; Overington, J.; Wilderspin, A.; Wood, S.; Merson, J. R.; Whittle, P. J.; Danley, D. E.; Geoghegan, K. F.; et al. X-Ray Analysis of Hiv-1 Proteinase at 2.7 a Resolution Confirms Structural Homology among Retroviral Enzymes. *Nature* **1989**, *342*, 299-302.

12. Meagher, K. L.; Lerner, M. G.; Carlson, H. A. Refining the Multiple Protein Structure Pharmacophore Method: Consistency across Three Independent Hiv-1 Protease Models. *J Med Chem* **2006**, *49*, 3478-3484.
13. Lerner, M. G.; Bowman, A. L.; Carlson, H. A. Incorporating dynamics in E. coli dihydrofolate reductase enhances structure-based drug discovery. *J. Chem. Info. Model.*, *submitted*.
14. Sawaya, M. R.; Kraut, J. Loop and Subdomain Movements in the Mechanism of Escherichia Coli Dihydrofolate Reductase: Crystallographic Evidence. *Biochemistry* **1997**, *36*, 586-603.
15. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A., A Second Generation Force-Field for the Simulation of Proteins, Nucleic-Acids, and Organic-Molecules. *J Am Chem Soc* **1995**, *117*, 5179-5197.
16. Case, D. A. P., D. A.; Caldwell, J. W.; Cheatham III, T. E.; Ross, W. S.; Simmerling, C. L.; Darden, T. A.; Merz, K. M.; Stanton, R. V.; Cheng, A. L.; Vincent, J. J.; Crowley, M.; Tsui, V.; Radmer, R. J.; Duan, Y.; Pitera, J.; Massova, I.; Seibel, G. L.; Singh, U. C.; Weiner, P. K.; Kollman, P. A. Amber6, University of California San Francisco: San Francisco, CA, 1996.
17. Carlson, H. A.; Masukawa, K. M.; McCammon, J. A., Method for Including the Dynamic Fluctuations of a Protein in Computer-Aided Drug Design. *J Phys Chem A* **1999**, *103*, 10213-10219.
18. Jorgensen, W. L.; Maxwell, D. S.; TiradoRives, J., Development and Testing of the Opls All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J Am Chem Soc* **1996**, *118*, 11225-11236.
19. Jorgensen, W. L. Boss, 4.2; Yale University: New Haven, CT, 2000.
20. Damm, K. L.; Carlson, H. A. Gaussian-Weighted Rmsd Superposition of Proteins: A Structural Comparison for Flexible Proteins and Predicted Protein Structures. *Biophys J* **2006**, *90*, 4558-4573.
21. Comprehensive Medicinal Chemistry; Hansch, C., Sammes, P. G., Taylor, J. B., Eds.; Pergamon Press: Oxford, 1990; Vol. 1-6.
22. Comprehensive Medicinal Chemistry Database; MDL Information Systems, Inc.: San Leandro, CA, 2003.
23. Meagher, K. L.; Carlson, H. A. Incorporating Protein Flexibility in Structure-Based Drug Discovery: Using Hiv-1 Protease as a Test Case. *J Am Chem Soc* **2004**, *126*, 13276-13281.

24. Bowman, A. L.; Lerner, M. G.; Carlson, H. A. Protein Flexibility and Species Specificity in Structure-Based Drug Discovery: Dihydrofolate Reductase as a Test System. *J Am Chem Soc* **2007**, *129*, 3634-3640.
25. Cummings, M. D.; DesJarlais, R. L.; Gibbs, A. C.; Mohan, V.; Jaeger, E. P. Comparison of Automated Docking Programs as Virtual Screening Tools. *J Med Chem* **2005**, *48*, 962-976.
26. Omega, 1.8.1; OpenEye Scientific Software: Santa Fe, New Mexico, 2004.
27. Lam, P. Y.; Jadhav, P. K.; Eyermann, C. J.; Hodge, C. N.; Ru, Y.; Bacheler, L. T.; Meek, J. L.; Otto, M. J.; Rayner, M. M.; Wong, Y. N.; et al. Rational Design of Potent, Bioavailable, Nonpeptide Cyclic Ureas as Hiv Protease Inhibitors. *Science* **1994**, *263*, 380-384.
28. Knegt, R. M.; Kuntz, I. D.; Oshiro, C. M. Molecular Docking to Ensembles of Protein Structures. *J Mol Biol* **1997**, *266*, 424-440.
29. Damm, K. L.; Carlson, H. A. Exploring Experimental Sources of Multiple Protein Conformations in Structure-Based Drug Design. *J Am Chem Soc* **2007**, *129*, 8225-8235.
30. Swets, J. A.; Dawes, R. M.; Monahan, J., Better Decisions through Science. *Scientific American* **2000**, *283*, 82-87.
31. Triballeau, N.; Acher, F.; Brabet, I.; Pin, J. P.; Bertrand, H. O. Virtual Screening Workflow Development Guided by the "Receiver Operating Characteristic" Curve Approach. Application to High-Throughput Docking on Metabotropic Glutamate Receptor Subtype 4. *J Med Chem* **2005**, *48*, 2534-2547.

CHAPTER 6

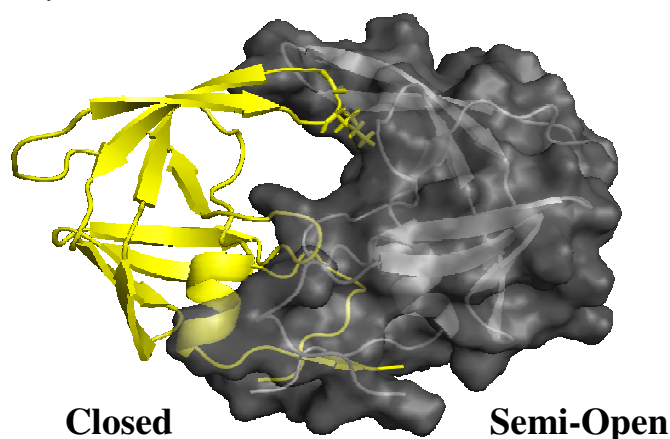
Inhibition of HIV-1p By Modulating its Conformational Behavior of the Flap Region

6.1 Introduction

The effectiveness and robustness of the MPS method has been adequately demonstrated¹⁻⁴, and we have begun to apply our technique to the discovery of novel inhibitors of HIV-1p. Targeting conserved residues that are essential for the activity and/or structural stabilization of HIV-1p is a potential approach to overcome the resistance associated with current PIs and can be exploited in the design of novel inhibitors.⁵

Here we present a novel mode of action for HIV-1p inhibitors – modulating the conformation behavior of HIV-1p by targeting the flap-recognition site. This region can also be thought of as the “eye” from recent HIV-1p naming convention (i.e., the front view of the protease resembles the face of a bulldog). Upon substrate binding, each flap closes down and positions itself in the highly conserved “eye” region of the opposite-side monomer. In Figure 6.1, the “eye” region of a monomer in the semi-open form (dark grey in both cartoon and surface representation) is shown occupied by the flap tip (residues 49-52) of the opposite monomer in a closed conformation (yellow in cartoon representation).

Figure 6.1. When a monomer closes, it places its flap tips against the “eyebrow” region of the other monomer. The right monomer is the apo, semi-open state and shown with a grey surface. The left monomer is in the bound, closed state and colored yellow. Ile 50 and Gly 51 are shown in stick representation in direct contact with the “eye”.



Targeting the “eye” has interesting ramifications for both the closed and open states. If the flaps cannot properly close and coordinate the central water molecule, the catalytic efficiency of HIV-1p drops.⁶⁻⁸ If the curling of the hydrophobic flap tips into the “eye” site drive the conformational change into the open state⁸⁻¹², blocking the interaction between residue I50 and the “turn” residues 79-81 may interfere with its ability to bind its large substrate. If either- or both - mechanisms are possible, an inhibitor bound to this site would alter the equilibrium of the system. Furthermore, it has been suggested that flap dynamics rather than sequence specificity plays a major role in the association and disassociation of substrates^{10,13}, and modulating the conformational behavior of the flaps may be a potential mechanism for eluding inhibitor resistance.

We show in this study that the addition of a small molecule into the flap-recognition pocket prevents the flap from assuming the proper closed conformation. Using solvent mapping of the binding site and the MPS method, we generated a receptor-based pharmacophore model that was screened against an in-house database of compounds. The chemical scaffold that best complemented the MPS model was chosen as a representative structure. We were able to demonstrate the stability of the bound complex through MD simulations of the complex in explicit solvent and multiple LD

simulations of the complex in implicit solvent. The MD simulation was run for 10 ns and five independent LD simulations were run for 5 ns each (total of 25-ns of simulation time). The inhibitory activity of our compound was subsequently confirmed through experimental testing.

6.2 Methods

Multiple Protein Structures Method

Previously, our group has conducted several 3-ns MD simulations of three unbound structures of HIV-1p (1HHP¹⁴, 3HVP¹⁵, and 3PHV¹⁶).¹ Each protein structure was solvated with explicit water and equilibrated using the AMBER94 force field¹⁷ and the AMBER6¹⁸ suite of programs. Counter ions were added to neutralize the systems. Multiple protein conformations were taken from those simulations after equilibration and every 600 ps along the 3-ns MD trajectory and used to generate a receptor-based pharmacophore model. Each monomer was considered a separate structure, resulting in a collection of 36 conformations from a total of 9-ns of simulation time. In our aforementioned work^{1,2}, we focused on mapping the bottom of the central cavity to describe the complementarity of competitive inhibitors. In fact, any features of the solvent mapping that were farther 9 Å from the catalytic acids were ignored.

The “eye” region of each structure was alternately flooded with 500 small molecule probes (benzene, ethane, and methanol) using a 14.5-Å radius flooding. Each structure was then subjected to a MUSIC simulation with the BOSS program¹⁹, using the OPLS force field²⁰, while the protein was held rigid. This resulted in clusters of small molecule probes at favorable interaction regions within the “eye” region of the protease. Probes were clustered using an in-house program based on Jarvis-Patrick methodology. If 8 probes appeared in a cluster, it was considered significant and included in the

consensus step below. For that step, each cluster was represented by the “parent”, the probe with the most favorable interactions with the protein.

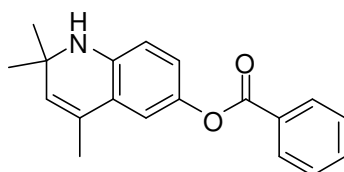
Similarly to Chapter 4, “consensus clusters” were determined by aligning the protein snapshots with the equilibrated 1HHP structure and looking for interactions (positions of parent probes) that were common over $\geq 50\%$ of the multiple protein conformations. Each consensus cluster was then represented in the pharmacophore model as a spherical element. The center of each element was defined by the average position of the probe molecules (benzene centroid, the midpoint of the carbon-carbon bond for ethane, and the oxygen atom of the methanol probe). The radius were based on the RMSD of the probe positions. Individual benzene clusters were labeled aromatic elements whereas ethane clusters were termed hydrophobic. Methanol clusters were classified as a hydrogen-bond donor or acceptor element, based on their interaction with the protein surface. This procedure resulted in a seven-site pharmacophore model of the eye region; the model coordinates are provided in Appendix 5. Full details of the MPS method can be found elsewhere.^{2,21}

Virtual Screening

The MPS pharmacophore model was screened against one of the databases (33,623 compounds) available from the University of Michigan’s Center for Chemical Genomics (CCG). The dataset was screened using the search option within the Pharmacophore Query Editor of MOE²². This is simply a fit/no-fit comparison based on the geometry of each conformer’s chemical features and the physical arrangement of the pharmacophore elements. Multiple conformations of each ligand were pre-generated using the default parameters of OMEGA²³ with the exception of the energy window and RMS threshold set to 14 kcal/mol and 1, respectively; the maximum number of conformations was 300. During the database search, the radii were multiplied by $1.3 \times \text{RMSD}$, and a compound was required to fulfill six of the seven features to count as a

hit; this produced small, tractable numbers of compounds that were strictly held to the pharmacophore features. The “eye” model identified 93 compounds within a 360 Da molecular weight filter. The predicted poses were manually viewed to ensure appropriate overlap with the pharmacophore model. Compound 1 (2,2,4-trimethyl-1,2-dihydroquinolin-6-yl benzoate), shown in Figure 6.2, was chosen as a model compound for the theoretical simulations because it best complemented the features of the model. The structures of all 93 identified compounds are provided in Appendix 5.

Figure 6.2. Compound 1 (2,2,4-trimethyl-1,2-dihydroquinolin-6-yl benzoate) was identified through a virtual screen and chosen for theoretical simulations.



Compound 1

Dynamics Simulations

Unrestrained, all-atom MD and LD simulations were conducted with AMBER8. The MD simulation used explicit solvation with TIP3P water³⁰ while aqueous solvation was implicitly modeled using the Generalized Born approach²⁴ for the LD simulations. Simulations were initiated from the crystallographic coordinates of the apo monomer of HIV-1p obtained from the PDB²⁵ (PDB ID: 1HHP¹⁴), and the homodimer was generated using C₂ symmetry operations in PyMOL²⁶. Hydrogens were added via the tLEaP module in AMBER8¹⁸ using the FF99SB force field²⁷. MD and LD simulations were performed using the 1HHP dimer in complex with Compound 1 in the “eye” site. The starting coordinates of Compound 1 were obtained from the binding pose generated in the MOE pharmacophore screen. The Antechamber module with the GAFF²⁸ force field and AM1-BCC charges²⁹ was used to determine force field parameters for Compound 1.

Explicit-solvent, MD simulations were performed for 10 ns using one random-number seed with HIV-1p in complex with Compound 1. The simulations were carried out using the FF99SB force field²⁷ and the sander module in the AMBER8 suite of programs¹⁸. The hydrogen atoms were first minimized, and then the system was solvated using truncated octahedral boundary conditions with TIP3P water molecules³⁰, a buffer distance of 12 Å, and closeness parameter of 0.5. The +4e charge of HIV-1p was neutralized by the addition of 4 chloride counter ions placed 10 Å from the protein surface in the most electropositive regions. The simulation was run in the NPT ensemble, and SHAKE was used to constrain all bonds to hydrogen atoms. A 2 fs time step was used, along with a 10-Å cutoff for nonbonded interactions and particle mesh Ewald (PME) for long-range electrostatics. For the solvated system, the hydrogen atoms were first minimized, followed by side chains, and lastly all atoms. The system was equilibrated in a series of four stages: a gradual heating of water from 10 to 310 K over 50 ps, followed by water equilibration with protein restrained for 250 ps at 310K, then a full system heating from 10 to 310 K over 50 ps, and finally, full system equilibration with the protein unrestrained at 310K for 250 ps. The production phase was run for 10 ns at 310 K.

Five independent LD simulations of HIV-1p in complex with Compound 1 were run for 5 ns starting from different random-number seeds using the FF99SB force field²⁷ and the sander module in the AMBER8 suite of programs¹⁸. Each simulation was run in the NPT ensemble using a 999 Å cutoff for nonbonded interactions, and a generalized Born solvation model²⁴ was employed. Default dielectric values were used: interior = 1 and exterior = 78.5. The hydrogen atoms were first minimized, followed by a minimization of all atoms. The system was equilibrated over a series of six steps; the first three equilibration steps were each performed for ten ps, steps four and five over 50 ps, and the sixth step for 100 ps. During the first two equilibration steps, the system was gradually heated from 100 K to 300 K and remained at 300 K for the subsequent steps.

Restraints were placed on all heavy atoms and gradually removed over the first four equilibration steps using force constants from 2.0 to 0.1 kcal/mol $\cdot\text{\AA}^2$. Throughout the fifth equilibration step, the backbone atoms remained restrained with a force constant of 0.1 kcal/mol $\cdot\text{\AA}^2$. In the sixth, and final, phase of equilibration, all force restraints were removed, and the system was run for 100 ps at 300 K. The subsequent production phase was run for 5 ns. A time step of 1 fs and 1 ps⁻¹ collision frequency were used, and SHAKE was employed to constraint hydrogens. This protocol is based on that used by Simmerling and coworkers for HIV-1p.^{31,32}

Analyses of the flap conformation and ligand position were performed using the Ptraj module from the AMBER8 suite of programs¹⁸. The MD trajectory was aligned to its average structure across the 10-ns simulation. However, for LD each trajectory was aligned to the fully minimized 1HHP, the last common structure between the simulations. RMSD traces were calculated for Compound 1 versus its initial position within the eye region. The following metrics were used to quantify the movement of the flaps: the distance between flap tips and catalytic acid (I50 to D25 and I50' to D25') and the distance between flap tips and eye pocket (G51 to T80 and G51' to T80'). C α atoms were used to measure distance and angles between residues.

Ruling out the “Elbow Region”

An additional site that might be appropriate for a small, hydrophobic molecule like Compound 1 to bind is the elbow region (residues 35/35'-42/42'). To rule out this site, we conducted five independent MD simulations of the 1HHP dimer in complex with Compound 1 in the elbow site. The explicit-solvent MD protocol described above was followed. AutoDock 3³³ was used to predict the binding pose of Compound 1 in the elbow region of 1HHP using the Lamarckian genetic algorithm. Polar hydrogens were added to both the protein and ligand while Kollman charges were assigned to the protein and Gasteiger charges and rotatable bonds to the ligand using AutoDockTools. Grids

encompassing the elbow region were calculated using 0.302 Å spacing with AutoGrid 3. Default docking parameters were employed with the exception of $ga_pop_size = 100$, $ga_num_evals = 750,000$, and $ga_run = 50$.

HIV-1 Protease Inhibition Assay

A FRET-based assay was available, but unfortunately Compound 1 was auto-fluorescent and could not be assayed. However, an analog of Compound 1 (Compound 2: 2,2,4-trimethyl-1,2-dihydroquinolin-6-yl 4-methoxybenzoate) was available in-house to experimentally test the chemical class. The employed assay is based on the previously described procedure for HIV-1p.^{34,35} The substrate used in the assay is an oligopeptide, RE(EDANS)SQNYPIVQK(dabcyl)R, purchased from Molecular Probes (Cat. No. H-2930)^{34,36}. All compounds were purchased from Chembridge; Compound 1 and Compound 2 are listed as catalog numbers 5493403 and 5303843, respectively. HIV-1p was purchased from Bachem Biosciences (Product H-9040). Pepstatin A was purchased from USB (lot #110018) and employed as a control.

Three fluorimetric assays were performed, in triplicate, in 384 well plates (Corning No. 3676) and read using a SpectraMax M5 from Molecular Devices. The excitation/emission wavelengths of the substrate are 340/490 nm and employed a cutoff filter at 475 nm. PEG-400 was diluted in Buffer A (20mM phosphate, 1 mM DTT, 1 mM EDTA, 20% glycerol and 0.1% CHAPS at pH 5.1), and 1 μ L was added to each well (PEG-400 final concentration, 0.1%) to counter HIV-1p precipitation. Each compound was diluted in water, and 2 μ L was added to each well (final concentration range of 1-78 μ M), followed by 5 μ L of the protease, diluted in Buffer A (final concentration of 30 nM). After 45 min of incubation at room temperature, 12 μ L of substrate (diluted in Buffer A, final concentration 2 μ M) was introduced to initiate the assay, and the fluorescence monitored for 5 min. The inhibition constant, K_i , was determined from a dose-response curve and the Cheng-Prusoff equation $K_i = IC_{50}/(1 + [S]/K_M)$, where [S] is

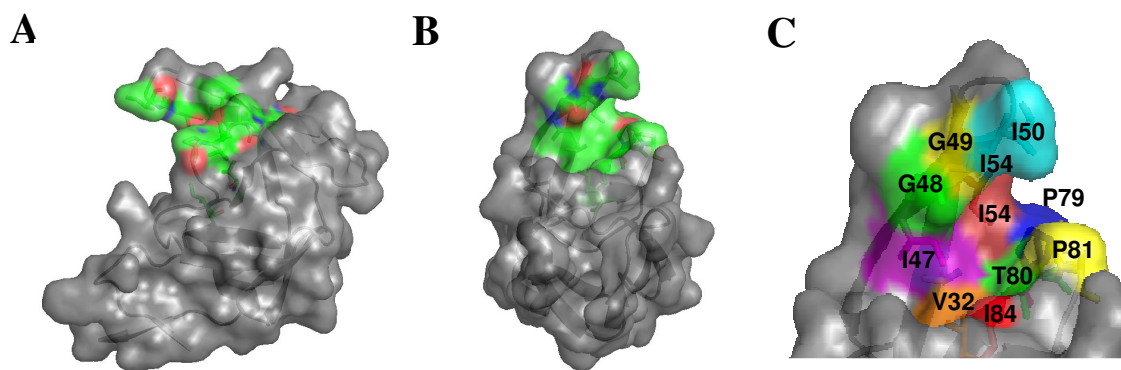
the substrate concentration.³⁷ The Michaelis constant K_M of the substrate was previously determined by Matayoshi et al. at $103 \pm 8 \mu\text{M}$.³⁴

6.3 Results and Discussion

Defining the Flap-Recognition Pocket

The flap-recognition pocket is only accessible in the semi-open and open conformations. Upon ligand binding the flap of the opposite monomer closes down and fills this site. The lower portion of the pocket is defined by I84, V32, P81, T80, P79, and G78 while the upper portion is defined by V56, I54, I47, G48, G49, and I50 (Figure 6.3C). More distal contacts may be possible with V82 and the backbone atoms of V77, L33, and K55. The large degree of green surface in Figure 6.3A,B indicates the hydrophobic character of the binding pocket.

Figure 6.3. The semi-open monomer is shown with the new site color-coded by atom type (green are carbons, red are oxygens, blue are nitrogens). **(A)** Front view. **(B)** 90 degree rotation. **(C)** The individual residues within the new site are each colored individually and labeled to show their placement within the cleft. G78 and V56 are not visible in this view.



Six of the twelve residues that define the flap-recognition pocket, G49, V56, G78, P79, T80, and P81, are highly conserved.^{38,39} A study by Foulkes et al. showed that mutations to the invariant residue T80 are detrimental to enzyme activity and

hypothesized that this may be due to alterations in the flexibility of the flap region.⁴⁰ The additional six residues, V32, I47, G48, I50, I54, and I84, are known to mutate to residues that confer drug resistance to protease inhibitors^{38,39}; the common mutations are provided in Table 6.1. Four of the six drug-resistant variants, V32I, I47V/A, G48V, and I50V/L, maintain their nonpolar nature in the mutant form but alter the size of the side chain. The additional two, I54 and I84, have been shown to mutate to a variety of residues, although the most common mutations are also hydrophobic, I54V and I84V/A.

Table 6.1. List of residues defining the flap-recognition pocket. Those in bold can mutate to residues that contribute to drug resistance.^{38,39}

Position #	Wild-Type Residue	Common Mutations
32	V	I
47	I	V, A
48	G	V
49	G	-
50	I	V, L
54	I	V, M, T, L, A, S
56	V	-
78	G	-
79	P	-
80	T	-
81	P	-
84	I	V, A, C

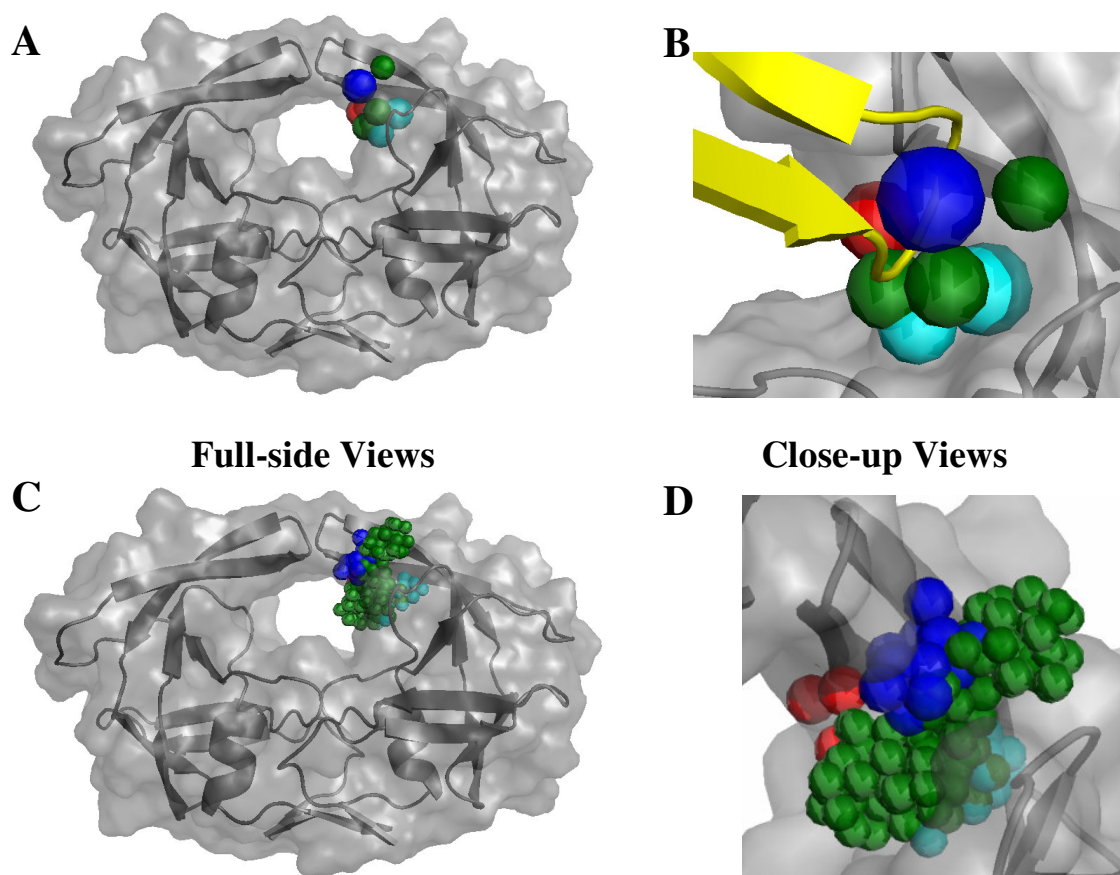
Virtual Screen using MPS Pharmacophore Model

An MPS pharmacophore model was generated to map the chemical character of the flap-recognition pocket in the semi-open conformation; it is shown in Figure 6.4A and B. The pharmacophore model has 7 sites: a hydrogen-bond donor element near the backbone carbonyl oxygen of G48, a hydrogen-bond acceptor element close to the backbone amine of I50, and three aromatic and two hydrophobic features that complement the hydrophobic nature of the cleft. If we compare the pharmacophore elements to the chemical features of the opposite side monomer in the closed state, the hydrogen-bond acceptor element perfectly reproduces the position of the backbone amide

of flap tip residue G51 (Figure 6.4B). Ethanes were seen to map out the bottom of the pocket where the side chain of I50 is known to occupy, but benzenes were too large showing that the interaction is more aliphatic than aromatic in nature.

In Chapter 5, an atomistic pharmacophore representation was developed that more specifically maps the contours of the interaction surface between each individual probe and protein. This technique was employed to provide further elucidation into the subtle contours of the “eye” site; the atomistic representation is shown in Figures 6.4C and D. However, this model was not used to predict novel compounds; it was generated solely to offer additional information about the interactions between the probe molecules and protein.

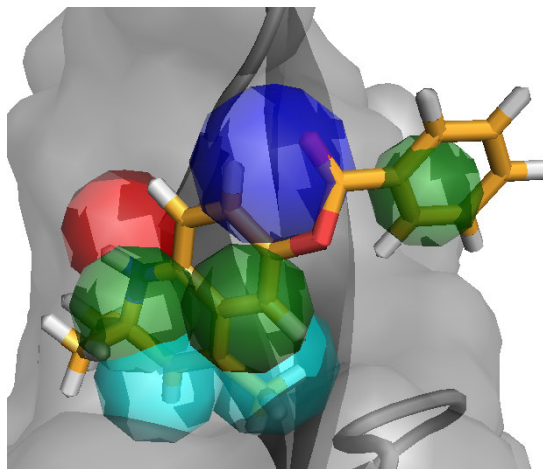
Figure 6.4. Two representative pharmacophore models are shown. Both are derived from the same solvent-mapping data. Elements are color-coded according to chemical functionality: red, hydrogen-bond donor; blue, hydrogen-bond acceptor; cyan, hydrophobic; green, aromatic. (A) Isotropic MPS Pharmacophore model (radii of $1.3 \times \text{RMSD}$) mapping the “eye” region of the semi-open conformation. (B) Close-up view of model and 90° rotation. Flap tip of the closed monomer is shown in yellow to demonstrate overlap with the pharmacophore model. (C) An atomistic representation of the solvent probes better shows the contour of the site. (D) Close-up view and 90° rotation.



The isotropic MPS model (Figure 6.4A and B) was screened against a subset of the CCG database using the Pharmacophore Query Editor option in MOE to predict compounds that could potentially complement the chemical features of the “eye” site. Using stringent searches, requiring 6 of 7 pharmacophore elements to be matched and radii equal to $1.3 \times \text{RMSD}$, 93 compounds were identified with a molecular weight of ≤ 360 Da. Each compound in its respective binding pose was visually inspected, and Compound 1 (2,2,4-trimethyl-1,2-dihydroquinolin-6-yl benzoate, Figure 6.2) was chosen as the representative compound because it best complemented the features of the model.

Shown in Figure 6.5 is the predicted binding pose for Compound 1 with the MPS pharmacophore model. The desired chemical features are present in the molecule and overlap well with their corresponding pharmacophore elements.

Figure 6.5. Compound 1, identified through the virtual screen, is shown overlaid with MPS pharmacophore model (radii of $1.3 \times \text{RMSD}$). The agreement between its chemical scaffold and the pharmacophore elements is demonstrated.



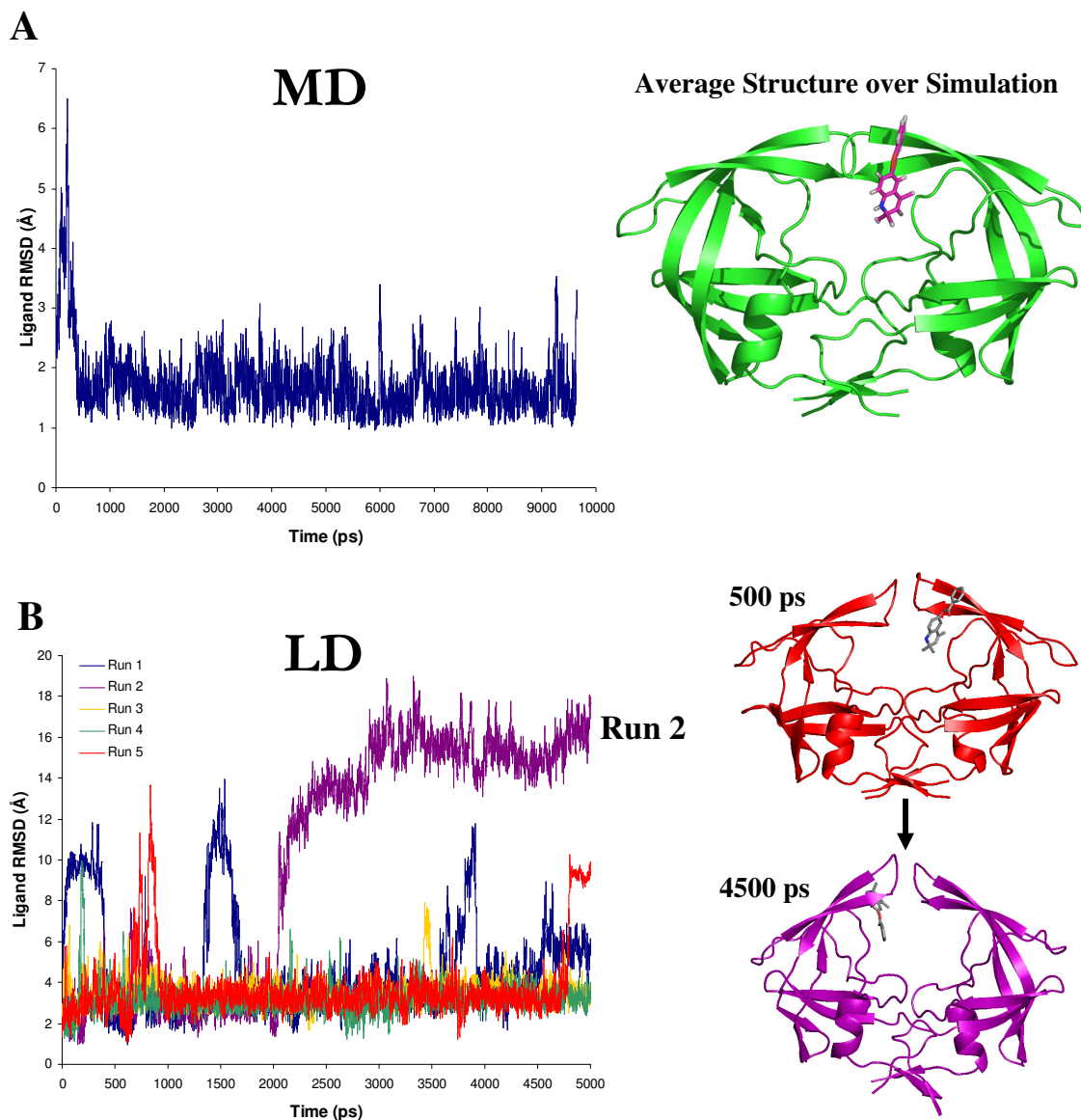
The molecules identified for binding in the eye pocket are significantly smaller than existing inhibitors; the molecular weight range for the protease inhibitors currently on the market is 505.2- 720.3 Da while the molecular weight of Compound 1 is 323.4 Da. Smaller molecules generally have better pharmacokinetic properties, and these entities could have a significant advantage in clinical use over existing HIV-1p inhibitors. Furthermore, both potential hydrogen bonds may be formed with backbone atoms of the protease which may be advantageous for overcoming potential escape mutants. In fact, the co-crystal structure of the recently approved nonpeptidic inhibitor, darunavir, demonstrated hydrogen-bonding between the bis-tetrahydrofuran oxygens and Asp 29 and Asp 30 backbone amides and between the aniline moiety and the carbonyl oxygen of Asp 30'.^{41,42} Experimental studies have shown that darunavir exhibits exceptional broad-spectrum activity against a large panel of MDR HIV-1 strains.⁴² Though these sites of

hydrogen bonding to the backbone are not the same residues as those in the eye site, it shows that targeting the backbone is a feasible way to counteract resistance mutations.

Ligand Behavior in the Dynamics Simulations

Analysis of the MD and LD simulations showed a stable trajectory. Compound 1 remained bound in the flap-recognition pocket in a stable fashion for the entire 10-ns MD simulation as characterized by the RMSD plot of Compound 1 (Figure 6.6A). At the beginning of the simulation, it did disassociate away from the protease and into the solvent, but after 250 ps, returned back to the eye pocket in its initial binding pose. After this point, a consistent binding pose throughout the trajectory was maintained; the average RMSD was $1.71 \pm 0.53 \text{ \AA}$ (the average position of Compound 1 over the MD simulation was used as a reference state). The 1,2-dihydroquinoline core remained in a stable position while the benzoate moiety continuously fluctuated throughout the simulation. Qualitatively, it appears that Compound 1 maintains a stable binding pose by interacting with the protease through van der Waals contacts. Additionally, there may be a favorable interaction between the backbone amide of I50 and π -electrons of the 1,2-dihydroquinoline core.

Figure 6.6. Ligand RMSD Plots. **(A)** Ligand RMSD values during the 10-ns MD simulation. Compound 1 is compared to its average position over the MD simulation. The average HIV-1p structure (green) and position of Compound 1 (pink) were calculated using data across the entire 10 ns. **(B)** Ligand RMSD values during the 5-ns LD simulations (5 random seeds). Compound 1 is compared to its minimized pose. In Run 2, Compound 1 starts in the flap-recognition pocket of Monomer A (red structure) but dissociates into the active center and binds in the opposite side pocket of Monomer B (purple structure). Several events are seen where the ligand dissociates and rebinds again in the same pocket (spikes up to 10/12 Å, which decrease again).



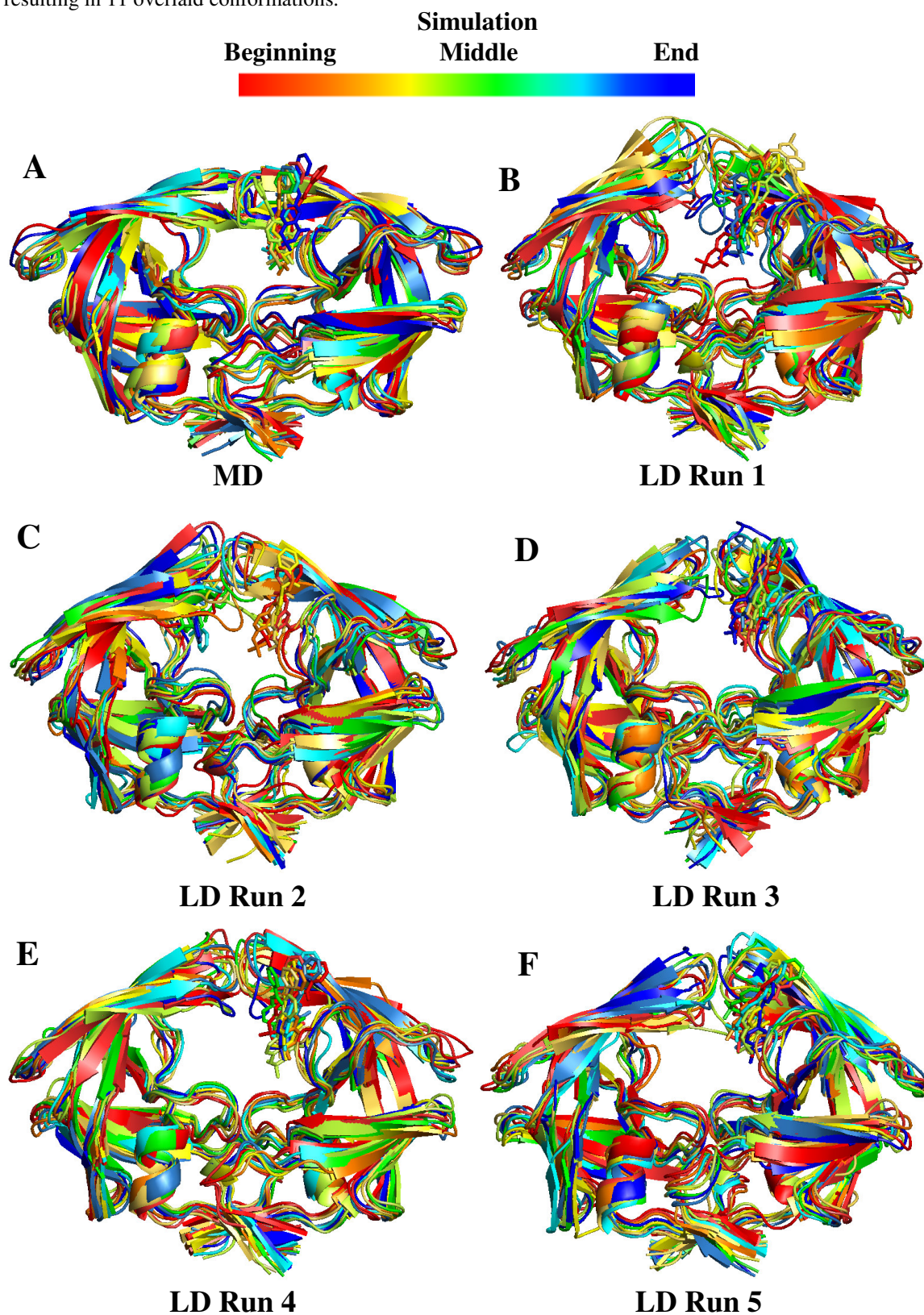
Five LD simulations were also performed for 5 ns each because LD samples greater conformational space than MD. During four of the five simulations, Compound 1 continued to interact in a stable fashion with the residues of the flap-recognition pocket as

demonstrated by Figure 6.6B. However, similar to the beginning of the MD simulation, Compound 1 was seen to dissociate temporarily and then rebind to the eye site. For example during Run 1 (blue trajectory) at ~1.3 ns, Compound 1 disassociated from the eye pocket into the solvent but returned to its original binding pose after 500 ps. However, Compound 1's behavior during Run 2 (purple trajectory) was the most intriguing. In Figure 6.6B, 1HHP colored in red demonstrates the initial binding pose (shown in grey) predicted from the MOE virtual screen using the MPS pharmacophore model. At ~ 2 ns, Compound 1 disassociated from the pocket and interacted with the flap tips at the center of the active site. After another ns, it entered the flap-recognition pocket of the opposite side monomer and assumed the initial binding pose! However, after ~100ps, Compound 1 flipped 180° and maintained this pose for the duration of the simulation (Figure 6.6B, 1HHP colored in purple). Docking studies with AutoDock 3 also predicted both binding modes (data not shown); hence, it may be possible for Compound 1 to adopt both poses.

Protein Behavior in the Dynamics Simulations

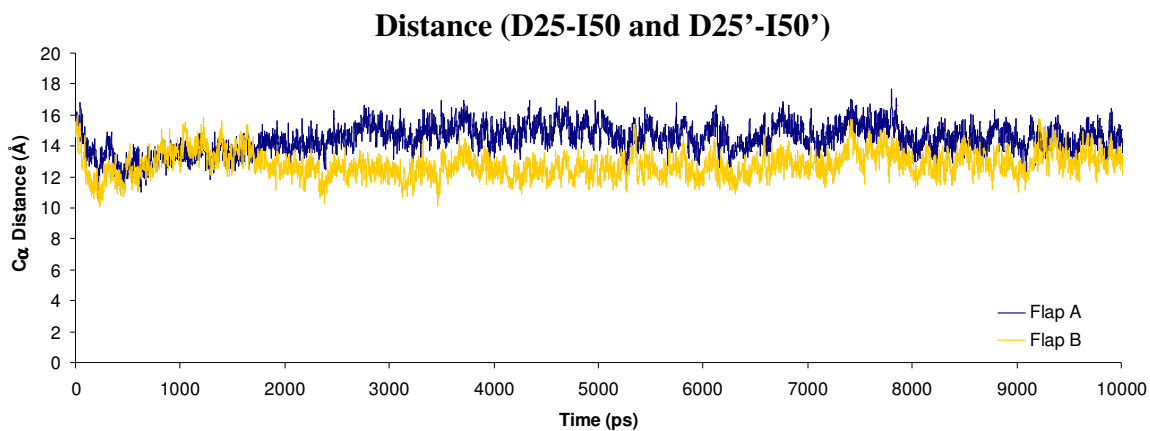
The effect of bound Compound 1 on the dynamics of the protease was also characterized. In Figure 6.7, an overlay of 11 snapshots taken at uniform time points across the trajectory for each MD and LD simulation demonstrates the conformational space sampled. The protease assumed a closed conformation throughout the MD simulation (Figure 6.7A). The LD simulations provided greater conformational variation as was expected. Although the protease generally remained in a semi-open state during LD, the closed conformation as also sampled as demonstrated in Figure 6.7B-F. It is very encouraging to see that Compound 1 can complement the eye site in many conformations. This entropic benefit is likely the result of our MPS models incorporating the behavior of the protein across an ensemble of conformational states.

Figure 6.7. Overlay of snapshots across dynamics simulations. The conformations are colored in order of time reference across the simulations (MD: 0-10 ns, LD: 0-5ns). (A) MD simulation, snapshot taken every 1 ns, resulting in 11 overlaid conformations. (B-F) LD Run 1-5, respectively. Snapshot taken every 0.5 ns, resulting in 11 overlaid conformations.



