FURTHERING MULTISCALE MEMBRANE PROTEIN PREDICTION AND DESIGN APPROACHES

By

Brittany Lasher

A thesis submitted to the Johns Hopkins University in conformity with the requirements for the degree of Master of Science in Engineering

Baltimore, Maryland

May 2019

Abstract

Over the past decade, there have been many advances in developing computational tools toward sub-Angstrom biomolecular structure prediction accuracy. A remaining challenge is capturing helix hinge dynamics within membrane proteins. Modeling these dynamics is challenging because their location and qualities are determined by a fine balance of intermolecular interactions with neighboring helices and the surrounding lipid bilayer. In this work, I aimed to enable sampling of kinked helices through the use of a kinked peptide fragment library. By first developing a classification method for kinked helices, I generated a kinked helix library. Exploration of this library revealed diverse helical representations which depended on the kink degree, resulting from the number of backbone hydrogen bonds present. I expect this library to allow for insertion of kinked protein fragments from the Protein Databank into membrane proteins. This library has the potential to significantly improve the accuracy of membrane protein structure prediction and enable *de novo* design of membrane proteins that contain flexible hinges.

Protein-protein interface prediction and design methods provide insight into protein function and guide protein engineering. For membrane proteins, this task is especially difficult because they reside in a heterogeneous lipid bilayer. In this work, I develop a multiscale modeling approach to dock membrane-anchored proteins. CYP76AD1 and NCP1 are redox enzymes which interact to produce potent small molecules. This system is challenging to model due to its many complexities: (i) membrane-anchored proteins, (ii) 600-700 residue proteins, and (iii) small molecules. I used a combination of molecular dynamics, global docking and local docking to predict the protein-protein interface region. Through experimental validation, I determined an important residue involved within the interface. Furthermore, I applied the change in binding energy calculations to guide structural predictions. This multiscale approach has the potential to predict interface regions between large membrane-anchored proteins which have posed a challenge in the past.

Advisor: Jeffrey Gray

Readers: Jeffrey Gray, Marc Ostermeier

Acknowledgments

I want to express my gratitude to professor Jeffrey Gray and my mentor Rebecca Alford for all their guidance and support through obtaining my Master's Degree. Thank you to the Dueber lab for their collaboration on the CYP76AD1-NCP1 docking project. I also want to thank all the Gray Lab team members who have always helped me with understanding concepts, writing or editing. Last, I want to thank my family and friends who have been my support system through the last two years. Thank you Liam and Kathy for constantly listening to me practice my presentations, and edit my writing. This has been a valuable experience where I have developed many new and important skills.

Table of contents

Abstract .... ***ii***

Acknowledgments .... ***iv***

Table of contents .... ***v***

Figure and table list .... ***vii***

Chapter 1: Introduction .... ***viii***

1.1 Motivation **1**

1.2 Macromolecular modeling approaches **1**

1.3 Significance of this work **2**

Chapter 2: A multiscale modeling approach for the prediction of membrane anchored protein-protein interface regions ***4***

2.1 Background **4**

2.1.1 Recent work resulted in an engineered S. cerevisiae that can generate backbone of benzylisoquinoline alkaloids: a molecule with potent pharmacological properties 4

2.1.2 Computational protein engineering is a potential route to improve catalysis and subsequently to increase yield 6

2.1.3 The complexity of the CYP76AD1 system poses a challenge to understand and engineer 8

2.1.4 A new approach for the prediction of membrane protein-protein interfaces. 10

2.2 Methods **11**

2.2.2 Protein structure generation and orientation 12

2.2.3 Molecular dynamics simulations 13

2.2.4 Prediction of the protein-protein interface 14

2.2.5 Experimental validation and structure prediction guidance 15

2.3 Results **16**

2.3.1 Protein structure generation and orientation within the membrane 16

2.3.2 Molecular dynamics simulations 18

2.3.3 Prediction of the protein-protein interface 21

2.3.4 Experimental validation and structure prediction guidance 24

2.4 Discussion **29**

2.4.1 Protein structure generation and orientation within the membrane 29

2.4.2 Molecular dynamics simulations 30

2.4.3 Prediction of the protein-protein interface 31

2.4.4 Experimental validation and structure prediction guidance 32

Chapter 3: Dynamic helical hinges in membrane proteins ***34***

3.1 Background **34**

3.1.1 Helical kinks are important for membrane protein function and are responsible for conformational changes 34
3.1.2 The origin of helix kinks 34
3.1.3 Classification Methods available to identify helical kinks in structures 35
3.1.4 Refinement methods are not proven to sample kinked helices in membrane proteins 37