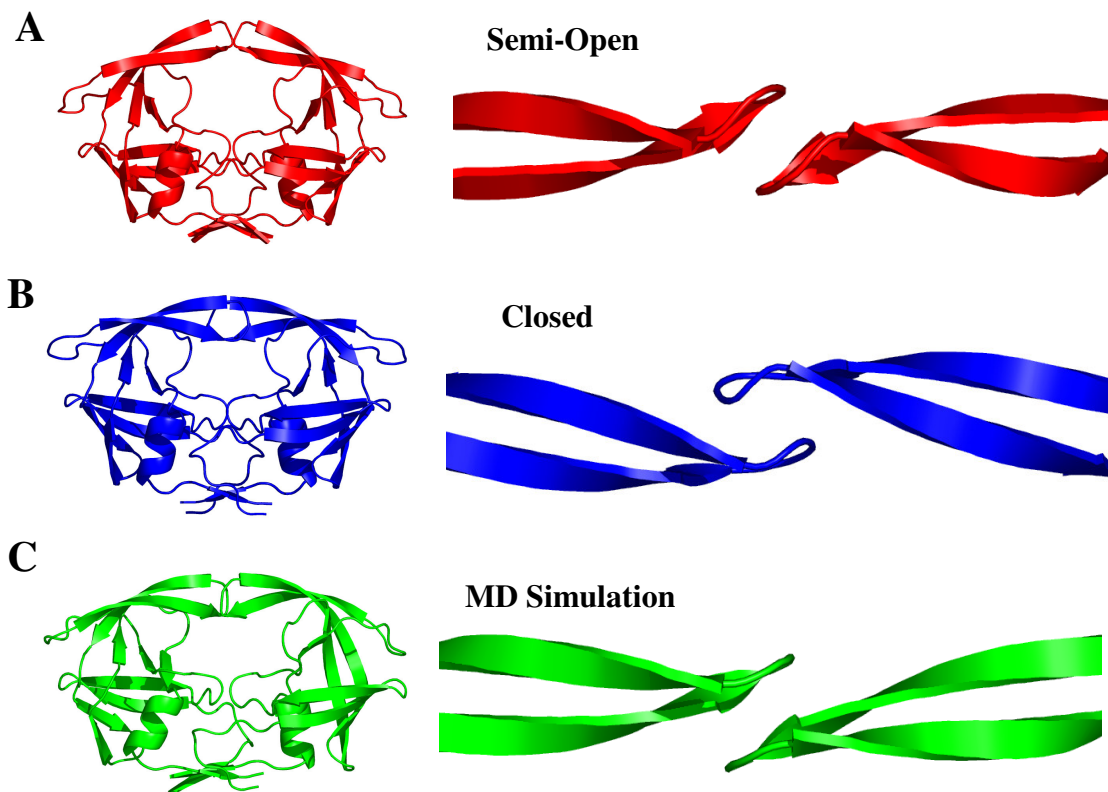
The structural alignments in Figure 6.7 raise an interesting question. How is the protease assuming a closed conformation with Compound 1 bound in the flap recognition pocket? A more detailed picture is possible with the MD simulation than the LD because of the explicit solvation. Furthermore, the hydrophobic effect is better represented, and that is a crucial component to real recognition. As such, a variety of metrics were calculated to quantify the conformational state of HIV-1p with Compound 1 bound using the trajectory from the MD simulation. To quantify the vertical movement of the flaps, the distance between the flap-tip residue I50/I50' and the C α atom of the catalytic residue D25/D25' was calculated for each monomer, as shown in Figure 6.8. The angle between residues D25-R57-I50/D25'-R57'-I50' could also be used to characterize the same motion as we found agreement between the two metrics. Flap A (monomer with Compound 1 bound) remained slightly more open than Flap B (ligand free); the average distance values were $14.47 \pm 0.92 \text{ \AA}$ and $12.80 \pm 0.84 \text{ \AA}$ for Flap A and Flap B, respectively. The presence of Compound 1 appears to create an asymmetry between the flap conformations as Flap B is assuming a slightly more closed state. To provide a reference, the distances were also calculated using crystal structures of HIV-1p in the semi-open (monomer 1HHP¹⁴) and closed forms (dimer 1PRO⁴³); the values were found to be 17.20 \AA and $14.14 \text{ \AA} / 14.12 \text{ \AA}$, respectively. This demonstrates that although Compound 1 is bound in the flap-recognition site of Flap B, the protease is still able to assume a closed conformation. In fact, Flap B is slightly more shut than the proper closed form. For completeness, analyses of the implicit-solvent LD are provided in the supplemental information.

Figure 6.8. Distance calculated between the flap-tip residue I50 C α to the catalytic residue D25 C α throughout the MD trajectory. This metric quantifies the flap movement in the vertical direction.



Further analysis determined that the closed state was possible due to the “handedness” of the flap-tip residues, shown in Figure 6.9. A representative structure taken from the MD simulation demonstrates the closed-flap conformation but semi-open handedness of the flap tips (Figure 6.9C). As previously mentioned in Chapter 1, upon ligand binding, the flaps assume a closed conformation over the active-site cavity^{44,45}, and the “handedness” of the flap tips (residues 49-52) orientation reverses upon closing^{11,31,32,46}. If the flaps do not change orientation, Flap B does not sit in its corresponding flap-recognition pocket. Due to the presence of Compound 1 in this site, the flaps are not able to properly shut; rather they form a new closed conformation. It is very likely that this state may render the protease inactive as the flap tips are not positioned correctly for substrate cleavage.

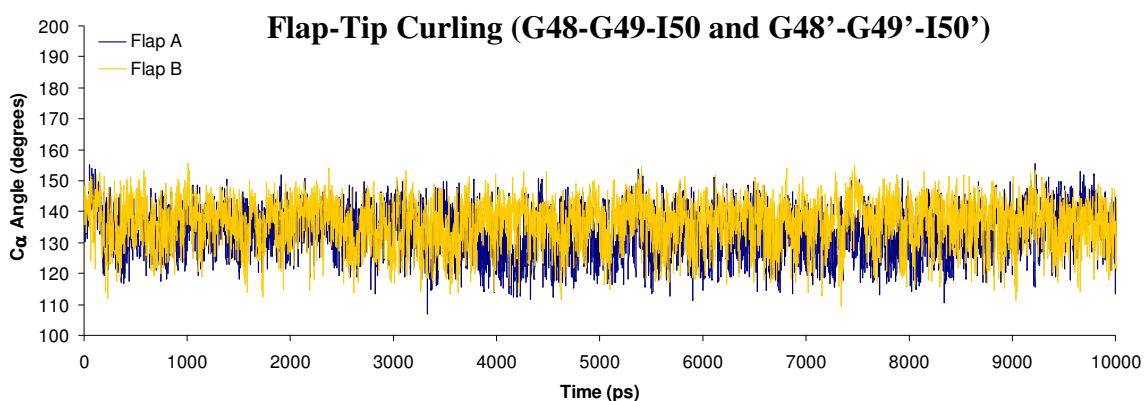
Figure 6.9. Structures of HIV-1p. A front and top view is provided to demonstrate the conformation of the flap region and change in handedness of the flap tips that occurs between the different states. **(A)** Semi-open conformation (PDB ID: 1HHP¹⁴). **(B)** Closed conformation (PDB ID: 1PRO⁴³) **(C)** Representative structure from the 10-ns MD in a closed-flap conformation but semi-open handedness of the flap tips.



As discussed in Chapter 1, the curling of the flap tips has recently been proposed as the driving force in flap opening.⁸⁻¹² This would place the flap tip of one monomer in its own “eye” site. The flap tips are curled 144.5° in the semi-open conformation, found by measuring the angle between the $C\alpha$ of residues G48-G49-I50/G48'-G49'-I50' of 1HHP¹⁴. Furthermore, a study by Perryman et al. suggested that the curled-in state occurred at $\sim 115^\circ$ and the curled-out state at $\sim 145^\circ$.¹⁰ Throughout the MD trajectory, the curling ranged from 155.3 - 107.3° for Flap A and 155.4 - 107.3° for Flap B as shown in Figure 6.10. However, the tips were only in the curling range $\sim 1\%$ of the simulation using Perryman’s definition. In addition, both monomers are sampling the same curling range, and an opening event was not observed in the 10 ns of simulation. Along with blocking Flap B from closing down into the Flap A recognition pocket, Compound 1 may

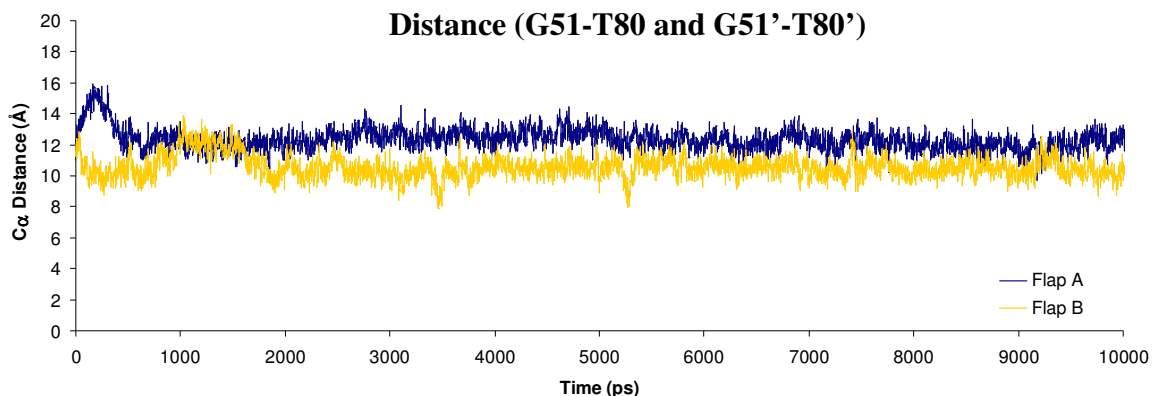
also be impeding the proposed interaction between the flap-tip residue I50 and the residues P79-T80-P81 of its own flap-recognition pocket.

Figure 6.10. Angle calculated between residues G48 C α - G49 C α - I50 C α to quantify the curling of the flap tips. An angle $\leq 115^\circ$ is defined as a curled state and $\geq 145^\circ$ as curled out.¹⁰ The blue curve displays the movement of Flap A and the yellow curve of Flap B.



The distance between flap tip residue G51/51' and the "eye" pocket residue T80/80' was also calculated to further quantify the movement of the flaps. It appears that having Compound 1 bound in Monomer A introduces an additional asymmetry across the system, as demonstrated by Figure 6.11. The average distance between the flap-tip residue G51 and "eye" pocket residue T80 for Monomer A (ligand bound) is 12.35 Å, while it is 10.57 Å for Monomer B (no ligand bound). Monomer B seems to have "collapsed" in order to accommodate the binding of Compound 1 in Monomer A and stabilize the closed, bound form. In fact, if a second ligand is introduced into the system (i.e. one compound in the eye pocket of each monomer), the complex is not stable (data not shown). The 1:1 stoichiometry (i.e. 1 compound bound per monomer) may not be possible due to lack of space.

Figure 6.11. Distance is calculated between residues G51 C α and T80 C α to quantify the position of the flaps. The blue curve displays the movement of Flap A and the yellow curve of Flap B.

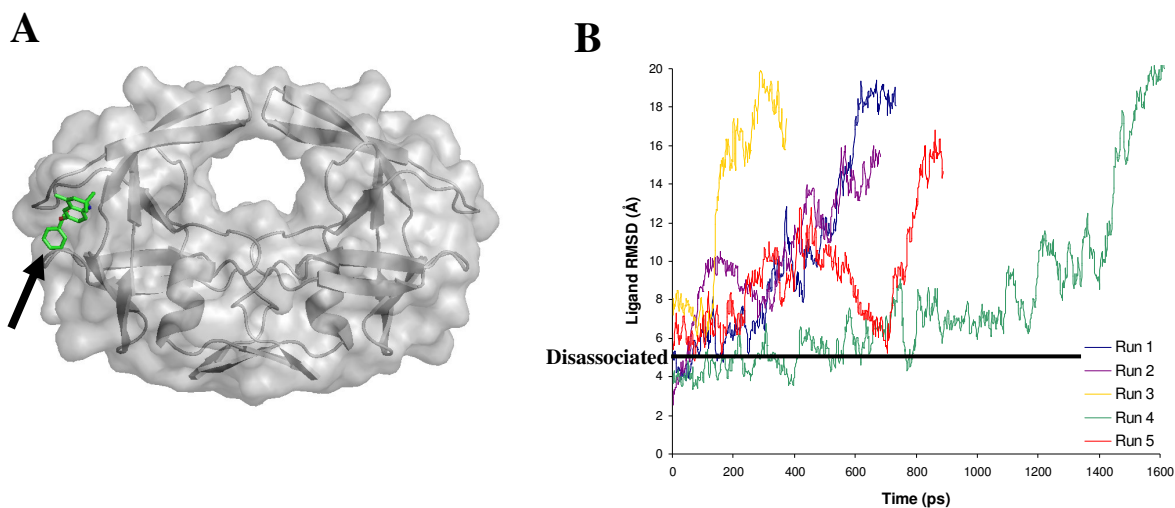


Ruling out the “Elbow Region”

The elbow region of HIV-1p (residues 35-42) is another small, flat, hydrophobic cleft and one of the recent hypotheses for inhibiting HIV-1p through new mechanisms. Anti-correlated motion has been identified between the flap and elbow regions through normal mode analysis and MD simulations^{7,10,47-49}, suggesting a potential site for allosteric control. It is the only other site on the surface of HIV-1p that may accommodate Compound 1. To provide further evidence that our compound is binding in the “eye” pocket and not the elbow region, we conducted five independent explicit-solvent MD simulations of the 1HHP in complex with Compound 1 in the elbow site. The protocol described in the methods was followed. AutoDock 3 was used to predict the binding pose of Compound 1, which provided the initial starting position for the simulations, shown in Figure 6.12A. Compound 1 disassociated from the elbow region in all five simulations as demonstrated in Figure 6.12B while HIV-1p itself remained stable. Furthermore in two of the five runs, the disassociation occurred during the equilibration phase and in another two, immediately after the restraints were removed from the protein. The ligand remained associated for almost 800 ps during one of the five simulations although not in a stable manner. Simulations would have been run for longer periods, as

is appropriate for the field, but they would not have provided useful information after the dissociation of the compound. Our analysis suggests that binding of Compound 1 in the elbow region is not favorable.

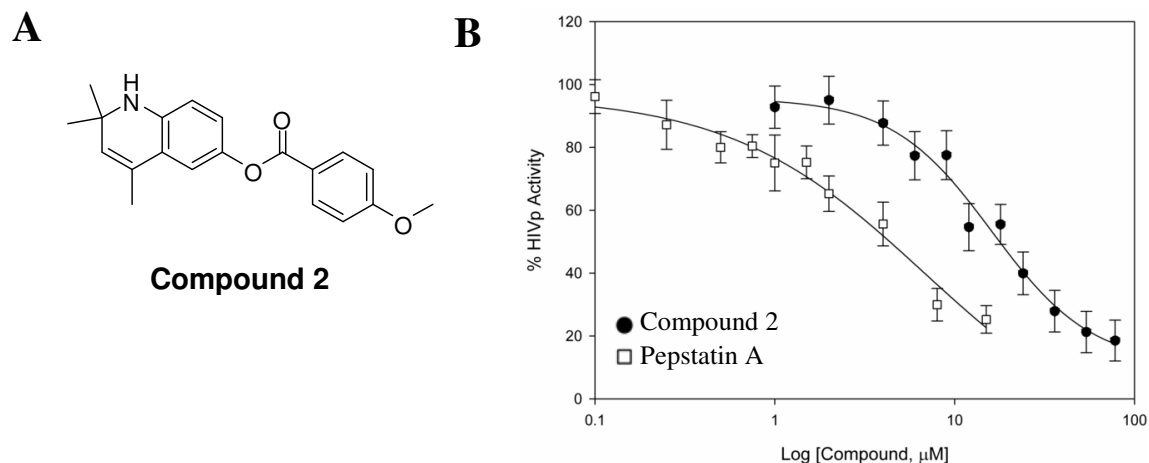
Figure 6.12. (A) HIV-1p shown in a surface representation. The predicted pose of Compound 1 (green, stick representation) by AutoDock 3 in elbow region is highlighted by an arrow. (B) RMSD values of Compound 1 during MD simulation (5 random seeds) showing ligand disassociating from contact with HIV-1p. Compound 1 is compared to the starting pose given in (A).



Experimental Verification of Predicted Compounds

Compound 1 was found to be auto-fluorescent and could not be evaluated by the employed assay. As such a para-methoxy analog also identified from the virtual screen of the CCG database screen was chosen to test (2,2,4-trimethyl-1,2-dihydroquinolin-6-yl 4-methoxybenzoate, provided in Figure 6.13A). Using the fluorescent assay previously described^{34,35}, Compound 2 was shown to inhibit HIV-1p activity; the IC₅₀ value was determined at ~18 μM and K_i as ~17.7 μM (Figure 6.13B). The binding affinity is modest and falls within the range of a lead-like compound, as does the molecular weight.⁵⁰⁻⁵² Oprea et al. noted that lead-like guidelines should be followed in initial phases of drug discovery to filter compounds, not drug-like profiles.^{51,53} If drug-like rules were employed, the identified lead compound may be difficult to optimize while remaining in “drug-like” space.

Figure 6.13. (A) Para-methoxy analog: 2,2,4-trimethyl-1,2-dihydroquinolin-6-yl 4-methoxybenzoate. (B) The activity of HIV-1p was monitored using a fluorimetric assay; upon HIV-1p cleavage of the FRET peptide substrate, fluorescence is recovered. Inhibition is measured as a result of the time-dependent decrease of fluorescence intensity that is linearly related to substrate cleavage. Each data point represents an average of three experiments, and the error bars reflect the standard deviation of observed values. Pepstatin A is shown as a control.



6.4 Conclusions

Finding novel mechanisms to inhibit HIV-1p is very important to overcoming the resistance associated with current inhibitors and discovering therapies with improved pharmacokinetic properties. We have shown in this study that the flap-recognition pocket can accommodate a small molecule, Compound 1, and maintains stable binding across both a 10-ns MD and five, 5-ns LD trajectories. Furthermore, the inhibition activity of an analog of our lead compound was experimentally verified ($K_i \sim 17.7 \mu\text{M}$ and $\text{IC}_{50} \sim 18 \mu\text{M}$) through a fluorimetric assay by preventing cleavage of the substrate. These inhibitors are much smaller than existing protease inhibitors and chemically very distinct. Hence, there is little likelihood that they are acting as traditional competitive inhibitors within the enzymatic binding site of HIV-1p. In fact modeling showed that Compound 1

is not stable within the central pocket; instead, it will migrate to the new site and form an appropriate complex.

The presence of a ligand in the flap-recognition pocket appears to alter the conformational behavior of the flap region, which may directly modify the kinetics of the system. Analysis of the conformational behavior of HIV-1p over an explicit-solvation MD simulation suggests that the protease assumes a closed conformation. We hypothesize two inhibition mechanisms: 1) Compound 1 binds in the flap-recognition pocket and the resulting closed conformation prevents the substrate accessibility to the active site or 2) the substrate may bind concurrently with Compound 1, but substrate cleavage cannot occur as the protease forms an inactive closed state (i.e. Flap B cannot properly occupy the recognition site of Monomer A). Furthermore, if the flap-tip curling hypothesis is correct, the flaps of HIV-1p may not be able to sample an open conformation when Compound 1 is bound.

There are several reasons why this new site is so attractive. First, it appears to be essential in forming the closed conformer. As previously mentioned, if the flaps cannot close appropriately, the substrate is not properly positioned for cleavage. Second, half of the residues defining the eye pocket are highly conserved and may be resistant to giving rise to escape mutants. Third, there is a good possibility that greater specificity for the site can be achieved because the ethane probes fit more deeply into the elbow than the flap tips. The most common mutant for the flap tip is I50V which shows that the eyebrow has some flexibility in binding hydrophobic moieties. Fourth, this site is much smaller than the central cavity, so it may yield inhibitors with low molecular weight which could have better pharmacokinetic properties than current HIV-1p drugs. Lastly, if co-administration proves synergistic, the new entities could be added to formulations of existing inhibitors as a combination therapy.

This study presents a new mode of inhibition of a key therapeutic target. In fact, this is the first new mechanism of action in ~15 years.^{54,55} Targeting the elbow and β -

sheet region are proposed in the literature as potential inhibition mechanisms, but no actual inhibitors have been identified. The dimerization inhibitors are the only experimentally verified alternative therapies to competitive, active-site inhibitors.

This work has been published as:

Damm, K.L., Quintero, J.J., Gestwicki, J.E. and Carlson, H.A. Inhibition of HIV-1 Protease by Modulating the Conformational Behavior of the Flap Region. *Manuscript in preparation.*

Carlson, H.A., Damm, K.L., and Meagher, K.L. "Compositions and Methods Relating to HIV Protease Inhibition" U.S. Provisional Patent Application No. 60/972,505 (Sept. 2007).

6.5 References

1. Meagher, K. L.; Lerner, M. G.; Carlson, H. A. Refining the Multiple Protein Structure Pharmacophore Method: Consistency across Three Independent Hiv-1 Protease Models. *J Med Chem* **2006**, *49*, 3478-3484.
2. Meagher, K. L.; Carlson, H. A. Incorporating Protein Flexibility in Structure-Based Drug Discovery: Using Hiv-1 Protease as a Test Case. *J Am Chem Soc* **2004**, *126*, 13276-13281.
3. Damm, K. L.; Carlson, H. A. Exploring Experimental Sources of Multiple Protein Conformations in Structure-Based Drug Design. *J Am Chem Soc* **2007**, *129*, 8225-8235.
4. Bowman, A. L.; Lerner, M. G.; Carlson, H. A. Protein Flexibility and Species Specificity in Structure-Based Drug Discovery: Dihydrofolate Reductase as a Test System. *J Am Chem Soc* **2007**, *129*, 3634-3640.
5. Ohtaka, H.; Freire, E. Adaptive Inhibitors of the Hiv-1 Protease. *Prog Biophys Mol Biol* **2005**, *88*, 193-208.
6. Baca, M.; Kent, S. B. Catalytic Contribution of Flap-Substrate Hydrogen Bonds in "Hiv-1 Protease" Explored by Chemical Synthesis. *Proc Natl Acad Sci U S A* **1993**, *90*, 11638-11642.
7. Piana, S.; Carloni, P.; Rothlisberger, U. Drug Resistance in Hiv-1 Protease: Flexibility-Assisted Mechanism of Compensatory Mutations. *Protein Sci* **2002**, *11*, 2393-2402.
8. Piana, S.; Carloni, P.; Parrinello, M. Role of Conformational Fluctuations in the Enzymatic Reaction of Hiv-1 Protease. *J Mol Biol* **2002**, *319*, 567-583.
9. Scott, W. R.; Schiffer, C. A. Curling of Flap Tips in Hiv-1 Protease as a Mechanism for Substrate Entry and Tolerance of Drug Resistance. *Structure* **2000**, *8*, 1259-1265.
10. Perryman, A. L.; Lin, J. H.; McCammon, J. A. Hiv-1 Protease Molecular Dynamics of a Wild-Type and of the V82f/I84v Mutant: Possible Contributions to Drug Resistance and a Potential New Target Site for Drugs. *Protein Sci* **2004**, *13*, 1108-1123.
11. Toth, G.; Borics, A. Flap Opening Mechanism of Hiv-1 Protease. *J Mol Graph Model* **2006**, *24*, 465-474.
12. Seibold, S. A.; Cukier, R. I. A Molecular Dynamics Study Comparing a Wild-Type with a Multiple Drug Resistant Hiv Protease: Differences in Flap and Aspartate 25 Cavity Dimensions. *Proteins* **2007**.

13. Katoh, E.; Louis, J. M.; Yamazaki, T.; Gronenborn, A. M.; Torchia, D. A.; Ishima, R. A Solution Nmr Study of the Binding Kinetics and the Internal Dynamics of an Hiv-1 Protease-Substrate Complex. *Protein Sci* **2003**, *12*, 1376-1385.
14. Spinelli, S.; Liu, Q. Z.; Alzari, P. M.; Hirel, P. H.; Poljak, R. J. The Three-Dimensional Structure of the Aspartyl Protease from the Hiv-1 Isolate Bru. *Biochimie* **1991**, *73*, 1391-1396.
15. Wlodawer, A.; Miller, M.; Jaskolski, M.; Sathyanarayana, B. K.; Baldwin, E.; Weber, I. T.; Selk, L. M.; Clawson, L.; Schneider, J.; Kent, S. B. Conserved Folding in Retroviral Proteases: Crystal Structure of a Synthetic Hiv-1 Protease. *Science* **1989**, *245*, 616-621.
16. Lapatto, R.; Blundell, T.; Hemmings, A.; Overington, J.; Wilderspin, A.; Wood, S.; Merson, J. R.; Whittle, P. J.; Danley, D. E.; Geoghegan, K. F.; et al. X-Ray Analysis of Hiv-1 Proteinase at 2.7 a Resolution Confirms Structural Homology among Retroviral Enzymes. *Nature* **1989**, *342*, 299-302.
17. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A., A Second Generation Force-Field for the Simulation of Proteins, Nucleic-Acids, and Organic-Molecules. *J Med Chem* **1995**, *117*, 5179-5197.
18. Case, D. A. P., D. A.; Caldwell, J. W.; Cheatham III, T. E.; Ross, W. S.; Simmerling, C. L.; Darden, T. A.; Merz, K. M.; Stanton, R. V.; Cheng, A. L.; Vincent, J. J.; Crowley, M.; Tsui, V.; Radmer, R. J.; Duan, Y.; Pitera, J.; Massova, I.; Seibel, G. L.; Singh, U. C.; Weiner, P. K.; Kollman, P. A. Amber6, University of California San Francisco: San Francisco, CA, 1996.
19. Jorgensen, W. L. Boss, 4.2; Yale University: New Haven, CT, 2000.
20. Jorgensen, W. L.; Maxwell, D. S.; TiradoRives, J., Development and Testing of the Opls All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J Med Chem* **1996**, *118*, 11225-11236.
21. Carlson, H. A.; Masukawa, K. M.; Rubins, K.; Bushman, F. D.; Jorgensen, W. L.; Lins, R. D.; Briggs, J. M.; McCammon, J. A. Developing a Dynamic Pharmacophore Model for Hiv-1 Integrase. *J Med Chem* **2000**, *43*, 2100-2114.
22. Molecular Operating Environment; Chemical Computing Group Inc.: Montreal, Canada, 2001.
23. Omega, 1.8.1; OpenEye Scientific Software: Santa Fe, New Mexico, 2004.
24. Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. Semianalytical treatment of solvation for molecular mechanics and dynamics *J Am Chem Soc* **1990**, *112*, 6127-6129.

25. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res* **2000**, *28*, 235-242.
26. DeLano, W. L. The PyMOL Molecular Graphics System. 2002. DeLano Scientific LLC, San Carlos, CA, USA. <http://www.pymol.org>.
27. Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of Multiple Amber Force Fields and Development of Improved Protein Backbone Parameters. *Proteins-Structure Function and Bioinformatics* **2006**, *65*, 712-725.
28. Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field. *J Comput Chem* **2004**, *25*, 1157-1174.
29. Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I. Fast, efficient generation of high-quality atomic charges. AM1-BCC model. *J Comp Chem* **2000**, *21*, 132-146.
30. Jorgensen, W. L.; Chandrasekhar, J. D.; Madura, R. W.; Impey, R. W.; Klein, M. L. Comparison of simple potential functions for simulating liquid water. *J Chem Phys* **1983**, *79*, 926-935.
31. Hornak, V.; Okur, A.; Rizzo, R. C.; Simmerling, C. Hiv-1 Protease Flaps Spontaneously Close to the Correct Structure in Simulations Following Manual Placement of an Inhibitor into the Open State. *J Am Chem Soc* **2006**, *128*, 2812-2813.
32. Hornak, V.; Okur, A.; Rizzo, R. C.; Simmerling, C. Hiv-1 Protease Flaps Spontaneously Open and Reclose in Molecular Dynamics Simulations. *Proc Natl Acad Sci U S A* **2006**, *103*, 915-920.
33. Morris, G. M.; Goodsell, D. S.; Halliday, R.S.; Huey, R.; Hart, W. E.; Belew, R. K.; Olson, A. J. Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function. *J Comp Chem* **1998**, *19*, 1639-1662.
34. Matayoshi, E. D.; Wang, G. T.; Krafft, G. A.; Erickson, J. Novel Fluorogenic Substrates for Assaying Retroviral Proteases by Resonance Energy Transfer. *Science* **1990**, *247*, 954-958.
35. Toth, M. V.; Marshall, G. R. A Simple, Continuous Fluorometric Assay for Hiv Protease. *Int J Pept Protein Res* **1990**, *36*, 544-550.
36. Wang, G.T.; Matayoshi, E.; Huffaker, H.J.; Krafft, G.A. Design and synthesis of new fluorogenic HIV protease substrates based on resonance energy transfer, *Tetrahedron Lett* **1990**, *31*, 6493-6492.

37. Cheng, Y.; Prosoff, W.H. Relationship between the inhibition constant (KI) and the concentration of inhibitor which causes 50 per cent inhibition (I50) of an enzymatic reaction. *Biochem Pharmacol.*, 1973, 22, 3099-3096.
38. Johnson, V. A.; Brun-Vezinet, F.; Clotet, B.; Kuritzkes, D. R.; Pillay, D.; Schapiro, J. M.; Richman, D. D. Update of the Drug Resistance Mutations in Hiv-1: Fall 2006. *Top HIV Med* **2006**, 14, 125-130.
39. Shafer, R. W.; Rhee, S. Y.; Pillay, D.; Miller, V.; Sandstrom, P.; Schapiro, J. M.; Kuritzkes, D. R.; Bennett, D. Hiv-1 Protease and Reverse Transcriptase Mutations for Drug Resistance Surveillance. *Aids* **2007**, 21, 215-223.
40. Foulkes, J. E.; Prabu-Jeyabalan, M.; Cooper, D.; Henderson, G. J.; Harris, J.; Swanstrom, R.; Schiffer, C. A. Role of Invariant Thr80 in Human Immunodeficiency Virus Type 1 Protease Structure, Function, and Viral Infectivity. *J Virol* **2006**, 80, 6906-6916.
41. Ghosh, A. K.; Ramu Sridhar, P.; Kumaragurubaran, N.; Koh, Y.; Weber, I. T.; Mitsuya, H. Bis-Tetrahydrofuran: A Privileged Ligand for Darunavir and a New Generation of Hiv Protease Inhibitors That Combat Drug Resistance. *ChemMedChem* **2006**, 1, 939-950.
42. Ghosh, A. K.; Sridhar, P. R.; Leshchenko, S.; Hussain, A. K.; Li, J.; Kovalevsky, A. Y.; Walters, D. E.; Wedekind, J. E.; Grum-Tokars, V.; Das, D.; Koh, Y.; Maeda, K.; Gatanaga, H.; Weber, I. T.; Mitsuya, H. Structure-Based Design of Novel Hiv-1 Protease Inhibitors to Combat Drug Resistance. *J Med Chem* **2006**, 49, 5252-5261.
43. Sham, H. L.; Zhao, C.; Stewart, K. D.; Betebenner, D. A.; Lin, S.; Park, C. H.; Kong, X. P.; Rosenbrook, W., Jr.; Herrin, T.; Madigan, D.; Vasavanonda, S.; Lyons, N.; Molla, A.; Saldivar, A.; Marsh, K. C.; McDonald, E.; Wideburg, N. E.; Denissen, J. F.; Robins, T.; Kempf, D. J.; Plattner, J. J.; Norbeck, D. W. A Novel, Picomolar Inhibitor of Human Immunodeficiency Virus Type 1 Protease. *J Med Chem* **1996**, 39, 392-397.
44. Ala, P. J.; Huston, E. E.; Klabe, R. M.; Jadhav, P. K.; Lam, P. Y.; Chang, C. H. Counteracting Hiv-1 Protease Drug Resistance: Structural Analysis of Mutant Proteases Complexed with Xv638 and Sd146, Cyclic Urea Amides with Broad Specificities. *Biochemistry* **1998**, 37, 15042-15049.
45. Wlodawer, A.; Gustchina, A. Structural and Biochemical Studies of Retroviral Proteases. *Biochim Biophys Acta* **2000**, 1477, 16-34.
46. Toth, G.; Borics, A. Closing of the Flaps of Hiv-1 Protease Induced by Substrate Binding: A Model of a Flap Closing Mechanism in Retroviral Aspartic Proteases. *Biochemistry* **2006**, 45, 6606-6614.

47. Zoete, V.; Michielin, O.; Karplus, M. Relation between Sequence and Structure of Hiv-1 Protease Inhibitor Complexes: A Model System for the Analysis of Protein Flexibility. *J Mol Biol* **2002**, *315*, 21-52.
48. Perryman, A. L.; Lin, J. H.; McCammon, J. A. Restrained Molecular Dynamics Simulations of Hiv-1 Protease: The First Step in Validating a New Target for Drug Design. *Biopolymers* **2006**, *82*, 272-284.
49. Layten, M.; Hornak, V.; Simmerling, C. The Open Structure of a Multi-Drug-Resistant Hiv-1 Protease Is Stabilized by Crystal Packing Contacts. *Journal of the American Chemical Society* **2006**, *128*, 13360-13361.
50. Hann, M. M.; Leach, A. R.; Harper, G. Molecular Complexity and Its Impact on the Probability of Finding Leads for Drug Discovery. *J Chem Inf Comput Sci* **2001**, *41*, 856-864.
51. Oprea, T. I.; Davis, A. M.; Teague, S. J.; Leeson, P. D. Is There a Difference between Leads and Drugs? A Historical Perspective. *J Chem Inf Comput Sci* **2001**, *41*, 1308-1315.
52. Scapin, G. Structural Biology and Drug Discovery. *Curr Pharm Des* **2006**, *12*, 2087-2097.
53. Oprea, T. I.; Allu, T. K.; Fara, D. C.; Rad, R. F.; Ostopovici, L.; Bologna, C. G. Lead-Like, Drug-Like or "Pub-Like": How Different Are They? *J Comput Aided Mol Des* **2007**, *21*, 113-119.
54. Weber, I. T. Comparison of the Crystal Structures and Intersubunit Interactions of Human Immunodeficiency and Rous Sarcoma Virus Proteases. *J Biol Chem* **1990**, *265*, 10492-10496.
55. Zhang, Z. Y.; Poorman, R. A.; Maggiora, L. L.; Heinrikson, R. L.; Kezdy, F. J. Dissociative Inhibition of Dimeric Enzymes. Kinetic Characterization of the Inhibition of Hiv-1 Protease by Its CooH-Terminal Tetrapeptide. *J Biol Chem* **1991**, *266*, 15591-15594.

APPENDICES

APPENDIX 1

Gaussian-Weighted RMSD Superposition of Proteins: A Structural Comparison for Flexible Proteins

A1.1 Global wRMSD code.

```
#!/usr/bin/env python

""" REQUIRED INSTALLATIONS:
    -Python 2.5
    -Scipy 0.5.2
    -NumPy 1.0.1
    -Biopython 1.42
    -Numerical 24.2 (install through Biopython website)
    -mxTextTools: http://www.egenix.com/files/python/mxTextTools.html

INPUT REQUIREMENTS:
    -2 PDB files
        PDB file must be in correct PDB format (i.e. chain ID's
        present, unique atom name within each residue, occupancy, etc...)

Global_wRMSD.py INFORMATION

    NOTE: For similar structures (characterized by a small sRMSD;
    example: sRMSD < 5), the scaling factor is set to 2
        For nonsimilar structures (characterized by a large sRMSD;
    example: sRMSD > 5), the scaling factor is set to 5

    TO RUN Global_wRMSD.py:

        Global_wRMSD.py Protein_1.pdb Protein_1_Chain_ID
        Protein_2.pdb Protein_2_Chain_ID

    Example:
        Global_wRMSD.py 3ERD.pdb A 3ERT.pdb A

    Example output:
        3ERD_sRMSD.pdb
        3ERD_wRMSD.pdb
        Calculated standard RMSD value
```

To run through cygwin:

In c:cygwin\etc\profile

Change PATH to local python (Python25):

```
PATH=/usr/local/bin:/cygdrive/c/Python25:/usr/bin:/bin:/usr/X11R6
/bin:$PATH
export PATH
```

For questions or comments:

Kelly Damm

kdammm@umich.edu

University of Michigan

Carlson Lab ""

```
#define global functions
from __future__ import division
import sys,re,cgi,os
from scipy import sort,transpose
import Numeric, LinearAlgebra
from Numeric import *

def run_Global_wRMSD(file1,file2,X_Chain_ID,Y_Chain_ID):
    #COMPARE RESIDUES FROM PROTEIN X AND PROTEIN Y BY CHAINS, REMOVES
    NONMATCHING RESIDUE COORDINATES AND RETURNS CA COORDINATES FROM
    MATCHING RESIDUES
    xlist,ylist =
    Ensure_Correspondence(X_Chain_ID,Y_Chain_ID,file1,file2)
    x = transpose(xlist)
    y = transpose(ylist)

    #RETURNS ALL COORDINATES OF PROTEIN X FOR TRANSFORMATION
    all = getAll_Coords(file1)
    set = getStructure(file1)
    title=(file1.split('.')[0], file1)

    #PERFORM STANDARD AND WEIGHTED RMSD CALCULATION
    allrot,SRMSD = weighted_alignment(x,y,all,set,title)
    print "The standard RMSD value is = ",SRMSD

    #OUTPUT TRANSFORMED STRUCTURE OF PROTEIN X
    s = getStructure(file1)
    setAll(s,allrot)
    writeStructure(s,'%s_wRMSD.pdb'%title[0])

#ENSURE RESIDUE CORRESPONDENCE
def Ensure_Correspondence(X_Chain_ID,Y_Chain_ID,file1,file2):
    xlist = []
    ylist = []
    x_CAResIDs= getCA_Resseq(file1, X_Chain_ID)
    y_CAResIDs= getCA_Resseq(file2, Y_Chain_ID)

    #REMOVE DISORDERED RESIDUES
    x_disordered = get_Disordered(file1,X_Chain_ID)
    y_disordered = get_Disordered(file2,Y_Chain_ID)
```

```

xremove_disorder,x_ResSeq1 =
remove_ResID(x_CAResIDs,x_disordered)
yremove_disorder,y_ResSeq1 =
remove_ResID(y_CAResIDs,y_disordered)

#COMPARE PROTEIN1 TO PROTEIN2 AND REMOVE NONMATCHING RESIDUES
x_Nonmatch, y_Nonmatch = compare(x_ResSeq1,y_ResSeq1)

#MAKE LIST OF ALL RESIDUES TO BE REMOVED
x_Remove = x_disordered + x_Nonmatch
x_Remove.sort()
y_Remove = y_disordered + y_Nonmatch
y_Remove.sort()

#DETERMINE POSITION OF RESIDUES TO BE REMOVED IN PDB FILES
x_Res_positions = get_Residue_Position(x_CAResIDs, x_Remove)
y_Res_positions = get_Residue_Position(y_CAResIDs, y_Remove)

#RETURNS ALL CA COORDINATES
x_CAcords = transpose(get_CAcords(file1,X_Chain_ID))
y_CAcords = transpose(get_CAcords(file2,Y_Chain_ID))

#Removes CA coordinates of nonmatching residues
xlist = zero_CAcords(x_CAcords,x_Res_positions)
ylist = zero_CAcords(y_CAcords,y_Res_positions)

#FINAL CHECK TO ENSURE RESIDUE CORRESPONDENCE
n = len(xlist)
j = len(ylist)
if n != j:
    sys.exit("Proteins do not have same number of atoms;
Protein X has",n,"atoms, while Protein Y has",j,"atoms")

#CHECK TO DETERMINE IF APPROPRIATE NUMBER OF COORDINATES PRESENT
if (len(xlist[0]) != 3):
    sys.exit("Protein X does not have a 3xn atom coordinate
set")
if (len(ylist[0]) != 3):
    sys.exit("Protein Y does not have a 3xn atom coordinate
set")

#CHECK TO DETERMINE IF >4 COORDINATES PRESENT FOR EACH PROTEIN
if n < 4:
    sys.exit("Protein X has 3 or less coordinates, 4 or more
needed to perform alignment")
if j < 4:
    sys.exit("Protein Y has 3 or less coordinates, 4 or more
needed to perform alignment")
return xlist,ylist

##WEIGHTED RMSD ALIGNMENT##
def weighted_alignment(x,y,all,set,title):
    atoms = len(x[0])

    #Initial standard alignment without weight

```

```

#TRANSLATE PROTEINS X AND Y TO CENTER
n = len(x[0])
x_mean = mean(x,n)
y_mean = mean(y,n)
x_trans = translation(x, x_mean)
x_translated = nested_list(x_trans,n)
y_trans = translation(y, y_mean)
y_translated = nested_list(y_trans,n)
x_transpose = transpose(x_translated)

#CALCULATE COVARIANCE MATRIX (y_translated *x_translated^t)
R = matrixmultiply(y_translated, x_transpose)
R_transpose = transpose(R)
R2 = matrixmultiply(R_transpose, R)

#DETERMINE THE EIGENVECTORS AND EIGENVALUES of R2
mu,A = LinearAlgebra.eigenvectors(R2)

#SORT EIGENVECTORS IN DECREASING ORDER OF EIGENVALUES
a = [(mu[i],A[i]) for i in range(len(A))]
a.sort()
a.reverse()
mu = [x[0] for x in a]
A = [x[1] for x in a]

#DETERMINE RIGHT-HANDED SYSTEM
A_3 = crossproduct(A[0], A[1])
A = [A[0], A[1], A_3]

#CALCULATE B, NORMALIZED PRODUCT OF (RxA)
B_1 = matrixmultiply(R, A[0])
B_2 = matrixmultiply(R, A[1])
norm_B_1 = normalize(B_1)
norm_B_2 = normalize(B_2)
norm_B_3 = crossproduct(norm_B_1,norm_B_2)
B = [norm_B_1,norm_B_2,norm_B_3]
B_transpose = transpose(B)

#CALCULATE ROTATION MATRIX, U
U = rotation_matrix(B_transpose, A)
x_rot = matrixmultiply(U,x_translated)

#CALCULATE STANDARD RMSD
standard_RMSD = sRMSD(x_rot, y_translated)

#ADD MEAN VALUES OF PROTEIN Y TO ROTATED COORDINATES OF PROTEIN X
x_coords = add_coords(x_rot, y_mean)
x = nested_list(x_coords,n)

#TRANSLATE ALL COORDINATES OF PROTEIN X
all_trans = translation(all, x_mean)
j = len(all[0])
all_translated = nested_list(all_trans,j)
all_rot = matrixmultiply(U,all_translated)

#ADD ALL MEAN VALUES OF PROTEIN Y TO ALL ROTATED COORDINATES OF
PROTEIN X

```

```

all_coords = add_coords(all_rot, y_mean)
all = nested_list(all_coords, j)
allrot = transpose(all)
setAll(set, allrot)

#OUTPUT STANDARD RMSD ALIGNMENT
writeStructure(set, '%s_sRMSD.pdb'%title[0])

#WEIGHTED RMSD CALCULATION, z = # of iterations
z = 1
weighted_rmsds = []
w_metric = []
all_list = []

#DETERMINE APPROPRIATE SCALING FACTOR
if standard_RMSD < 5:
    scaling_factor = 2
elif standard_RMSD >= 5:
    scaling_factor = 5
while z < 5001:
    n = len(x[0])
    #TRANSLATE WEIGHTED CENTROIDS TO ORIGIN
    #CALCULATE WEIGHTS (protein1, protein2, scaling_factor)
    weights = weight(x, y, scaling_factor)
    weighted_x_mean = weight_trans(x, weights, n)
    weighted_y_mean = weight_trans(y, weights, n)
    x_trans = translation(x, weighted_x_mean)
    y_trans = translation(y, weighted_y_mean)
    x_translated = nested_list(x_trans, n)
    y_translated = nested_list(y_trans, n)

    #CALCULATE WEIGHTED COVARIANCE MATRIX
    weighted_rot =
weight(x_translated, y_translated, scaling_factor)
    weighted_x_translated = multiply(weighted_rot, x_translated)
    wx_transpose = transpose(weighted_x_translated)
    R = matrixmultiply(y_translated, wx_transpose)
    R_transpose = transpose(R)
    R2 = matrixmultiply(R_transpose, R)

    #DETERMINE THE EIGENVECTORS AND EIGENVALUES of R2
    mu, A = LinearAlgebra.eigenvectors(R2)

    #SORT EIGENVECTORS IN DECREASING ORDER OF EIGENVALUES
    a = [(mu[i], A[i]) for i in range(len(A))]
    a.sort()
    a.reverse()
    mu = [x[0] for x in a]
    A = [x[1] for x in a]

    #DETERMINE RIGHT-HANDED SYSTEM
    A_3 = crossproduct(A[0], A[1])
    A = [A[0], A[1], A_3]

    #CALCULATE B, NORMALIZED PRODUCT OF (RxA)
    B_1 = matrixmultiply(R, A[0])
    B_2 = matrixmultiply(R, A[1])