3.2 Methods part 1 38
3.2.1 Testing Rosetta refinement on systems with helical hinge dynamics 38
3.2.2 Testing Rosetta score functions for biasing kinked or straight helical conformations 42
3.2.3 Manual straightening of kinked helices 42

3.3 Results Part 1 44
3.3.1 Need for sampling between kinked and non-kinked helical conformational forms in membrane proteins 44
3.3.2 Do score functions favor kinked or straight helices? 48

3.4 Discussion Part 1 53
3.4.1 Need for sampling between kinked and non-kinked helical conformational forms in membrane proteins 53
3.4.2 Do score functions favor kinked or straight helices? 54

3.5 Methods Part 2 55
3.5.1 Development of a classification method for kinked helices 55
3.5.2 Generation of a kinked helical fragment library of different lengths 56

3.6 Results Part 2 58
3.6.1 A new classification method for kinked helices 58
3.6.2 Examination of a new kinked helix fragment library 58

3.7 Discussion Part 2 61
3.7.1 A novel classification method for kinked helices 61
3.7.2 A fragment library of kinked helices 61
3.7.3 Future steps 61

Chapter 4: Conclusion 62

Appendix 64
MARCC script 64
Relax refinement script 64
Straighten kink script 65
Classification script 65
Kinked fragment collection script 69

References 79

Curriculum Vitae 85

Table list

Table 1: Structural model generated for CYP76AD1 (blue), NCP1 (tan), the percent identity for each model and template PDB ID(s) for NCP1 and CYP76AD1. .... *17*

Table 2: Mutations suggested for each predicted complex model. Residues within NCP1 are shown in red and residues within CYP76AD1 are shown in blue. For each mutation site, there is two proposed mutations, a conservative choice and a less conservative choice. *25*

Figure list

Figure 1: Three benzylisoquinoline examples commonly found in nature............4
Figure 2: Experimental method for evaluating mutant oxidase catalytic activity....6
Figure 3: Octopus topology prediction results for CYP76AD1 and NCP1. Residues on the outside of the membrane are labeled in blue, residues on the inside of the membrane are labeled in black and residues within the membrane are labeled in red. 18
Figure 4: (A) CYP76AD1 (purple) and (B) NCP1 (tan), with tracked residues represented as spheres. The colors correspond to the residue being tracked. Residue 10 in blue, residue 21 in brown, residue 118 in green, residue 235 in purple, residue 364 in orange, and residue 471 in red......19
Figure 5: Distance measurement from specified residues to the lipid bilayer center of mass over the last 20 nanoseconds of each protein's trajectory. The colors correspond to the residue being tracked. Residue 10 in blue, residue 21 in brown, residue 118 in green, residue 235 in purple, residue 364 in orange, and residue 471 in red...20
Figure 6: Backbone RMSD of CYP76AD1 and NCP1, relative to the starting conformation, over last 20 nanoseconds of the trajectory. 21
Figure 7: Panels A-B show the top 4 complex model predictions through ClusPro. In these panels, NCP1 is aligned for each complex model and shown in tan. CYP76AD1 is shown in blue scaled colors, depending on the complex model. Panel B shows the heme cofactor colored in blue scale within its complex model and represented in spheres...22
Figure 8: Panel A shows the top 5 residue predictions of InterEVDock2 for CYP76AD1, represented in spheres and colored in cyan. Panel B shows the top 5 residue predictions of INterEVDock2 for NCP1, represented in spheres and colored in cyan. Panel C shows the involvement these coevolutionary interface residue predictions for NCP1, within the ClusPro globally docked top 4 models. The top 4 globally docked complexes are in panel C, with CYP76AD1 shown in different shades of blue, depending on the complex model and NCP1 aligned for each complex model and shown in tan. 23
Figure 9: Percent activity, compared to the wildtype, for each mutant oxidase variant obtained by John Dueber's Lab. 26
Figure 10: Flex ΔΔG calculated results representing the change in binding energy due to each mutation. Panels A-D illustrate results for the top models, respectively 1-4. 27
Figure 11: Plots comparing the change in calculated binding energy due to each mutation and the measured catalytic activity of each CYP76AD1 Variant. The top 4 complex models 1-4 are shown in panels A-D, respectively. ...28
Figure 12: The structural conformation of both the kinked and straight conformation of human adiponectin receptor 1 is shown in panel A, with the helical portion highlighted in purple, the straight protein conformation

in teal and the kinked protein conformation in pink. Panel B illustrates the closed and open conformational structures from a top down view, with zinc shown in blue. Panel C shows the alignment of both conformational variants with the kinked helix shown in pink and the straight helix shown in teal. 39