```

```

norm_B_1 = normalize(B_1)
norm_B_2 = normalize(B_2)
norm_B_3 = crossproduct(norm_B_1,norm_B_2)
B = [norm_B_1,norm_B_2,norm_B_3]
B_transpose = transpose(B)

#CALCULATE WEIGHTED ROTATION MATRIX, U
U = rotation_matrix(B_transpose, A)
x_rot = matrixmultiply(U,x_translated)

#CALCULATE WEIGHTED RMSD
weighted_rmsds.append(wRMSD(x_rot,
y_translated,scaling_factor))
w_metric.append(wSUM(x_rot,
y_translated,scaling_factor,atoms,z))

#DETERMINE IF CONVERGENCE IS REACHED
if z > 1:
    wrmsd_diff = weighted_rmsds[-2] - weighted_rmsds[-1]
else:
    wrmsd_diff = []

if 0 < wrmsd_diff < 0.000001:

    #ADD MEAN VALUES OF PROTEIN Y TO ROTATED COORDINATES
OF PROTEIN X
    x_coords = add_coords(x_rot, weighted_y_mean)
    x = nested_list(x_coords,n)

    #TRANSFORM ALL COORDINATES OF PROTEIN X
    all_trans = translation(all, weighted_x_mean)
    j = len(all[0])
    all_translated = nested_list(all_trans,j)
    all_rot = matrixmultiply(U,all_translated)

    #ADD ALL MEAN VALUES OF PROTEIN Y TO ALL ROTATED
COORDINATES OF PROTEIN X
    all_coords = add_coords(all_rot, weighted_y_mean)
    all = nested_list(all_coords,j)
    allrot = transpose(all)
    break

else:

    #ADD MEAN VALUES OF PROTEIN Y TO ROTATED COORDINATES
OF PROTEIN X
    x_coords = add_coords(x_rot, weighted_y_mean)
    x = nested_list(x_coords,n)

    #TRANSFORM ALL COORDINATES OF PROTEIN X
    all_trans = translation(all, weighted_x_mean)
    j = len(all[0])
    all_translated = nested_list(all_trans,j)
    all_rot = matrixmultiply(U,all_translated)

    #ADD ALL MEAN VALUES OF PROTEIN Y TO ALL ROTATED
COORDINATES OF PROTEIN X
    all_coords = add_coords(all_rot, weighted_y_mean)

```

```

        all = nested_list(all_coords, j)
        z = z + 1
    else:
        print "Alignment stopped after 5000 iterations, convergence
was never reached"
        allrot = transpose(all)
        return allrot, standard_RMSD

#####HELPER FUNCTIONS#####
def getCA_Resseq(filename, chain_key):
    x = []
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            if residue.has_id("CA"):
                resseq=residue.get_id()[1]
                x.append(resseq)
    return x

def get_Disordered(filename, chain_key):
    x = []
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            if residue.is_disordered():
                resseq = residue.get_id()[1]
                x.append(resseq)
    return x

def remove_ResID(filename, list):
    result=[]
    result1=[]
    for value in filename:
        if value in list:
            result.append(value)
        else:
            result1.append(value)
    return result, result1

def compare(x, y):
    x_list = []
    y_list = []
    for each in x:
        if each not in y:
            x_list.append(each)
    for each in y:
        if each not in x:
            y_list.append(each)
    x_list = zero2(x_list)
    y_list = zero2(y_list)
    x_list = sort(x_list)
    y_list = sort(y_list)

```

```

x_list = list(x_list)
y_list = list(y_list)
return x_list, y_list

def zero2(x):
    result = []
    for each in x:
        if each != 0:
            result.append(each)
        else:
            continue
    return result

def get_Residue_Position(first, list):
    c = 0
    q = len(list)
    result2 = []
    while c < q:
        result = []
        for each in first:
            result.append(each)
            if each != list[c]:
                continue
            else:
                break
        ans = len(result)
        ans = ans - 1
        result2.append(ans)
        c = c + 1
    return result2

def get_CAcords(filename, chain_key):
    x = []
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            if residue.has_id("CA"):
                ca=residue["CA"]
                x.append(ca.get_coord())
    x_t = transpose(x)
    return x_t

###THANK YOU TO MARK BENSON FOR UPDATING THIS HELPER FUNCTION###
def zero_CAcords(first, list):
    c = 0
    q = len(list)
    r = len(first)
    s = r - q
    result = array([[s]])
    import copy
    temp = copy.copy(first)
    if q != 0:
        result = copy.copy(first)
        for i in range(len(first)) :
            if i not in list :

```



```

        if i + c < r :
            temp[i] = first[i+c]
    else:
        c = c + 1

        result[i] = [0,0,0]
        if i + c < r :
            temp[i] = first[i+c]
    a = temp[:s]
    return a
else:
    return first

def getAll_Coords(filename):
    result = []
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain = model.get_list()
        for each in chain:
            for res in each.get_list():
                for x in res.get_list():
                    result.append(x.get_coord())
    x_t = transpose(result)
    return x_t

def getStructure(filename):
    parser = PDBParser()
    structure = parser.get_structure(filename.split('.')[0],
filename)
    return structure

def writeStructure(structure,filename):
    from Bio.PDB.PDBIO import PDBIO
    import sys
    io = PDBIO()
    io.set_structure(structure)
    io.save(filename)

def setAll(structure,newCACoords):
    allCAs = []
    for model in structure.get_list():
        chain = model.get_list()
        for each in chain:
            for res in each.get_list():
                for x in res.get_list():
                    allCAs.append(x)
    if len(allCAs) != len(newCACoords):
        print "wrong number of atoms .. structure
had",len(allCAs),"you gave me",len(newCACoords)
        raise Exception("wrong number of atoms")
    for newCoords,ca in zip(newCACoords,allCAs):
        #print newCoords,ca.get_coord()
        ca.set_coord(newCoords)
        #print ca.get_coord()
        #raise Exception("time to stop")

```

```

def mean(first,n):
    return [sum(each)/n for each in first]

def translation(first,second):
    c = 0
    k = []
    q = len(first)
    while c < q:
        for each in first[c]:
            subtr = each - second[c]
            k.append(subtr)
        c = c + 1
    return k

def nested_list(name,n):
    first1 = name[0:n]
    first2 = name[n:2*n]
    first3 = name[2*n:3*n]
    first_translated =[first1,first2,first3]
    return first_translated

def crossproduct(a,b):
    C_0 = a[1]*b[2] - a[2]*b[1]
    C_1 = a[2]*b[0] - a[0]*b[2]
    C_2 = a[0]*b[1] - a[1]*b[0]
    return [C_0, C_1, C_2]

def normalize(a):
    B_0 = (a[0])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    B_1 = (a[1])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    B_2 = (a[2])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    return [B_0, B_1, B_2]

def rotation_matrix(first, second):
    U = matrixmultiply(first, second)
    return U

def sqr(matrix):
    k = []
    for each in matrix:
        sq = (each)**2
        k.append(sq)
    return k

def sqroot(matrix):
    j = []
    for each in matrix:
        sqroot = sqrt(each)
        j.append(sqroot)
    return j

def sRMSD(first,second):
    first = array(first)
    second = array(second)
    subtr = first - second
    def sqr(matrix):
        k = []

```

```

        for each in matrix:
            sq = (each)**2
            k.append(sq)
        return k
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    def sqroot(matrix):
        j = []
        for each in matrix:
            sqroot = sqrt(each)
            j.append(sqroot)
        return j
    d = sqroot(sum_subtr_s)
    sq_d = sqr(d)
    s_sq_d = sum(sq_d)
    tot = (len(first[0]))
    value = sqrt(s_sq_d/tot)
    return value

def add_coords(first,second):
    c = 0
    k = []
    q = len(first)
    while c < q:
        for each in first[c]:
            add = each + second[c]
            k.append(add)
        c = c + 1
    return k

def weight_trans(first,weight,n):
    mult = multiply(first, weight)
    sum_mult = sum(mult)
    mean = [sum(each)/n for each in mult]
    return mean

def weight(first,second,constant):
    first = array(first)
    second = array(second)
    subtr = first - second
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    d = sqroot(sum_subtr_s)
    weighted_d = Gaussian(d,constant)
    weighted_d = Gaussian2(weighted_d)
    return weighted_d

def wSUM(first,second,constant,atoms,z):
    first = array(first)
    second = array(second)
    subtr = first - second
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    d = sqroot(sum_subtr_s)
    weighted_d = Gaussian(d,constant)
    weighted_d = Gaussian2(weighted_d)
    sum_weighted_d = sum(weighted_d)

```

```

    value = sum_weighted_d/atoms
    return value

def wRMSD(first, second, constant):
    first = array(first)
    second = array(second)
    subtr = first - second
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    d = sqrt(sum_subtr_s)
    weighted_d = Gaussian(d, constant)
    weights = Gaussian2(weighted_d)
    sq_d = sqr(d)
    wd = multiply(sq_d, weights)
    s_wd = sum(wd)
    n = len(d)
    s_sq_d_divide = s_wd/n
    value = sqrt(s_sq_d_divide)
    return value

def Gaussian(first, z):
    value = []
    for each in first:
        weight = (-((each)**2)/z)
        value.append(weight)
    return value

def Gaussian2(first):
    value = []
    for each in first:
        weight = exp(each)
        value.append(weight)
    return value

##### HELPER FUNCTIONS IN ENSURE RESIDUE CORRESPONDENCE FUNCTION #####
##### THIS HAS BEEN MODIFIED TO WORK WITH WRMSD CODE #####
# Copyright (C) 2002, Thomas Hamelryck (thamelry@vub.ac.be)
# This code is part of the Biopython distribution and governed by its
# license. Please see the LICENSE file that should have been included
# as part of this package.

# Python stuff
import sys
from string import split
from Numeric import array, Float0

# My stuff
from Bio.PDB.StructureBuilder import StructureBuilder
from Bio.PDB.PDBExceptions import PDBConstructionException
from Bio.PDB.parse_pdb_header import _parse_pdb_header_list

__doc__="Parser for PDB files."

# If PDB spec says "COLUMNS 18-20" this means line[17:20]

```

```

class PDBParser:
    """
    Parse a PDB file and return a Structure object.
    """

    def __init__(self, PERMISSIVE=0, get_header=0,
structure_builder=None):
    """
    The PDB parser call a number of standard methods in an
aggregated
    StructureBuilder object. Normally this object is instanciated
by the
    PDBParser object itself, but if the user provides his own
StructureBuilder
    object, the latter is used instead.

    Arguments:
    o PERMISSIVE - int, if this is 0 exceptions in constructing the
SMCRA data structure are fatal. If 1 (DEFAULT), the exceptions
are
    caught, but some residues or atoms will be missing. THESE
EXCEPTIONS
    ARE DUE TO PROBLEMS IN THE PDB FILE!.
    o structure_builder - an optional user implemented
StructureBuilder class.
    """
    if structure_builder!=None:
        self.structure_builder=structure_builder
    else:
        self.structure_builder=StructureBuilder()
    self.header=None
    self.trailer=None
    self.line_counter=0
    self.PERMISSIVE=PERMISSIVE

# Public methods

def get_structure(self, id, file):
    """Return the structure.

    Arguments:
    o id - string, the id that will be used for the structure
    o file - name of the PDB file OR an open filehandle
    """
    self.header=None
    self.trailer=None
    # Make a StructureBuilder instance (pass id of structure as
parameter)
    self.structure_builder.init_structure(id)
    if isinstance(file, basestring):
        file=open(file)
    self._parse(file.readlines())
    file.close()
    self.structure_builder.set_header(self.header)
    # Return the Structure instance

```

```

return self.structure_builder.get_structure()

def get_header(self):
    "Return the header."
    return self.header

def get_trailer(self):
    "Return the trailer."
    return self.trailer

# Private methods

def _parse(self, header_coords_trailer):
    "Parse the PDB file."
    # Extract the header; return the rest of the file
    self.header,
coords_trailer=self._get_header(header_coords_trailer)
    # Parse the atomic data; return the PDB file trailer
    self.trailer=self._parse_coordinates(coords_trailer)

def _get_header(self, header_coords_trailer):
    "Get the header of the PDB file, return the rest."
    structure_builder=self.structure_builder
    for i in range(0, len(header_coords_trailer)):
        structure_builder.set_line_counter(i+1)
        line=header_coords_trailer[i]
        record_type=line[0:6]
        if(record_type=='ATOM  ' or record_type=='HETATM' or
record_type=='MODEL  '):
            break
    header=header_coords_trailer[0:i]
    # Return the rest of the coords+trailer for further processing
    self.line_counter=i
    coords_trailer=header_coords_trailer[i:]
    header_dict=_parse_pdb_header_list(header)
    return header_dict, coords_trailer

def _parse_coordinates(self, coords_trailer):
    "Parse the atomic data in the PDB file."
    local_line_counter=0
    structure_builder=self.structure_builder
    current_model_id=0
    # Flag we have an open model
    model_open=0
    current_chain_id=None
    current_segid=None
    current_residue_id=None
    current_resname=None
    for i in range(0, len(coords_trailer)):
        line=coords_trailer[i]
        record_type=line[0:6]
        global_line_counter=self.line_counter+local_line_counter+1
        structure_builder.set_line_counter(global_line_counter)
        if(record_type=='ATOM  ' or record_type=='HETATM'):
            # Initialize the Model - there was no explicit MODEL
record
            if not model_open:

```

```

        structure_builder.init_model(current_model_id)
        current_model_id+=1
        model_open=1
    fullname=line[12:16]
    # get rid of whitespace in atom names
    split_list=split(fullname)
    if len(split_list)!=1:
        # atom name has internal spaces, e.g. " N B ", so
        # we do not strip spaces
        name=fullname
    else:
        # atom name is like " CA ", so we can strip spaces
        name=split_list[0]
    altloc=line[16:17]
    resname=line[17:20]
    chainid=line[21:22]
    try:
        serial_number=int(line[6:11])
    except:
        serial_number=0
    resseq=int(split(line[22:26])[0]) # sequence

    icode=line[26:27] # insertion code
    if record_type=='HETATM': # hetero atom flag
        if resname=="HOH" or resname=="WAT":
            hetero_flag="W"
        else:
            hetero_flag="H"
    else:
        hetero_flag=" "
    residue_id=(hetero_flag, resseq, icode)
    # atomic coordinates
    x=float(line[30:38])
    y=float(line[38:46])
    z=float(line[46:54])
    coord=array((x, y, z), Float0)
    # occupancy & B factor
    occupancy=float(line[54:60])
    bfactor=float(line[60:66])
    segid=line[72:76]
    if current_segid!=segid:
        current_segid=segid
        structure_builder.init_seg(current_segid)
    if current_chain_id!=chainid:
        current_chain_id=chainid
        structure_builder.init_chain(current_chain_id)
    current_residue_id=residue_id
    current_resname=resname
    try:
        structure_builder.init_residue(resname,
hetero_flag, resseq, icode)
    except PDBConstructionException, message:
        self._handle_PDB_exception(message,
global_line_counter)
    elif current_residue_id!=residue_id or
current_resname!=resname:
        current_residue_id=residue_id

```

```

        current_resname=resname
        try:
            structure_builder.init_residue(resname,
hetero_flag, resseq, icode)
        except PDBConstructionException, message:
            self._handle_PDB_exception(message,
global_line_counter)
            # init atom
            try:
                structure_builder.init_atom(name, coord, bfactor,
occupancy, altloc, fullname, serial_number)
            except PDBConstructionException, message:
                self._handle_PDB_exception(message,
global_line_counter)
            elif(record_type=='ANISOU'):
                anisou=map(float, (line[28:35], line[35:42],
line[43:49], line[49:56], line[56:63], line[63:70]))
                # U's are scaled by 10^4
                anisou_array=(array(anisou,
Float0)/10000.0).astype(Float0)
                structure_builder.set_anisou(anisou_array)
            elif(record_type=='MODEL '):
                structure_builder.init_model(current_model_id)
                current_model_id+=1
                model_open=1
                current_chain_id=None
                current_residue_id=None
            elif(record_type=='END ' or record_type=='CONNECT'):
                # End of atomic data, return the trailer
                self.line_counter=self.line_counter+local_line_counter
                return coords_trailer[local_line_counter:]
            elif(record_type=='ENDMDL'):
                model_open=0
                current_chain_id=None
                current_residue_id=None
            elif(record_type=='SIGUIJ'):
                # standard deviation of anisotropic B factor
                siguij=map(float, (line[28:35], line[35:42],
line[42:49], line[49:56], line[56:63], line[63:70]))
                # U sigma's are scaled by 10^4
                siguij_array=(array(siguij,
Float0)/10000.0).astype(Float0)
                structure_builder.set_siguij(siguij_array)
            elif(record_type=='SIGATM'):
                # standard deviation of atomic positions
                sigatm=map(float, (line[30:38], line[38:45],
line[46:54], line[54:60], line[60:66]))
                sigatm_array=array(sigatm, Float0)
                structure_builder.set_sigatm(sigatm_array)
            local_line_counter=local_line_counter+1
            # EOF (does not end in END or CONNECT)
            self.line_counter=self.line_counter+local_line_counter
            return []

def _handle_PDB_exception(self, message, line_counter):
    """

```



```

        This method catches an exception that occurs in the
StructureBuilder
        object (if PERMISSIVE==1), or raises it again, this time adding
the
        PDB line number to the error message.
"""
        message="%s at line %i." % (message, line_counter)
        if self.PERMISSIVE:
            # just print a warning - some residues/atoms will be
missing
            print "PDBConstructionException: %s" % message
            print "Exception ignored.\nSome atoms or residues will be
missing in the data structure."
        else:
            # exceptions are fatal - raise again with new message
(including line nr)
            raise PDBConstructionException, message

if __name__=="__main__":

    import sys

    p=PDBParser(PERMISSIVE=1)

    s=p.get_structure("scr", sys.argv[1])

    for m in s.get_iterator():
        p=m.get_parent()
        assert(p is s)
        for c in m.get_iterator():
            p=c.get_parent()
            assert(p is m)
            for r in c.get_iterator():
                p=r.get_parent()
                assert(p is c)
                for a in r.get_iterator():
                    p=a.get_parent()
                    if not p is r:
                        print p, r

#RUN Global_wRMSD.py
if __name__ == "__main__":
    if len(sys.argv) != 5:
        print "usage: Global_wRMSD.py Protein_X.pdb
Protein_X_ChainID Protein_Y.pdb Protein_Y_ChainID"
        sys.exit()
    filename1 = sys.argv[1]
    filename2 = sys.argv[3]
    X_Chain_ID = sys.argv[2]
    Y_Chain_ID = sys.argv[4]
    run_Global_wRMSD(filename1, filename2, X_Chain_ID, Y_Chain_ID)

```

A1.2 Local wRMSD code.

```
#!/usr/bin/env python

""" REQUIRED INSTALLATIONS:
    -Python 2.5
    -Scipy 0.5.2
    -NumPy 1.0.1
    -Biopython 1.42
    -Numerical 24.2 (install through Biopython website)
    -mxTextTools: http://www.egenix.com/files/python/mxTextTools.html

INPUT REQUIREMENTS:
    -2 PDB files
        PDB file must be in correct PDB format (i.e. chain ID's
        present, unique atom name within each residue, occupancy, etc...)

Local_wRMSD.py INFORMATION

    NOTE: Scaling factor is set to 2 for local alignments.

    TO RUN Local_wRMSD.py:

        Local_wRMSD.py Protein_1.pdb Protein_1_Chain_ID
        Protein_2.pdb Protein_2_Chain_ID

    Example:
        Local_wRMSD.py 3ERD.pdb A 3ERT.pdb A

    Example output:
        Unique weighted solutions with a corresponding %wSUM

To run through cygwin:
    In c:cygwin\etc\profile

    Change PATH to local python (Python25):

        PATH=/usr/local/bin:/cygdrive/c/Python25:/usr/bin:/bin:/usr/X11R6
        /bin:$PATH
        export PATH

For questions or comments:
Kelly Damm
kdamm@umich.edu

University of Michigan
Carlson Lab """

#define global functions
from __future__ import division
import sys, re, cgi, os
from scipy import sort, transpose
```

```

import Numeric, LinearAlgebra
from Numeric import *

def run_Local_wRMSD(file1,file2,X_Chain_ID,Y_Chain_ID):
    #COMPARE RESIDUES FROM PROTEIN X AND PROTEIN Y BY CHAINS, REMOVES
    NONMATCHING RESIDUE COORDINATES AND RETURNS CA COORDINATES FROM
    MATCHING RESIDUES
    xlist,ylist =
    Ensure_Correspondence(X_Chain_ID,Y_Chain_ID,file1,file2)
    x = transpose(xlist)
    y = transpose(ylist)

    #RETURNS ALL COORDINATES OF PROTEIN X FOR TRANSFORMATION
    all = getAll_Coords(file1)
    title=(file1.split('.')[0], file1)

    #PERFORM STANDARD AND WEIGHTED RMSD CALCULATION
    final_wSUM,All_final_Coords = weighted_alignment(x,y,all,2)

    #DETERMINE UNIQUE LOCAL SOLUTIONS
    all_answer = Unique_Coords(All_final_Coords)
    s = getStructure(file1)
    all_sorted_wSUM =
    Coord_Positions(all_answer,All_final_Coords,s,final_wSUM,title)
    rounded_wSUM = round_decimal(all_sorted_wSUM)
    solns = percent(rounded_wSUM)
    print_pos_soln(solns,title)

#ENSURE RESIDUE CORRESPONDENCE
def Ensure_Correspondence(X_Chain_ID,Y_Chain_ID,file1,file2):
    xlist =[]
    ylist =[]
    x_CAResIDs= getCA_Resseq(file1, X_Chain_ID)
    y_CAResIDs= getCA_Resseq(file2, Y_Chain_ID)

    #REMOVE DISORDERED RESIDUES
    x_disordered = get_Disordered(file1,X_Chain_ID)
    y_disordered = get_Disordered(file2,Y_Chain_ID)

    xremove_disorder,x_ResSeq1 =
    remove_ResID(x_CAResIDs,x_disordered)
    yremove_disorder,y_ResSeq1 =
    remove_ResID(y_CAResIDs,y_disordered)

    #COMPARE PROTEIN1 TO PROTEIN2 AND REMOVE NONMATCHING RESIDUES
    x_Nonmatch, y_Nonmatch = compare(x_ResSeq1,y_ResSeq1)

    #MAKE LIST OF ALL RESIDUES TO BE REMOVED
    x_Remove = x_disordered + x_Nonmatch
    x_Remove.sort()
    y_Remove = y_disordered + y_Nonmatch
    y_Remove.sort()

    #DETERMINE POSITION OF RESIDUES TO BE REMOVED IN PDB FILES
    x_Res_positions = get_Residue_Position(x_CAResIDs, x_Remove)
    y_Res_positions = get_Residue_Position(y_CAResIDs, y_Remove)

```

```

#RETURNS ALL CA COORDINATES
x_CAcoords = transpose(get_CAcoords(file1,X_Chain_ID))
y_CAcoords = transpose(get_CAcoords(file2,Y_Chain_ID))

#Removes CA coordinates of nonmatching residues
xlist = zero_CAcoords(x_CAcoords,x_Res_positions)
ylist = zero_CAcoords(y_CAcoords,y_Res_positions)

#FINAL CHECK TO ENSURE RESIDUE CORRESPONDENCE
n = len(xlist)
j = len(ylist)
if n != j:
    sys.exit("Proteins do not have same number of atoms;
Protein X has",n,"atoms, while Protein Y has",j,"atoms")

#CHECK TO DETERMINE IF APPROPRIATE NUMBER OF COORDINATES PRESENT
if (len(xlist[0]) != 3):
    sys.exit("Protein X does not have a 3xn atom coordinate
set")
if (len(ylist[0]) != 3):
    sys.exit("Protein Y does not have a 3xn atom coordinate
set")

#CHECK TO DETERMINE IF >4 COORDINATES PRESENT FOR EACH PROTEIN
if n < 4:
    sys.exit("Protein X has 3 or less coordinates, 4 or more
needed to perform alignment")
if j < 4:
    sys.exit("Protein Y has 3 or less coordinates, 4 or more
needed to perform alignment")
return xlist,ylist

##WEIGHTED RMSD ALIGNMENT##
def weighted_alignment(x,y,all,scaling_factor):
    All_final_Coords = []

    #TOTAL NUMBER OF ATOMS
    atoms = len(x[0])
    residue = int(atoms * (.10))
    final_wSUM = []

    #10 LOCAL STANDARD ALIGNMENTS (t = counter, goes through 10
iterations; begin = first residue in local section; end = last residue
in local section)
    t = 1
    begin = 0
    end = 10
    while t < 11:
        x_domain = [coords[begin:end] for coords in x]
        y_domain = [coords[begin:end] for coords in y]
        n = len(x_domain[0])

    #TRANSLATE PROTEINS X AND Y TO CENTER
        x_mean = mean(x_domain,n)
        y_mean = mean(y_domain,n)

```

```

x_trans = translation(x_domain, x_mean)
x_translated = nested_list(x_trans,n)
y_trans = translation(y_domain, y_mean)
y_translated = nested_list(y_trans,n)
x_transpose = transpose(x_translated)

#CALCULATE COVARIANCE MATRIX (y_translated *x_translated^t)
R = matrixmultiply(y_translated, x_transpose)
R_transpose = transpose(R)
R2 = matrixmultiply(R_transpose, R)

#DETERMINE THE EIGENVECTORS AND EIGENVALUES of R2
mu,A = LinearAlgebra.eigenvectors(R2)

#SORT EIGENVECTORS IN DECREASING ORDER OF EIGENVALUES
a = [(mu[i],A[i]) for i in range(len(A))]
a.sort()
a.reverse()
mu = [s[0] for s in a]
A = [s[1] for s in a]

#DETERMINE RIGHT-HANDED SYSTEM
A_3 = crossproduct(A[0], A[1])
A = [A[0], A[1], A_3]

#CALCULATE B, NORMALIZED PRODUCT OF (RxA)
B_1 = matrixmultiply(R, A[0])
B_2 = matrixmultiply(R, A[1])
norm_B_1 = normalize(B_1)
norm_B_2 = normalize(B_2)
norm_B_3 = crossproduct(norm_B_1,norm_B_2)
B = [norm_B_1,norm_B_2,norm_B_3]
B_transpose = transpose(B)

#CALCULATE ROTATION MATRIX, U
U = rotation_matrix(B_transpose, A)

#TRANSLATE ALL COORDINATES OF PROTEIN X
s_all_trans = translation(all, x_mean)
j = len(all[0])
s_all_translated = nested_list(s_all_trans,j)
s_all_rot = matrixmultiply(U,s_all_translated)

#ADD MEAN VALUES OF PROTEIN Y TO ROTATED COORDINATES OF
PROTEIN X
s_all_coords = add_coords(s_all_rot, y_mean)
s_all2 = nested_list(s_all_coords,j)
s_allrot = transpose(s_all2)

#TRANSFORM CA COORDINATES ONLY
g = len(x[0])
s_x_ca_trans = translation(x, x_mean)
s_x_ca_translated = nested_list(s_x_ca_trans,g)
s_x_ca_rot = matrixmultiply(U,s_x_ca_translated)
s_x_ca_coords = add_coords(s_x_ca_rot, y_mean)
xt = nested_list(s_x_ca_coords,g)
S_RMSD = sRMSD(xt, y)

```

```

#AFTER LOCAL STANDARD ALIGNMENT, WEIGHTED ALIGNMENT
PERFORMED USING ENTIRE PROTEIN
#z = number of iterations
z = 1
weighted_rmsds = []
w_metric = []
all_list = []

#DETERMINE APPROPRIATE SCALING FACTOR
while z < 501:
    p = len(xt[0])

    #TRANSLATE WEIGHTED CENTROIDS TO ORIGIN
    #CALCULATE WEIGHTS (protein1, protein2,
scaling_factor)
    weights = weight(xt,y,scaling_factor)
    weighted_x_mean = weight_trans(xt,weights,p)
    weighted_y_mean = weight_trans(y,weights,p)

    #USING NEW WEIGHTED TRANSLATION
    w_x_trans = translation(xt, weighted_x_mean)
    w_y_trans = translation(y, weighted_y_mean)
    w_x_translated = nested_list(w_x_trans,p)
    w_y_translated = nested_list(w_y_trans,p)

    #CALCULATE WEIGHTED COVARIENCE MATRIX
    weighted_rot =
weight(w_x_translated,w_y_translated,scaling_factor)
    weighted_x_translated =
multiply(weighted_rot,w_x_translated)
    wx_transpose = transpose(weighted_x_translated)
    R = matrixmultiply(w_y_translated,wx_transpose)
    R_transpose = transpose(R)
    R2 = matrixmultiply(R_transpose,R)

    #DETERMINE THE EIGENVECTORS AND EIGENVALUES of R2
    mu,A = LinearAlgebra.eigenvectors(R2)

    #SORT EIGENVECTORS IN DECREASING ORDER OF EIGENVALUES
    a = [(mu[k],A[k]) for k in range(len(A))]
    a.sort()
    a.reverse()
    mu = [b[0] for b in a]
    A = [b[1] for b in a]

    #DETERMINE RIGHT-HANDED SYSTEM
    A_3 = crossproduct(A[0], A[1])
    A = [A[0], A[1], A_3]

    #CALCULATE B, NORMALIZED PRODUCT OF (RxA)
    B_1 = matrixmultiply(R, A[0])
    B_2 = matrixmultiply(R, A[1])
    norm_B_1 = normalize(B_1)
    norm_B_2 = normalize(B_2)
    norm_B_3 = crossproduct(norm_B_1,norm_B_2)
    B = [norm_B_1,norm_B_2,norm_B_3]

```

```

B_transpose = transpose(B)

#CALCULATE WEIGHTED ROTATION MATRIX, U
U = rotation_matrix(B_transpose, A)
w_x_rot = matrixmultiply(U,w_x_translated)

#CALCULATE WEIGHTED RMSD
w_y_translated,scaling_factor))
weighted_rmsds.append(wRMSD(w_x_rot,

#CALCULATE %wSUM
w_metric.append(wSUM(w_x_rot,
w_y_translated,scaling_factor,p))

#DETERMINE IF CONVERGENCE IS REACHED
if z > 1:
    wRMSD_diff = w_metric[-2] - w_metric[-1]

else:
    wRMSD_diff = []

if 0 < wRMSD_diff < 0.000001:
    #TRANSFORM ALL COORDINATES OF PROTEIN X
    w_all_trans = translation(s_all2,
weighted_x_mean)
    i = len(all[0])
    w_all_translated = nested_list(w_all_trans,i)
    w_all_rot = matrixmultiply(U,w_all_translated)

    #ADD ALL MEAN VALUES OF PROTEIN Y TO ALL
ROTATED COORDINATES OF PROTEIN X
    w_all_coords = add_coords(w_all_rot,
weighted_y_mean)
    w_all = nested_list(w_all_coords,i)
    w_allrot = transpose(w_all)
    All_final_Coords.append(w_allrot)
    iters = z - 1

    #WRITE OUT WEIGHTED RMSD SOLUTION
    final_wSUM.append(w_metric[-1])
    begin = begin + residue
    end = end + residue
    t = t + 1
    break

else:
    #ADD MEAN VALUES OF PROTEIN Y TO ROTATED
COORDINATES OF PROTEIN X
    w_x_coords = add_coords(w_x_rot,
weighted_y_mean)
    xt = nested_list(w_x_coords,p)

    #TRANSFORM ALL COORDINATES OF PROTEIN X
    all_trans = translation(s_all2,
weighted_x_mean)
    i = len(all[0])
    all_translated = nested_list(all_trans,i)

```

```

        all_rot = matrixmultiply(U,all_translated)

        #ADD ALL MEAN VALUES OF PROTEIN Y TO ALL
ROTATED COORDINATES OF PROTEIN X
        all_coords = add_coords(all_rot,
weighted_y_mean)
        s_all2 = nested_list(all_coords,i)
        z = z + 1
    else:
        #print "Solution",t,":Alignment stopped after 1000
iterations, convergence was never reached"
        allrot_nocovg = transpose(s_all2)
        All_final_Coords.append(allrot_nocovg)
        iters = z - 1

        #WRITE OUT WEIGHTED RMSD SOLUTION
        final_wSUM.append(w_metric[-1])
        begin = begin + residue
        end = end + residue
        t = t + 1
    else:
        end
    return final_wSUM,All_final_Coords

#####HELPER FUNCTIONS#####
def getCA_Resseq(filename,chain_key):
    x =[]
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            if residue.has_id("CA"):
                resseq=residue.get_id()[1]
                x.append(resseq)
    return x

def get_Disordered(filename,chain_key):
    x =[]
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            if residue.is_disordered():
                resseq = residue.get_id()[1]
                x.append(resseq)
    return x

def remove_ResID(filename,list):
    result=[]
    result1=[]
    for value in filename:
        if value in list:
            result.append(value)

```



```

        else:
            result1.append(value)
    return result,result1

def compare(x,y):
    x_list = []
    y_list = []
    for each in x:
        if each not in y:
            x_list.append(each)
    for each in y:
        if each not in x:
            y_list.append(each)
    x_list = zero2(x_list)
    y_list = zero2(y_list)
    x_list = sort(x_list)
    y_list = sort(y_list)
    x_list = list(x_list)
    y_list = list(y_list)
    return x_list, y_list

def zero2(x):
    result = []
    for each in x:
        if each != 0:
            result.append(each)
        else:
            continue
    return result

def get_Residue_Position(first,list):
    c = 0
    q = len(list)
    result2 = []
    while c < q:
        result = []
        for each in first:
            result.append(each)
            if each != list[c]:
                continue
        else:
            break
        ans = len(result)
        ans = ans - 1
        result2.append(ans)
        c = c + 1
    return result2

def get_CAcoords(filename,chain_key):
    x =[]
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            if residue.has_id("CA"):
                ca=residue["CA"]

```

```

        x.append(ca.get_coord())
    x_t = transpose(x)
    return x_t

###THANK YOU TO MARK BENSON FOR UPDATING THIS HELPER FUNCTION###
def zero_CAcoords(first,list):
    c = 0
    q = len(list)
    r = len(first)
    s = r - q
    result = array([[s]])
    import copy
    temp = copy.copy(first)
    if q != 0:
        result = copy.copy(first)
        for i in range(len(first)) :
            if i not in list :
                if i + c < r :
                    temp[i] = first[i+c]
            else:
                c = c + 1

                result[i] = [0,0,0]
                if i + c < r :
                    temp[i] = first[i+c]
        a = temp[:s]
        return a
    else:
        return first

def getAll_Coords(filename):
    result =[]
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain = model.get_list()
        for each in chain:
            for res in each.get_list():
                for x in res.get_list():
                    result.append(x.get_coord())
    x_t = transpose(result)
    return x_t

def getStructure(filename):
    parser = PDBParser()
    structure = parser.get_structure(filename.split('.')[0],
filename)
    return structure

def writeStructure(structure,filename):
    from Bio.PDB.PDBIO import PDBIO
    import sys
    io = PDBIO()
    io.set_structure(structure)
    io.save(filename)

def setAll(structure,newCACoords):

```

```

allCAs = []
for model in structure.get_list():
    chain = model.get_list()
    for each in chain:
        for res in each.get_list():
            for x in res.get_list():
                allCAs.append(x)
if len(allCAs) != len(newCACoords):
    print "wrong number of atoms .. structure
had", len(allCAs), "you gave me", len(newCACoords)
    raise Exception("wrong number of atoms")
for newCoords, ca in zip(newCACoords, allCAs):
    #print newCoords, ca.get_coord()
    ca.set_coord(newCoords)
    #print ca.get_coord()
    #raise Exception("time to stop")

def mean(first, n):
    return [sum(each)/n for each in first]

def translation(first, second):
    c = 0
    k = []
    q = len(first)
    while c < q:
        for each in first[c]:
            subtr = each - second[c]
            k.append(subtr)
        c = c + 1
    return k

def nested_list(name, n):
    first1 = name[0:n]
    first2 = name[n:2*n]
    first3 = name[2*n:3*n]
    first_translated = [first1, first2, first3]
    return first_translated

def crossproduct(a, b):
    C_0 = a[1]*b[2] - a[2]*b[1]
    C_1 = a[2]*b[0] - a[0]*b[2]
    C_2 = a[0]*b[1] - a[1]*b[0]
    return [C_0, C_1, C_2]

def normalize(a):
    B_0 = (a[0])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    B_1 = (a[1])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    B_2 = (a[2])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    return [B_0, B_1, B_2]

def rotation_matrix(first, second):
    U = matrixmultiply(first, second)
    return U

def sqr(matrix):
    k = []
    for each in matrix:

```

```

        sq = (each)**2
        k.append(sq)
    return k

def sqroot(matrix):
    j = []
    for each in matrix:
        sqroot = sqrt(each)
        j.append(sqroot)
    return j

def sRMSD(first,second):
    first = array(first)
    second = array(second)
    subtr = first - second
    def sqr(matrix):
        k = []
        for each in matrix:
            sq = (each)**2
            k.append(sq)
        return k
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    def sqroot(matrix):
        j = []
        for each in matrix:
            sqroot = sqrt(each)
            j.append(sqroot)
        return j
    d = sqroot(sum_subtr_s)
    sq_d = sqr(d)
    s_sq_d = sum(sq_d)
    tot = (len(first[0]))
    value = sqrt(s_sq_d/tot)
    return value

def add_coords(first,second):
    c = 0
    k = []
    q = len(first)
    while c < q:
        for each in first[c]:
            add = each + second[c]
            k.append(add)
        c = c + 1
    return k

def weight_trans(first,weight,n):
    mult = multiply(first, weight)
    sum_mult = sum(mult)
    mean = [sum(each)/n for each in mult]
    return mean

def weight(first,second,constant):
    first = array(first)
    second = array(second)
    subtr = first - second

```

```

subtr_s = sqr(subtr)
sum_subtr_s = sum(subtr_s)
d = sqrt(sum_subtr_s)
weighted_d = Gaussian(d, constant)
weighted_d = Gaussian2(weighted_d)
return weighted_d

def wSUM(first, second, constant, atoms):
    first = array(first)
    second = array(second)
    subtr = first - second
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    d = sqrt(sum_subtr_s)
    weighted_d = Gaussian(d, constant)
    weighted_d = Gaussian2(weighted_d)
    sum_weighted_d = sum(weighted_d)
    value = sum_weighted_d/atoms
    return value

def wRMSD(first, second, constant):
    first = array(first)
    second = array(second)
    subtr = first - second
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    d = sqrt(sum_subtr_s)
    weighted_d = Gaussian(d, constant)
    weights = Gaussian2(weighted_d)
    sq_d = sqr(d)
    wd = multiply(sq_d, weights)
    s_wd = sum(wd)
    n = len(d)
    s_sq_d_divide = s_wd/n
    value = sqrt(s_sq_d_divide)
    return value

def Gaussian(first, z):
    value = []
    for each in first:
        weight = (-((each)**2)/z)
        value.append(weight)
    return value

def Gaussian2(first):
    value = []
    for each in first:
        weight = exp(each)
        value.append(weight)
    return value

def Compare_Solns(file, t):
    list = [t + 1]
    c = t + 1
    while c < 10:
        SRMSD = sRMSD(transpose(file[c]), transpose(file[t]))
        c = c + 1

```

```

        if SRMSD < 0.5:
            list.append(c)
    return list

def Unique_Coords(file):
    t = 0
    all_answer = []
    while t < 9:
        if t == 0:
            answer = Compare_Solns(file,t)
            all_answer.append(answer)
        else:
            for each in all_answer:
                if t + 1 not in each:
                    answer = Compare_Solns(file,t)
                else:
                    answer = [0]
                    break
            all_answer.append(answer)
        t = t + 1
    for each in all_answer:
        if 10 not in each:
            all_answer.append([10])
        break
    return all_answer

def round_decimal(file):
    rounded_wSUM = []
    for each in file:
        rounded_wSUM.append(round(each,4))
    return rounded_wSUM

def percent(file):
    catch = []
    for each in file:
        a = each * 100
        catch.append(a)
    return catch

def Coord_Positions(file, coords, s, final_wSUM, title):
    new_list = []
    all_wSUM = []
    ALL_first = []
    all_soln_coords = []
    for each in file:
        if each != [0]:
            new_list.append(each)
    for num in new_list:
        first = num[0]
        ALL_first.append(first)
    for value in ALL_first:
        wSUM = final_wSUM[value -1]
        all_wSUM.append(wSUM)
        soln_coords = (coords[value - 1])
        all_soln_coords.append(soln_coords)
    #Sort coords in order of %wSUM

```

```

    a = [(all_wSUM[k],all_soln_coords[k]) for k in
range(len(all_soln_coords))]
    a.sort()
    a.reverse()
    all_wSUM = [b[0] for b in a]
    all_soln_coords = [b[1] for b in a]
    count = 1
    #Write out weighted solns
    for sorted_coords in all_soln_coords:
        setAll(s,sorted_coords)
        writeStructure(s,'%s_wRMSD_%s.pdb'%(title[0],count))
        count = count + 1
    return all_wSUM

def print_pos_soln(file1,title):
    n = len(file1)
    c = 0
    while c < n:
        print title[0],"wRMSD",c+1,"corresponds to %wSUM
of",file1[c],"%"
        c = c + 1

##### HELPER FUNCTIONS IN ENSURE RESIDUE CORRESPONDENCE FUNCTION #####
##### THIS HAS BEEN MODIFIED TO WORK WITH WRMSD CODE #####
# Copyright (C) 2002, Thomas Hamelryck (thamelry@vub.ac.be)
# This code is part of the Biopython distribution and governed by its
# license. Please see the LICENSE file that should have been included
# as part of this package.

# Python stuff
import sys
from string import split
from Numeric import array, Float0

# My stuff
from Bio.PDB.StructureBuilder import StructureBuilder
from Bio.PDB.PDBExceptions import PDBConstructionException
from Bio.PDB.parse_pdb_header import _parse_pdb_header_list

__doc__="Parser for PDB files."

# If PDB spec says "COLUMNS 18-20" this means line[17:20]

class PDBParser:
    """
    Parse a PDB file and return a Structure object.
    """

    def __init__(self, PERMISSIVE=0, get_header=0,
structure_builder=None):
        """
        The PDB parser call a number of standard methods in an
aggregated

```

StructureBuilder object. Normally this object is instantiated by the PDBParser object itself, but if the user provides his own StructureBuilder object, the latter is used instead.

Arguments:

o PERMISSIVE - int, if this is 0 exceptions in constructing the SMCRA data structure are fatal. If 1 (DEFAULT), the exceptions are

caught, but some residues or atoms will be missing. THESE EXCEPTIONS

ARE DUE TO PROBLEMS IN THE PDB FILE!.

o structure_builder - an optional user implemented StructureBuilder class.

```
"""
if structure_builder!=None:
    self.structure_builder=structure_builder
else:
    self.structure_builder=StructureBuilder()
self.header=None
self.trailer=None
self.line_counter=0
self.PERMISSIVE=PERMISSIVE
```

Public methods

```
def get_structure(self, id, file):
    """Return the structure.
```

Arguments:

o id - string, the id that will be used for the structure
o file - name of the PDB file OR an open filehandle

```
"""
self.header=None
self.trailer=None
# Make a StructureBuilder instance (pass id of structure as
parameter)
self.structure_builder.init_structure(id)
if isinstance(file, basestring):
    file=open(file)
self._parse(file.readlines())
file.close()
self.structure_builder.set_header(self.header)
# Return the Structure instance
return self.structure_builder.get_structure()
```

```
def get_header(self):
    "Return the header."
    return self.header
```

```
def get_trailer(self):
    "Return the trailer."
    return self.trailer
```

Private methods


```

def _parse(self, header_coords_trailer):
    "Parse the PDB file."
    # Extract the header; return the rest of the file
    self.header,
coords_trailer=self._get_header(header_coords_trailer)
    # Parse the atomic data; return the PDB file trailer
    self.trailer=self._parse_coordinates(coords_trailer)

def _get_header(self, header_coords_trailer):
    "Get the header of the PDB file, return the rest."
    structure_builder=self.structure_builder
    for i in range(0, len(header_coords_trailer)):
        structure_builder.set_line_counter(i+1)
        line=header_coords_trailer[i]
        record_type=line[0:6]
        if(record_type=='ATOM ' or record_type=='HETATM' or
record_type=='MODEL '):
            break
        header=header_coords_trailer[0:i]
        # Return the rest of the coords+trailer for further processing
        self.line_counter=i
        coords_trailer=header_coords_trailer[i:]
        header_dict=_parse_pdb_header_list(header)
        return header_dict, coords_trailer

def _parse_coordinates(self, coords_trailer):
    "Parse the atomic data in the PDB file."
    local_line_counter=0
    structure_builder=self.structure_builder
    current_model_id=0
    # Flag we have an open model
    model_open=0
    current_chain_id=None
    current_segid=None
    current_residue_id=None
    current_resname=None
    for i in range(0, len(coords_trailer)):
        line=coords_trailer[i]
        record_type=line[0:6]
        global_line_counter=self.line_counter+local_line_counter+1
        structure_builder.set_line_counter(global_line_counter)
        if(record_type=='ATOM ' or record_type=='HETATM'):
            # Initialize the Model - there was no explicit MODEL
record
            if not model_open:
                structure_builder.init_model(current_model_id)
                current_model_id+=1
                model_open=1
            fullname=line[12:16]
            # get rid of whitespace in atom names
            split_list=split(fullname)
            if len(split_list)!=1:
                # atom name has internal spaces, e.g. " N B ", so
                # we do not strip spaces
                name=fullname
            else:
                # atom name is like " CA ", so we can strip spaces

```

```

        name=split_list[0]
    altloc=line[16:17]
    resname=line[17:20]
    chainid=line[21:22]
    try:
        serial_number=int(line[6:11])
    except:
        serial_number=0
    resseq=int(split(line[22:26])[0]) # sequence
identifier
    icode=line[26:27] # insertion code
    if record_type=='HETATM': # hetero atom flag
        if resname=="HOH" or resname=="WAT":
            hetero_flag="W"
        else:
            hetero_flag="H"
    else:
        hetero_flag=" "
    residue_id=(hetero_flag, resseq, icode)
    # atomic coordinates
    x=float(line[30:38])
    y=float(line[38:46])
    z=float(line[46:54])
    coord=array((x, y, z), Float0)
    # occupancy & B factor
    occupancy=float(line[54:60])
    bfactor=float(line[60:66])
    segid=line[72:76]
    if current_segid!=segid:
        current_segid=segid
        structure_builder.init_seg(current_segid)
    if current_chain_id!=chainid:
        current_chain_id=chainid
        structure_builder.init_chain(current_chain_id)
        current_residue_id=residue_id
        current_resname=resname
    try:
        structure_builder.init_residue(resname,
hetero_flag, resseq, icode)
    except PDBConstructionException, message:
        self._handle_PDB_exception(message,
global_line_counter)
        elif current_residue_id!=residue_id or
current_resname!=resname:
            current_residue_id=residue_id
            current_resname=resname
        try:
            structure_builder.init_residue(resname,
hetero_flag, resseq, icode)
        except PDBConstructionException, message:
            self._handle_PDB_exception(message,
global_line_counter)
        # init atom
        try:
            structure_builder.init_atom(name, coord, bfactor,
occupancy, altloc, fullname, serial_number)
        except PDBConstructionException, message:

```

```

        self._handle_PDB_exception(message,
global_line_counter)
        elif(record_type=='ANISOU'):
            anisou=map(float, (line[28:35], line[35:42],
line[43:49], line[49:56], line[56:63], line[63:70]))
            # U's are scaled by 10^4
            anisou_array=(array(anisou,
Float0)/10000.0).astype(Float0)
            structure_builder.set_anisou(anisou_array)
        elif(record_type=='MODEL '):
            structure_builder.init_model(current_model_id)
            current_model_id+=1
            model_open=1
            current_chain_id=None
            current_residue_id=None
        elif(record_type=='END ' or record_type=='CONNECT'):
            # End of atomic data, return the trailer
            self.line_counter=self.line_counter+local_line_counter
            return coords_trailer[local_line_counter:]
        elif(record_type=='ENDMDL'):
            model_open=0
            current_chain_id=None
            current_residue_id=None
        elif(record_type=='SIGUIJ'):
            # standard deviation of anisotropic B factor
            siguij=map(float, (line[28:35], line[35:42],
line[42:49], line[49:56], line[56:63], line[63:70]))
            # U sigma's are scaled by 10^4
            siguij_array=(array(siguij,
Float0)/10000.0).astype(Float0)
            structure_builder.set_siguij(siguij_array)
        elif(record_type=='SIGATM'):
            # standard deviation of atomic positions
            sigatm=map(float, (line[30:38], line[38:45],
line[46:54], line[54:60], line[60:66]))
            sigatm_array=array(sigatm, Float0)
            structure_builder.set_sigatm(sigatm_array)
            local_line_counter=local_line_counter+1
        # EOF (does not end in END or CONNECT)
        self.line_counter=self.line_counter+local_line_counter
        return []

def _handle_PDB_exception(self, message, line_counter):
    """
    This method catches an exception that occurs in the
StructureBuilder
    object (if PERMISSIVE==1), or raises it again, this time adding
the
    PDB line number to the error message.
    """
    message="%s at line %i." % (message, line_counter)
    if self.PERMISSIVE:
        # just print a warning - some residues/atoms will be
missing
        print "PDBConstructionException: %s" % message
        print "Exception ignored.\nSome atoms or residues will be
missing in the data structure."

```

```

        else:
            # exceptions are fatal - raise again with new message
            (including line nr)
            raise PDBConstructionException, message

if __name__=="__main__":

    import sys

    p=PDBParser(PERMISSIVE=1)

    s=p.get_structure("scr", sys.argv[1])

    for m in s.get_iterator():
        p=m.get_parent()
        assert(p is s)
        for c in m.get_iterator():
            p=c.get_parent()
            assert(p is m)
            for r in c.get_iterator():
                p=r.get_parent()
                assert(p is c)
                for a in r.get_iterator():
                    p=a.get_parent()
                    if not p is r:
                        print p, r

#RUN Local_wRMSD.py
if __name__ == "__main__":
    if len(sys.argv) != 5:
        print "usage: Local_wRMSD.py Protein_X.pdb
Protein_X_ChainID Protein_Y.pdb Protein_Y_ChainID"
        sys.exit()
    filename1 = sys.argv[1]
    filename2 = sys.argv[3]
    X_Chain_ID = sys.argv[2]
    Y_Chain_ID = sys.argv[4]
    run_Local_wRMSD(filename1,filename2,X_Chain_ID,Y_Chain_ID)

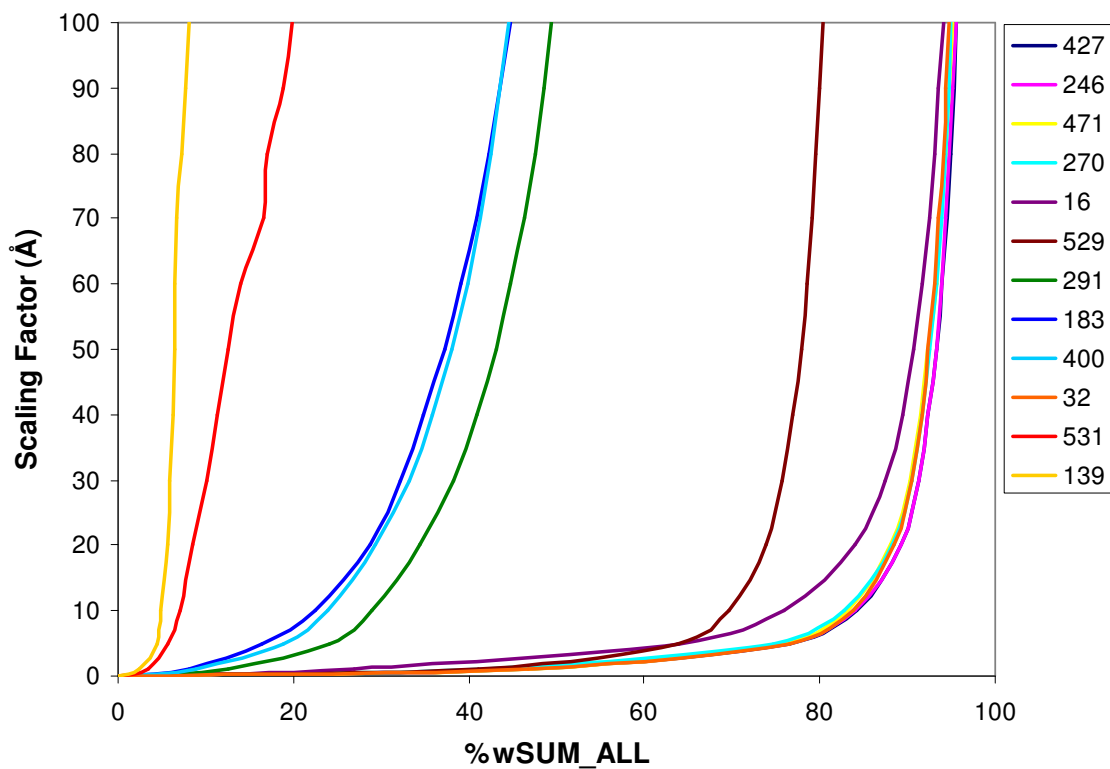
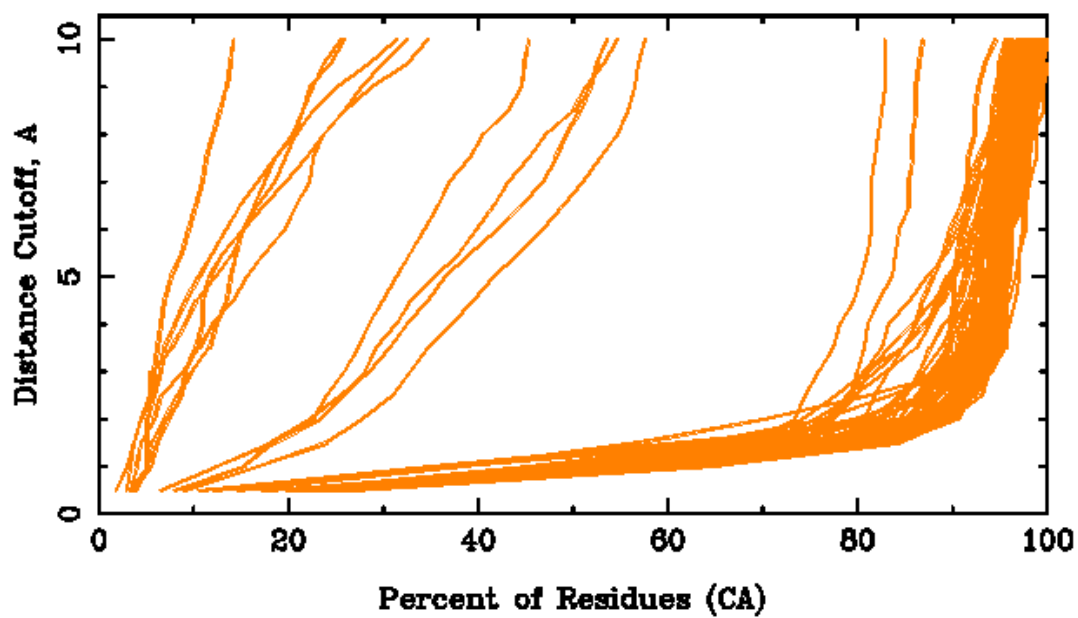
```

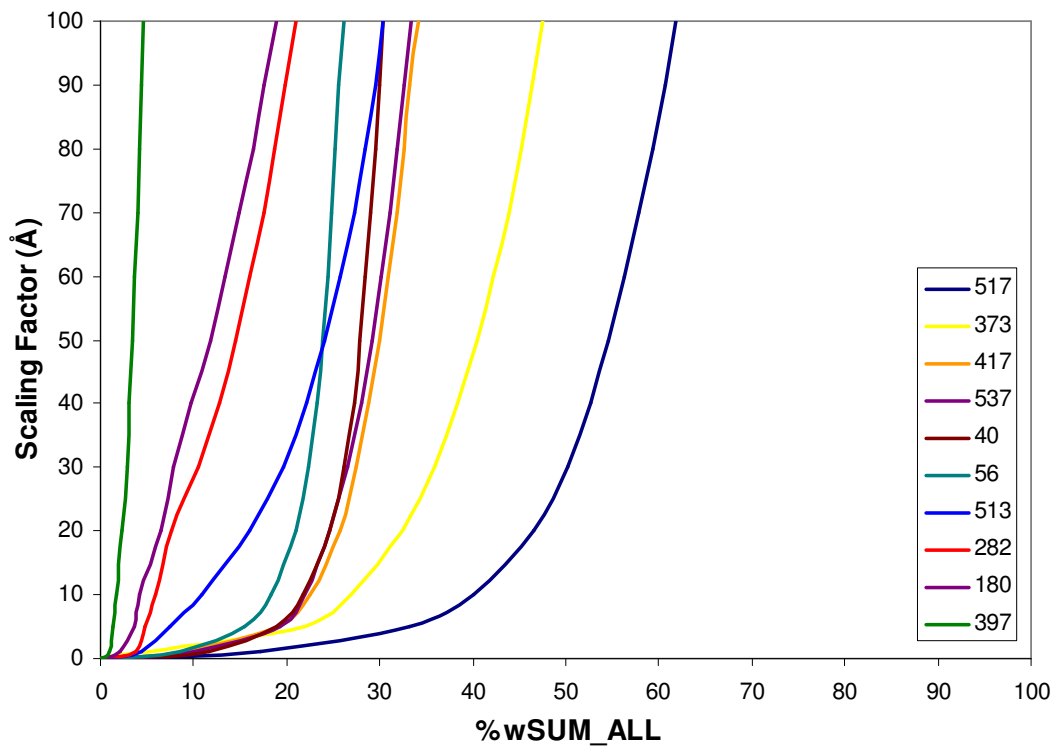
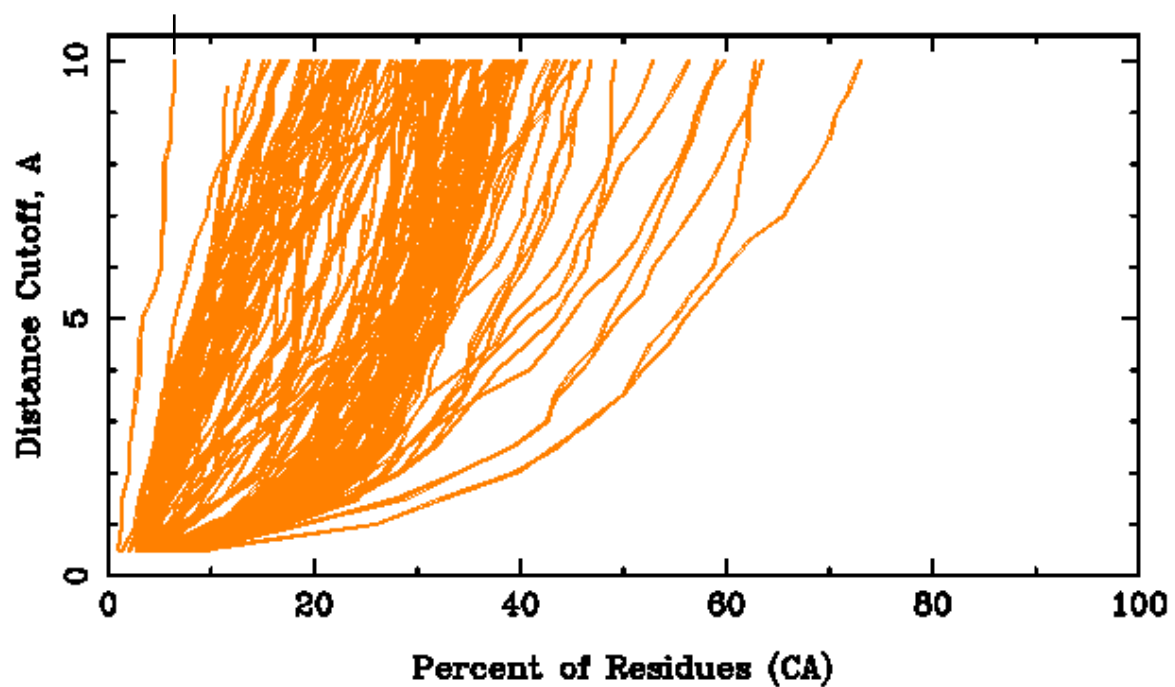
APPENDIX 2

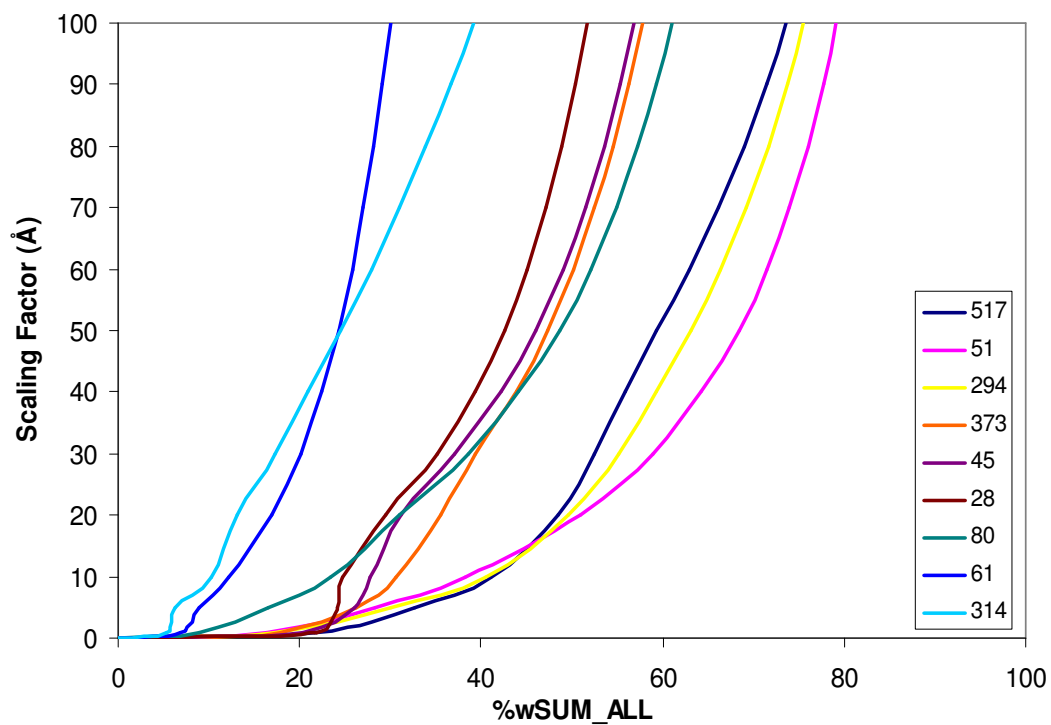
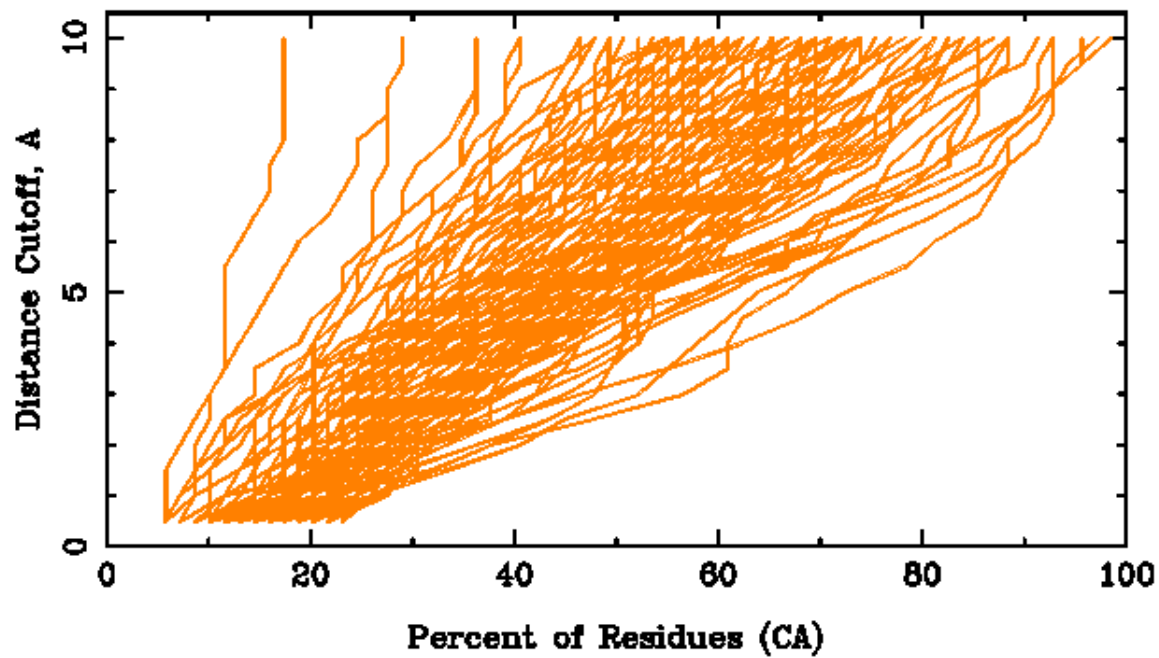
Application of the wRMSD Method to Predicted Protein Structures and Homologous Proteins

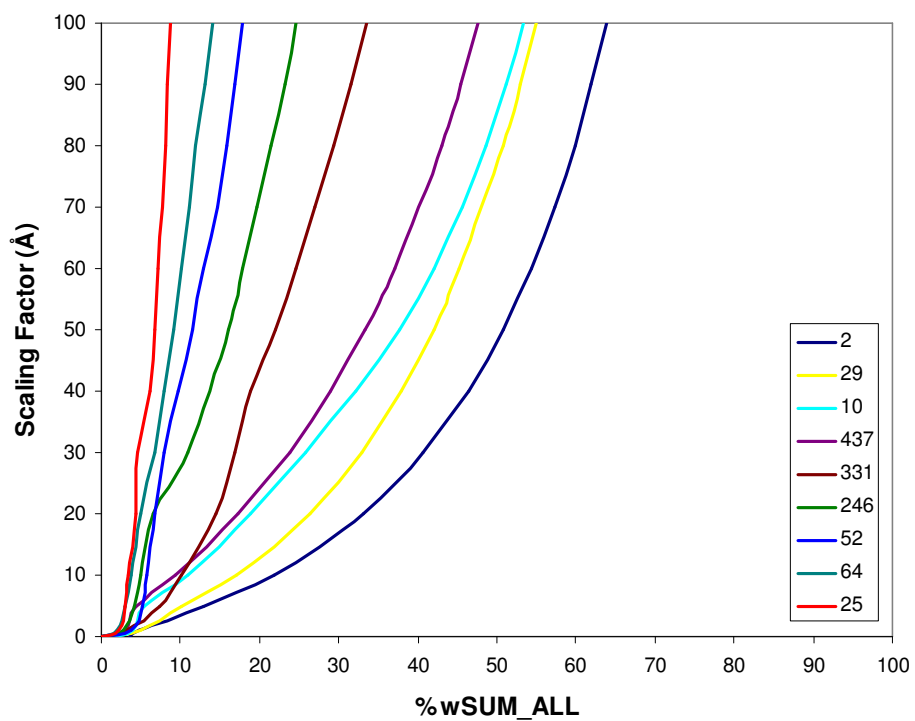
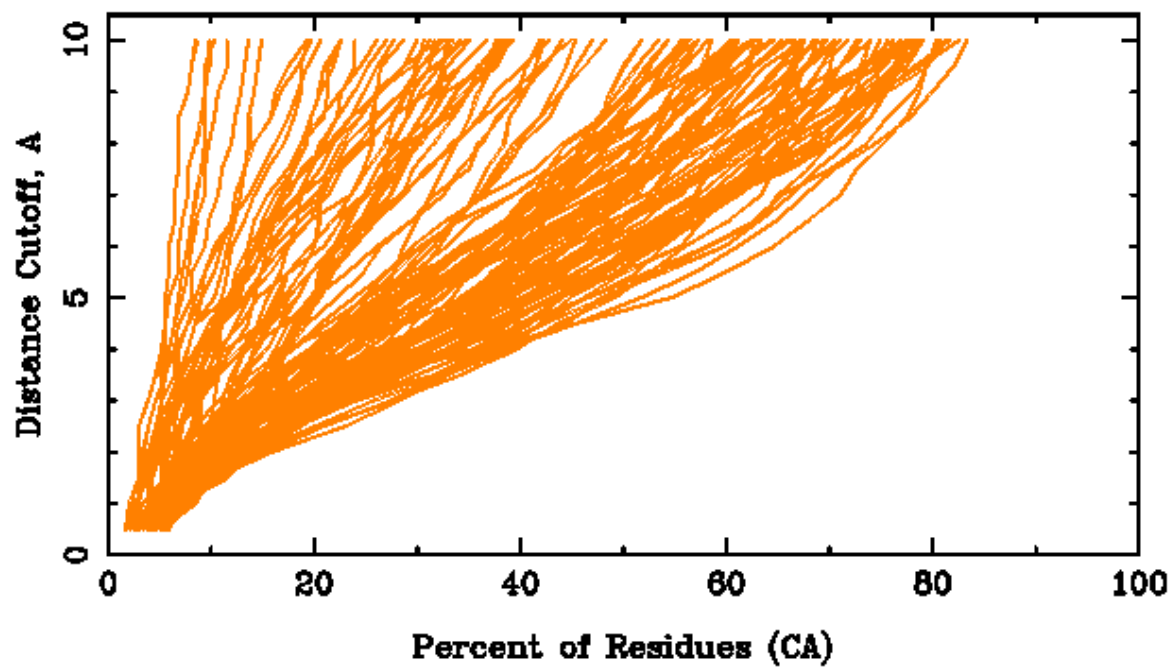
A2.1 RMS/coverage graphs.

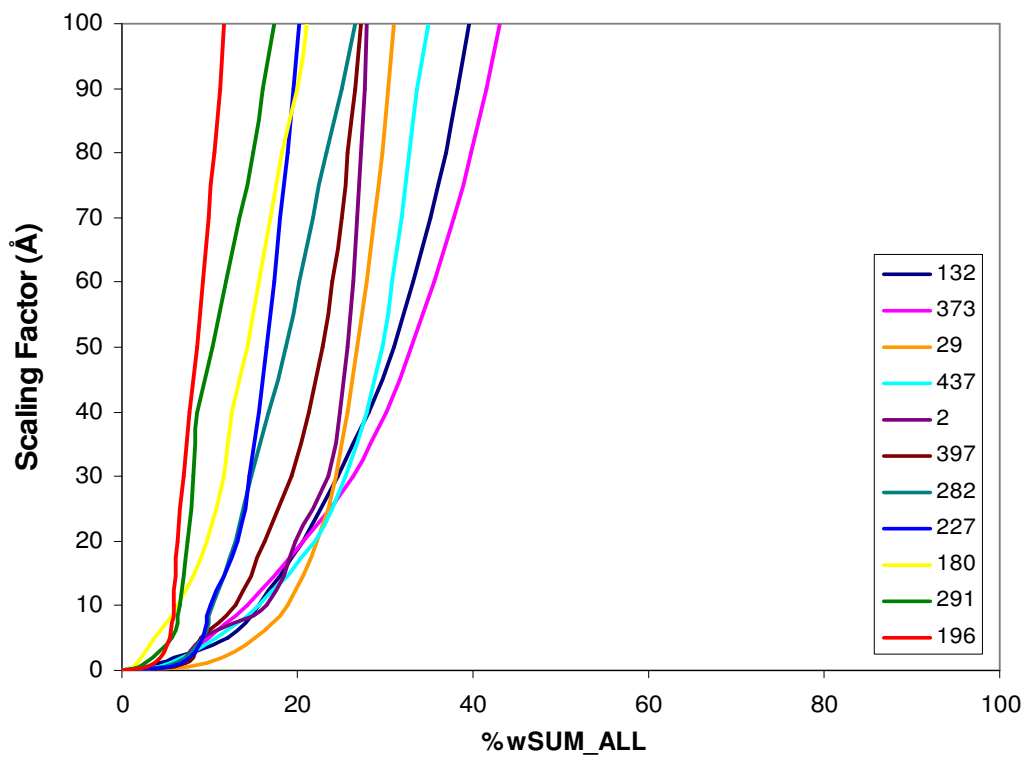
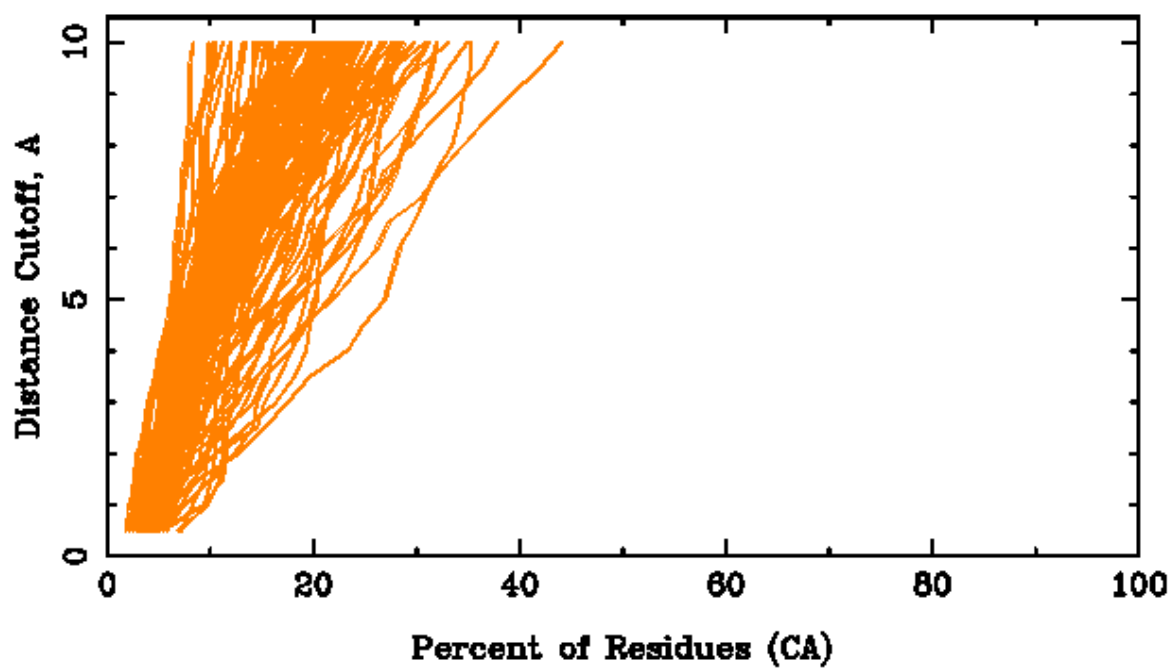
The wRMSD technique was used to create RMS/coverage graphs by plotting the %wSUM versus c . As the scaling factor c increases, %wSUM also increases in a manner similar to RMS/coverage graphs from GDT. The top graph corresponds to the wRMSD metric and the bottom to the GDT_TS metric.

TARGET 179 (Comparative Modeling)**T0179**

TARGET 172 (Comparative Modeling/Fold Recognition)**T0172**

TARGET 170 (Fold Recognition/New Fold)**T0170**

TARGET 147 (Fold Recognition)**T0147**

TARGET 162-3 (New Fold)**T0162**

A2.2 PDB IDs and references for crystal structures obtained from the HOMSTRAD Database

1FJM-Goldberg, J., Huang, H. B., Kwon, Y. G., Greengard, P., Nairn, A. C., Kuriyan, J. (1995). Three-dimensional structure of the catalytic subunit of protein serine/threonine phosphatase-1. *Nature* 376, 745-753.

1TCO- Griffith, J. P., Kim, J. L., Kim, E. E., Sintchak, M. D., Thomson, J. A., Fitzgibbon, M. J., Fleming, M. A., Caron, P. R., Hsiao, K., Navia, M. A. (1995). X-ray structure of calcineurin inhibited by the immunophilin-immunosuppressant FKBP12-FK506 complex. *Cell* 82, 507-522.

1M9W- Monleon, D., Celda, B. Study of electrostatic potential surface distribution using high resolution side-chain conformation determined by NMR. To be Published.

2HGS- Polekhina, G., Board, P. G., Gali, R. R., Rossjohn, J., Parker, M. W. (1999). Molecular basis of glutathione synthetase deficiency and a rare gene permutation event. *EMBO J* 18, 3204-3213.

1AU1- Karpusas, M., Nolte, M., Benton, C. B., Meier, W., Lipscomb, W. N., Goelz, S. (1997). The crystal structure of human interferon beta at 2.2-Å resolution. *Proc Natl Acad Sci USA* 94, 11813-11818.

1ITF- Klaus, W., Gsell, B., Labhardt, A. M., Wipf, B., Senn, H. (1997). The three-dimensional high resolution structure of human interferon alpha-2a determined by heteronuclear NMR spectroscopy in solution. *J Mol Biol* 274, 661-675.

1I7B- Tolbert, W. D., Ekstrom, J. L., Mathews, I. I., Secrist 3rd., J. A., Kapoor, P., Pegg, A. E., Ealick, S. E. (2001). The structural basis for substrate specificity and inhibition of human S-adenosylmethionine decarboxylase. *Biochemistry* 40, 9484-9494.

1MHM- Bennett, E. M., Ekstrom, J. L., Pegg, A. E., Ealick, S. E. (2002). Monomeric S-Adenosylmethionine Decarboxylase from Plants Provides an Alternative to Putrescine Stimulation. *Biochemistry* 41, 14509-14517.

1EPW- Swaminathan, S., Eswaramoorthy, S. (2000). Structural analysis of the catalytic and binding sites of Clostridium botulinum neurotoxin B. *Nat Struct Biol* 4, 693-699.

3BTA- Lacy, D. B., Tepp, W., Cohen, A. C., DasGupta, B. R., Stevens, R. (1998). Crystal structure of botulinum neurotoxin type A and implications for toxicity. *Nat Struct Biol* 5, 898-902.

1AUK- Lukatela, G., Krauss, N., Theis, K., Selmer, T., Gieselmann, V., von Figura, K., Saenger, W. (1998). Crystal structure of human arylsulfatase A: the aldehyde function

and the metal ion at the active site suggest a novel mechanism for sulfate ester hydrolysis. *Biochemistry* 37, 3654-3664.

1FSU- Bond, C. S., Clements, P. R., Ashby, S. J., Collyer, C. A., Harrop, S. J., Hopwood, J. J., Guss, J. M. (1997). Structure of a human lysosomal sulfatase. *Structure* 5, 277-289.

3PCG- Orville, A. M., Elango, N., Lipscomb, J. D., Ohlendorf, D. H. (1997). Structures of competitive inhibitor complexes of protocatechuate 3,4-dioxygenase: multiple exogenous ligand binding orientations within the active site. *Biochemistry* 36, 10039-10051.

1A3G- Okada, K., Hirotsu, K., Sato, M., Hayashi, H., Kagamiyama, H. (1997). Three-dimensional structure of *Escherichia coli* branched-chain amino acid aminotransferase at 2.5 Å resolution. *J Biochem (Tokyo)* 121, 637-641.

5DAA- van Ophem, P.W., Peisach, D., Erickson, S.D., Soda, K., Ringe, D., Manning, J.M. (1999). Effects of the E177K mutation in D-amino acid transaminase. Studies on an essential coenzyme anchoring group that contributes to stereochemical fidelity. *Biochemistry* 38, 1323-1331.

1IPA- Nureki, O., Shirouzu, M., Hashimoto, K., Ishitani, R., Terada, T., Tamakoshi, M., Oshima, T., Chijimatsu, M., Takio, K., Vassylyev, D. G., Shibata, T., Inoue, Y., Kuramitsu, S. & Yokoyama, S. (2002). An enzyme with a deep trefoil knot for the active-site architecture. *Acta Crystallogr D Biol Crystallogr* 58, 1129-37.

1GZ0- Michel, G., Sauve, V., Larocque, R., Li, Y., Matte, A. & Cygler, M. (2002). The structure of the RlmB 23S rRNA methyltransferase reveals a new methyltransferase fold with a unique knot. *Structure (Camb)* 10, 1303-15.

1OYC- Fox, K. M., Karplus, P. A. (1994). Old yellow enzyme at 2 Å resolution: overall structure, ligand binding, and comparison with related flavoproteins. *Structure* 2, 1089-1105.

2TMD- Barber, M. J., Neame, P. J., Lim, L. W., White, S., Matthews, F. S. (1992). Correlation of x-ray deduced and experimental amino acid sequences of trimethylamine dehydrogenase. *J Biol Chem* 267, 6611-6619.

1IQ8- Ishitani, R., Nureki, O., Fukai, S., Kijimoto, T., Nameki, N., Watanabe, M., Kondo, H., Sekine, M., Okada, N., Nishimura, S., Yokoyama, S. (2002). Crystal structure of archaeosine tRNA-guanine transglycosylase. *J Mol Biol* 318, 665-677.

1K4G- Meyer, E. A., Brenk, R., Castellano, R. K., Furler, M., Klebe, G., Diederich, F. (2002). De novo design, synthesis, and in vitro evaluation of inhibitors for prokaryotic tRNA-guanine transglycosylase: a dramatic sulfur effect on binding affinity. *ChemBioChem* 3, 250-253.

- 1GLN-** Nureki, O., Vassylyev, D. G., Katayanagi, K., Shimizu, T., Sekine, S., Kigawa, T., Miyazawa, T., Yokoyama, S., Morikawa, K. (1995). Architectures of class-defining and specific domains of glutamyl-tRNA synthetase. *Science* 267, 1958-1965.
- 1QTQ-** Rath, V. L., Silvan, L. F., Beijer, B., Sproat, B. S., Steitz, T. A. (1998). How glutaminyl-tRNA synthetase selects glutamine. *Structure* 6, 439-449.
- 1BOO-** Gong, W., O'Gara, M., Blumenthal, R. M., Cheng, X. (1997). Structure of pvu II DNA-(cytosine N4) methyltransferase, an example of domain permutation and protein fold assignment. *Nucleic Acids Res* 25, 2702-2715.
- 1EG2-** Scavetta, R. D., Thomas, C. B., Walsh, M. A., Szegedi, S., Joachimiak, A., Gumpert, R. I., Churchill, M. E. (2000). Structure of RsrI methyltransferase, a member of the N6-adenine beta class of DNA methyltransferases. *Nucleic Acids Res* 28, 3950-3961.
- 1AB4-** Cabral, J. H., Jackson, A. P., Smith, C. V., Shikotra, N., Maxwell, A., Liddington, R. C. (1997). Crystal structure of the breakage-reunion domain of DNA gyrase. *Nature* 388, 903-906.
- 1BJT-** Fass, D., Bogden, C. E., Berger, J. M. (1999). Quaternary changes in topoisomerase II may direct orthogonal movement of two DNA strands. *Nat Struct Biol* 6, 322-326.
- 1TDJ-** Gallagher, D. T., Gilliland, G. L., Xiao, G., Zondlo, J., Fisher, K. E., Chinchilla, D., Eisenstein, E. (1998). Structure and control of pyridoxal phosphate dependent allosteric threonine deaminase. *Structure* 6, 465-475.
- 2TYS-** Rhee, S., Parris, K. D., Hyde, C. C., Ahmed, S. A., Miles, E. W., Davies, D. R. (1997). Crystal structures of a mutant (betaK87T) tryptophan synthase alpha2beta2 complex with ligands bound to the active sites of the alpha- and beta-subunits reveal ligand-induced conformational changes. *Biochemistry* 36, 7664-7680.
- 1BK0-** Roach, P. L., Clifton, I. J., Hensgens, C. M., Shibata, N., Schofield, C. J., Hajdu, J., Baldwin, J. E. (1997). Structure of isopenicillin N synthase complexed with substrate and the mechanism of penicillin formation. *Nature* 387, 827-830.
- 1DCS-** Valegard, K., van Scheltinga, A. C., Lloyd, M. D., Hara, T., Ramaswamy, S., Perrakis, A., Thompson, A., Lee, H. J., Baldwin, J. E., Schofield, C. J., Hajdu, J., Andersson, I. (1998). Structure of a cephalosporin synthase. *Nature* 394, 805-809.
- 1FFV-** Hanzelmann, P., Dobbek, H., Gremer, L., Huber, R., Meyer, O. (2000). The effect of intracellular molybdenum in *Hydrogenophaga pseudoflava* on the crystallographic structure of the seleno-molybdo-iron-sulfur flavoenzyme carbon monoxide dehydrogenase. *J Mol Biol* 301, 1221-1235.

- 1FO4-** Enroth, C., Eger, B. T., Okamoto, K., Nishino, T., Nishino, T., Pai, E. F. (2000). Crystal structures of bovine milk xanthine dehydrogenase and xanthine oxidase: structure-based mechanism of conversion. *Proc Natl Acad Sci USA* 97, 10723-10728.
- 1A9N-** Price, S. R., Evans, P. R., Nagai, K. (1998). Crystal structure of the spliceosomal U2B"-U2A' protein complex bound to a fragment of U2 small nuclear RNA. *Nature* 394, 645-650.
- 1D0B-** Marino, M., Braun, L., Cossart, P., Ghosh, P. (1999). Structure of the InlB leucine-rich repeats, a domain that triggers host cell invasion by the bacterial pathogen *L. monocytogenes*. *Mol Cell* 4, 1063-1072.
- 1B74-** Hwang, K. Y., Cho, C. S., Kim, S. S., Sung, H. C., Yu, Y. G., Cho, Y. (1999). Structure and mechanism of glutamate racemase from *Aquifex pyrophilus*. *Nat Struct Biol* 6, 422-426.
- 1JFL-** Liu, L., Iwata, K., Kita, A., Kawarabayasi, Y., Yohda, M., Miki, K. (2002). Crystal structure of aspartate racemase from *Pyrococcus horikoshii* OT3 and its implications for molecular mechanism of PLP-independent racemization. *J Mol Biol* 319, 479-489.
- 1CB8-** Fethiere, J., Eggimann, B., Cygler, M. (1999). Crystal structure of chondroitin AC lyase, a representative of a family of glycosaminoglycan degrading enzymes. *J Mol Biol* 288, 635-647.
- 1EGU-** Li, S., Kelly, S. J., Lamani, E., Ferraroni, M., Jedrzejewski, M. J. (2000). Structural basis of hyaluronan degradation by *Streptococcus pneumoniae* hyaluronate lyase. *Embo. J.* 19, 1228-1240.
- 1FOH-** Enroth, C., Neujahr, H., Schneider, G., Lindqvist, Y. (1998). The crystal structure of phenol hydroxylase in complex with FAD and phenol provides evidence for a concerted conformational change in the enzyme and its cofactor during catalysis. *Structure* 6, 605-617.
- 1PB3-** Mesecar, A. D., Koshland Jr., D. E. (2000). Sites of Binding and Orientation in a Four-Location Model for Protein Stereospecificity. *IUBMB Life* 49, 457-466.