Figure 13: The structural conformation of both the kinked and straight conformation of KcsA potassium channel is shown in panel A, with the helical portion highlighted in gray, the straight protein conformation in teal and the kinked protein conformation in pink. Panel B illustrates the closed and open conformational structures from a top down view, with ions shown in gray. Panel C shows the alignment of both conformational variants for a single subunit, with the kinked helix shown in teal and the straight helix shown in pink. 40

Figure 14: : The structural conformation of both the kinked and straight conformation of human platelet activating factor receptor is shown in panel A, with the straight helical portion highlighted in green and the straight helical portion shown in purple. Panel B illustrates the closed and open conformational structures from a top down view, with supporting cofactors shown in sphere representation. Panel C shows the alignment of both conformational variants, with the kinked helix shown in purple and the straight helix shown in green.. 41

Figure 15: Kinked and straightened helical conformations of four test proteins. (A) 5tud, a GPCR protein, (B) 4h33 a voltage gated potassium channel, (C) 2irv, a rhomboid protease protein, and (D) 5h35, a TRIC trimer cation channel.. 43

Figure 16: Panels A-C show plots of the density of each ensemble vs. RMSD to each native state. The line shown in red is the density of the non-kinked ensemble vs. the RMSD to the non-kinked native. The line shown in blue is the density of the non-kinked ensemble vs. the RMSD to the kinked native. The line shown in green is the density of the kinked ensemble vs. the RMSD to the kinked native. The line shown in purple is the density of the kinked ensemble vs. the RMSD to the non-kinked native. 45

Figure 17: Panels A-C illustrate the plot of the RMSD of each ensemble vs. RMSD to the kink native for each protein. For each protein, the kinked ensemble is shown in blue and the non-kinked ensemble is shown in gray. 46

Figure 18: Panels A-C illustrate the plot of the total energy vs. RMSD to the kink native. For each protein, the kinked ensemble is shown in blue and the non-kinked ensemble is shown in gray.. 47

Figure 19: Plots for the MP07 score function for the four hydrogen bonding terms and the total-score within Rosetta. The pink line shows the kinked ensemble and the teal line shows the straight ensemble for each protein in the testcase set. 49

Figure 20: Plots for the MP12 score function for the four hydrogen bonding terms and the total score within Rosetta. The pink line shows the kinked ensemble and the teal line shows the straight ensemble for each protein in the testcase set. 51
Figure 21: Plots for the Ref2015 score function for the four hydrogen bonding terms and total score within Rosetta. The pink line shows the kinked ensemble and the teal line shows the straight ensemble for each protein in the testcase set. 53
Figure 22: Steps for the helical classification method. Step 1 locate of the hinge-point, step2 use PCA to gain two vectors representing the helix, and step 3 use of the two vectors to determine the helix kink angle. 56
Figure 23: The steps involved in adding kinked fragments to the library. Step 1, pull a fragment of 24 residues from the full PDB library. Step 2, eliminate ends that are not helical secondary structure. Step 3, count secondary structure and allow only two non-helical residues. Step 4-5, utilize classification method to determine the angle of the kinked helix. Step 6, remove edges of the segment to reach the desired length.......................57
Figure 24: Number of hydrogen bonds changes as the angle of helical kinks increases from 20 to 160°. The total number of hydrogen backbone bonds capable of being satisfied is 10 and the minimum number is 0. 59
Figure 25: Types of helical kinks found within the kink fragment library.............60
Figure 26: Density of angles for the kinked fragment library. The percent of possible backbone hydrogen bonds satisfied is shown in purple and the general shape of helices at three angles across the range of angles is shown in teal. 60

Chapter 1: Introduction

1.1 Motivation

Membrane proteins are critical and play an important role in functions of transduction ion regulation and signaling [1]. Protein structure prediction and design can provide insight into the structural basis of protein function. This is especially important for membrane proteins, as they comprise 30% of all proteins [2]. This task is difficult for membrane proteins because they reside in a heterogeneous lipid bilayer. The bilayer results in challenges in experimental characterization [3], and parametrization of score terms computationally. Over the past decade, there have been advances in developing computational tools toward to sub-Angstrom prediction accuracy of membrane proteins [4], but challenges still remain. Two challenges examined in this study, are that of (i) hinge dynamics within transmembrane helices, and (ii) protein-protein interface prediction.