A2.3 Homologous wRMSD code.

```
#!/usr/bin/env python

""" REQUIRED INSTALLATIONS FOR A PC:
    -Python 2.5
    -Scipy 0.5.2
    -NumPy 1.0.1
    -Biopython 1.42
    -Numerical 24.2 (install through Biopython website)
    -mxTextTools: http://www.egenix.com/files/python/mxTextTools.html
    -BLAST (bl2seq), Tutorial on downloading BLAST on a PC-
http://www.people.vcu.edu/~elhaij/IntroBioinf/Links/DownloadBlast.html

INPUT REQUIREMENTS:
    -2 PDB files
        PDB file must be in correct PDB format (i.e. chain ID's
        present, unique atom name within each residue, occupancy, etc...)

    -This script ignores Hetgroups. If you have an atypical residue
    in your sequence that is labeled as an HETATM (i.e. MSE), it will be
    ignored. To have it included in the sequence alignment as an 'X',
    manually change the label from HETATM to ATOM.

Global_HwRMSD.py INFORMATION
    NOTE: For similar structures (characterized by a small sRMSD;
    example: sRMSD < 5), the scaling factor is set to 2
        For nonsimilar structures (characterized by a large sRMSD;
    example: sRMSD > 5), the scaling factor is set to 5

    TO RUN Global_wRMSD.py:

        Global_HwRMSD.py bl2seq_Location Protein_1.pdb
        Protein_1_Chain_ID Protein_2.pdb Protein_2_Chain_ID

    Example:
        Global_HwRMSD.py bl2seq 3ERD.pdb A 3ERT.pdb A

    Example output:
        3ERD_sRMSD.pdb
        3ERD_wRMSD.pdb
        3ERD_FASTA.txt
        3ERT_FASTA.txt
        3ERD_3ERT_BLAST.out
        Calculated standard RMSD value

To run through cygwin:
    In c:cygwin\etc\profile

    Change PATH to local python (Python25):
```

```

PATH=/usr/local/bin:/cygdrive/c/Python25:/usr/bin:/bin:/usr/X11R6
/bin:$PATH
export PATH

```

For questions or comments:
 Kelly Damm
 kdamm@umich.edu

University of Michigan
 Carlson Lab ""

```

#define global functions
from __future__ import division
import sys,re,cgi,os
from scipy import sort,transpose
import Numeric, LinearAlgebra
from Numeric import *

def run_Global_HwRMSD(file1,file2,X_Chain_ID,Y_Chain_ID,bl_loc):
    #CREATE FASTA FILES FROM PDB FILES
    x_FASTA,y_FASTA =
get_FASTA_Files(file1,file2,X_Chain_ID,Y_Chain_ID)

    #RUN BLAST (bl2seq) USING FASTA FILES
    os.system('%s -p blastp -i ProteinX_FASTA.txt -j
ProteinY_FASTA.txt -o BLAST.out -F F'%bl_loc)

    #DETERMINES RESIDUE CORRESPONDENCE USING BLAST SEQUENCE ALIGNMENT
    xlist,ylist =
BLAST_Coordinates(file1,file2,X_Chain_ID,Y_Chain_ID)
    x = transpose(xlist)
    y = transpose(ylist)

    #RETURNS ALL COORDINATES OF PROTEIN X FOR TRANSFORMATION
    all = getAll_Coords(file1)
    set = getStructure(file1)
    title=(file1.split('.')[0], file1)
    title2=(file2.split('.')[0], file2)

    #PERFORM STANDARD AND WEIGHTED RMSD CALCULATION
    allrot,SRMSD = weighted_alignment(x,y,all,set,title)
    print "The standard RMSD value is = ",SRMSD

    #OUTPUT TRANSFORMED STRUCTURE OF PROTEIN X
    s = getStructure(file1)
    setAll(s,allrot)
    writeStructure(s,'%s_wRMSD.pdb'%title[0])

    #REMOVE FILES GENERATED DURING ALIGNMENT
    os.system('mv ProteinX_FASTA.txt %s_FASTA.txt'%title[0])
    os.system('mv ProteinY_FASTA.txt %s_FASTA.txt'%title2[0])
    os.system('mv BLAST.out %s_%s_BLAST.out'%(title[0],title2[0]))

```



```

#CREATE FASTA FILES
def get_FASTA_Files(file1,file2,X_Chain_ID,Y_Chain_ID):
    x_ResidueID = get_ResidueName(file1,X_Chain_ID)
    y_ResidueID = get_ResidueName(file2,Y_Chain_ID)
    #CHANGE 3 LETTER AA CODE TO 1 LETTER AA CODE
    x_AminoAcid = AminoAcids(x_ResidueID)
    y_AminoAcid = AminoAcids(y_ResidueID)
    x_FASTA = "".join(x_AminoAcid)
    y_FASTA = "".join(y_AminoAcid)
    x_output = open('ProteinX_FASTA.txt','w')
    x_output.write(x_FASTA)
    x_output.close()
    y_output = open('ProteinY_FASTA.txt','w')
    y_output.write(y_FASTA)
    y_output.close()
    return x_FASTA,y_FASTA

#ENSURE RESIDUE CORRESPONDENCE
def BLAST_Coordinates(file1,file2,X_Chain_ID,Y_Chain_ID):
    ## GET PDB COORDINATE START NUMBERS
    input = open('BLAST.out','r')
    filelist = input.readlines()
    X_start = X_startnum(filelist)
    Y_start = Y_startnum(filelist)

    ## GET AA LIST FROM BLAST OUTPUT
    X_all_aa_list = X_AAlist(filelist)
    Y_all_aa_list = Y_AAlist(filelist)

    ##GET CA RES ID (WHAT WE NEED FOR ALIGNMENT) AND ALL RES ID (USED
    IN TO CREATE FASTA FILE FOR BLAST INPUT); HETS NOT INCLUDED
    X_CA_resID = get_CA_ResID(file1,X_Chain_ID)
    Y_CA_resID = get_CA_ResID(file2,Y_Chain_ID)

    x_FASTA_IDs = get_ResidueID(file1,X_Chain_ID)
    y_FASTA_IDs = get_ResidueID(file2,Y_Chain_ID)

    ##COMPARE CA RESIDUES TO RESIDUES (FASTA FILE), REMOVE THOSE IN
    X_FASTA AND Y_FASTA FROM X_FASTA_IDs AND Y_FASTA_IDs
    x_CA,x_CA_fasta = compare(X_CA_resID,x_FASTA_IDs)
    y_CA,y_CA_fasta = compare(Y_CA_resID,y_FASTA_IDs)
    X_AA_positions = AA_to_Position(X_all_aa_list, X_start)
    Y_AA_positions = AA_to_Position(Y_all_aa_list, Y_start)
    X_noCA_AA_positions = CA_to_Dash(X_AA_positions,x_CA_fasta)
    Y_noCA_AA_positions = CA_to_Dash(Y_AA_positions,y_CA_fasta)

    ##GET POSITION OF GAPS
    Gap_list = find_Gaps(X_noCA_AA_positions,Y_noCA_AA_positions)
    Gap_list_X, Gap_list_Y = unzip(Gap_list)

    #GET PDB COORD END NUMBER
    X_Gap_length = len(Gap_list_X)
    Y_Gap_length = len(Gap_list_Y)

```

```

#ADD CORRECT START POSITIONS (PDB file and BLAST output) TO
GAP_LIST
Final_X_list = Add_Start_Value(Gap_list_X,X_start)
Final_Y_list = Add_Start_Value(Gap_list_Y,Y_start)
X_ResID_f = Minus_Start_Value(Final_X_list)
Y_ResID_f = Minus_Start_Value(Final_Y_list)

#GET ALL CA COORDINATES FROM PDB FILES, EXCEPT THOSE OF HET
GROUPS
x_CAcords = get_CA_Coords(file1,X_Chain_ID)
y_CAcords = get_CA_Coords(file2,Y_Chain_ID)

#GET COORDINATES THAT COORESPOND TO POSITIONS FROM SEQUENCE
ALIGNMENT
X_Seq_CA_Coords = Seq_CACoords(x_CAcords,X_ResID_f)
Y_Seq_CA_Coords = Seq_CACoords(y_CAcords,Y_ResID_f)

#FINAL CHECK TO ENSURE RESIDUE CORRESPONDENCE
n = len(X_Seq_CA_Coords)
j = len(Y_Seq_CA_Coords)
if n != j:
    sys.exit("Proteins do not have same number of atoms;
Protein X has",n,"atoms, while Protein Y has",j,"atoms")

#CHECK TO DETERMINE IF APPROPRIATE NUMBER OF COORDINATES PRESENT
if (len(X_Seq_CA_Coords[0]) != 3):
    sys.exit("Protein X does not have a 3xn atom coordinate
set")
if (len(Y_Seq_CA_Coords[0]) != 3):
    sys.exit("Protein Y does not have a 3xn atom coordinate
set")

#CHECK TO DETERMINE IF >4 COORDINATES PRESENT FOR EACH PROTEIN
if n < 4:
    sys.exit("Protein X has 3 or less coordinates, 4 or more
needed to perform alignment")
if j < 4:
    sys.exit("Protein Y has 3 or less coordinates, 4 or more
needed to perform alignment")

return X_Seq_CA_Coords,Y_Seq_CA_Coords

##WEIGHTED RMSD ALIGNMENT##
def weighted_alignment(x,y,all,set,title):
    atoms = len(x[0])

    #Initial standard alignment without weight
    #TRANSLATE PROTEINS X AND Y TO CENTER
    n = len(x[0])
    x_mean = mean(x,n)
    y_mean = mean(y,n)
    x_trans = translation(x, x_mean)
    x_translated = nested_list(x_trans,n)
    y_trans = translation(y, y_mean)
    y_translated = nested_list(y_trans,n)

```

```

x_transpose = transpose(x_translated)

#CALCULATE COVARIANCE MATRIX (y_translated *x_translated^t)
R = matrixmultiply(y_translated, x_transpose)
R_transpose = transpose(R)
R2 = matrixmultiply(R_transpose, R)

#DETERMINE THE EIGENVECTORS AND EIGENVALUES of R2
mu,A = LinearAlgebra.eigenvalues(R2)

#SORT EIGENVECTORS IN DECREASING ORDER OF EIGENVALUES
a = [(mu[i],A[i]) for i in range(len(A))]
a.sort()
a.reverse()
mu = [x[0] for x in a]
A = [x[1] for x in a]

#DETERMINE RIGHT-HANDED SYSTEM
A_3 = crossproduct(A[0], A[1])
A = [A[0], A[1], A_3]

#CALCULATE B, NORMALIZED PRODUCT OF (RxA)
B_1 = matrixmultiply(R, A[0])
B_2 = matrixmultiply(R, A[1])
norm_B_1 = normalize(B_1)
norm_B_2 = normalize(B_2)
norm_B_3 = crossproduct(norm_B_1,norm_B_2)
B = [norm_B_1,norm_B_2,norm_B_3]
B_transpose = transpose(B)

#CALCULATE ROTATION MATRIX, U
U = rotation_matrix(B_transpose, A)
x_rot = matrixmultiply(U,x_translated)

#CALCULATE STANDARD RMSD
standard_RMSD = sRMSD(x_rot, y_translated)

#ADD MEAN VALUES OF PROTEIN Y TO ROTATED COORDINATES OF PROTEIN X
x_coords = add_coords(x_rot, y_mean)
x = nested_list(x_coords,n)

#TRANSLATE ALL COORDINATES OF PROTEIN X
all_trans = translation(all, x_mean)
j = len(all[0])
all_translated = nested_list(all_trans,j)
all_rot = matrixmultiply(U,all_translated)

#ADD ALL MEAN VALUES OF PROTEIN Y TO ALL ROTATED COORDINATES OF
PROTEIN X
all_coords = add_coords(all_rot, y_mean)
all = nested_list(all_coords,j)
allrot = transpose(all)
setAll(set,allrot)

#OUTPUT STANDARD RMSD ALIGNMENT
writeStructure(set, '%s_sRMSD.pdb'%title[0])

```

```

#WEIGHTED RMSD CALCULATION, z = # of iterations
z = 1
weighted_rmsds = []
w_metric = []
all_list = []

#DETERMINE APPROPRIATE SCALING FACTOR
if standard_RMSD < 5:
    scaling_factor = 2
elif standard_RMSD >= 5:
    scaling_factor = 5
while z < 5001:
    n = len(x[0])
    #TRANSLATE WEIGHTED CENTROIDS TO ORIGIN
    #CALCULATE WEIGHTS (protein1, protein2, scaling_factor)
    weights = weight(x,y,scaling_factor)
    weighted_x_mean = weight_trans(x,weights,n)
    weighted_y_mean = weight_trans(y,weights,n)
    x_trans = translation(x, weighted_x_mean)
    y_trans = translation(y, weighted_y_mean)
    x_translated = nested_list(x_trans,n)
    y_translated = nested_list(y_trans,n)

    #CALCULATE WEIGHTED COVARIANCE MATRIX
    weighted_rot =
weight(x_translated,y_translated,scaling_factor)
    weighted_x_translated = multiply(weighted_rot,x_translated)
    wx_transpose = transpose(weighted_x_translated)
    R = matrixmultiply(y_translated,wx_transpose)
    R_transpose = transpose(R)
    R2 = matrixmultiply(R_transpose, R)

    #DETERMINE THE EIGENVECTORS AND EIGENVALUES of R2
    mu,A = LinearAlgebra.eigenvectors(R2)

    #SORT EIGENVECTORS IN DECREASING ORDER OF EIGENVALUES
    a = [(mu[i],A[i]) for i in range(len(A))]
    a.sort()
    a.reverse()
    mu = [x[0] for x in a]
    A = [x[1] for x in a]

    #DETERMINE RIGHT-HANDED SYSTEM
    A_3 = crossproduct(A[0], A[1])
    A = [A[0], A[1], A_3]

    #CALCULATE B, NORMALIZED PRODUCT OF (RxA)
    B_1 = matrixmultiply(R, A[0])
    B_2 = matrixmultiply(R, A[1])
    norm_B_1 = normalize(B_1)
    norm_B_2 = normalize(B_2)
    norm_B_3 = crossproduct(norm_B_1,norm_B_2)
    B = [norm_B_1,norm_B_2,norm_B_3]
    B_transpose = transpose(B)

    #CALCULATE WEIGHTED ROTATION MATRIX, U
    U = rotation_matrix(B_transpose, A)

```

```

x_rot = matrixmultiply(U,x_translated)

#CALCULATE WEIGHTED RMSD
weighted_rmsds.append(wRMSD(x_rot,
y_translated,scaling_factor))
w_metric.append(wSUM(x_rot,
y_translated,scaling_factor,atoms,z))

#DETERMINE IF CONVERGENCE IS REACHED
if z > 1:
    wrmsd_diff = weighted_rmsds[-2] - weighted_rmsds[-1]
else:
    wrmsd_diff = []

if 0 < wrmsd_diff < 0.000001:

    #ADD MEAN VALUES OF PROTEIN Y TO ROTATED COORDINATES
OF PROTEIN X
    x_coords = add_coords(x_rot, weighted_y_mean)
    x = nested_list(x_coords,n)

    #TRANSFORM ALL COORDINATES OF PROTEIN X
    all_trans = translation(all, weighted_x_mean)
    j = len(all[0])
    all_translated = nested_list(all_trans,j)
    all_rot = matrixmultiply(U,all_translated)

    #ADD ALL MEAN VALUES OF PROTEIN Y TO ALL ROTATED
COORDINATES OF PROTEIN X
    all_coords = add_coords(all_rot, weighted_y_mean)
    all = nested_list(all_coords,j)
    allrot = transpose(all)
    break

else:
    #ADD MEAN VALUES OF PROTEIN Y TO ROTATED COORDINATES
OF PROTEIN X
    x_coords = add_coords(x_rot, weighted_y_mean)
    x = nested_list(x_coords,n)

    #TRANSFORM ALL COORDINATES OF PROTEIN X
    all_trans = translation(all, weighted_x_mean)
    j = len(all[0])
    all_translated = nested_list(all_trans,j)
    all_rot = matrixmultiply(U,all_translated)

    #ADD ALL MEAN VALUES OF PROTEIN Y TO ALL ROTATED
COORDINATES OF PROTEIN X
    all_coords = add_coords(all_rot, weighted_y_mean)
    all = nested_list(all_coords,j)
    z = z + 1

else:
    print "Alignment stopped after 5000 iterations, convergence
was never reached"
    allrot = transpose(all)
    return allrot,standard_RMSD

```

```
##### HELPER FUNCTIONS IN ENSURE RESIDUE CORRESPONDENCE FUNCTION #####
```

```
###PDB_Parser.py HAS BEEN MODIFIED FOR USAGE WITH KLD HWRMSD CODE###
# Copyright (C) 2002, Thomas Hamelryck (thamelry@vub.ac.be)
# This code is part of the Biopython distribution and governed by its
# license. Please see the LICENSE file that should have been included
# as part of this package.
```

```
# Python stuff
import sys
from string import split
from Numeric import array, Float0
```

```
# My stuff
from Bio.PDB.StructureBuilder import StructureBuilder
from Bio.PDB.PDBExceptions import PDBConstructionException
```

```
# If PDB spec says "COLUMNS 18-20" this means line[17:20]
```

```
class PDBParser:
    """
    Parse a PDB file and return a Structure object.
    """

    def __init__(self, PERMISSIVE=0, structure_builder=None):
        """
        The PDB parser call a number of standard methods in an
        aggregated
        StructureBuilder object. Normally this object is
        instanciated by the
        PDBParser object itself, but if the user provides his own
        StructureBuilder
        object, the latter is used instead.

        Arguments:
        o PERMISSIVE - int, if this is 0 (default) exceptions in
        constructing the
        SMCRA data structure are fatal. If 1, the exceptions are
        caught, but some
        residues or atoms will be missing.
        o structure_builder - an optional user implemented
        StructureBuilder class.
        """
        if structure_builder!=None:
            self.structure_builder=structure_builder
        else:
            self.structure_builder=StructureBuilder()
        self.header=None
        self.trailer=None
        self.line_counter=0
        self.PERMISSIVE=PERMISSIVE
        #
        # We added repeatedResidues to keep track of repeated
        residues. _handle_PDBException now
```

```

        # just adds repeated residues to this list and ignores the
exception even when PERMISSIVE is not
        # set. The user is then responsible for removing the
repeated residues herself.
        #
        self.repeatedResidues = {}
# Public methods

def get_structure(self, id, filename):
    """Return the structure.

    Arguments:
    o id - string, the id that will be used for the structure
    o filename - name of the PDB file
    """
    self.header=None
    self.trailer=None
    # Make a StructureBuilder instance (pass id of structure as
parameter)
    self.structure_builder.init_structure(id)
    file=open(filename)
    self._parse(file.readlines())
    file.close()
    # Return the Structure instance
    return self.structure_builder.get_structure()

def get_header(self):
    "Return the header."
    return self.header

def get_trailer(self):
    "Return the trailer."
    return self.trailer

# Private methods

def _parse(self, header_coords_trailer):
    "Parse the PDB file."
    # Extract the header; return the rest of the file
    self.header,
coords_trailer=self._get_header(header_coords_trailer)
    # Parse the atomic data; return the PDB file trailer
    self.trailer=self._parse_coordinates(coords_trailer)

def _get_header(self, header_coords_trailer):
    "Get the header of the PDB file, return the rest."
    structure_builder=self.structure_builder
    for i in range(0, len(header_coords_trailer)):
        structure_builder.set_line_counter(i+1)
        line=header_coords_trailer[i]
        record_type=line[0:6]
        if(record_type=='ATOM ' or record_type=='HETATM' or
record_type=='MODEL '):
            break
    header=header_coords_trailer[0:i]
    # Return the rest of the coords+trailer for further
processing

```

```

self.line_counter=i
coords_trailer=header_coords_trailer[i:]
return header, coords_trailer

def _parse_coordinates(self, coords_trailer):
    "Parse the atomic data in the PDB file."
    local_line_counter=0
    structure_builder=self.structure_builder
    current_model_id=0
    current_chain_id=None
    current_segid=None
    current_residue_id=None
    current_resname=None
    structure_builder.init_model(current_model_id)
    for i in range(0, len(coords_trailer)):
        line=coords_trailer[i]
        record_type=line[0:6]

global_line_counter=self.line_counter+local_line_counter+1

structure_builder.set_line_counter(global_line_counter)
if(record_type=='ATOM ' or record_type=='HETATM'):
    fullname=line[12:16]
    # get rid of whitespace in atom names
    split_list=split(fullname)
    if len(split_list)!=1:
        # atom name has internal spaces, e.g. " N
B ", so
        # we do not strip spaces
        name=fullname
    else:
        # atom name is like " CA ", so we can
strip spaces
        name=split_list[0]
    altloc=line[16:17]
    resname=line[17:20]
    chainid=line[21:22]
    resseq=int(split(line[22:26])[0]) # sequence
identifier
    icode=line[26:27] # insertion code
atom flag
    if record_type=='HETATM': # hetero
        if resname=="HOH" or resname=="WAT":
            hetero_flag="W"
        else:
            hetero_flag="H"
    else:
        hetero_flag=" "
    residue_id=(hetero_flag, resseq, icode)
    # atomic coordinates
    x=float(line[30:38])
    y=float(line[38:46])
    z=float(line[46:54])
    coord=array((x, y, z), Float0)
    # occupancy & B factor
    occupancy=float(line[54:60])
    bfactor=float(line[60:66])

```



```

        segid=line[72:76]
        if current_segid!=segid:
            current_segid=segid
            structure_builder.init_seg(current_segid)
        if current_chain_id!=chainid:
            current_chain_id=chainid

        structure_builder.init_chain(current_chain_id)
            current_residue_id=residue_id
            current_resname=resname
        try:

            structure_builder.init_residue(resname, hetero_flag, resseq,
icode)
                except PDBConstructionException, message:
                    self._handle_PDB_exception(message,
global_line_counter, chainid)
            elif current_residue_id!=residue_id or
current_resname!=resname:
                current_residue_id=residue_id
                current_resname=resname
            try:

                structure_builder.init_residue(resname, hetero_flag, resseq,
icode)
                    except PDBConstructionException, message:
                        self._handle_PDB_exception(message,
global_line_counter, chainid)
                # init atom
                try:
                    structure_builder.init_atom(name, coord,
bfactor, occupancy, altloc, fullname)
                except PDBConstructionException, message:
                    self._handle_PDB_exception(message,
global_line_counter, chainid)
                elif(record_type=='ANISOU'):
                    anisou=map(float, (line[28:35], line[35:42],
line[43:49], line[49:56], line[56:63], line[63:70]))
                    # U's are scaled by 10^4
                    anisou_array=(array(anisou,
Float0)/10000.0).astype(Float0)
                    structure_builder.set_anisou(anisou_array)
                elif(record_type=='ENDMDL'):
                    current_model_id=current_model_id+1
                    structure_builder.init_model(current_model_id)
                    current_chain_id=None
                    current_residue_id=None
                elif(record_type=='END ' or record_type=='CONNECT'):
                    # End of atomic data, return the trailer

            self.line_counter=self.line_counter+local_line_counter
            return coords_trailer[local_line_counter:]
        elif(record_type=='SIGUIJ'):
            # standard deviation of anisotropic B factor
            siguij=map(float, (line[28:35], line[35:42],
line[42:49], line[49:56], line[56:63], line[63:70]))
            # U sigma's are scaled by 10^4

```

```

        siguij_array=(array(siguij,
Float0)/10000.0).astype(Float0)
        structure_builder.set_siguij(siguij_array)
        elif(record_type=='SIGATM'):
            # standard deviation of atomic positions
            sigatm=map(float, (line[30:38], line[38:45],
line[46:54], line[54:60], line[60:66]))
            sigatm_array=array(sigatm, Float0)
            structure_builder.set_sigatm(sigatm_array)
            local_line_counter=local_line_counter+1
        # EOF (does not end in END or CONECT)
        self.line_counter=self.line_counter+local_line_counter
        return []

    def _handle_PDB_exception(self, message, line_counter,
chainid=None):
        """
        This method catches an exception that occurs in the
StructureBuilder
        object (if PERMISSIVE==1), or raises it again, this time
adding the
        PDB line number to the error message.
        """
        message="%s at line %i." % (message, line_counter)
        if self.PERMISSIVE:
            # just print a warning - some residues/atoms will be
missing
            print "PDBConstructionException: %s" % message
            print "Exception ignored.\nSome atoms or residues
will be missing in the data structure."
        else:
            # exceptions are fatal - raise again with new message
(including line nr)
            try:
                if PDBConstructionException.reason ==
'repeated':
                    self.repeatedResidues.setdefault(chainid, []).append(PDBConstructi
onException.resseq)

                    #self.repeatedResidues.append(PDBConstructionException.resseq)
                    #print "PDBConstructionException: %s" %
message
                    #print "Exception ignored.\nSome atoms or
residues will be missing in the data structure."
            else:
                raise PDBConstructionException, message
            except AttributeError:
                # if PDBConstructionException doesn't have a
reason, raise it like normal
                print "caught an attribute error .. no reason?"
                #print dir(PDBConstructionException)
                #raise
                raise PDBConstructionException, message

if __name__=="__main__":

```

```

import sys

p=PDBParser(PERMISSIVE=1)

s=p.get_structure("scr", sys.argv[1])

for m in s.get_iterator():
    p=m.get_parent()
    assert(p is s)
    for c in m.get_iterator():
        p=c.get_parent()
        assert(p is m)
        for r in c.get_iterator():
            p=r.get_parent()
            assert(p is c)
            for a in r.get_iterator():
                p=a.get_parent()
                if not p is r:
                    print p, r

def AminoAcids(file):
    list = []
    for each in file:
        if each == 'GLY':
            each = 'G'
            list.append(each)
        elif each == 'PRO':
            each = 'P'
            list.append(each)
        elif each == 'ALA':
            each = 'A'
            list.append(each)
        elif each == 'VAL':
            each = 'V'
            list.append(each)
        elif each == 'HIS':
            each = 'H'
            list.append(each)
        elif each == 'HID':
            each = 'H'
            list.append(each)
        elif each == 'HIE':
            each = 'H'
            list.append(each)
        elif each == 'HIP':
            each = 'H'
            list.append(each)
        elif each == 'LEU':
            each = 'L'
            list.append(each)
        elif each == 'ILE':
            each = 'I'
            list.append(each)
        elif each == 'MET':
            each = 'M'

```

```
        list.append(each)
elif each == 'CYS':
    each = 'C'
    list.append(each)
elif each == 'CYX':
    each = 'C'
    list.append(each)
elif each == 'CYM':
    each = 'C'
    list.append(each)
elif each == 'PHE':
    each = 'F'
    list.append(each)
elif each == 'TYR':
    each = 'Y'
    list.append(each)
elif each == 'TYM':
    each = 'Y'
    list.append(each)
elif each == 'TRP':
    each = 'W'
    list.append(each)
elif each == 'LYS':
    each = 'K'
    list.append(each)
elif each == 'LYN':
    each = 'K'
    list.append(each)
elif each == 'ARG':
    each = 'R'
    list.append(each)
elif each == 'GLN':
    each = 'Q'
    list.append(each)
elif each == 'ASN':
    each = 'N'
    list.append(each)
elif each == 'GLU':
    each = 'E'
    list.append(each)
elif each == 'GLH':
    each = 'E'
    list.append(each)
elif each == 'ASP':
    each = 'D'
    list.append(each)
elif each == 'ASH':
    each = 'D'
    list.append(each)
elif each == 'SER':
    each = 'S'
    list.append(each)
elif each == 'THR':
    each = 'T'
    list.append(each)
else:
    each = 'X'
```

```

        list.append(each)
    return list

def get_ResidueName(filename, chain_key):
    x = []
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            residue_id = residue.get_id()
            if residue_id[0] == ' ':
                resid=residue.get_resname()
                x.append(resid)
    return x

def X_startnum(filelist):
    querycount = 0
    for fileline in filelist:
        if fileline.strip().lower().startswith('query'):
            querycount += 1
        if querycount == 2:
            parts = fileline.split()
            num = int(parts[1])
            break
    return num

def Y_startnum(filelist):
    #querycount = 0
    for fileline in filelist:
        if fileline.strip().lower().startswith('sbjct'):
            parts = fileline.split()
            num = int(parts[1])
            break
    return num

def X_AAlist(filelist):
    scorecount = 0
    all_aa_list = []
    for fileline in filelist:
        if fileline.strip().lower().startswith('score'):
            scorecount += 1
        if scorecount == 1:
            if fileline.strip().lower().startswith('query:'):
                parts = fileline.split()
                #print "parts=",parts
                aa_list = parts[2]
                all_aa_list.append(aa_list)
                AA_string = "".join(all_aa_list)
        if scorecount != 1:
            quit
    return AA_string

def Y_AAlist(filelist):
    scorecount = 0
    all_aa_list = []
    for fileline in filelist:

```

```

    if fileline.strip().lower().startswith('score'):
        scorecount += 1
    if scorecount == 1:
        if fileline.strip().lower().startswith('sbjct:'):
            parts = fileline.split()
            #print "parts=",parts
            aa_list = parts[2]
            all_aa_list.append(aa_list)
            AA_string = "".join(all_aa_list)
        if scorecount != 1:
            quit
    return AA_string

def get_CA_ResID(filename, chain_key):
    x = []
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            if residue.has_id("CA"):
                residue_id = residue.get_id()
                if residue_id[0] == ' ':
                    resid = residue_id[1]
                    x.append(residue.get_id()[1])
    return x

def get_ResidueID(filename, chain_key):
    x = []
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            residue_id = residue.get_id()
            if residue_id[0] == ' ':
                resid=residue.get_id()[1]
                x.append(resid)
    return x

def compare(x,y):
    x_list = []
    y_list = []
    for each in x:
        if each not in y:
            x_list.append(each)
    for each in y:
        if each not in x:
            y_list.append(each)
    x_list = zero2(x_list)
    y_list = zero2(y_list)
    x_list = sort(x_list)
    y_list = sort(y_list)
    x_list = list(x_list)
    y_list = list(y_list)
    return x_list, y_list

```

```

def zero2(x):
    result = []
    for each in x:
        if each != 0:
            result.append(each)
        else:
            continue
    return result

def AA_to_Position(filename, start):
    idx1 = start - 1
    result = []
    for value in filename:
        if value == '-':
            result.append(value)
        else:
            idx1 += 1
            result.append(idx1)
    return result

def CA_to_Dash(filename, CA_list):
    result = []
    for value in filename:
        if value in CA_list:
            value = '-'
            result.append(value)
        else:
            result.append(value)
    return result

def find_Gaps(seq1, seq2):
    idx1, idx2 = 1, 1
    result = []
    for res1, res2 in zip(seq1, seq2):
        if res1 != '-' and res2 != '-':
            result += [(idx1, idx2)]
            if res1 != '-': idx1 += 1
            if res2 != '-': idx2 += 1
    return result

def unzip(l, *jj):
    #Christopher P. Smith 7/13/2001
    if jj==():
        jj=range(len(l[0]))
    rl = [[li[j] for li in l] for j in jj]
    if len(rl) == 1:
        rl=rl[0]
    return rl

def Add_Start_Value(file, Blast_start):
    new_count = []
    for each in file:
        grab = each - 1 + Blast_start
        new_count.append(grab)
    return new_count

def Minus_Start_Value(file):

```

```

new_count = []
for each in file:
    grab = each - 1
    new_count.append(grab)
return new_count

def get_CA_Coords(filename, chain_key):
    x = []
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain_A = model[chain_key]
        for residue in chain_A.get_list():
            if residue.has_id("CA"):
                residue_id = residue.get_id()
                if residue_id[0] == ' ':
                    ca=residue["CA"]
                    x.append(ca.get_coord())

    return x

def Seq_CACoords(first, list):
    result = []
    for each in list:
        result.append(first[each])
    return result

def getAll_Coords(filename):
    result =[]
    parser=PDBParser()
    structure=parser.get_structure(filename.split('.')[0], filename)
    for model in structure.get_list():
        chain = model.get_list()
        for each in chain:
            for res in each.get_list():
                for x in res.get_list():
                    result.append(x.get_coord())
    x_t = transpose(result)
    return x_t

def getStructure(filename):
    parser = PDBParser()
    structure = parser.get_structure(filename.split('.')[0],
filename)
    return structure

def writeStructure(structure, filename):
    from Bio.PDB.PDBIO import PDBIO
    import sys
    io = PDBIO()
    io.set_structure(structure)
    io.save(filename)

def setAll(structure, newCACoords):
    allCAs =[]
    for model in structure.get_list():
        chain = model.get_list()
        for each in chain:

```



```

        for res in each.get_list():
            for x in res.get_list():
                allCAs.append(x)
    if len(allCAs) != len(newCACoords):
        print "wrong number of atoms .. structure
had",len(allCAs),"you gave me",len(newCACoords)
        raise Exception("wrong number of atoms")
    for newCoords,ca in zip(newCACoords,allCAs):
        ca.set_coord(newCoords)

##### HELPER FUNCTIONS IN WEIGHTED ALIGNMENT FUNCTION #####

def mean(first,n):
    return [sum(each)/n for each in first]

def nested_list(name,n):
    first1 = name[0:n]
    first2 = name[n:2*n]
    first3 = name[2*n:3*n]
    first_translated =[first1,first2,first3]
    return first_translated

def translation(first,second):
    c = 0
    k = []
    q = len(first)
    while c < q:
        for each in first[c]:
            subtr = each - second[c]
            k.append(subtr)
        c = c + 1
    return k

def crossproduct(a,b):
    C_0 = a[1]*b[2] - a[2]*b[1]
    C_1 = a[2]*b[0] - a[0]*b[2]
    C_2 = a[0]*b[1] - a[1]*b[0]
    return [C_0, C_1, C_2]

def normalize(a):
    B_0 = (a[0])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    B_1 = (a[1])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    B_2 = (a[2])/(((a[0]**2)+(a[1]**2)+((a[2])**2))**(1/2))
    return [B_0, B_1, B_2]

def rotation_matrix(first, second):
    U = matrixmultiply(first, second)
    return U

def sRMSD(first,second):
    first = array(first)
    second = array(second)
    subtr = first - second
    def sqr(matrix):

```

```

        k = []
        for each in matrix:
            sq = (each)**2
            k.append(sq)
        return k
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    d = sqrt(sum_subtr_s)
    sq_d = sqr(d)
    s_sq_d = sum(sq_d)
    tot = (len(first[0]))
    value = sqrt(s_sq_d/tot)
    return value

def add_coords(first, second):
    c = 0
    k = []
    q = len(first)
    while c < q:
        for each in first[c]:
            add = each + second[c]
            k.append(add)
        c = c + 1
    return k

def weight_trans(first, weight, n):
    mult = multiply(first, weight)
    sum_mult = sum(mult)
    mean = [sum(each)/n for each in mult]
    return mean

def weight(first, second, constant):
    first = array(first)
    second = array(second)
    subtr = first - second
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    d = sqrt(sum_subtr_s)
    weighted_d = Gaussian(d, constant)
    weighted_d = Gaussian2(weighted_d)
    return weighted_d

def sqr(matrix):
    k = []
    for each in matrix:
        sq = (each)**2
        k.append(sq)
    return k

def sqrt(matrix):
    j = []
    for each in matrix:
        sqrt = sqrt(each)
        j.append(sqrt)
    return j

def wSUM(first, second, constant, atoms, z):

```

```

j = []
k = []
first = array(first)
second = array(second)
subtr = first - second
subtr_s = sqr(subtr)
sum_subtr_s = sum(subtr_s)
d = sqrt(sum_subtr_s)
weighted_d = Gaussian(d, constant)
weighted_d = Gaussian2(weighted_d)
for each in weighted_d:
    mult100 = (each * 100)
    j.append(mult100)
for each in j:
    subtr100 = (100 - each)
    k.append(subtr100)
sum_weighted_d = sum(weighted_d)
value = sum_weighted_d/atoms
return value

def wRMSD(first, second, constant):
    first = array(first)
    second = array(second)
    subtr = first - second
    subtr_s = sqr(subtr)
    sum_subtr_s = sum(subtr_s)
    d = sqrt(sum_subtr_s)
    weighted_d = Gaussian(d, constant)
    weights = Gaussian2(weighted_d)
    sq_d = sqr(d)
    wd = multiply(sq_d, weights)
    s_wd = sum(wd)
    n = len(d)
    s_sq_d_divide = s_wd/n
    value = sqrt(s_sq_d_divide)
    return value

def Gaussian(first, z):
    value = []
    for each in first:
        weight = (-((each)**2)/z)
        value.append(weight)
    return value

def Gaussian2(first):
    value = []
    for each in first:
        weight = exp(each)
        value.append(weight)
    return value

def get_Adjusted_Res_Position(position, listname):
    while position in listname:
        position += 1
    return position

def getHetGroups(filename, chain_key):

```

```
result = []
result2 = []
parser=PDBParser()
structure=parser.get_structure(filename.split('.')[0], filename)
for model in structure.get_list():
    chain_A = model[chain_key]
    for residue in chain_A.get_list():
        if residue.has_id("CA"):
            residue_id = residue.get_id()
            resid = residue_id[1]
            if residue_id[0] == ' ':
                result.append(resid)
            else:
                result2.append(resid)
    return result2

#RUN Global_HwRMSD.py
if __name__ == "__main__":
    if len(sys.argv) != 6:
        print "usage: Global_HwRMSD.py bl2seq_Location
Protein_X.pdb Protein_X_ChainID Protein_Y.pdb Protein_Y_ChainID"
        sys.exit()
    bl_loc = sys.argv[1]
    filename1 = sys.argv[2]
    filename2 = sys.argv[4]
    X_Chain_ID = sys.argv[3]
    Y_Chain_ID = sys.argv[5]
    run_Global_HwRMSD(filename1,filename2,X_Chain_ID,Y_Chain_ID,bl_lo
c)
```

A2.4 Raw Data from differences in superpositions.

Both standard and weighted superpositions were generated from a variety of sequence alignments. The sequence alignments were altered by varying the parameters within BLAST or varying the code used for the alignment. The differences across the superpositions were measured in RMSD (Å) between the coordinates. The bolded heading are the default BLAST parameters.

A. Varying Parameters within BLAST (Scoring matrix, Gap Opening Penalty (G), Gap Extension Penalty (E))- Standard RMSD Superposition.

Standard RMSD						AVG
Serine/Threonine Protein Phosphatase						
39%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	0.494
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.577	0				
BLOSUM62, G6, E2	0.000	0.577	0			
BLOSUM45, G11, E1	0.577	0.000	0.577	0		
PAM30, G11, E1	0.846	0.471	0.846	0.471	0	
Eukaryotic Glutathione Synthase						0.600
37%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.083	0				
BLOSUM62, G6, E2	0.304	0.325	0			
BLOSUM45, G11, E1	0.080	0.016	0.326	0		
PAM30, G11, E1	1.280	1.221	1.134	1.227	0	
Interferon alpha, beta, and delta						
35%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62,	1.561	0				

G13, E1					
BLOSUM62, G6, E2	1.561	0.000	0		
BLOSUM45, G11, E1	1.561	0.000	0.000	0	
PAM30, G11, E1	0.917	2.212	2.212	2.212	0

1.224

Adenosylmethionine Decarboxylase

33%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	0.183	0.183	0		
BLOSUM45, G11, E1	0.729	0.729	0.883	0	
PAM30, G11, E1	0.967	0.967	1.011	0.901	0

0.655

Clostridial Neurotoxin Zinc Protease

31%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.443	0			
BLOSUM62, G6, E2	0.609	0.826	0		
BLOSUM45, G11, E1	0.607	0.785	0.108	0	
PAM30, G11, E1	0.816	0.557	1.241	1.201	0

0.719

Sulfatase

29%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	1.282	0			
BLOSUM62, G6, E2	1.746	1.599	0		
BLOSUM45, G11, E1	1.448	1.607	0.727	0	
PAM30, G11, E1	1.633	0.768	2.219	2.205	0

1.523

Protocatechuate-3,4-Dioxygenase, alpha and beta chains

28%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62,	0.560	0.560	0		

G6, E2					
BLOSUM45, G11, E1	0.438	0.438	0.329	0	
PAM30, G11, E1	2.361	2.361	2.170	2.385	0

1.160

Aminotransferase Class IV

27%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	1.704	1.704	0		
BLOSUM45, G11, E1	1.699	1.699	0.115	0	
PAM30, G11, E1	1.742	1.742	0.717	0.686	0

1.181

SpoU rRNA Methylase Family

26%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	5.408	0			
BLOSUM62, G6, E2	1.499	6.098	0		
BLOSUM45, G11, E1	3.171	6.792	1.911	0	
PAM30, G11, E1	6.551	1.589	7.401	8.239	0

4.866

FMN Oxidoreductase

25%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.627	0			
BLOSUM62, G6, E2	0.473	0.958	0		
BLOSUM45, G11, E1	1.988	1.554	2.282	0	
PAM30, G11, E1	2.554	2.426	2.792	1.701	0

1.736

Queuine tRNA-Ribosyltransferase

25%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.238	0			
BLOSUM62, G6, E2	0.342	0.456	0		
BLOSUM45, G11, E1	0.207	0.064	0.451	0	

G11, E1					
PAM30, G11, E1	1.622	1.704	1.400	1.701	0

0.819

tRNA Synthetase Class I (E and Q)

24%D	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	4.038	0			
BLOSUM62, G6, E2	0.350	3.967	0		
BLOSUM45, G11, E1	1.181	3.669	1.234	0	
PAM30, G11, E1	na	na	na	na	0

2.407

DNA Methylase

23%D	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	1.281	0			
BLOSUM62, G6, E2	1.244	1.857	0		
BLOSUM45, G11, E1	1.266	1.700	0.600	0	
PAM30, G11, E1	na	na	na	na	0

1.325

Type II DNA Topoisomerase, Domains 2-4

22%D	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	2.222	2.222	0		
BLOSUM45, G11, E1	2.324	2.324	0.696	0	
PAM30, G11, E1	4.947	4.947	3.999	3.735	0

2.742

Pyridoxal-Phosphate Dependent Enzymes

21%D	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.333	0			
BLOSUM62, G6, E2	1.436	1.407	0		
BLOSUM45, G11, E1	1.317	1.242	0.397	0	
PAM30, G11, E1	na	na	na	na	0

1.022

E1					
----	--	--	--	--	--

Iron/Ascorbate Oxidoreductase

20%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.833	0			
BLOSUM62, G6, E2	3.386	3.359	0		
BLOSUM45, G11, E1	3.467	3.469	0.216	0	
PAM30, G11, E1	na	na	na	na	0

2.455

FAD Binding Domain in Molybdopterin Dehydrogenase

19%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	0.696	0.696	0		
BLOSUM45, G11, E1	0.462	0.462	0.867	0	
PAM30, G11, E1	na	na	na	na	0

0.531

Leucine Rich Repeats in Splicesomal U2A' Protein and Internalin B

19%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	3.020	0			
BLOSUM62, G6, E2	0.000	3.020	0		
BLOSUM45, G11, E1	1.619	3.475	1.619	0	
PAM30, G11, E1	na	na	na	na	0

2.126

Asp/Glu/Hydantoin Racemase

18%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	1.762	0			
BLOSUM62, G6, E2	1.126	2.168	0		
BLOSUM45, G11, E1	1.737	0.170	2.117	0	
PAM30, G11, E1	na	na	na	na	0

1.513

Polysaccharide Lyase Family 8, N Terminal Domain

18%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	1.074	1.074	0		
BLOSUM45, G11, E1	0.608	0.608	1.387	0	
PAM30, G11, E1	na	na	na	na	0

0.792

PHBH-like

17%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	2.900	2.900	0		
BLOSUM45, G11, E1	2.992	2.992	1.936	0	
PAM30, G11, E1	na	na	na	na	0

2.287

B. Varying Parameters within BLAST (Scoring matrix, Gap Opening Penalty (G), Gap Extension Penalty (E))- Weighted RMSD Superposition.**Weighted
RMSD****Serine/Threonine Protein Phosphatase****AVG**

39%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.035	0			
BLOSUM62, G6, E2	0.000	0.035	0		
BLOSUM45, G11, E1	0.040	0.005	0.039	0	
PAM30, G11, E1	0.126	0.107	0.126	0.110	0

0.062

Eukaryotic Glutathione Synthase

37%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				

BLOSUM62, G13, E1	0.023	0				
BLOSUM62, G6, E2	0.038	0.019	0			
BLOSUM45, G11, E1	0.009	0.002	0.021	0		
PAM30, G11, E1	0.166	0.171	0.184	0.171	0	0.080

Interferon alpha, beta, and delta

35%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.525	0				
BLOSUM62, G6, E2	0.525	0.000	0			
BLOSUM45, G11, E1	0.508	0.025	0.025	0		
PAM30, G11, E1	0.161	0.420	0.420	0.400	0	0.301

Adenosylmethionine Decarboxylase

33%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.000	0				
BLOSUM62, G6, E2	0.119	0.119	0			
BLOSUM45, G11, E1	0.030	0.030	0.117	0		
PAM30, G11, E1	0.203	0.203	0.314	0.216	0	0.135

Clostridial Neurotoxin Zinc Protease

31%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.009	0				
BLOSUM62, G6, E2	0.007	0.016	0			
BLOSUM45, G11, E1	0.053	0.062	0.045	0		
PAM30, G11, E1	0.205	0.210	0.202	0.171	0	0.098

Sulfatase

29%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.583	0				

BLOSUM62, G6, E2	0.021	0.586	0		
BLOSUM45, G11, E1	0.044	0.555	0.058	0	
PAM30, G11, E1	0.440	0.356	0.452	0.421	0

0.352

Protocatechuate-3,4-Dioxygenase, alpha and beta chains

28%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	0.100	0.044	0		
BLOSUM45, G11, E1	0.112	0.054	0.029	0	
PAM30, G11, E1	0.243	0.218	0.195		0

0.111

Aminotransferase Class IV

27%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	0.145	0.145	0		
BLOSUM45, G11, E1	0.124	0.124	0.118	0	
PAM30, G11, E1	0.601	0.627	0.720	0.689	0

0.329

SpoU rRNA Methylase Family

26%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.323	0			
BLOSUM62, G6, E2	0.265	0.222	0		
BLOSUM45, G11, E1	0.336	0.408	0.236	0	
PAM30, G11, E1	1.251	1.059	1.074	1.269	0

0.644

FMN Oxidoreductase

25%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.060	0			
BLOSUM62, G6, E2	0.093	0.123	0		

BLOSUM45, G11, E1	0.046	0.018	0.114	0	
PAM30, G11, E1	0.960	1.300	1.347	1.300	0

0.536

Queuine tRNA-Ribosyltransferase

25%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.094	0			
BLOSUM62, G6, E2	0.021	0.102	0		
BLOSUM45, G11, E1	0.096	0.016	0.102	0	
PAM30, G11, E1	na	na	na	na	0

0.072

tRNA Synthetase Class I (E and Q)

24%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.700	0			
BLOSUM62, G6, E2	0.021	0.712	0		
BLOSUM45, G11, E1	0.306	0.720	0.293	0	
PAM30, G11, E1	na	na	na	na	0

0.459

**DNA
Methylase**

23%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.077	0			
BLOSUM62, G6, E2	0.092	0.162	0		
BLOSUM45, G11, E1	0.134	0.202	0.049	0	
PAM30, G11, E1	na	na	na	na	0

0.119

Type II DNA Topoisomerase, Domains 2-4

22%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	0.682	0.682	0		
BLOSUM45, G11, E1	0.560	0.560	0.154	0	

PAM30, G11, E1	1.034	1.149	1.159	1.443	0	0.742
----------------	-------	-------	-------	-------	---	-------

Pyridoxal-Phosphate Dependent Enzymes

21%D	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.247	0				
BLOSUM62, G6, E2	0.376	0.466	0			
BLOSUM45, G11, E1	0.281	0.101	0.512	0		
PAM30, G11, E1	na	na	na	na	0	0.331

Iron/Ascorbate Oxidoreductase

20%D	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.542	0				
BLOSUM62, G6, E2	0.464	0.919	0			
BLOSUM45, G11, E1	0.257	0.393	0.761	0		
PAM30, G11, E1	na	na	na	na	0	0.556

FAD Binding Domain in Molybdopterin Dehydrogenase

19%D	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.000	0				
BLOSUM62, G6, E2	0.001	0.000	0			
BLOSUM45, G11, E1	0.196	0.196	0.195	0		
PAM30, G11, E1	na	na	na	na	0	0.098

Leucine Rich Repeats in Spliceosomal U2A' Protein and Internalin B

19%D	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1	
BLOSUM62, G11, E1	0					
BLOSUM62, G13, E1	0.832	0				
BLOSUM62, G6, E2	0.000	0.832	0			
BLOSUM45, G11, E1	0.755	0.078	0.755	0		
PAM30, G11, E1	na	na	na	na	0	0.542

Asp/Glu/Hydantoin Racemase

18%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.314	0			
BLOSUM62, G6, E2	0.157	0.199	0		
BLOSUM45, G11, E1	0.331	0.263	0.322	0	
PAM30, G11, E1	na	na	na	na	0

0.264

Polysaccharide Lyase Family 8, N Terminal Domain

18%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	0.095	0.095	0		
BLOSUM45, G11, E1	0.075	0.075	0.147	0	
PAM30, G11, E1	na	na	na	na	0

0.081

PHBH-like

17%ID	BLOSUM62, G11, E1	BLOSUM62, G13, E1	BLOSUM62, G6, E2	BLOSUM45, G11, E1	PAM30, G11, E1
BLOSUM62, G11, E1	0				
BLOSUM62, G13, E1	0.000	0			
BLOSUM62, G6, E2	0.513	0.513	0		
BLOSUM45, G11, E1	0.170	0.990	0.433	0	
PAM30, G11, E1	na	na	na	na	0

0.437

C. Varying Code used for Sequence Alignment- Standard RMSD Superposition.**Standard
RMSD****Serine/Threonine Protein Phosphatase****AVG**

39%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.656	0				

FASTA	1.119	1.010	0			
ALIGN	1.882	1.824	1.279	0		
CLUSTALW	1.836	1.645	1.317	0.488	0	
TCOFFEE	1.119	1.010	0.000	1.279	1.317	0

1.185

Eukaryotic Glutathione Synthase

37%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.061	0				
FASTA	0.157	0.190	0			
ALIGN	0.204	0.211	0.153	0		
CLUSTALW	0.115	0.112	0.169	0.253	0	
TCOFFEE	0.126	0.075	0.221	0.229	0.149	0

0.162

Interferon alpha, beta, and delta

35%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.789	0				
FASTA	0.730	1.737	0			
ALIGN	0.611	0.461	1.626	0		
CLUSTALW	1.706	1.072	1.481	1.200	0	
TCOFFEE	1.704	1.979	1.488	1.196	0.042	0

1.188

Adenosylmethionine Decarboxylase

33%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.947	0				
FASTA	0.424	0.918	0			
ALIGN	0.376	0.902	0.063	0		
CLUSTALW	0.377	0.924	0.157	0.122	0	
TCOFFEE	0.826	1.532	1.112	1.056	1.000	0

0.716

Clostridial Neurotoxin Zinc Protease

31%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.883	0				
FASTA	0.506	0.881	0			
ALIGN	0.506	0.881	0.000	0		
CLUSTALW	0.665	1.433	0.728	0.728	0	
TCOFFEE	0.341	0.829	0.192	0.192	0.717	0

0.632

Sulfatase

29%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
-------	-------	-----	-------	-------	----------	---------

BLAST	0						
SIM	0.551	0					
FASTA	1.898	1.773	0				
ALIGN	2.346	2.258	0.675	0			
CLUSTALW	0.685	0.697	1.375	1.781	0		
TCOFFEE	1.073	0.960	0.964	1.408	0.658	0	1.273

Protocatechuate-3,4-Dioxygenase, alpha and beta chains

28%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	3.215	0					
FASTA	0.705	3.123	0				
ALIGN	1.415	4.096	1.273	0			
CLUSTALW	6.027	5.482	6.001	7.182	0		
TCOFFEE	0.906	3.123	0.590	0.724	6.513	0	3.358

Aminotransferase Class IV

27%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	3.044	0					
FASTA	1.728	2.320	0				
ALIGN	1.787	2.135	0.237	0			
CLUSTALW	1.738	2.251	0.090	0.189	0		
TCOFFEE	1.368	3.266	1.840	1.907	1.888	0	1.719

SpoU rRNA Methylase Family

26%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	6.619	0					
FASTA	3.174	8.156	0				
ALIGN	3.244	8.037	0.304	0			
CLUSTALW	3.322	7.816	0.819	0.873	0		
TCOFFEE	3.264	8.090	0.403	0.550	0.462	0	3.676

FMN Oxidoreductase

25%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	5.619	0					
FASTA	0.669	5.159	0				
ALIGN	4.263	2.618	4.014	0			
CLUSTALW	2.039	5.616	1.825	4.551	0		
TCOFFEE	1.461	4.729	0.830	3.925	2.149	0	3.298

Queuine tRNA-Ribosyltransferase

25%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.792	0					
FASTA	0.363	0.678	0				
ALIGN	0.364	0.662	0.039	0			
CLUSTALW	0.681	0.877	0.363	0.385	0		
TCOFFEE	0.606	0.884	0.329	0.342	0.203	0	0.505

tRNA Synthetase Class I (E and Q)

24%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	2.600	0					
FASTA	2.662	2.108	0				
ALIGN	2.054	1.595	1.107	0			
CLUSTALW	1.880	1.532	1.470	1.053	0		
TCOFFEE	2.275	2.114	1.712	1.851	1.303	0	1.821

DNA Methylase

23%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	1.375	0					
FASTA	3.442	2.056	0				
ALIGN	2.871	3.523	0.626	0			
CLUSTALW	3.354	2.984	1.687	1.590	0		
TCOFFEE	2.709	3.211	1.833	1.544	2.069	0	2.325

Type II DNA Topoisomerase, Domains 2-4

22%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.000	0					
FASTA	2.110	2.110	0				
ALIGN	2.129	2.129	0.129	0			
CLUSTALW	2.557	2.557	0.805	0.836	0		
TCOFFEE	2.403	2.403	0.486	0.521	0.503	0	1.445

Pyridoxal-Phosphate Dependent Enzymes

21%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	6.141	0					
FASTA	3.761	6.370	0				
ALIGN	3.045	3.770	4.030	0			
CLUSTALW	1.698	5.344	3.735	2.224	0		

TCOFFEE	3.425	3.536	4.622	0.762	2.755	0	3.681
---------	-------	-------	-------	-------	-------	---	-------

Iron/Ascorbate Oxidoreductase

20%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	6.000	0					
FASTA	11.757	8.990	0				
ALIGN	3.924	2.287	9.824	0			
CLUSTALW	3.850	2.365	9.864	0.239	0		
TCOFFEE	4.055	2.176	9.759	0.474	0.466	0	5.069

FAD Binding Domain in Molybdopterin Dehydrogenase

19%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	2.655	0					
FASTA	0.490	2.917	0				
ALIGN	0.481	2.790	0.352	0			
CLUSTALW	0.699	2.975	0.324	0.422	0		
TCOFFEE	0.638	3.000	0.301	0.327	0.199	0	1.238

Leucine Rich Repeats in Splicesomal U2A' Protein and Internalin B

19%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	4.404	0					
FASTA	1.960	3.793	0				
ALIGN	3.057	3.996	2.641	0			
CLUSTALW	3.636	4.122	2.689	1.967	0		
TCOFFEE	2.751	3.870	1.891	1.026	1.552	0	2.890

Asp/Glu/Hydantoin Racemase

18%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	6.438	0					
FASTA	1.294	7.294	0				
ALIGN	4.732	3.049	5.701	0			
CLUSTALW	4.860	3.012	5.855	0.663	0		
TCOFFEE	8.863	4.500	9.698	4.275	4.180	0	4.961

Polysaccharide Lyase Family 8, N Terminal Domain

18%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	2.572	0					
FASTA	1.206	3.132	0				

ALIGN	2.082	2.367	2.254	0		
CLUSTALW	1.149	3.125	1.169	1.959	0	
TCOFFEE	2.230	1.670	2.798	1.258	2.367	0

2.089

PHBH-like

17%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	3.707	0				
FASTA	7.070	5.627	0			
ALIGN	4.703	4.881	6.505	0		
CLUSTALW	4.822	3.729	5.811	3.830	0	
TCOFFEE	5.285	4.515	5.962	3.199	1.517	0

4.744

D. Varying Code used for Sequence Alignment- Weighted RMSD Superposition.**Weighted
RMSD****Serine/Threonine Protein Phosphatase****AVG**

39%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.035	0				
FASTA	0.004	0.033	0			
ALIGN	0.009	0.041	0.009	0		
CLUSTALW	0.034	0.001	0.033	0.041	0	
TCOFFEE	0.004	0.034	0.000	0.009	0.033	0

0.021

Eukaryotic Glutathione Synthase

37%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.027	0				
FASTA	0.045	0.045	0			
ALIGN	0.032	0.030	0.047	0		
CLUSTALW	0.061	0.083	0.075	0.074	0	
TCOFFEE	0.030	0.034	0.062	0.022	0.070	0

0.049

Interferon alpha, beta, and delta

35%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.292	0				
FASTA	0.345	0.306	0			
ALIGN	0.220	0.217	0.228	0		
CLUSTALW	0.529	0.438	0.449	0.330	0	
TCOFFEE	0.520	0.430	0.445	0.322	0.011	0

0.339

Adenosylmethionine Decarboxylase

33%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.102	0					
FASTA	0.057	0.105	0				
ALIGN	0.055	0.107	0.014	0			
CLUSTALW	0.124	0.153	0.101	0.113	0		
TCOFFEE	0.105	0.117	0.104	0.099	0.174	0	0.102

Clostridial Neurotoxin Zinc Protease

31%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.240	0					
FASTA	0.031	0.224	0				
ALIGN	0.031	0.224	0.000	0			
CLUSTALW	0.011	0.235	0.031	0.031	0		
TCOFFEE	0.011	0.232	0.028	0.028	0.010	0	0.091

Sulfatase

29%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.231	0					
FASTA	0.072	0.169	0				
ALIGN	1.478	1.652	1.542	0			
CLUSTALW	0.184	0.088	0.120	1.643	0		
TCOFFEE	0.043	0.046	0.046	1.515	0.148	0	0.598

Protocatechuate-3,4-Dioxygenase, alpha and beta chains

28%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.230	0					
FASTA	0.087	0.240	0				
ALIGN	0.056	0.214	0.055	0			
CLUSTALW	0.150	0.252	0.090	0.133	0		
TCOFFEE	0.086	0.240	0.006	0.052	0.092	0	0.132

Aminotransferase Class IV

27%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.159	0					
FASTA	0.156	0.225	0				
ALIGN	0.152	0.243	0.037	0			
CLUSTALW	0.155	0.236	0.024	0.020	0		
TCOFFEE	0.057	0.204	0.139	0.124	0.130	0	0.137

SpoU rRNA Methylase Family

26%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.398	0				
FASTA	0.361	0.486	0			
ALIGN	0.361	0.495	0.046	0		
CLUSTALW	0.297	0.528	0.215	0.213	0	
TCOFFEE	0.258	0.207	0.428	0.449	0.457	0

0.347

FMN Oxidoreductase

25%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.418	0				
FASTA	0.117	0.443	0			
ALIGN	0.639	0.618	0.618	0		
CLUSTALW	0.085	0.369	0.140	0.577	0	
TCOFFEE	0.064	0.402	0.167	0.677	0.118	0

0.363

Queuine tRNA-Ribosyltransferase

25%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.370	0				
FASTA	0.029	0.352	0			
ALIGN	0.029	0.352	0.000	0		
CLUSTALW	0.079	0.331	0.058	0.058	0	
TCOFFEE	0.102	0.420	0.109	0.109	0.149	0

0.170

tRNA Synthetase Class I (E and Q)

24%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.629	0				
FASTA	0.088	0.870	0			
ALIGN	0.406	1.051	0.412	0		
CLUSTALW	0.521	0.982	0.503	0.173	0	
TCOFFEE	0.377	0.935	0.363	0.195		0

0.536

DNA Methylase

23%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.144	0				
FASTA	0.026	0.141	0			
ALIGN	0.057	0.125	0.061	0		
CLUSTALW	0.159	0.442	0.157	0.125	0	
TCOFFEE	0.081	0.203	0.095	0.083	0.203	0

0.140

Type II DNA Topoisomerase, Domains 2-4

22%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					

SIM	0.000	0					
FASTA	0.682	0.682	0				
ALIGN	0.700	0.700	0.054	0			
CLUSTALW	1.356	1.356	1.600	1.581	0		
TCOFFEE	1.368	1.368	1.503	1.485	0.307	0	0.983

Pyridoxal-Phosphate Dependent Enzymes

21%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.520	0					
FASTA	0.438	0.671	0				
ALIGN	0.625	0.925	0.918	0			
CLUSTALW	0.640	0.887	0.850	0.328	0		
TCOFFEE	0.667	0.974	0.903	0.182	0.204	0	0.649

Iron/Ascorbate Oxidoreductase

20%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.870	0					
FASTA	0.684	1.272	0				
ALIGN	1.662	0.951	2.123	0			
CLUSTALW	1.741	0.997	2.226	0.155	0		
TCOFFEE	1.703	1.051	2.163	0.074	0.104	0	1.185

FAD Binding Domain in Molybdopterin Dehydrogenase

19%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.904	0					
FASTA	0.675	1.242	0				
ALIGN	0.245	0.688	0.736	0			
CLUSTALW	0.320	0.611	0.844	0.118	0		
TCOFFEE	0.328	0.597	0.834	0.138	0.052	0	0.555

Leucine Rich Repeats in Splicesomal U2A' Protein and Internalin B

19%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.751	0					
FASTA	0.139	0.869	0				
ALIGN	1.067	1.614	1.037	0			
CLUSTALW	1.067	0.325	1.183	1.916	0		
TCOFFEE	0.761	0.801	0.872	1.032	1.048	0	0.965

Asp/Glu/Hydontoin Racemase

18%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE	
BLAST	0						
SIM	0.384	0					
FASTA	0.343	0.494	0				

ALIGN	0.355	0.247	0.318	0		
CLUSTALW	0.614	0.549	0.364	0.346	0	
TCOFFEE	0.584	0.624	0.290	0.426	0.267	0

0.414

Polysaccharide Lyase Family 8, N Terminal Domain

18%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	0.341	0				
FASTA	0.445	0.231	0			
ALIGN	0.751	0.502	0.336	0		
CLUSTALW	0.508	0.331	0.250	0.334	0	
TCOFFEE	0.840	0.586	0.476	0.201	0.408	0

0.436

PHBH-like

17%ID	BLAST	SIM	FASTA	ALIGN	CLUSTALW	TCOFFEE
BLAST	0					
SIM	1.074	0				
FASTA	0.942	0.651	0			
ALIGN	1.540	1.737	2.096	0		
CLUSTALW	1.134	0.661	0.474	2.245	0	
TCOFFEE	1.494	0.553	1.056	1.745	1.069	0

1.231

A2.5 An alternate version of Figure 3.4 with selenomethionine added to the sequence of 1GZ0.

Correcting the β -sheet in Figure 4 of the text. Python bioparsers ignore HET groups, regardless of their inclusion in the protein chain. In the PDB file 1GZ0, methionine residues (MET) are replaced with selenomethionine (MSE) and labeled HETATM. (A) The sequence alignment changes when X (bold, blue font) is used to represent MSE. Residues with a weighting of 40% or greater in the wRMSD overlay are shown with red asterisks, which now includes atoms from the β -sheet region (blue box). (B, next page) The weighted superposition is shown for both sequence alignments, the one without MSE as shown in Figure 4 of the paper (left) and the one with MSE included (right). The superposition is unchanged because the wRMSD procedure can compensate for the error, but the resulting weights in the β -sheet region (highlighted by the arrow) are affected. The color code of the weights is the same as in Figures 2 and 4 in the paper.

```

1IPA IKELARLLERKHRDSQRRFLIEGAREIERALQAGIELEQALVWEGGLNPPEEQQVYAALLA
      :   :   :   :   :   :   :   :   :   :   :   :   :   :   :   :
1GZ0 IHAVQALLERAPERFQEVFILKG-REDKRLLP----LIHALESQGVVIQLANRQY-----

1IPA  LLEVSEAVLKKLSVRDNPAGLIALARMPERTLEEYRSPDAL-----ILVAVGLEKPG
      : : :           : : :   : :   : :   : :   : :   : :   : :
1GZ0  LDEKSDGAVHQ-----GI IARVK-PGRQYQENDLPDLIASLDQPFLILIDGVTDPH
                        *                               *****

1IPA  NLGAVLRSADAAGAEAVLV---AGVDLYSPQVIRNSTGVVFSLRTLAASESEVLDWIK
      : : : : : : : : : : : : : : : : : : : : : : : : : : : :
1GZ0  NLGACLRSAAGVHAVIVPKDRSAQLNATAKKVACGAAESVPLIRV-TNLARTXXLQ
      *****                                           *      *

1IPA  QHNLPLVATTPHAEALYWEANLRPEVAIAVGPEHEGLRAAWLEAAQTQVRIPMQGQADSL
      :   :   :   :   :   :   :   :   :   :   :   :   :   :   :   :
1GZ0  EENIWIVGTAGEADHTLYQSKXTGFLALVXGAEGEGXRRLTREHCDELISIPXAGSVSSL
      * ***** X***** X***** *

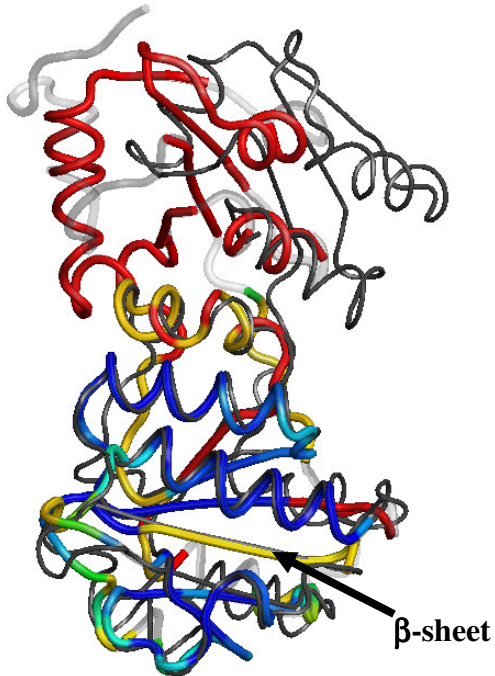
```

β -sheet

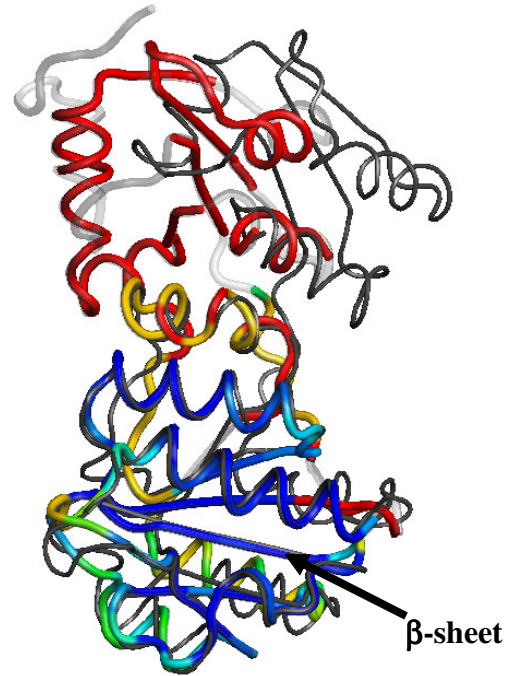
```

1IPA  NVSVSAALLLYEALRQR
      : : :   : : : :
1GZ0  NVSVATGICLFEAVRQR
      *****

```

B

wRMSD superposition when MSE **IS NOT** included in the initial sequence alignment (as in Figure 4 of the paper)



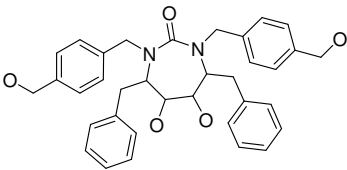
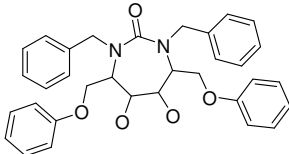
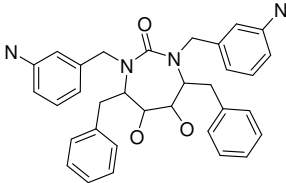
wRMSD superposition when MSE **IS** included in the initial sequence alignment (as given in section S3A above)

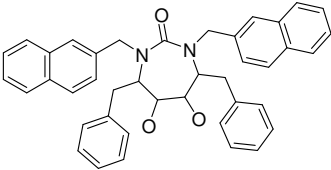
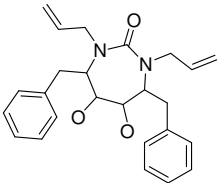
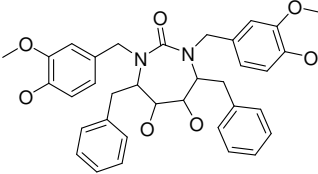
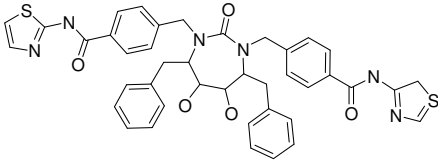
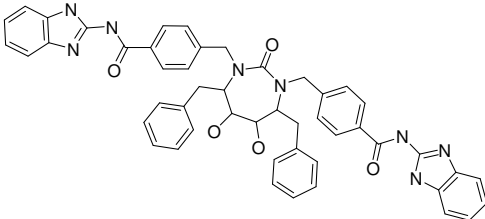
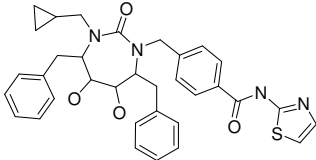
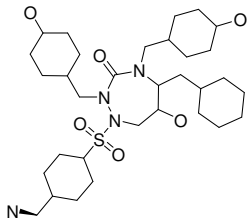
APPENDIX 3

**Exploring Experimental Sources of Multiple Protein Conformations in
Structure-Based Drug Design**

A3.1 Structures and inhibition constants of the ten unique cyclic urea ligands.

1BVE (NMR structure) and 1QBS (crystal structure) are bound to same cyclic urea ligand. Inhibition constants were obtained from the Binding MOAD database; values are pulled from the crystal structure papers referenced in A3.2.

PDB ID	K _i (nM)	Cyclic Urea Ligand
1BVE/1QBS	0.33	
1AJX	12.0	
1DMP	0.34	

1HVR	0.31	
1HWR	4.70	
1PRO	0.01	
1QBR	0.03	
1QBT	0.02	
1QBU	0.06	
1T7K	N/A	

A3.2 PDB IDs and references of 90 HIV-1 protease crystal structures.

- 1A30-** Louis, J. M.; Dyda, F.; Nashed, N. T.; Kimmel, A. R.; Davies, D. R. *Biochemistry* **1998**, *37*, 2105-2110.
- 1A8G-** Ringhofer, S.; Kallen, J.; Dutzler, R.; Billich, A.; Visser, A. J.; Scholz, D.; Steinhauser, O.; Schreiber, H.; Auer, M.; Kungl, A.J. *J. Mol. Biol.* **1999**, *286*, 1147-1159.
- 1A8K-** Weber, I. T.; Wu, J.; Adomat, J.; Harrison, R. W.; Kimmel, A. R.; Wondrak, E. M.; Louis, J. M. *Eur. J. Biochem.* **1997**, *249*, 523-530.
- 1A94-** Wu, J.; Adomat, J. M.; Ridky, T. W.; Louis, J. M.; Leis, J.; Harrison, R.W.; Weber, I. T. *Biochemistry* **1998**, *37*, 518-4526.
- 1AAQ-** Dreyer, G. B.; Lambert, D. M.; Meek, T. D.; Carr, T. J.; Tomaszek Jr. T. A.; Fernandez, A. V.; Bartus, H.; Cacciavillani, E.; Hassell, A. M.; Minnich, M.; Petteway Jr. S. R.; Metcalf, B. W. *Biochemistry* **1992**, *31*, 6646-6659.
- 1AJV-** Backbro, K.; Lowgren, S.; Osterlund, K.; Atepo, J.; Unge, T.; Hulten, J.; Bonham, N. M.; Schaal, W.; Karlen, A.; Hallberg, A. *J. Med. Chem.* **1997**, *40*, 898-902.
- 1AJX-** Backbro, K.; Lowgren, S.; Osterlund, K.; Atepo, J.; Unge, T.; Hulten, J.; Bonham, N. M.; Schaal, W.; Karlen, A.; Hallberg, A. *J. Med. Chem.* **1997**, *40*, 898-902.
- 1B6J-** Martin, J. L.; Begun, J.; Schindeler, A.; Wickramasinghe, W. A.; Alewood, D.; Alewood, P. F.; Bergman, D. A.; Brinkworth, R. I.; Abbenante, G.; March, D. R.; Reid, R. C.; Fairlie, D. P. *Biochemistry* **1999**, *38*, 7978-7988.
- 1C70-** Munshi, S.; Chen, Z.; Yan, Y.; Li, Y.; Olsen, D. B.; Schock, H. B.; Galvin, B. B.; Dorsey, B.; Kuo, L. C. *Acta Crystallogr. Sect. D* **2000**, *56*, 381-388.
- 1CPI-** Abbenante, G.; March, D. R.; Bergman, D. A.; Hunt, P. A.; Garnham, B.; Dancer, R. J.; Martin, J. L.; Fairlie, D. P. *J. Am. Chem. Soc.* **1995**, *117*, 10220-10226.
- 1D4H-** Andersson, H. O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U. H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.
- 1D4I-** Andersson, H. O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U. H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.

1D4J- Andersson, H. O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U. H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.

1D4Y- Thaisrivongs, S.; Skulnick, H.I.; Turner, S.R.; Strohbach, J.W.; Tommasi, R.A.; Johnson, P. D.; Aristoff, P. A.; Judge, T. M.; Gammill, R. B.; Morris, J. K.; Romines, K. R.; Chrusciel, R. A.; Hinshaw, R. R.; Chong, K. T.; Tarpley, W. G.; Poppe, S. M.; Slade, D. E.; Lynn, J. C.; Horng, M. M.; Tomich, P. K.; Seest, E. P.; Dolak, L. A.; Howe, W. J.; Howard, G. M.; Schwende, F. M.; Toth, L. N.; Padbury, G. E.; Wilson, G. J.; Shiou, L.; Zipp, G. L.; Wilkinson, K. F.; Rush, B. D.; Ruwart, M. J.; Koeplinger, K. A.; Zhao, Z.; Cole, S.; Zaya, R. M.; Kakuk, T. J.; Janakiraman, N. M.; Watenpugh, K.D. *J. Med. Chem.* **1996**, *39*, 4349-4353.

1DIF- Silva, A. M.; Cachau, R. E. Sham, H. L.; Erickson, J. W. *J. Mol. Biol.* **1996**, *255*, 321-346.

1DMP- Hodge, C. N.; Aldrich, P. E.; Bacheler, L. T.; Chang, C. H.; Eyermann, C. J.; Garber, S.; Grubb, M.; Jackson, D. A.; Jadhav, P. K.; Korant, B.; Lam, P. Y.; Maurin, M. B.; Meek, J. L.; Otto, M. J.; Rayner, M. M.; Reid, C.; Sharpe, T. R.; Shum, L.; Winslow, D. L.; Erickson-Viitanen, S. *Chem. Biol.* **1996**, *3*, 301-314.

1EBW- Andersson, H. O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U. H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.

1EBY- Andersson, H. O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U. H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.

1EBZ- Andersson, H. O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U. H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.

1EC0- Lindberg, J.; Pyring, D.; Lowgren, S.; Rosenquist, A.; Zuccarello, G.; Kvarnstrom, I.; Zhang, H.; Vrang, L.; Classon, B.; Hallberg, A.; Samuelsson, B.; Unge, T. *Eur. J. Biochem.* **2004**, *271*, 4594-4602.

1EC1- Andersson, H.O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U.H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.

1EC2- Andersson, H. O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U. H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.

1EC3- Andersson, H. O.; Fridborg, K.; Lowgren, S.; Alterman, M.; Muhlman, A.; Bjorsne, M.; Garg, N.; Kvarnstrom, I.; Schaal, W.; Classon, B.; Karlen, A.; Danielsson, U. H.; Ahlsen, G.; Nillroth, U.; Vrang, L.; Oberg, B.; Samuelsson, B.; Hallberg, A.; Unge, T. *Eur. J. Biochem.* **2003**, *270*, 1746-1758.

1FGC- Mahalingam, B.; Louis, J. M.; Hung, J.; Harrison, R. W.; Weber, I. T. *Proteins: Struct., Funct., Bioinf.* **2001**, *43*, 455-464.

1G2K- Schaal, W.; Karlsson, A.; Ahlsen, G.; Lindberg, J.; Andersson, H. O.; Danielson, U. H.; Classon, B.; Unge, T.; Samuelsson, B.; Hulten, J.; Hallberg, A.; Karlen, A. *J. Med. Chem.* **2001**, *44*, 155-169.

1G35- Schaal, W.; Karlsson, A.; Ahlsen, G.; Lindberg, J.; Andersson, H. O.; Danielson, U. H.; Classon, B.; Unge, T.; Samuelsson, B.; Hulten, J.; Hallberg, A.; Karlen, A. *J. Med. Chem.* **2001**, *44*, 155-169.

1HBV- Hoog, S. S.; Zhao, B.; Winborne, E.; Fisher, S.; Green, D. W.; DesJarlais, R. L.; Newlander, K. A.; Callahan, J. F.; Abdel-Meguid, S. S.; Moore, M. L.; Huffman, W. F. *J. Med. Chem.* **1995**, *38*, 3246-3252.

1HEF- Murthy, K. H.; Winborne, E. L.; Minnich, M. D.; Culp, J. S.; Debouck, C. J. *Biol. Chem.* **1992**, *267*, 22770-22778.

1HEG- Murthy, K. H.; Winborne, E. L.; Minnich, M. D.; Culp, J. S.; Debouck, C. J. *Biol. Chem.* **1992**, *267*, 22770-22778.

1HIH- Priestle, J. P.; Fassler, A.; Rosel, J.; Tintelnot-Blomley, M.; Strop, P.; Grutter, M. G. *Structure* **1995**, *3*, 381-389.

1HIV- Thanki, N.; Rao, J. K.; Foundling, S. I.; Howe, W. J.; Moon, J. B.; Hui, J. O.; Tomasselli, A. G.; Henrikson, R. L.; Thaisrivongs, S.; Wlodawer, A. *Protein Sci.* **1992**, *1*, 1061-1072.

1HOS- Abdel-Meguid, S. S.; Zhao, B.; Murthy, K. H.; Winborne, E.; Choi, J. K.; DesJarlais, R. L.; Minnich, M. D.; Culp, J. S.; Debouck, C.; Tomaszek Jr., T. A.; Meek, T. D.; Dreyer, G. B. *Biochemistry* **1993**, *32*, 7972-7980.

1HPO- Skulnick, H. I.; Johnson, P. D.; Aristoff, P. A.; Morris, J. K.; Lovasz, K. D.; Howe, W. J.; Watenpaugh, K. D.; Janakiraman, M. N.; Anderson, D. J.; Reischer, R. J.; Schwartz, T. M.; Banitt, L. S.; Tomich, P. K.; Lynn, J. C.; Horng, M. M.; Chong, K. T.; Hinshaw, R. R.; Dolak, L. A.; Seest, E. P.; Schwende, F. J.; Rush,

B. D.; Howard, G. M.; Toth, L. N.; Wilkinson, K. R.; Kakuk, T. J.; Johnson, C. W.; Cole, S. L.; Zaya, R. M.; Zipp, G. L.; Possert, P. L.; Dalga, R. J.; Zhong, W-Z.; Williams, M. G.; Romines, K. R. *J. Med. Chem.* **1997**, *40*, 1149-1164.

1HPS- Thompson, S. K.; Murthy, K. H.; Zhao, B.; Winborne, E.; Green, D. W.; Fisher, S. M.; DesJarlais, R. L.; Tomaszek Jr. T. A.; Meek, T. D.; Gleason, J. G.; Abdel-Meguid, S. S. *J. Med. Chem.* **1994**, *37*, 3100-3107.

1HPV- Kim, E. E.; Baker, C. T.; Dwyer, M. D.; Murcko, M. A.; Rao, B. G.; Tung, R. D.; Navia, M. A. *J. Am. Chem. Soc.* **1995**, *117*, 1181.

1HPX- Baldwin, E. T.; Bhat, T. N.; Gulnik, S.; Liu, B.; Topol, I. A.; Kiso, Y.; Mimoto, T.; Mitsuya, H.; Erickson, J.W. *Structure* **1995**, *3*, 581-590.

1HSG- Chen, Z.; Li, Y.; Chen, E.; Hall, D. L.; Darke, P. L.; Culberson, C.; Shafer, J. A.; Kuo, L. C. *J. Biol. Chem.* **1994**, *269*, 26344-26348.

1HTF- Jhoti, H.; Singh, O. M.; Weir, M. P.; Cooke, R.; Murray-Rust, P.; Wonacott, A. *Biochemistry* **1994**, *33*, 8417-8427.

1HTG- Jhoti, H.; Singh, O. M.; Weir, M. P.; Cooke, R.; Murray-Rust, P.; Wonacott, A. *Biochemistry* **1994**, *33*, 8417-8427.

1HVI- Hosur, M. V.; Bhat, T. N.; Kempf, D.; Baldwin, E. T.; Liu, B.; Gulnik, S.; Wideburg, N. E.; Norbeck, D. W.; Appelt, K.; Erickson, J. W. *J. Am. Chem. Soc.* **1994**, *116*, 847.

1HVJ- Hosur, M. V.; Bhat, T. N.; Kempf, D.; Baldwin, E. T.; Liu, B.; Gulnik, S.; Wideburg, N. E.; Norbeck, D. W.; Appelt, K.; Erickson, J. W. *J. Am. Chem. Soc.* **1994**, *116*, 847.

1HVK- Hosur, M. V.; Bhat, T. N.; Kempf, D.; Baldwin, E. T.; Liu, B.; Gulnik, S.; Wideburg, N. E.; Norbeck, D. W.; Appelt, K.; Erickson, J. W. *J. Am. Chem. Soc.* **1994**, *116*, 847.

1HVL- Hosur, M. V.; Bhat, T. N.; Kempf, D.; Baldwin, E. T.; Liu, B.; Gulnik, S.; Wideburg, N. E.; Norbeck, D. W.; Appelt, K.; Erickson, J. W. *J. Am. Chem. Soc.* **1994**, *116*, 847.

1HVR- Lam, P. Y.; Jadhav, P. K.; Eyermann, C. J.; Hodge, C. N.; Ru, Y., Bachelier, L. T.; Meek, J. L.; Otto, M. J.; Rayner, M. M.; Wong, Y. N.; Chang, C-H.; Weber, P. C.; Jackson, D. A.; Sharpe, T. R.; Erickson-Viitanen, S. *Science* **1994**, *263*, 380-384.

1HWR- Ala, P. J.; Huston, E. E.; Klabe, R. M.; Jadhav, P. K.; Lam, P. Y.; Chang, C. H. *Biochemistry* **1998**, *37*, 15042-15049.

1HXB- Krohn, A.; Redshaw, S.; Ritchie, J. C.; Graves, B. J.; Hatada, M. H. *J. Med. Chem.* **1991**, *34*, 3340-3342.

1HXW- Kempf, D. J.; Marsh, K. C.; Denissen, J. F.; McDonald, E.; Vasavanonda, S.; Flentge, C. A.; Green, B. E.; Fino, L.; Park, C. H.; Kong, X. P. Wideburg, N. E.; Saldivar, A.; Ruiz, L.; Kati, W. M.; Sham, H. L.; Robins, T.; Stewart, K. D.; Hsu, A.; Plattner, J. J.; Leonard, J. M.; Norbeck, D. W. *Proc. Natl. Acad. Sci. USA* **1995**, *92*, 2484-2488.

1IZH- Weber, J.; Mesters, J. R.; Lepšik, M.; Prejdova, J.; Svec, M.; Sponarova, J.; Mlcochova, P.; Skalicka, K.; Strisovsky, K.; Uhlikova, T.; Soucek, M.; Machala, L.; Stankova, M.; Vondrasek, J.; Klimkait, T.; Kraeusslich, H-G.; Hilgenfeld, R.; Konvalinka, J. *J. Mol. Biol.* **2002**, *324*, 739-754.

1MRW- Vega, S.; Kang, L-W.; Velazquez-Campoy, A.; Kiso, Y.; Amzel, L. M.; Freire, E. *Proteins: Struct., Funct., Bioinf.* **2004**, *55*, 594-602.

1MSM- Vega, S.; Kang, L-W.; Velazquez-Campoy, A.; Kiso, Y.; Amzel, L. M.; Freire, E. *Proteins: Struct., Funct., Bioinf.* **2004**, *55*, 594-602.

1NPA- Smith III, A. B.; Hirschmann, R.; Pasternak, A.; Yao, W.; Sprengeler, P. A.; Holloway, M. K.; Kuo, L. C.; Chen, Z.; Darke, P. L.; Schleif, W. A. *J. Med. Chem.* **1997**, *40*, 2440-2444.

1NPV- Smith III, A. B.; Cantin, L. D.; Pasternak, A.; Guise-Zawacki, L.; Yao, W.; Charnley, A. K.; Barbosa, J.; Sprengeler, P. A.; Hirschmann, R.; Munshi, S.; Olsen, D. B.; Schleif, W. A.; Kuo, L. C. *J. Med. Chem.* **2003**, *46*, 1831-1844.

1NPW- Smith III, A. B.; Cantin, L. D.; Pasternak, A.; Guise-Zawacki, L.; Yao, W.; Charnley, A. K.; Barbosa, J.; Sprengeler, P. A.; Hirschmann, R.; Munshi, S.; Olsen, D. B.; Schleif, W. A.; Kuo, L. C. *J. Med. Chem.* **2003**, *46*, 1831-1844.

1ODW- Kervinen, J.; Thanki, N.; Zdanov, A.; Tino, J.; Barrish, J.; Lin, P. F.; Colonno, R.; Riccardi, K.; Samanta, H.; Wlodawer, A. *Protein Pept. Lett.* **1996**, *3*, 399.

1ODY- Kervinen, J.; Lubkowski, J.; Zdanov, A.; Bhatt, D.; Dunn, B. M.; Hui, K. Y.; Powell, D. J.; Kay, J.; Wlodawer, A.; Gustchina, A. *Protein Sci.* **1998**, *7*, 2314-2323.

1OHR- Kaldor, S. W.; Kalish, V. J.; Davies 2nd., J. F.; Shetty, B. V.; Fritz, J. E.; Appelt, K.; Burgess, J. A.; Campanale, K. M.; Chirgadze, N. Y.; Clawson, D. K.; Dressman, B. A.; Hatch, S. D.; Khalil, D. A.; Kosa, M. B.; Lubbehusen, P. P.; Muesing, M. A.; Patick, A. K.; Reich, S. H.; Su, K. S.; Tatlock, J. H. *J. Med. Chem.* **1997**, *40*, 3979-3985.

1PRO- Sham, H. L.; Zhao, C.; Stewart, K. D.; Betebenner, D. A.; Lin, S.; Park, C. H.; Kong, X. P.; Rosenbrook, W., Jr. Herrin, T.; Madigan, D.; Vasavanonda, S.; Lyons, N.; Molla, A.; Saldivar, A.; Marsh, K. C.; McDonald, E.; Wideburg, N. E.; Denissen, J. F.; Robins, T.; Kempf, D. J.; Plattner, J. J.; Norbeck, D. W. *J. Med. Chem.* **1996**, *39*, 392-397.

1QBR- Jadhav, P. K.; Ala, P.; Woerner, F. J.; Chang, C. H.; Garber, S. S.; Anton, E. D.; Bacheler, L. T. *J. Med. Chem.* **1997**, *40*, 181-191.

1QBS- Lam, P. Y.; Ru, Y.; Jadhav, P. K.; Aldrich, P. E.; DeLucca, G. V.; Eyermann, C. J.; Chang, C. H.; Emmett, G.; Holler, E. R.; Daneker, W. F.; Li, L.; Confalone, P. N.; McHugh, R. J.; Han, Q.; Li, R.; Markwalder, J. A.; Seitz, S. P.; Sharpe, T. R.; Bacheler, L. T.; Rayner, M. M.; Klabe, R. M.; Shum, L.; Winslow, D. L.; Kornhauser, D. M.; Jackson, D. A.; Erickson-Viitanen, S.; Hodge, C. N. *J. Med. Chem.* **1996**, *39*, 3514-3525.

1QBT- Jadhav, P. K.; Ala, P.; Woerner, F. J.; Chang, C. H.; Garber, S. S.; Anton, E. D.; Bacheler, L. T. *J. Med. Chem.* **1997**, *40*, 181-191.

1QBU- Jadhav, P. K.; Ala, P.; Woerner, F. J.; Chang, C. H.; Garber, S. S.; Anton, E. D.; Bacheler, L. T. *J. Med. Chem.* **1997**, *40*, 181-191.

1SBG- Abdel-Meguid, S. S.; Metcalf, B. W.; Carr, T. J.; Demarsh, P.; DesJarlais, R. L.; Fisher, S.; Green, D. W.; Ivanoff, L.; Lambert, D. M.; Murthy, K. H.; Petteway Jr., S. R.; Pitts, W. J.; Tomaszek Jr., T. A.; Winborne, E.; Zhao, B.; Dreyer, G. B.; Meek, T. D. *Biochemistry* **1994**, *33*, 11671-11677.

1T7K- Huang, P. P.; Randolph, J. T.; Klein, L. L.; Vasavanonda, S.; Dekhtyar, T.; Stoll, V. S.; Kempf, D. J. *Bioorg. Med. Chem. Lett.* **2004**, *14*, 4075-4078.

1UPJ- Thaisrivongs, S.; Watenpaugh, K. D.; Howe, W. J.; Tomich, P. K.; Dolak, L. A.; Chong, K. T.; Tomich, C. C.; Tomasselli, A. G.; Turner, S. R.; Strohbach, J. W.; Mulichak, A. M.; Janakiraman, M. N.; Moon, J. B.; Lynn, J. C.; Horng, M-M.; Hinshaw, R. R.; Curry, K. A.; Rothrock, D. J. *J. Med. Chem.* **1995**, *38*, 3624-3637.

1VIJ- Lange-Savage, G.; Berchtold, H.; Liesum, A.; Budt, K. H.; Peyman, A.; Knolle, J.; Sedlacek, J.; Fabry, M.; Hilgenfeld, R. *Eur. J. Biochem.* **1997**, *248*, 313-322.

1VIK- Lange-Savage, G.; Berchtold, H.; Liesum, A.; Budt, K. H.; Peyman, A.; Knolle, J.; Sedlacek, J.; Fabry, M.; Hilgenfeld, R. *Eur. J. Biochem.* **1997**, *248*, 313-322.

1W5V- Lindberg, J.; Pyring, D.; Loewgren, S.; Rosenquist, A.; Zuccarello, G.; Kvarnstrom, I.; Zhang, H.; Vrang, L.; Claesson, B.; Hallberg, A.; Samuelsson, B.; Unge, T. *Eur. J. Biochem.* **2004**, *271*, 4594.

1W5W- Lindberg, J.; Pyring, D.; Loewgren, S.; Rosenquist, A.; Zuccarello, G.; Kvarnstrom, I.; Zhang, H.; Vrang, L.; Claesson, B.; Hallberg, A.; Samuelsson, B.; Unge, T. *Eur. J. Biochem.* **2004**, *271*, 4594.

1W5X- Lindberg, J.; Pyring, D.; Loewgren, S.; Rosenquist, A.; Zuccarello, G.; Kvarnstrom, I.; Zhang, H.; Vrang, L.; Claesson, B.; Hallberg, A.; Samuelsson, B.; Unge, T. *Eur. J. Biochem.* **2004**, *271*, 4594.

1W5Y- Lindberg, J.; Pyring, D.; Loewgren, S.; Rosenquist, A.; Zuccarello, G.; Kvarnstrom, I.; Zhang, H.; Vrang, L.; Claesson, B.; Hallberg, A.; Samuelsson, B.; Unge, T. *Eur. J. Biochem.* **2004**, *271*, 4594.

1WBK- Lindberg, J.; Unge, T. To be Published.

1WBM- Lindberg, J.; Unge, T. To be Published.

1XL5- Specker, E.; Boettcher, J.; Lilie, H.; Heine, A.; Schoop, A.; Muller, G.; Griebenow, N.; Klebe, G. *Angew Chem. Int. Ed. Engl.* **2005**, *44*, 3140-3144.