1.2 Macromolecular modeling approaches

Two popular approaches for macromolecular modeling are molecular dynamics (MD) [5,6] and Monte Carlo sampling (MC) [7]. Molecular dynamics simulations are driven by Newton's equations of motion which allow atoms to be tracked through time. MD simulations can provide insight into protein systems which lack experimental results and help guide future discovery through experimental validation. These simulations provide success when examining dynamics of small molecule-protein interactions [8], surface-protein interactions [9] and peptide/protein equilibration [10]. However, MD is computationally heavy, as calculations for each atom within a system can be slow, limiting the amount of time that can be simulated. With most computational resources, simulations must be at small timescales of nanoseconds to microseconds and larger systems may require even shorter simulation times. Because MD is only feasible at short timescales, it is often incapable of capturing binding dynamics or protein conformational changes. Consequently, Monte Carlo methods [11] have been used to overcome some of the challenges resulting with MD.

Monte Carlo is a method used to sample protein conformational states. Within this method of sampling, random moves are applied and accepted or rejected based on the Metropolis criterion [12]. Rosetta is a Monte Carlo (MC) based protein modeling software suite designed for the prediction and design of proteins. This method is used to sample the energy landscape, finding low-energy wells, with the goal of predicting models representative of native structures.

In comparison to MD, MC-based methods are advantageous because they are not time based and allow for sampling the low-energy landscape faster. Although visualization of dynamics in action is not possible, such as results seen with MD simulations. Ultimately, MD and MC methods can both provide valuable insight into various aspects of proteins, and each method of approach should be determined on a case-by-case basis.

1.3 Significance of this work

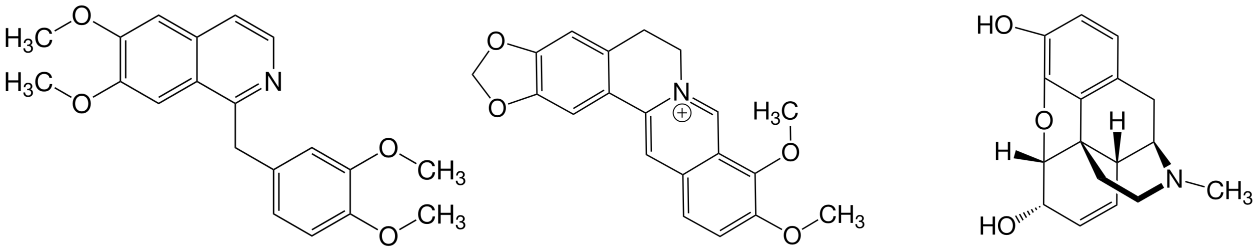
In this work, **Chapter 2** explains a multiscale modeling approach for docking membrane anchored proteins. Using this approach, I modeled a large protein system comprising two redox enzymes within a lipid bilayer. **Chapter 3** explains (i) the ability of current refinement methods within Rosetta to sample dynamic helical hinges, (ii) whether Rosetta score functions bias specific helical conformations, (iii) a degree classification method for helical kinks, and (iv) the generation of a kinked fragment library. Lastly, I discuss the effect that each of these approaches may have on future efforts.

Chapter 2: A multiscale modeling approach for the prediction of membrane anchored protein-protein interface regions

2.1 Background

2.1.1 Recent work resulted in an engineered S. cerevisiae that can generate backbone of benzylisoquinoline alkaloids: a molecule with potent pharmacological properties

Benzylisoquinoline alkaloids (BIAs) are a group of naturally-occurring small molecules that have been extracted for wide-ranging pharmaceutical purposes including pain relief, fighting bacterial infections, healing skin abrasions, and use as a muscle relaxant [13] (**Figure 1**). Their structures include a common backbone of rings with multiple chiral centers [14]. Their history pre-dates the modern era; for instance, greater celandine (Chelidonium majus) has been used for healing purposes throughout Europe and Asia since the Imperial Roman period. There are currently over 2500 known structures with potentially many more BIAs still undiscovered [15].



Berberine Papaverine Morphine

Figure 1: Three benzylisoquinoline examples commonly found in nature

Despite their importance, BIAs can only be extracted from plants in minute amounts. Extraction from plants requires significant land, time for growth, and resources for cultivation [16]. In addition, environmental factors result in significant batch-to-batch variation of BIA purity and production quantities [17]. Chemical synthesis is a common alternate route for pharmaceutical production. However, the large number of chiral centers complicates this approach [18]. Thus far, there are thirty feasible chemical syntheses of different BIAs [19]. Still, the high cost and low yield makes the process suboptimal for batch production. As a result, the scientific community is in search of a more efficient route to BIA production.