1YTG- Rose, R. B.; Craik, C. S.; Douglas, N. L.; Stroud, R. M. *Biochemistry* **1996**, *35*, 12933-12944.

1YTH- Rose, R. B.; Craik, C. S.; Douglas, N. L.; Stroud, R. M. *Biochemistry* **1996**, *35*, 12933-12944

1ZP8- Brik, A.; Alexandratos, J. N.; Elder, J. H.; Olson, A. J.; Wlodawer, A.; Goodsell, D. S.; Wong, C. H. *ChemBiochem* **2005**, *6*, 1-4.

1ZPA- Brik, A.; Alexandratos, J. N.; Elder, J. H.; Olson, A. J.; Wlodawer, A.; Goodsell, D. S.; Wong, C. H. *ChemBiochem* **2005**, *6*, 1-4.

2AID- Rutenber, E.; Fauman, E. B.; Keenan, R. J.; Fong, S.; Furth, P. S.; Ortiz de Montellano, P. R.; Meng, E.; Kuntz, I. D.; DeCamp, D. L.; Salto, R.; Rose, J. R.; Craik, C. S.; Stroud, R. M. *J. Biol. Chem.* **1993**, *268*, 15343-15346.

2BB9- Smith III, A. B.; Charnley, A. K.; Harada, H.; Beiger, J. J.; Cantin, L. D.; Kenesky, C. S.; Hirschmann, R.; Munshi, S.; Olsen, D. B.; Stahlhut, M. W.; Schleif, W. A.; Kuo, L. C. *Bioorg. Med. Chem. Lett.* **2006**, *16*, 859-863.

2BBB- Smith III, A. B.; Charnley, A. K.; Harada, H.; Beiger, J. J.; Cantin, L. D.; Kenesky, C. S.; Hirschmann, R.; Munshi, S.; Olsen, D. B.; Stahlhut, M. W.; Schleif, W. A.; Kuo, L. C. *Bioorg. Med. Chem. Lett.* **2006**, *16*, 859-863.

2BPV- Munshi, S.; Chen, Z.; Li, Y.; Olsen, D. B.; Fraley, M. E.; Hungate, R. W.; Kuo, L. C. *Acta Crystallogr. Sect. D* **1998**, *54*, 1053-1060.

2BPY- Munshi, S.; Chen, Z.; Li, Y.; Olsen, D. B.; Fraley, M. E.; Hungate, R. W.; Kuo, L. C. *Acta Crystallogr. Sect. D* **1998**, *54*, 1053-1060.

2BPZ- Munshi, S.; Chen, Z.; Li, Y.; Olsen, D. B.; Fraley, M. E.; Hungate, R. W.; Kuo, L. C. *Acta Crystallogr. Sect. D* **1998**, *54*, 1053-1060.

3AID- Rutenber, E. E.; McPhee, F.; Kaplan, A. P.; Gallion, S. L.; Hogan Jr. J. C.; Craik, C. S.; Stroud, R. M. *Bioorg. Med. Chem.* **1996**, *4*, 1545-1558.

2BPV- Munshi, S.; Chen, Z.; Li, Y.; Olsen, D. B.; Fraley, M. E.; Hungate, R. W.; Kuo, L. C. *Acta Crystallogr. Sect. D* **1998**, *54*, 1053-1060.

3TLH- Li, M.; Morris, G. M.; Lee, T.; Laco, G. S.; Wong, C. H.; Olson, A. J.; Elder, J. H.; Wlodawer, A.; Gustchina, A. *Proteins: Struct., Funct., Bioinf.* **2000**, *38*, 29-40.

4HVP- Miller, M.; Schneider, J.; Sathyanarayana, B. K.; Toth, M. V.; Marshall, G. R.; Clawson, L.; Selk, L.; Kent, S. B.; Wlodawer, A. *Science* **1989**, *246*, 1149-1152.

4PHV- Bone, R.; Vacca, J. P.; Anderson, P. S.; Holloway, M. K. *J. Am. Chem. Soc.* **1991**, *113*, 9382.

7HVP- Swain, A. L.; Miller, M. M.; Green, J.; Rich, D. H.; Schneider, J.; Kent, S. B.; Wlodawer, A. *Proc. Natl. Acad. Sci. USA* **1990**, *87*, 8805-8809.

7UPJ- Skulnick, H.I.; Johnson, P.D.; Aristoff, P.A.; Morris, J.K.; Lovasz, K.D.; Howe, W.J.; Watenpaugh, K.D.; Janakiraman, M.N.; Anderson, D.J.; Reischer, R.J.; Schwartz, T.M.; Banitt, L.S.; Tomich, P.K.; Lynn, J.C.; Horng, M.M.; Chong, K.T.; Hinshaw, R.R.; Dolak, L.A.; Seest, E.P.; Schwende, F.J.; Rush, B.D.; Howard, G.M.; Toth, L.N.; Wilkinson, K.R.; Kakuk, T. J.; Johnson, C. W.; Cole, S. L.; Zaya, R. M.; Zipp, G. L. Possert, P. L. Dalga, R. J. Zhong, W-Z.; Williams, M. G.; Romines, K.R. *J. Am. Chem. Soc.* **1997**, *40*, 1149-1164.

8HVP- Jaskolski, M.; Tomasselli, A. G.; Sawyer, T. K.; Staples, D. G.; Heinrikson, R. L.; Schneider, J.; Kent, S. B.; Wlodawer, A. *Biochemistry* **1991**, *30*, 1600-1609.

A3.3 MPS NMR and crystal structure pharmacophore models.

Coordinates and RMSD for the pharmacophore models (radii of the elements are determined from the RMSD). Coordinates provided are relative to the restrained minimized average NMR structure.

A. MPS NMR pharmacophore model.

Elem. Type	Coordinates (x, y, z)	RMSD, Å
Donor	-0.1611, -3.0595, -5.2083	1.04
Donor	2.6602, -3.2735, -4.1087	0.90
AroHyd	-4.1929, -2.7413, -9.9160	1.07
AroHyd	6.6589, -4.9417, -0.1864	1.04
AroHyd	1.3605, -4.1510, -2.3492	0.88
AroHyd	1.3024, -2.7065, -7.4530	0.91
AroHyd	2.7215, -3.2545, -9.4311	0.87
AroHyd	-0.9180, -6.3930, -1.3636	1.08
Ex Vol	-0.9630, -0.0937, -3.6953	1.50
Ex Vol	3.1599, 0.1924, -4.0376	1.50

B. Average NMR pharmacophore model.

Elem. Type	Coordinates (x, y, z)	RMSD, Å
Donor	2.2589, -3.4029, -4.6702	0.75
Donor	-0.3882, -3.3633, -5.5292	0.66
Donor	-0.1471, -0.7050, -11.8712	1.01
Donor	3.5497, -4.8143, 2.1128	0.50
Aromatic	-0.0177, -5.6019, -0.9173	0.44
Aromatic	2.5906, -3.1309, -9.3923	0.36
AroHyd	-3.9996, -2.7591, -10.0458	0.70
AroHyd	0.9839, -4.1726, -2.5660	0.65
AroHyd	1.1833, -2.9642, -7.6859	0.74
AroHyd	6.6757, -5.0684, 0.0844	0.63
Ex Vol	-1.3370, -0.3760, -4.0790	1.50
Ex Vol	2.9700, 0.1380, -4.2960	1.50

C. MPS cu-crystal pharmacophore model.

Elem. Type	Coordinates (x, y, z)	RMSD, Å
Donor	2.4152, -2.4095, -4.7692	0.62
Donor	-0.0731, -2.5531, -4.5108	0.56
Acceptor	1.3437, -6.6043, -5.6621	0.80
Aromatic	0.6738, -6.7983, 2.2868	1.35
Aromatic	1.7913, -2.4492, -11.7215	0.60
AroHyd	-3.3554, -2.9588, -9.7970	1.04
AroHyd	1.4747, -3.4045, -2.8660	0.75
AroHyd	0.6975, -2.5271, -6.4520	0.75
AroHyd	5.9341, -5.1557, -0.3034	0.89
AroHyd	3.2395, -4.1574, -9.0405	1.11
AroHyd	-0.5012, -5.9069, -1.6563	1.17
Ex Vol	-1.4028, 0.4679, -3.4277	1.50
Ex Vol	3.5914, 0.7441, -4.1807	1.50

D. MPS all-crystal pharmacophore model.

Elem. Type	Coordinates (x, y, z)	RMSD, Å
Donor	-0.1078, -2.6451, -4.6182	0.68
Donor	2.4448, -2.4559, -4.8834	0.70
Acceptor	1.5220, -6.3965, -5.6347	0.53
Aromatic	-0.1250, -3.3718, -6.9840	1.05
Aromatic	2.7582, -4.2537, -2.9076	0.96
Aromatic	1.8569, -1.9610, -12.0313	1.01
Aromatic	0.6104, -5.7236, 2.1739	1.06
AroHyd	-3.6092, -3.0254, -9.5565	0.99
AroHyd	6.0061, -5.0509, -0.4301	0.98
AroHyd	-0.7590, -5.8742, -1.8651	1.09
AroHyd	3.2866, -3.6945, -8.8517	1.18
Ex Vol	-1.3470, 0.4254, -3.4447	1.50
Ex Vol	3.5239, 0.8270, -4.1748	1.50

A3.4 Raw pharmacophore screening data.

The raw data from all pharmacophore models screens. The first column, W, provides the factor that was multiplied by the consensus cluster RMSD. The second column contains the % true positives values found by dividing the number of known active hits by 89 and multiplying by 100. The third column reports the % inactive hits divided by 85 and multiplied by 100. The last column contains distance values ($\sqrt{(0 - \% \text{falsepositives})^2 + (1 - \% \text{truepositives})^2}$); the smaller the value the closer the point on the ROC plot is to (100% true positives, 0% false positives) indicating the most optimal pharmacophore models. The bolded numbers represent the optimal pharmacophore models, again optimal being defined as the models that reduce the amount of false positives while sacrificing the least true positives.

A. MPS NMR pharmacophore model- 89 Known Inhibitors vs. 85 Decoys.

W	% True Positives			% False Positives			distance from (0,100)		
	6 of 8	7 of 8	8 of 8	6 of 8	7 of 8	8 of 8	6 of 8	7 of 8	8 of 8
1	65	22	0	5	0	0	35	78	100
1 1/3	87	55	15	9	1	0	16	45	85
1 2/3	94	80	43	22	6	0	23	21	57
2	99	90	66	32	11	0	32	15	34
2 1/3	99	92	75	36	14	5	36	16	25
2 2/3	99	93	87	41	20	9	41	21	16
3	99	95	90	44	27	13	44	27	16

B. Average NMR pharmacophore model- 89 Known Inhibitors vs. 85 Decoys.

W	% True Positives			% False Positives			distance from (0,100)		
	8 of 10	9 of 10	10 of 10	8 of 10	9 of 10	10 of 10	8 of 10	9 of 10	10 of 10
1	0	0	0	0	0	0	100	100	100
1 1/3	0	0	0	0	0	0	100	100	100
1 2/3	1	0	0	0	0	0	99	100	100
2	6	0	0	1	0	0	74	100	100
2 1/3	26	1	0	1	0	0	74	99	100
2 2/3	54	8	0	2	0	0	46	92	100
3	69	14	0	4	1	0	32	87	100

C. MPS cu-crystal pharmacophore model- 89 Known Inhibitors vs. 85 Decoys.

W	% True Positives			% False Positives			distance from (0,100)		
	9 of 10	10 of 11	11 of 11	9 of 10	10 of 11	11 of 11	9 of 10	10 of 11	11 of 11
1	0	0	0	0	0	0	100	100	100
1 1/3	0	0	0	0	0	0	100	100	100
1 2/3	10	0	0	0	0	0	90	100	100
2	34	7	0	0	0	0	66	93	100
2 1/3	71	22	1	0	0	0	29	78	99
2 2/3	89	54	11	6	0	0	13	46	89
3	90	72	21	7	1	0	12	28	79
3 1/3	92	84	39	13	6	0	15	17	61
3 2/3	96	85	56	17	7	1	17	16	44
4	96	87	65	22	8	4	23	16	35

D. MPS all-crystal pharmacophore model- 89 Known Inhibitors vs. 85 Decoys.

W	% True Positives				% False Positives			
	8 of 11	9 of 11	10 of 11	11 of 11	8 of 11	9 of 11	10 of 11	11 of 11
1	0	0	0	0	0	0	0	0
1 1/3	0	0	0	0	0	0	0	0
1 2/3	15	0	0	0	0	0	0	0
2	50	5	0	0	2	0	0	0
2 1/3	71	24	1	0	8	1	0	0
2 2/3	87	56	11	0	14	2	0	0
3	87	67	34	3	19	7	4	0

D. cont.

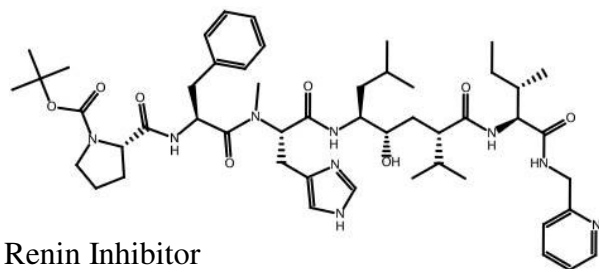
W	distance from (0,100)			
	8 of 11	9 of 11	10 of 11	11 of 11
1	100	100	100	100
1 1/3	100	100	100	100
1 2/3	85	100	100	100
2	51	96	100	100
2 1/3	30	76	99	100
2 2/3	20	44	89	100
3	23	32	66	97

E. MPS NMR pharmacophore model- 89 Known Inhibitors vs. 2322 Compounds.

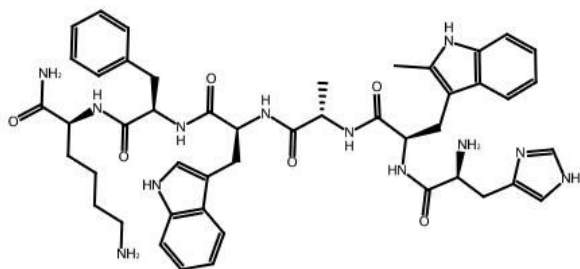
W	% True Positives			% False Positives			distance from (0,100)		
	6 of 8	7 of 8	8 of 8	6 of 8	7 of 8	8 of 8	6 of 8	7 of 8	8 of 8
1	65	22	0	0	0	0	34	78	100
1 1/3	87	55	15	2	0	0	14	45	85
1 2/3	94	80	43	12	1	0	13	20	57
2	99	90	66	27	3	0	27	10	34
2 1/3	99	92	75	43	8	0	43	11	25
2 2/3	99	93	87	64	17	2	64	19	14
3	99	95	90	76	29	4	76	29	11

F. MPS cu-crystal pharmacophore model- 89 Known Inhibitors vs. 2322 Compounds.

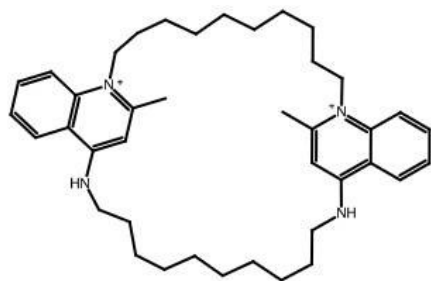
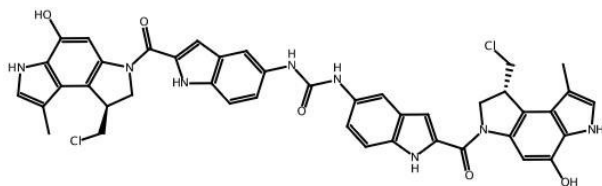
W	% True Positives			% False Positives			distance from (0,100)		
	9 of 10	10 of 11	11 of 11	9 of 10	10 of 11	11 of 11	9 of 10	10 of 11	11 of 11
1	0	0	0	0	0	0	100	100	100
1 1/3	0	0	0	0	0	0	100	100	100
1 2/3	10	0	0	0	0	0	90	100	100
2	34	7	0	0	0	0	66	93	100
2 1/3	71	22	1	0	0	0	29	78	99
2 2/3	89	54	11	3	0	0	12	46	89
3	90	72	21	6	1	0	12	28	79
3 1/3	92	84	39	14	2	0	16	16	61
3 2/3	96	85	56	25	4	0	25	15	44
4	96	87	65	35	8	1	35	16	35

A3.5 Identified false positives.**A. 4 False Positives for the optimal average NMR pharmacophore model: 8/10, 3×RMSD.**

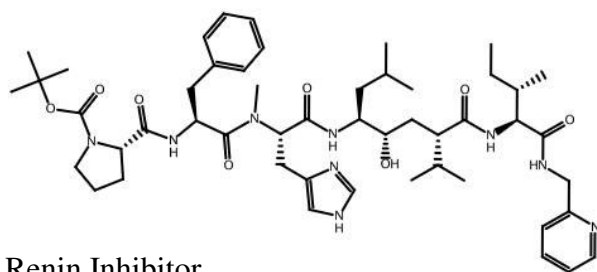
Renin Inhibitor



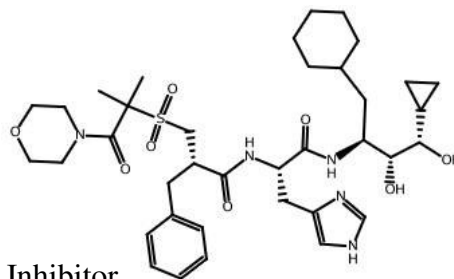
Growth Promoter



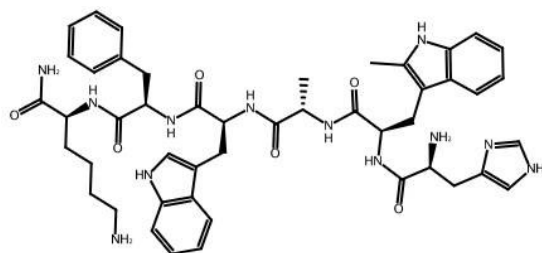
B. 9 False Positives for the optimal NMR pharmacophore model: 7/8, 2.0×RMSD.



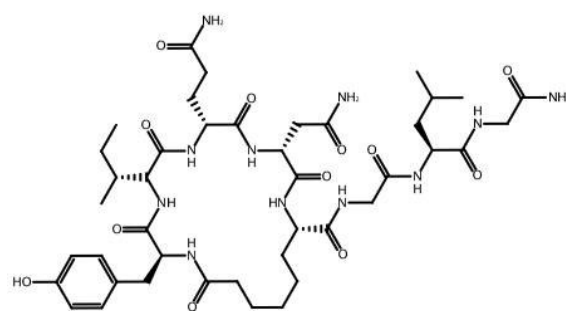
Renin Inhibitor



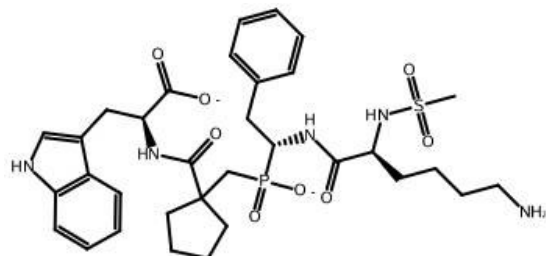
Renin Inhibitor



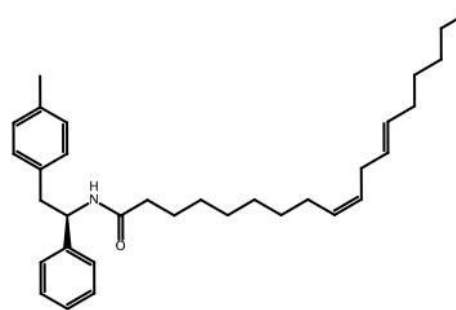
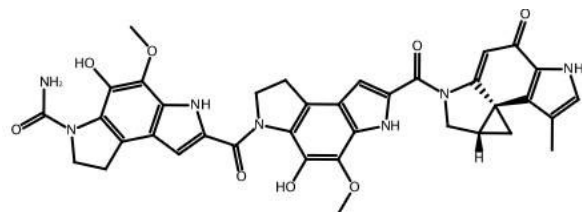
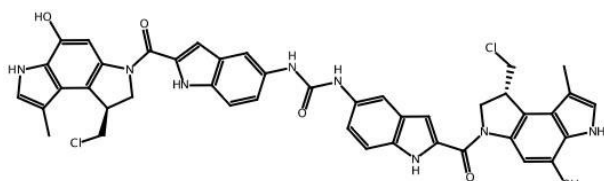
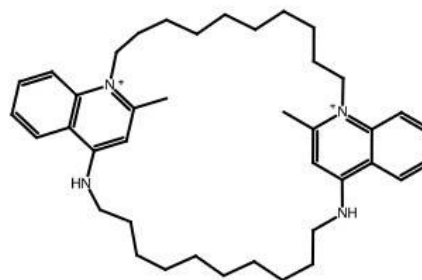
Growth Promoter



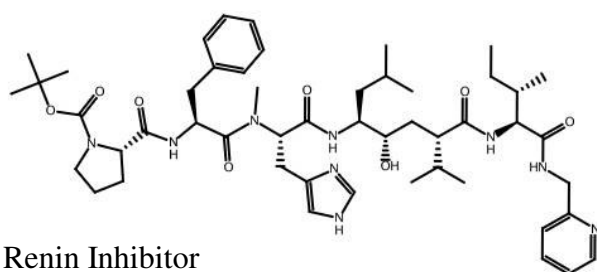
Oxytocin Peptide



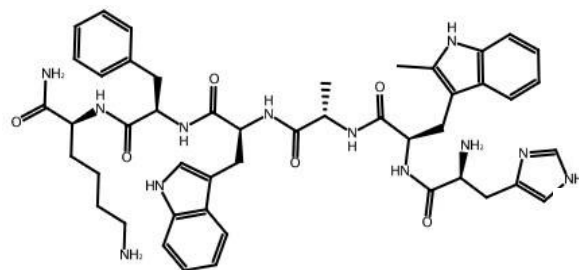
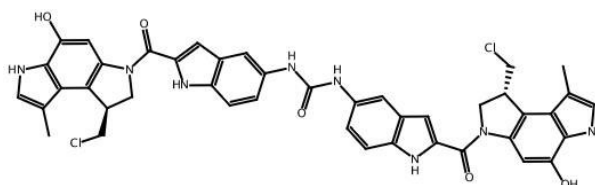
Transition State Mimic- Endothelin Inhibitor



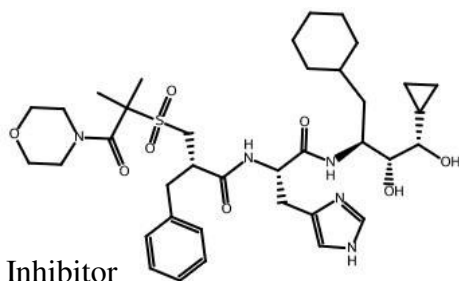
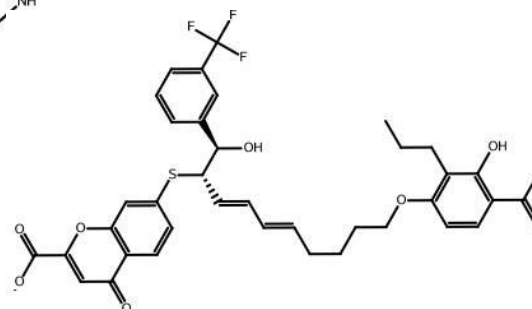
**C. 6 False Positives for the optimal cu-crystal pharmacophore model, 9/11:
3×RMSD.**



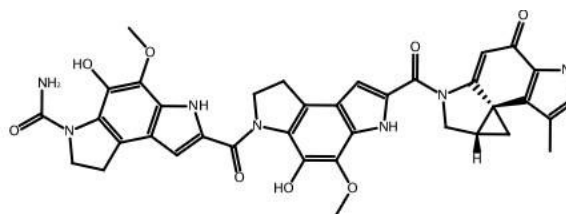
Renin Inhibitor



Growth Promoter



Renin Inhibitor



APPENDIX 4

Accounting for Multiple Protein Conformations in Ranking Ligand Databases

A4.1 DOCK code modifications.

Global variables added to: global.h, dock.c, grid.c

```
/* KLM global vbles */
extern float sphere_array_KLM[1500][8];
extern int sphere_number_KLM;
extern char sphere_name_KLM[81];
```

Code added to dock.c

```
/*KLM file for sphere */
FILE *klm_file;

/* KLM read sphere file */

printf ("Please enter sphere file (full path):\n");
gets (sphere_name_KLM);
printf ("you entered: %s\n", sphere_name_KLM);
printf ("Please enter number of spheres (max 1500):\n");
scanf ("%d", &sphere_number_KLM);
printf ("you entered: %d\n", sphere_number_KLM);

if ( (klm_file = fopen(sphere_name_KLM, "r")) != NULL )
{
    printf ("file DID open\n");
}

for (i=0; i<sphere_number_KLM; i++)
{
    fscanf (klm_file, "%d %f %f %f %f %f %f %f",
&sphere_array_KLM[i][0], &sphere_array_KLM[i][1],
&sphere_array_KLM[i][2]
```

```

], &sphere_array_KLM[i][3], &sphere_array_KLM[i][4],
&sphere_array_KLM[i][5], &sphere_array_KLM[i][6], &sphere_array_KLM[
i][7]);

}

fclose (klm_file);

```

Code added to score.c

```

/* ////////////////////////////////////////////////////////////////////

Routine to identify all receptor atoms near the ligand atom
and compute a continuous intermolecular score.

2/97 te

////////////////////////////////////////////////////////////////// */

void calc_inter_score_cont
(
  SCORE_GRID      *grid,
  void            *score,
  float           distance_cutoff,
  void            calc_inter_score
                  (SCORE_GRID *, void *, LABEL *, MOLECULE *,
                  int, int, SCORE_PART *),
  LABEL           *label,
  MOLECULE        *molecule,
  int             atom,
  SCORE_PART      *inter
)
{
  int i, j, k;
  int ilo, jlo, klo;
  int ihi, jhi, khi;
  int grid_cutoff;
  int grid_coord[3];
  int index;
  int rec_atom;
  SLINT *ptr;

  /*printf ("I'm in calc_inter_score_cont \n");
  */

  get_grid_coordinate (grid, molecule->coord[atom], grid_coord);
  grid_cutoff = (int) (distance_cutoff / grid->spacing) + 1,

  /* original code
  ilo = MAX (0, (grid_coord[0] - grid_cutoff));
  jlo = MAX (0, (grid_coord[1] - grid_cutoff));
  klo = MAX (0, (grid_coord[2] - grid_cutoff));
  ihi = MIN (grid->span[0], (grid_coord[0] + grid_cutoff + 1));
  jhi = MIN (grid->span[1], (grid_coord[1] + grid_cutoff + 1));
  */

```

```

khi = MIN (grid->span[2], (grid_coord[2] + grid_cutoff + 1));

for (i = ilo; i < ihi; i++)
  for (j = jlo; j < jhi; j++)
    for (k = klo; k < khi; k++)
    {
      index =
        grid->span[0] * grid->span[1] * k +
        grid->span[0] * j + i;

      for (ptr = grid->atom[index]; ptr; ptr = ptr->next)
      {
        rec_atom = ptr->value;
        calc_inter_score
          (
            grid,
            score,
            label,
            molecule,
            atom,
            rec_atom,
            inter
          );
      }
    }
*/

/* KLM changes */
/* change so it only goes through once */

ilo = MAX (0, (grid_coord[0] - grid_cutoff));
jlo = MAX (0, (grid_coord[1] - grid_cutoff));
klo = MAX (0, (grid_coord[2] - grid_cutoff));
ihi = MIN (grid->span[0], (grid_coord[0] + grid_cutoff + 1));
jhi = MIN (grid->span[1], (grid_coord[1] + grid_cutoff + 1));
khi = MIN (grid->span[2], (grid_coord[2] + grid_cutoff + 1));

rec_atom = 1;

calc_inter_score
(
  grid,
  score,
  label,
  molecule,
  atom,
  rec_atom,
  inter
);

/* KLM commented out this grid searching thing - above just forced it
to only call calc_inter_score once
*/
/* for (i = ilo; i < ihi; i++)
   for (j = jlo; j < jhi; j++)
     for (k = klo; k < khi; k++)

```



```

    {
        index =
            grid->span[0] * grid->span[1] * k +
            grid->span[0] * j + i;
        if (ptr = grid->atom[index])
        {
            rec_atom = ptr->value;
printf ("index is %d, rec_atom is %d, gridatom index is %d \n", index,
rec_atom, *ptr);

            calc_inter_score
            (
                grid,
                score,
                label,
                molecule,
                atom,
                rec_atom,
                inter
            );
        }
    }
*/
}

/* ////////////////////////////////////////

Routine to compute the intermolecular chemical score in a continuous
fashion given a ligand atom and a receptor atom.

2/97 te

//////////////////////////////////// */

void calc_inter_chemical_cont
(
    SCORE_GRID      *grid,
    SCORE_ENERGY    *energy,
    LABEL           *label,
    MOLECULE        *molecule,
    int             atom,
    int             rec_atom,
    SCORE_PART      *inter
)
{
/* KLM modifications */
extern int sphere_number_KLM;
extern float sphere_array_KLM[1500][8];
int i;
float xlig, ylig, zlig, xdiff, ydiff, zdiff, rad, dist;
FILE *file;
float array [397][8];
int flush;
char Caro[10] = "C.ar";
char Naro[10] = "N.ar";

```

```

char Chyd[10] = "C.3";
char Nam[10] = "N.am";
char N4[10] = "N.4";
char N3[10] = "N.3";
char O3[10] = "O.3";
char *type[10];
/**/

float vdwA, vdwB, electro;
int count = 0;

if (label->vdw.member[grid->receptor.atom[rec_atom].vdw_id].well_depth != 0.0)
{
/*   printf ("if clause value is %f \n",label->vdw.member[grid->receptor.atom[rec_atom].vdw_id].well_depth);
*/
  calc_pairwise_energy
  (
    energy,
    label,
    molecule,
    &grid->receptor,
    atom,
    rec_atom,
    &vdwA,
    &vdwB,
    &electro
  );

/* commenting out original function
inter->vdw += (vdwA - vdwB *
  label->chemical.score_table
  [molecule->atom[atom].chem_id]
  [grid->receptor.atom[rec_atom].chem_id]) * energy->scale_vdw;
*/
/* leave electrostatic part alone */

inter->electro += electro * energy->scale_electro;

/*   inter->total = inter->vdw + inter->electro;
*/

/* KLM scoring function */

xlig = molecule->coord[atom][0];
ylig = molecule->coord[atom][1];
zlig = molecule->coord[atom][2];

if (molecule->atom[atom].heavy_flag == TRUE)
{
/*printf ("starting if loop ");
printf ("xcoord is %f %f %f  ", xlig, ylig, zlig);
printf ("atomType is %s \n", molecule->atom[atom].type);
*/
}

```

```

*type = molecule->atom[atom].type;

for (i=0; i<sphere_number_KLM; i++)
{

/* printf ("sphere is %f, %f, %f\n", sphere_array_KLM[i][1],
sphere_array_KLM[i][2], sphere_array_KLM[i][3]);
printf ("rad is %f next is %d next is %d next is %f \n",
sphere_array_KLM[i][4], sphere_array_KLM[i][5], sphere_array_KLM[i][6],
sphere_array_KLM[i][7]);

printf ("sphere_array_KLM is %f sphere number is %d\n",
sphere_array_KLM[i][7], i);
*/
    xdiff = (xlig-sphere_array_KLM[i][1]);
    ydiff = (ylig-sphere_array_KLM[i][2]);
    zdiff = (zlig-sphere_array_KLM[i][3]);

/*printf ("diffs are: %f, %f, %f", xdiff, ydiff, zdiff);
*/
    dist = sqrt( (xdiff*xdiff) + (ydiff*ydiff) + (zdiff*zdiff) );

/*printf ("dist is %f\n", dist);
*/

    if (dist <= sphere_array_KLM[i][4])
    {
/*      printf ("dist is within rad %f type is %s label is %f
Caro is %s\n", dist, *type, sphere_array_KLM[i][7], Caro);

if ( strcmp(*type, Caro) == 0)
{
printf ("type and Caro agree!\n");
}
*/
        if ( sphere_array_KLM[i][7] == 5 )
        {
/*          printf ("too close to ex vol :( id is %f spheres num is
%d sphere xcoord is %f %f %f dist is %f\n", sphere_array_KLM[i][7], i,
sphere_array_KLM[i][1], sphere_array_KLM[i][2], sphere_array_KLM[i][3],
dist);
*/
            inter->vdw += sphere_array_KLM[i][5] *
sphere_array_KLM[i][6];
        }

        if ( ((strcmp(*type, Caro) == 0) || (strcmp(*type, Naro) ==
0)) && (sphere_array_KLM[i][7] == 1) )
        {
/*          printf ("aromatic chemistry agrees :) id is %f sphere
number is %d\n", sphere_array_KLM[i][7], i);
*/

```

```

        inter->vdw -= sphere_array_KLM[i][5] *
sphere_array_KLM[i][6];
/*          printf ("inter->vdw w/in loop is %f\n", inter->vdw);
*/
    }

    if ( ((strcmp(*type, Caro) == 0) || (strcmp(*type, Chyd) ==
0)) && (sphere_array_KLM[i][7] == 2) )
    {
/*          printf ("hydrophobic chemistry agrees :) id is %f sphere
number is %d\n", sphere_array_KLM[i][7], i);
*/
        inter->vdw -= sphere_array_KLM[i][5] *
sphere_array_KLM[i][6];
/*          printf ("inter->vdw w/in loop is %f\n", inter->vdw);
*/
    }

    if ( ((strcmp(*type, Nam) == 0) || (strcmp(*type, O3) == 0)
|| (strcmp(*type, N4) == 0) || (strcmp(*type, N3) == 0))
&& (sphere_array_KLM[i][7] == 3) )
    {
/*          printf ("hydrogen bond chemistry agrees :) id is %f
sphere number is %d\n", sphere_array_KLM[i][7], i);
*/
        inter->vdw -= sphere_array_KLM[i][5] *
sphere_array_KLM[i][6];
/*          printf ("inter->vdw w/in loop is %f\n", inter->vdw);
*/
    }
    }
else
{
/*printf ("dist is too big\n");
*/
    }
}
}

inter->total = inter->vdw + inter->electro;

/*printf ("inter->total score is: %f\n", inter->total);
*/

}

}

```

A4.2 Modified DOCK parameter files.**A. Chem_score**

```
label null
label hydrophobic
label aromatic
label donor
```

```
table
1
0 5
0 0 1

0 0 0 10
```

B. Chem_match

```
label null
label hydrophobic
label aromatic
label donor
label acceptor
```

```
table
1
0 1
0 0 1
0 0 0 1

0 0 0 0 1
```

C. Chem.defn

```
name hydrophobic

definition C. [ O. ] [ N. [ 2 O.2 ] [ 2 C. ] ] ( * )
definition C.ar
definition N.pl3 ( 3 C. )
definition Cl ( C. )
definition Br ( C. )
definition I ( C. )
definition C.3 [ * ]
```

```
name aromatic
```

definition C.ar

name donor

definition N. (H)

definition O. (H)

definition N.4 (H)

name acceptor

definition O. [H] [N.] (*)

definition O.3 (1 *) [N.]

definition O.co2 (C.2 (O.co2))

definition N. [H] [N.] [O.] [3 .] (*)

definition O.2 [*]

A4.3 Example INDOCK file.

```

flexible_ligand          no
orient_ligand            yes
score_ligand             yes
minimize_ligand         no
multiple_ligands        yes
parallel_jobs            no
random_seed              293847
match_receptor_sites    yes
random_search            no
automated_matching      no
maximum_orientations    5000
write_orientations      yes
rank_orientations       yes
rank_orientation_total  1
nodes_minimum           4
nodes_maximum           1000
distance_tolerance      0.25
distance_minimum        2
check_degeneracy        no
reflect_ligand          no
critical_points         no
chemical_match          yes
intermolecular_score    yes
gridded_score           no
contact_score           no
chemical_score          yes
energy_score            no
energy_cutoff_distance  999
distance_dielectric     yes
dielectric_factor       4
attractive_exponent     6
repulsive_exponent     12
atom_model              u
vdw_scale               1
electrostatic_scale     1
ligands_maximum         500000
initial_skip            0
interval_skip           0
heavy_atoms_minimum     0
heavy_atoms_maximum    5000
rank_ligands            no
ligand_atom_file        ../general_DHFR_inhibs_peoe.mol2
receptor_site_file      dhfr_c10_a125_RMSD05.sph
receptor_atom_file      receptor.mol2
vdw_definition_file     /users/kdamm/DOCK/kld_dock/parameter/vdw.defn
chemical_definition_file ../chem_all.defn
chemical_match_file      ../chem_match_all.tbl
chemical_score_file      ../chem_score.tbl
ligand_chemical_file     LOW_dhfr_c10_4-2.mol2
info_file                LOW_dhfr_c10_4-2.info

```

A4.4 Raw atomistic pharmacophore screening data.

Tables A-E: Effect of varying the number of minimum required nodes (m), inter-node distance (d), the distance tolerance (t), and cluster size cut-off (c) on virtual screening performance using the 1HHP model. Two excluded volumes were used to represent the floor of the active site with a scoring penalty of 100. The Cummings et al. data set was used consisting of 1025 compounds seeded with 5 HIV-1p inhibitors. For a given fraction of the ranked database, the number of known HIV-1p inhibitors identified is shown along with the sum of the ranks of the 5 inhibitors and the number of compounds scored by DOCK.

Table A: DOCK parameters $3m - 4d$ and $0.25t$.

c (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
15 (110)	0	1	2	2	5	1048	963	50,63,257,288,390
14 (114)	0	0	1	2	5	1163	965	91,121,273,307,371
13 (117)	0	1	1	2	5	1028	967	33,134, 246,267,348
12 (124)	0	0	1	3	5	829	970	100,112,114,236,267
11 (131)	0	1	1	3	5	861	981	38,128,188,241,271
10 (140)	0	0	1	3	5	829	982	48,141,193,212,225
9 (148)	1	1	3	5	5	744	981	17,56,100,230,341
8 (157)	1	2	3	4	5	476	988	20,44,89,118,205
7 (170)	0	1	3	4	5	705	991	47,79,95,157,327
6 (185)	1	1	2	4	5	645	994	18,70,133,133,291
5 (205)	1	2	3	3	5	689	993	10,40,82,222,335
4 (237)	0	1	3	4	5	768	998	38,54,56,147,473
3 (291)	0	1	3	4	5	800	999	25,66,99,193,417
1 (399)	0	1	2	3	4	960	1000	37,70,132,207,514

Table B: DOCK parameters $3m - 4d$ and $0.30t$.

<i>c</i> (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
15 (110)	0	0	2	2	5	1059	968	54,89,209,295,415
14 (114)	0	0	1	2	5	1192	969	95,177,290,292,338
13 (117)	0	0	1	2	5	1150	971	99,160,257,258,376
12 (124)	0	0	2	3	5	843	972	76,101,138,239,289
11 (131)	0	1	1	2	5	946	985	37,118,247,255,289
10 (140)	0	0	1	2	5	863	986	54,106,209,223,271
9 (148)	0	0	2	3	5	763	986	66,97,131,219,250
8 (157)	0	1	2	5	5	560	991	29,86,111,132,202
7 (170)	0	1	3	4	5	729	994	51,80,88,182,328
6 (185)	1	1	1	4	5	722	996	16,122,125,128,331
5 (205)	1	3	4	4	5	599	994	12,41,30,92,414
4 (237)	0	3	3	4	5	706	999	30,35,49,129,463
3 (291)	1	2	2	4	4	937	1000	7,33,148,197,552
1 (399)	0	0	1	2	4	1165	1002	52,110,218,220,564

Table C: DOCK parameters $4m - 2d$ and $0.25t$.

<i>c</i> (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
15 (110)	0	1	1	2	5	913	924	23,141,207,256,276
14 (114)	0	1	1	2	5	1069	932	33,137,215,260,424
13 (117)	0	0	1	2	5	1143	943	85,127,248,288,395
12 (124)	0	1	1	3	5	903	966	43,152,161,240,307
11 (131)	0	0	2	2	5	1100	971	56,75,244,352,373
10 (140)	0	0	1	2	5	1282	978	53,112,344,382,391
9 (148)	0	1	2	4	5	799	980	22,85,104,124,463
8 (157)	1	2	2	3	5	898	991	10,32,104,328,424
7 (170)	0	2	2	4	5	864	996	22,39,130,169,504
6 (185)	1	2	2	3	5	774	1003	5,27,171,271,300
5 (205)	2	2	3	4	5	517	1009	1,20,77,198,221
4 (237)	0	1	3	3	5	792	1011	31,54,62,330,315
3 (291)	0	2	3	3	5	884	1020	21,30,77,338,418
1 (399)	0	1	3	4	5	595	1023	47,56,84,199,209

Table D: DOCK parameters $4m - 2d$ and $0.30t$.

<i>c</i> (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
15 (110)	0	0	2	2	5	1059	968	54,86,209,295,415
14 (114)	0	0	1	2	5	1192	969	95,117,290,292,338
13 (117)	0	0	1	2	5	1044	957	86,108,259,264,327
12 (124)	0	1	2	3	5	840	981	24,70,150,262,334
11 (131)	0	1	2	2	5	983	981	48,85,212,271,367
10 (140)	0	0	1	1	5	1182	986	74,218,263,284,343
9 (148)	0	1	3	3	5	686	990	41,64,71,224,286
8 (157)	1	1	2	3	5	989	1004	18,75,142,353,401
7 (170)	1	2	3	4	5	715	1007	18,47,95,157,398
6 (185)	1	1	2	4	5	552	1008	14,55,116,148,219
5 (205)	2	2	4	5	5	314	1016	2,14,55,70,173
4 (237)	2	3	3	3	5	694	1018	9,11,26,302,346
3 (291)	0	1	3	3	5	950	1022	23,81,98,368,380
1 (399)	0	1	1	3	5	902	1024	32,118,142,217,393

Table E: DOCK parameters $4m - 3d$ and $0.30t$.

<i>c</i> (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
15 (110)	0	2	2	4	5	633	754	27,33,116,135,322
14 (114)	1	2	2	4	5	628	763	13,24,128,157,306
13 (117)	0	2	2	3	5	723	769	40,46,118,221,298
12 (124)	1	2	3	4	5	578	781	13,21,65,136,343
11 (131)	0	2	2	3	5	707	828	28,29,140,209,301
10 (140)	1	2	2	4	5	616	838	17,49,137,189,224
9 (148)	0	2	2	2	5	995	846	28,34,258,305,370
8 (157)	1	2	2	3	5	716	853	7,29,129,229,322
7 (170)	1	2	3	4	5	516	870	16,26,100,168,206
6 (185)	2	2	3	5	5	352	890	2,18,60,132,140
5 (205)	2	2	2	4	5	635	897	3,9,112,204,307
4 (237)	1	1	2	3	5	891	922	2,95,204,224,366
3 (291)	1	1	3	4	5	568	942	2,84,96,121,265
1 (399)	2	2	4	4	5	529	948	1,17,83,94,334

Tables F-I: Virtual screening performance of the four HIV-1p pharmacophore models (1HHP, 3HVP, 3PHV, and CONS) using the optimal dock parameters, 4m - 3d and 0.25t along with an excluded volume penalty of 10. The Cummings et al. data set was used consisting of 1025 compounds seeded with 5 HIV-1p inhibitors. For a given fraction of the ranked database, the number of known HIV-1p inhibitors identified is shown along with the sum of the ranks of the 5 inhibitors and the number of compounds scored by DOCK.

Table F: 1HHP, RMSD cut-off = 1.00 Å (755 spheres)

<i>c</i> (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
18 (98)	2	2	2	2	5	983	669	9,10,269,278,417
17 (103)	2	2	3	4	5	665	681	3,19,85,153,405
16 (107)	2	2	4	4	5	575	706	5,16,76,80,398
15 (110)	2	2	4	4	5	584	722	5,20,75,90,394
14 (114)	2	2	3	4	5	624	730	3,15,89,116,401
13 (117)	2	2	3	4	5	583	737	5,15,89,172,302
12 (124)	2	3	4	4	5	459	751	5,19,45,98,292
11 (131)	2	2	2	3	5	730	792	10,17,167,251,285
10 (140)	0	2	2	2	5	780	799	21,47,207,230,275
9 (148)	1	2	2	3	5	820	807	20,41,200,243,316
8 (157)	1	2	3	4	5	642	820	10,41,94,180,317
7 (170)	0	2	3	3	5	688	841	21,28,73,256,310
6 (185)	1	2	2	3	5	688	862	3,22,108,219,336
5 (205)	1	2	2	2	5	792	878	3,32,207,271,279
4 (237)	1	2	2	2	5	930	907	4,30,246,374,76
3 (291)	2	2	2	3	5	726	929	3,10,132,254,327
1 (399)	2	3	3	4	5	517	937	1,20,29,131,336

Table G: 3HVP RMSD cut-off = 0.75 Å (768 spheres)

<i>c</i> (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
20 (94)	1	2	2	2	5	1030	746	9,28,260,292,441
19 (102)	1	2	2	2	5	890	783	3,35,232,298,322
18 (107)	1	2	2	2	5	897	840	1,26,254,298,318
17 (109)	1	2	2	2	5	918	843	1,37,259,299,322
16 (116)	1	2	3	3	5	788	855	3,36,189,215,345
15 (122)	2	2	2	3	5	670	857	5,15,129,239,282
14 (128)	2	2	2	5	5	490	856	3,14,109,170,194
13 (134)	2	2	2	3	5	781	872	5,18,165,240,353
12 (137)	1	2	2	3	5	751	874	3,22,164,273,289
11 (144)	1	2	2	2	5	1009	885	19,37,283,315,355
10 (154)	1	1	3	3	5	675	902	5,76,90,236,268
9 (167)	1	1	3	3	5	711	906	13,68,98,262,270
8 (176)	1	1	3	3	5	848	909	10,76,98,294,370
7 (192)	1	2	2	2	5	884	922	16,43,236,282,307
6 (215)	1	2	2	3	5	807	924	1,51,156,274,325
5 (246)	1	1	2	3	5	753	934	2,54,157,251,289
4 (285)	1	2	4	4	5	529	941	16,29,83,85,316
3 (347)	0	0	0	4	5	774	950	115,135,138,141,245
1 (478)	0	1	3	4	5	582	958	30,75,56,168,233

Table H: 3PHV RMSD cut-off = 1.00 Å (750 spheres)

<i>c</i> (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
15 (90)	1	2	3	3	4	879	765	1,25,55,274,524
14 (97)	1	2	3	5	5	385	806	3,22,75,134,151
13 (99)	2	2	3	4	5	532	810	3,19,96,126,288
12 (106)	1	3	3	5	5	416	828	6,46,47,145,172
11 (111)	1	2	2	3	5	684	837	3,29,122,214,316
10 (117)	1	1	2	3	5	876	853	12,57,202,273,332
9 (126)	2	2	2	3	5	796	878	5,11,196,279,305
8 (132)	1	2	2	2	5	1101	885	1,47,272,333,448
7 (142)	1	2	2	2	5	960	902	3,44,273,274,366
6 (156)	1	2	2	2	5	980	918	2,46,285,309,338
5 (177)	1	1	2	3	5	1074	928	7,69,192,382,424
4 (213)	1	2	2	2	5	1140	946	8,27,275,390,440
3 (264)	1	2	2	3	5	970	959	6,22,148,370,424
1 (365)	2	2	2	2	5	868	970	2,11,210,207,338

Table I: CONS RMSD cut-off = 1.25 Å (807 spheres)

<i>c</i> (# spheres)	Top Ranked Compounds					Sum of Ranks	# Ranked Cmpds	RANK
	2%	5%	10%	20%	50%			
100 (183)	0	2	2	2	5	1261	764	26,46,303,435,451
65 (204)	0	0	2	2	4	1412	767	61,73,292,458,528
60 (212)	0	0	2	2	5	1274	828	75,88,273,405,433
50 (218)	0	2	2	3	5	706	856	26,29,114,206,331
45 (233)	0	2	2	3	5	754	910	23,27,121,213,370
40 (253)	1	1	2	2	5	1025	937	8,72,226,347,372
35 (267)	1	2	3	3	5	743	945	6,43,68,298,328
30 (289)	1	2	2	2	5	917	957	6,31,245,257,378
27 (304)	1	1	2	3	5	809	963	10,53,198,224,329
25 (315)	1	2	2	3	5	897	962	19,39,150,269,420
22 (332)	1	2	2	4	5	813	962	6,31,181,198,397
20 (342)	1	2	2	2	5	1029	968	14,47,228,335,405
18 (352)	1	2	2	2	5	984	970	10,48,205,332,389
17 (358)	1	2	2	2	5	921	971	7,33,207,292,382
16 (371)	1	2	2	2	5	976	972	9,44,281,306,336
15 (389)	1	2	2	3	5	752	974	13,27,116,269,327
14 (396)	1	1	2	3	5	896	976	4,56,194,283,359
13 (408)	2	2	2	2	5	963	978	3,17,253,319,371
12 (423)	1	1	2	3	5	903	978	2,58,169,309,365
11 (432)	1	2	2	3	5	802	978	2,30,151,300,319
10 (449)	1	2	3	3	5	644	980	1,25,102,231,285
9 (465)	1	2	2	3	5	741	981	2,44,135,272,288
8 (486)	2	2	2	2	5	870	981	7,8,240,252,363
7 (513)	2	2	3	4	5	581	981	6,20,102,132,321
6 (556)	2	2	3	4	5	713	982	4,8,101,198,402
5 (615)	1	3	3	4	5	642	981	5,40,42,143,412

APPENDIX 5

Inhibition of HIV-1p By Modulating its Conformational Behavior of the Flap Region

A5.1 MPS “eye” pharmacophore model.

Coordinates and RMSD for the pharmacophore model (radii of the elements are determined from the RMSD). Coordinates provided are relative to 1HHP monomer.

A. Isotropic pharmacophore model.

Elem. Type	Coordinates (x, y, z)			RMSD, Å
Aromatic	37.223	38.637	-5.538	1.22
Aromatic	34.737	39.215	-6.802	1.22
Aromatic	30.129	37.435	-7.590	1.05
Hydrophobic	37.650	39.514	-7.048	1.21
Hydrophobic	35.959	38.190	-9.007	1.41
Donor	37.526	35.713	-4.966	1.26
Acceptor	32.651	37.225	-4.985	1.51

B. Atomistic pharmacophore model.

Sphere Number	x	y	z	r	weight
AROMATIC					
1	34.617	39.721	-8.997	0.750	1
2	35.586	40.821	-4.707	0.750	1
3	34.727	39.960	-3.889	0.750	1
4	36.015	39.878	-9.076	0.750	1
5	28.907	36.396	-9.749	0.750	1
6	28.271	36.629	-8.703	0.750	1
7	36.305	40.830	-7.549	0.750	1

8	29.287	39.621	-8.291	0.750	1
9	33.197	37.494	-5.658	0.750	1
10	38.409	40.425	-4.640	0.750	3
11	36.068	40.062	-3.472	0.750	1
12	36.845	39.726	-2.518	0.750	1
13	37.552	40.574	-3.364	0.750	2
14	35.667	41.339	-6.331	0.750	2
15	35.589	37.851	-3.766	0.750	1
16	33.086	39.978	-6.213	0.750	8
17	38.638	39.318	-3.830	0.750	2
18	37.446	39.480	-3.203	0.750	1
19	32.090	38.284	-5.288	0.750	1
20	30.656	37.540	-5.556	0.750	1
21	30.759	37.707	-6.391	0.750	1
22	36.420	36.973	-4.491	0.750	1
23	29.221	36.767	-6.601	0.750	9
24	28.450	38.043	-7.536	0.750	6
25	29.742	39.294	-6.770	0.750	4
26	28.978	35.503	-8.385	0.750	2
27	34.631	41.328	-5.281	0.750	1
28	30.105	37.515	-6.912	0.750	1
29	35.515	37.452	-7.328	0.750	1
30	30.933	37.717	-9.421	0.750	1
31	30.142	38.927	-9.001	0.750	2
32	35.840	38.753	-8.847	0.750	1
33	36.590	38.802	-3.418	0.750	3
34	36.376	39.318	-8.123	0.750	7
35	34.429	40.245	-5.431	0.750	6
36	30.491	38.738	-7.347	0.750	8
37	29.164	38.383	-6.014	0.750	4
38	36.022	38.518	-4.254	0.750	2
39	29.970	35.927	-6.552	0.750	1
40	29.432	37.875	-8.767	0.750	2
41	39.345	37.980	-6.110	0.750	1
42	33.066	38.755	-5.135	0.750	2
43	37.970	36.463	-5.886	0.750	1
44	34.623	38.624	-4.438	0.750	1
45	36.985	39.375	-4.353	0.750	3
46	30.056	35.243	-7.784	0.750	1
47	29.827	37.641	-9.922	0.750	2
48	37.129	36.804	-6.758	0.750	2
49	38.142	38.129	-4.166	0.750	2
50	37.891	37.913	-5.630	0.750	3
51	37.173	36.947	-5.829	0.750	1
52	29.807	36.541	-9.131	0.750	1
53	36.667	37.436	-5.960	0.750	6
54	30.982	36.278	-8.024	0.750	1
55	30.872	36.445	-6.533	0.750	5
56	34.385	38.572	-7.475	0.750	2
57	38.791	40.005	-6.018	0.750	1
58	31.887	37.325	-6.476	0.750	4
59	36.812	39.564	-5.744	0.750	1
60	32.512	38.649	-6.108	0.750	4
61	35.056	40.659	-6.963	0.750	2
62	32.941	38.763	-7.497	0.750	6
63	34.402	38.535	-5.461	0.750	5
64	30.175	36.017	-7.739	0.750	8

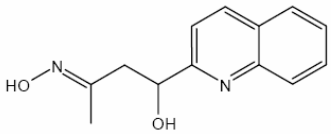
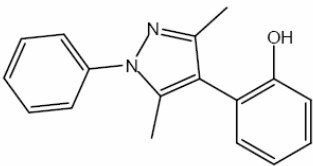
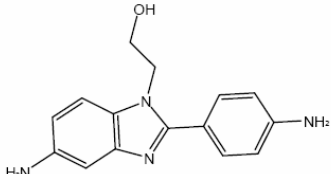
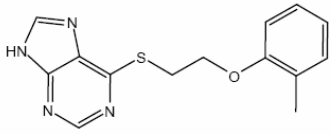
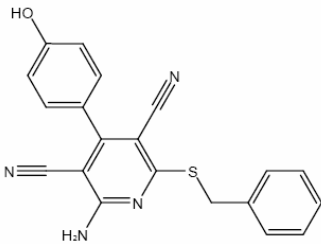
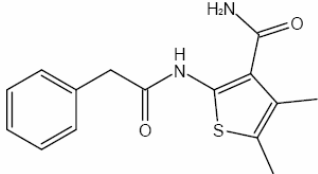
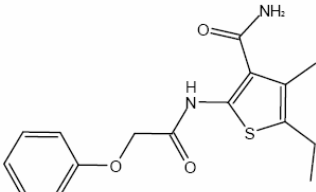
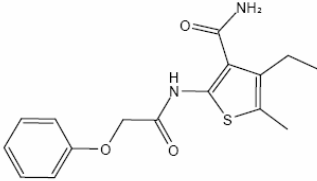
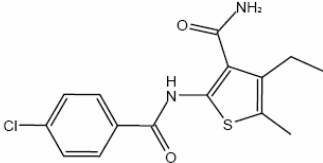
65	29.312	36.721	-8.430	0.750	2
66	37.034	37.977	-7.524	0.750	2
67	37.851	40.159	-7.799	0.750	1
68	31.092	37.992	-8.018	0.750	9
69	31.993	38.263	-7.215	0.750	2
70	33.563	37.795	-6.622	0.750	1
71	37.442	40.546	-6.507	0.750	1
72	35.648	38.008	-6.064	0.750	7
73	35.251	37.505	-6.499	0.750	3
74	34.098	37.922	-7.139	0.750	7
75	36.021	37.764	-5.081	0.750	2
76	37.234	37.643	-4.823	0.750	6
77	37.627	39.529	-5.744	0.750	3
78	34.845	38.828	-7.951	0.750	4
79	34.669	39.942	-7.116	0.750	1
80	38.177	39.180	-4.843	0.750	2
81	37.342	40.055	-7.429	0.750	1
82	33.742	38.880	-7.104	0.750	1
83	34.735	39.293	-6.000	0.750	2
84	38.988	38.954	-4.926	0.750	2
85	38.203	39.416	-6.268	0.750	1
86	38.635	38.043	-5.352	0.750	5
87	38.176	37.262	-6.051	0.750	3
88	37.598	38.418	-4.628	0.750	2
89	30.712	36.906	-8.483	0.750	7
90	30.399	37.482	-8.837	0.750	3
91	36.455	38.515	-5.850	0.750	5
92	38.718	38.988	-6.366	0.750	8
93	37.716	38.715	-6.778	0.750	3
94	35.482	39.435	-6.201	0.750	2
95	37.568	39.706	-6.895	0.750	7
96	34.025	39.257	-7.805	0.750	1
97	36.867	39.032	-6.956	0.750	5
98	36.115	38.624	-6.825	0.750	1
99	35.225	39.716	-6.752	0.750	2
100	34.565	39.320	-6.865	0.750	1
101	35.164	38.912	-6.569	0.750	5
102	34.130	39.599	-7.293	0.750	5
103	35.160	38.569	-7.187	0.750	2
104	36.467	40.154	-5.582	0.750	1
105	36.069	39.961	-6.555	0.750	9
106	36.345	39.177	-6.527	0.750	3

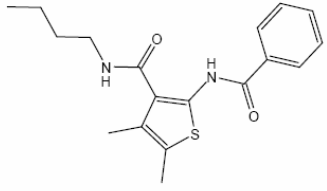
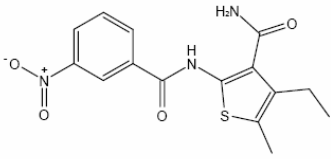
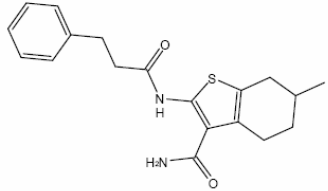
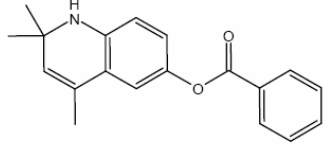
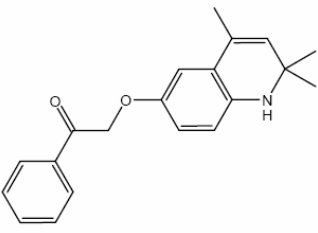
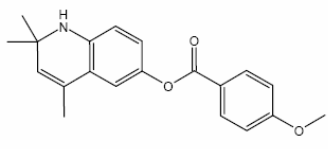
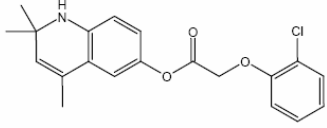
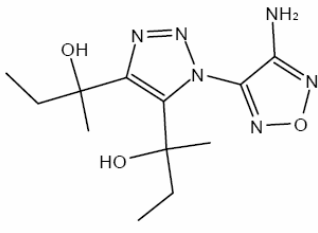
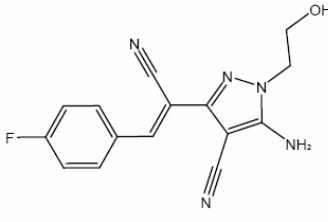
HYDROPHOBIC

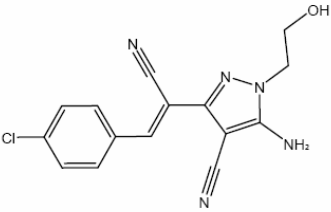
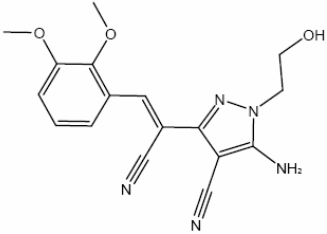
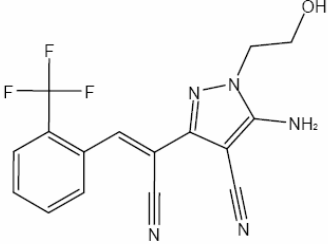
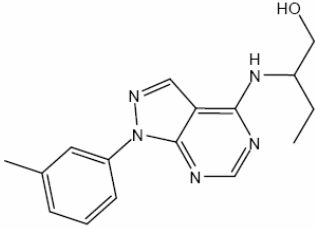
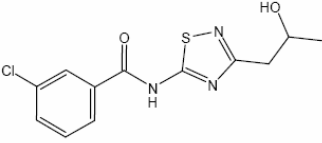
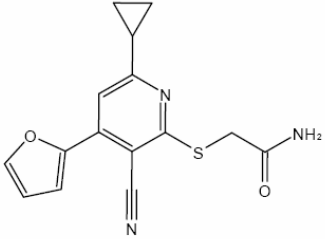
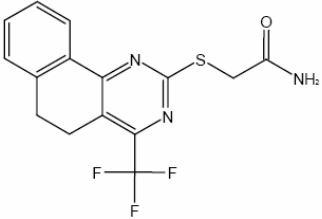
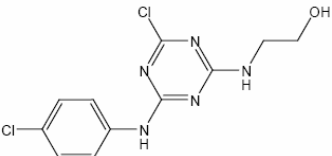
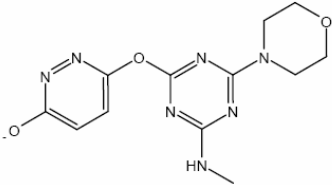
107	34.945	38.251	-10.549	0.750	1
108	36.254	40.103	-9.109	0.750	1
109	34.798	39.729	-8.809	0.750	1
110	36.295	40.436	-7.560	0.750	1
111	34.719	37.631	-9.508	0.750	1
112	35.986	38.076	-11.380	0.750	1
113	34.851	38.652	-9.284	0.750	1
114	37.378	38.535	-6.062	0.750	1
115	36.103	39.173	-10.316	0.750	1
116	36.997	37.871	-6.734	0.750	1
117	36.159	37.156	-10.296	0.750	6
118	36.295	37.869	-9.048	0.750	7
119	39.385	39.349	-6.428	0.750	1

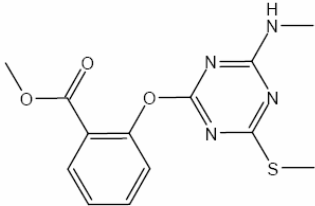
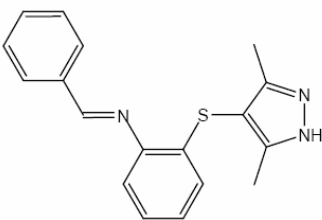
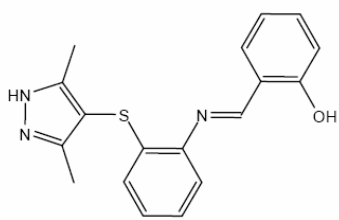
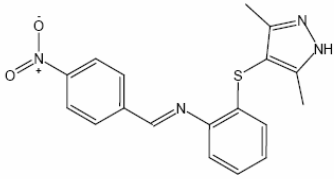
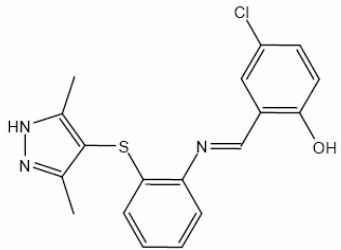
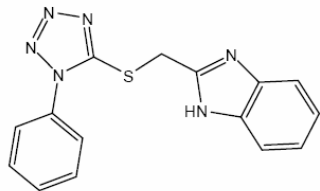
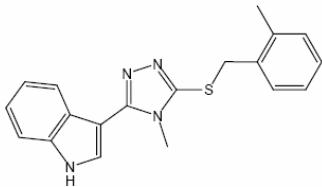
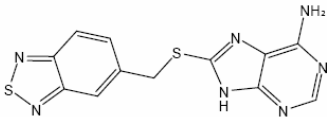
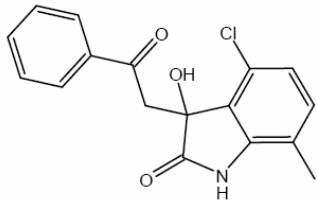
120	37.943	39.936	-6.939	0.750	8
121	37.297	41.269	-7.276	0.750	3
122	34.999	38.496	-7.894	0.750	6
123	38.468	38.631	-6.733	0.750	3
124	37.165	38.644	-7.859	0.750	8
125	36.064	38.118	-7.322	0.750	2
DONOR					
126	37.679	34.664	-4.203	0.750	1
127	38.167	36.500	-5.268	0.750	8
128	36.154	35.397	-4.390	0.750	1
ACCEPTOR					
129	31.138	34.840	-5.183	0.750	1
130	33.272	36.039	-4.427	0.750	1
131	32.817	35.236	-5.980	0.750	2
132	32.828	38.203	-2.985	0.750	3
133	32.476	37.262	-3.603	0.750	1
134	31.840	37.594	-4.579	0.750	1
135	30.757	36.944	-6.538	0.750	1
136	32.331	38.696	-4.593	0.750	1
137	31.932	37.441	-5.936	0.750	1
138	33.550	38.456	-4.635	0.750	3
139	33.133	37.272	-5.994	0.750	9
140	31.474	36.435	-5.441	0.750	2

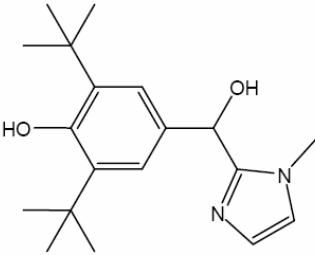
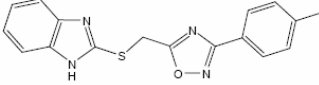
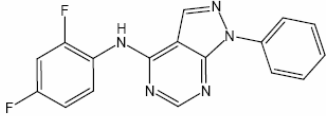
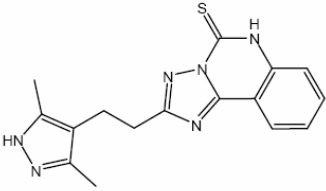
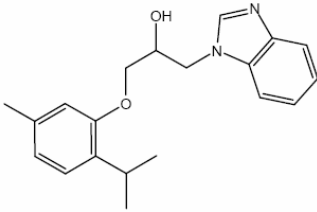
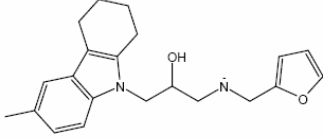
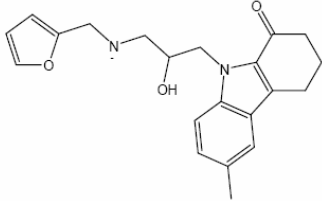
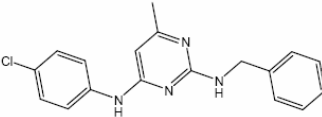
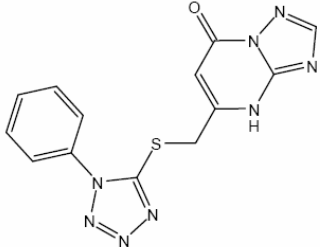
**A5.2 93 Identified Compounds from MPS pharmacophore screen grouped by 60%
chemical similarity and overlap.**

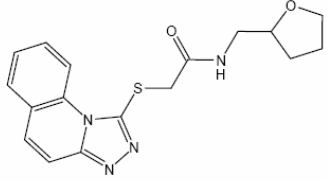
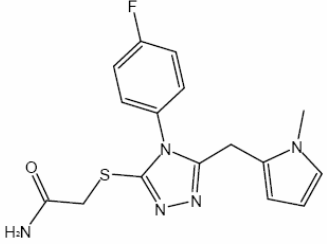
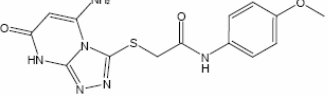
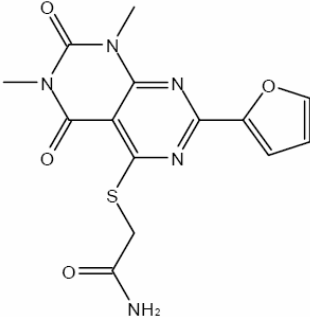
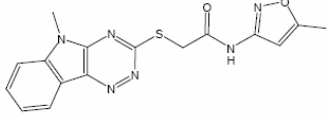
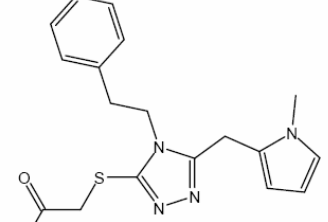
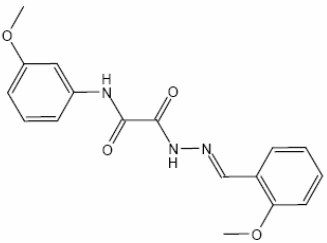
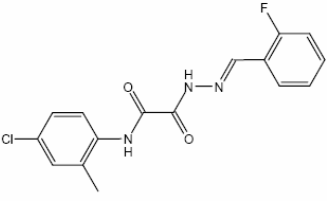
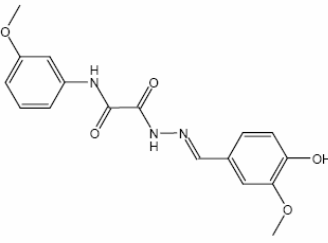
<p>1</p> <p>5562086</p>  <p>Weight: 230.267</p>	<p>2</p> <p>1499-0598</p>  <p>Weight: 264.328</p>	<p>3</p> <p>3769-2047</p>  <p>Weight: 268.32</p>
<p>4</p> <p>1465-0022</p>  <p>Weight: 286.359</p>	<p>5</p> <p>3448-5797</p>  <p>Weight: 358.425</p>	<p>6</p> <p>5524366</p>  <p>Weight: 288.371</p>
<p>7</p> <p>5539991</p>  <p>Weight: 318.397</p>	<p>8</p> <p>5555993</p>  <p>Weight: 318.397</p>	<p>9</p> <p>5546084</p>  <p>Weight: 322.816</p>

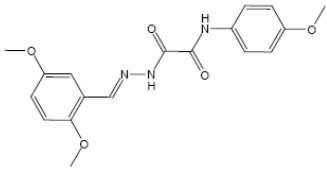
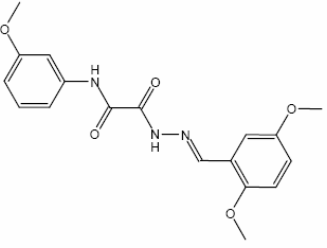
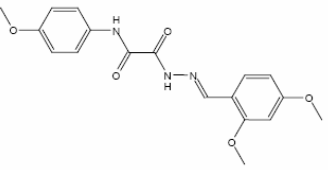
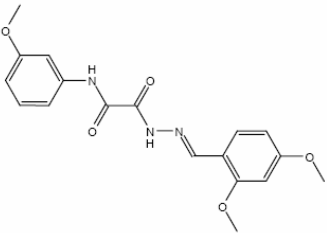
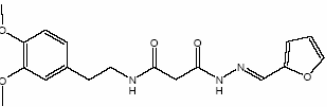
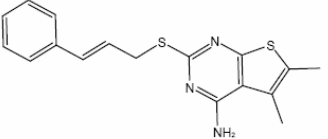
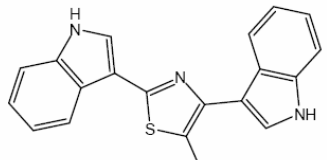
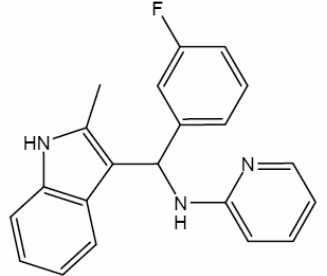
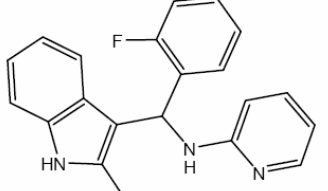
<p>10</p> <p>5373736</p>  <p>Weight: 330.452</p>	<p>11</p> <p>5552988</p>  <p>Weight: 333.368</p>	<p>12</p> <p>5574999</p>  <p>Weight: 342.463</p>
<p>13</p> <p>5493403</p>  <p>Weight: 293.366</p>	<p>14</p> <p>5483428</p>  <p>Weight: 307.393</p>	<p>15</p> <p>5303843</p>  <p>Weight: 323.392</p>
<p>16</p> <p>5469524</p>  <p>Weight: 357.837</p>	<p>17</p> <p>1307-0007</p>  <p>Weight: 296.331</p>	<p>18</p> <p>5561288</p>  <p>Weight: 297.293</p>

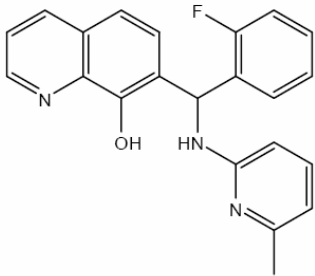
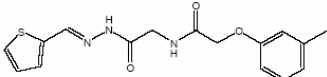
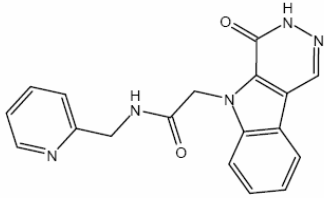
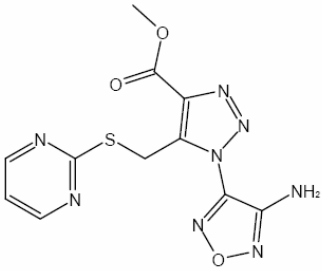
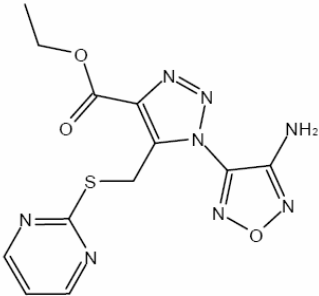
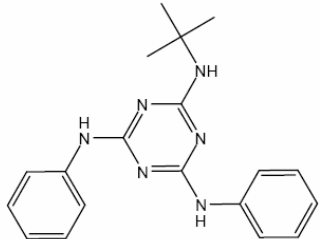
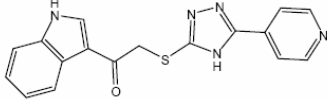
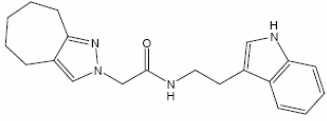
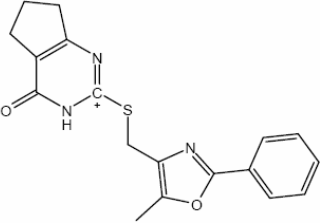
<p>19</p> <p>5524374</p>  <p>Weight: 313.748</p>	<p>20</p> <p>5626286</p>  <p>Weight: 339.355</p>	<p>21</p> <p>5522294</p>  <p>Weight: 347.3</p>
<p>22</p> <p>K402-0900</p>  <p>Weight: 297.362</p>	<p>23</p> <p>C200-1280</p>  <p>Weight: 297.766</p>	<p>24</p> <p>3343-2780</p>  <p>Weight: 299.354</p>
<p>25</p> <p>6466-0177</p>  <p>Weight: 339.341</p>	<p>26</p> <p>K890-0002</p>  <p>Weight: 300.149</p>	<p>27</p> <p>6989-0080</p>  <p>Weight: 304.29</p>

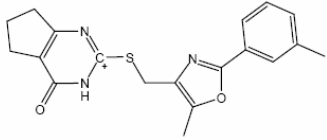
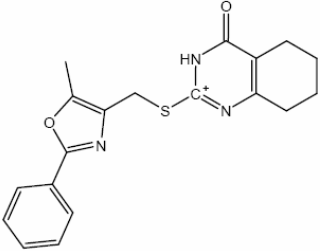
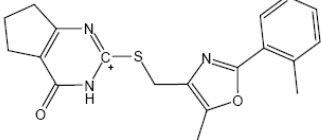
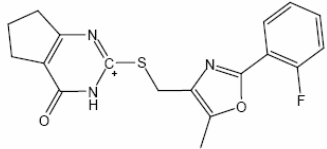
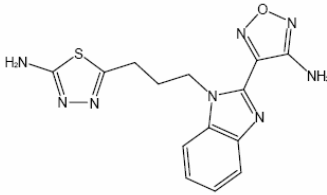
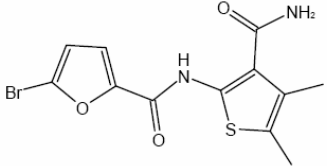
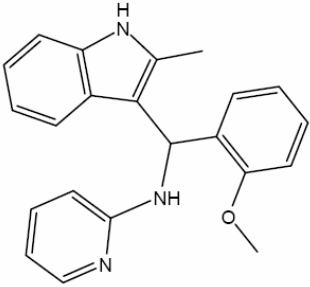
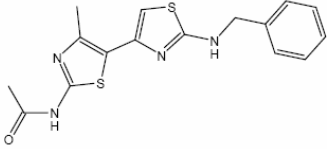
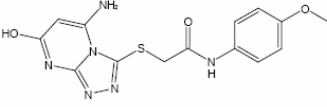
<p>28</p> <p>3379-0757</p>  <p>Weight: 306.346</p>	<p>29</p> <p>5483692</p>  <p>Weight: 307.421</p>	<p>30</p> <p>5493267</p>  <p>Weight: 323.42</p>
<p>31</p> <p>5486930</p>  <p>Weight: 352.418</p>	<p>32</p> <p>5485189</p>  <p>Weight: 357.865</p>	<p>33</p> <p>5491484</p>  <p>Weight: 308.369</p>
<p>34</p> <p>C592-0086</p>  <p>Weight: 334.447</p>	<p>35</p> <p>1323-0109</p>  <p>Weight: 315.385</p>	<p>36</p> <p>5056-0111</p>  <p>Weight: 315.756</p>

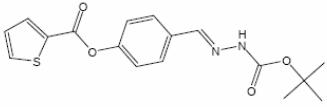
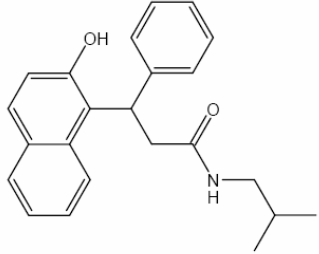
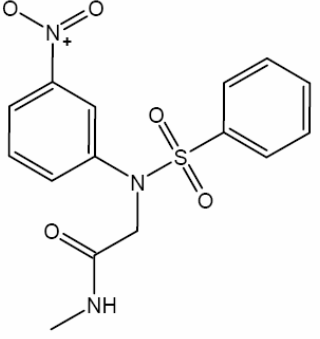
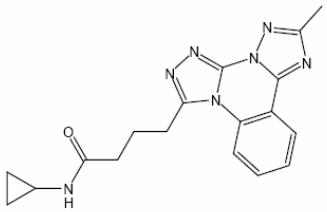
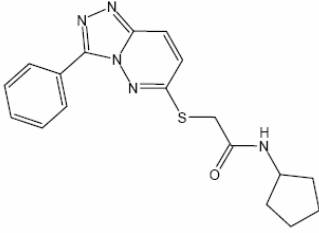
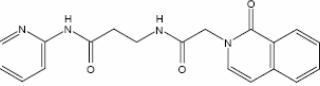
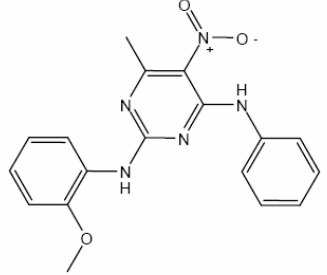
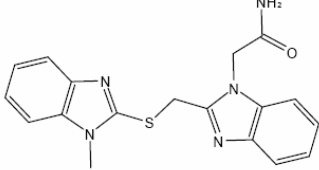
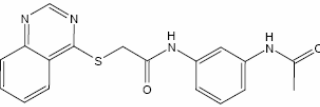
<p>37</p> <p>5529870</p>  <p>Weight: 316.445</p>	<p>38</p> <p>5371-4626</p>  <p>Weight: 322.392</p>	<p>39</p> <p>K402-0245</p>  <p>Weight: 323.306</p>
<p>40</p> <p>K280-0545</p>  <p>Weight: 324.412</p>	<p>41</p> <p>3910-0351</p>  <p>Weight: 324.424</p>	<p>42</p> <p>6658-0012</p>  <p>Weight: 337.443</p>
<p>43</p> <p>5926-0037</p>  <p>Weight: 351.426</p>	<p>44</p> <p>4466-2067</p>  <p>Weight: 324.815</p>	<p>45</p> <p>6456-0643</p>  <p>Weight: 326.344</p>

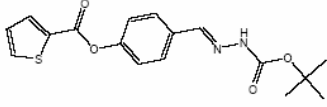
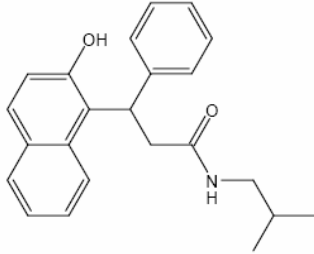
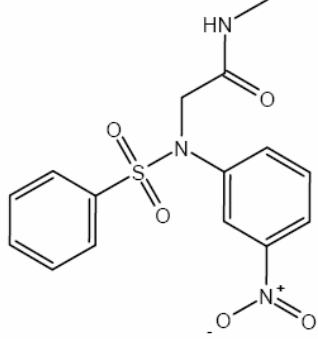
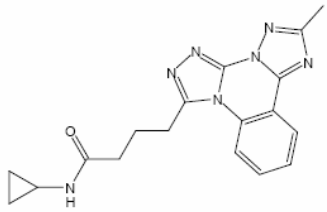
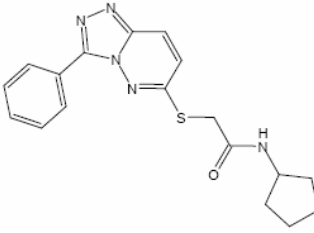
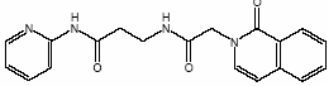
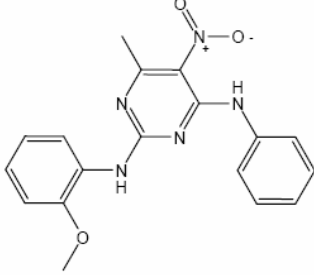
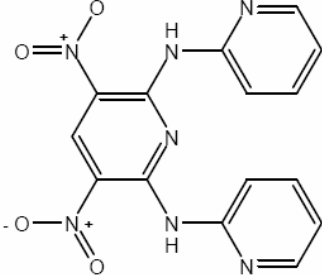
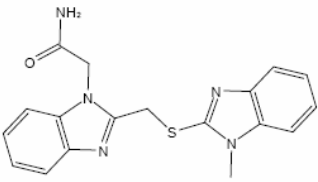
<p>46</p> <p>5867-3931</p>  <p>Weight: 342.423</p>	<p>47</p> <p>C677-0232</p>  <p>Weight: 345.402</p>	<p>48</p> <p>6737-0622</p>  <p>Weight: 346.371</p>
<p>49</p> <p>C610-0367</p>  <p>Weight: 347.355</p>	<p>50</p> <p>6218637</p>  <p>Weight: 354.394</p>	<p>51</p> <p>C677-0010</p>  <p>Weight: 355.466</p>
<p>52</p> <p>5522450</p>  <p>Weight: 327.34</p>	<p>53</p> <p>5526815</p>  <p>Weight: 333.75</p>	<p>54</p> <p>5545228</p>  <p>Weight: 343.339</p>

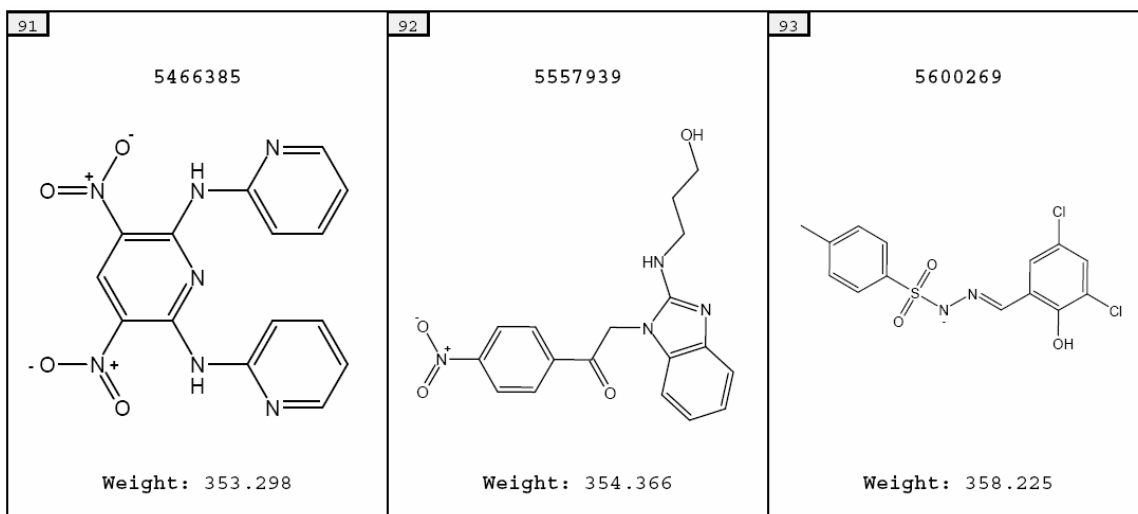
<p>55</p> <p>5559290</p>  <p>Weight: 357.366</p>	<p>56</p> <p>5523166</p>  <p>Weight: 357.366</p>	<p>57</p> <p>5556943</p>  <p>Weight: 357.366</p>
<p>58</p> <p>5539286</p>  <p>Weight: 357.366</p>	<p>59</p> <p>5655180</p>  <p>Weight: 359.382</p>	<p>60</p> <p>4236-0436</p>  <p>Weight: 327.476</p>
<p>61</p> <p>6597798</p>  <p>Weight: 329.427</p>	<p>62</p> <p>5228-4175</p>  <p>Weight: 331.394</p>	<p>63</p> <p>5228-4184</p>  <p>Weight: 331.394</p>

<p>64</p> <p>4896-4230</p>  <p>Weight: 359.404</p>	<p>65</p> <p>5468265</p>  <p>Weight: 331.396</p>	<p>66</p> <p>C884-0070</p>  <p>Weight: 333.351</p>
<p>67</p> <p>6703-1500</p>  <p>Weight: 334.32</p>	<p>68</p> <p>6703-1782</p>  <p>Weight: 348.347</p>	<p>69</p> <p>5469915</p>  <p>Weight: 334.427</p>
<p>70</p> <p>6585837</p>  <p>Weight: 335.391</p>	<p>71</p> <p>C881-1035</p>  <p>Weight: 336.439</p>	<p>72</p> <p>C700-1363</p>  <p>Weight: 339.419</p>

<p>73</p> <p>C700-1362</p>  <p>Weight: 353.446</p>	<p>74</p> <p>C700-1271</p>  <p>Weight: 353.446</p>	<p>75</p> <p>C700-1355</p>  <p>Weight: 353.446</p>
<p>76</p> <p>C700-1364</p>  <p>Weight: 357.409</p>	<p>77</p> <p>3448-7570</p>  <p>Weight: 342.387</p>	<p>78</p> <p>5542560</p>  <p>Weight: 343.201</p>
<p>79</p> <p>5228-4182</p>  <p>Weight: 343.43</p>	<p>80</p> <p>6504928</p>  <p>Weight: 344.463</p>	<p>81</p> <p>6143-0142</p>  <p>Weight: 346.371</p>

<p>82</p> <p>6261334</p>  <p>Weight: 346.407</p>	<p>83</p> <p>5489937</p>  <p>Weight: 347.458</p>	<p>84</p> <p>6197605</p>  <p>Weight: 349.367</p>
<p>85</p> <p>C684-0061</p>  <p>Weight: 349.398</p>	<p>86</p> <p>C742-0059</p>  <p>Weight: 353.45</p>	<p>87</p> <p>C066-1486</p>  <p>Weight: 350.378</p>
<p>88</p> <p>5307415</p>  <p>Weight: 351.366</p>	<p>89</p> <p>3448-9010</p>  <p>Weight: 351.434</p>	<p>90</p> <p>C725-0034</p>  <p>Weight: 352.418</p>

<p>6261334</p>  <p>Weight: 346.407</p>	<p>5489937</p>  <p>Weight: 347.458</p>	<p>6197605</p>  <p>Weight: 349.367</p>
<p>C684-0061</p>  <p>Weight: 349.398</p>	<p>C742-0059</p>  <p>Weight: 353.45</p>	<p>C066-1486</p>  <p>Weight: 350.378</p>
<p>5307415</p>  <p>Weight: 351.366</p>	<p>5466385</p>  <p>Weight: 353.298</p>	<p>3448-9010</p>  <p>Weight: 351.434</p>



A5.3 Analysis of implicit-solvation Langevin Dynamics simulations.

Distance calculated between the flap tip residue I50 C α to the catalytic residue D25 C α throughout the MD trajectory. This metric quantifies the flap movement in the vertical direction. Compound 1 is bound to Monomer A (i.e., Flap A).

Distance (D25-I50 and D25'-I50')