Fermentation of microbes such as *E. coli, or S. cerevisiae* are already used to produce amino acids, vitamins, and antibiotics. Recently, genetic engineering technologies have revealed a pathway to synthesis of more complex biomolecules through microbial hosts [20,21]. Specifically, we can transplant a desirable biosynthetic pathway from one organism into a microbial host system such as *E. coli, or S. cerevisiae* that are easier to cultivate in the lab [22]. As a result, we can take advantage of previously-established batch-production techniques to scale production [22].

The technology was first applied by DuPont and Genencor International, Inc. in 2002. Nakamura et al. used metabolic engineering to develop a single organism capable of creating 1,3-propanediol, known for its use in polymers, from D-glucose [23]. This process resulted in the use of an inexpensive feedstock of D-glucose, allowing for increased production through renewable resources [23]. Although microbial engineering is appealing for its short cultivation periods and controlled environment, it is still challenging to optimize the synthetic biology pathways. Even so, due to the possibility of low cost high yield, synthetic biology along with microbial engineering is a promising approach for the production of complex small molecules [24,25].

Recently, Deloache *et al.* applied metabolic engineering to modify a microbial host capable of synthesizing the backbone of BIA molecules [26]. First, they transplanted the cytochrome P450 variant CYP76AD1 from *B. vulgaris* into *S. cerevisiae*. The cytochrome oxidase (CYP) partners with a native reductase (NCP1) to perform the conversion by selectively adding a hydroxyl group onto the aromatic ring. Then, they established a reporting assay that converts L-DOPA to fluorescent Betaxanthin, providing a measure of the catalytic activity (**Figure 2**). Although the CYP variant shows promising activity, its performance remains low, emphasizing the need for enzymatic optimization.

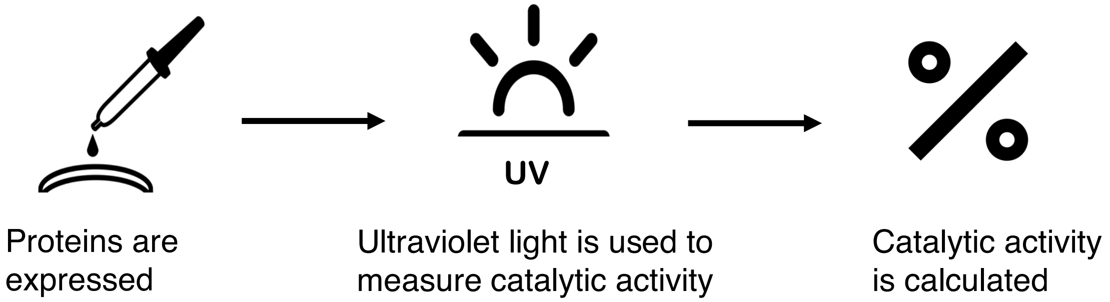

UV
Proteins are expressed
Ultraviolet light is used to measure catalytic activity
Catalytic activity is calculated

Figure 2: Experimental method for evaluating mutant oxidase catalytic activity

2.1.2 Computational protein engineering is a potential route to improve catalysis and subsequently to increase yield

Bioinformatics and molecular modeling techniques have advanced significantly in the past decade and are useful for understanding and engineering enzymes. These tools and methods have been developed to aid in specific design and engineering of enzymes relating to activity, selectivity, and stability. An especially successful enzyme engineering example is the design of formolase, a cornerstone enzyme performing the carboligation reaction step within a novel metabolic pathway to more efficiently use carbon [27]. To design this enzyme, a starting structure was chosen from an enzyme which performs a similar reaction on a different substrate. To increase the activity of the enzyme and specificity for the desired substrate, computational tools were used to redesign the binding pocket for the new substrate. The new substrate was modeled within the binding site, followed by Rosetta Design and Foldit methods to fill in the hollow spaces and increase the binding affinity. Four iterations of computational designs followed by experimental evaluations were performed, resulting in a total of 121 unique designs. To test these designs, enzyme assays were applied to measure the amount of desired product produced, quantifying the catalytic activity [27]. As a result of these methods, a variant with four residue mutations was discovered that yielded 26-fold higher catalytic activity. The promising results from this design methodology demonstrate the possibilities of designing a new enzyme to increase affinity and specificity to catalyze specific reactions.

Another promising example of enzyme engineering includes the thermostability enhancement of a pullulanase enzyme, responsible for the hydrolysis of glycosidic linkages in specific polymers [28]. Because of their poor thermostability, these enzymes result in decreased activity, making them suboptimal. To increase the thermostability of a pullulanase enzyme, four data driven rational design methods (B-FITTER, proline theory, PoPMuSiC-2.1 and sequence consensus approach) [29–31] were used with the goal of generating a highly active enzyme at higher temperatures [28]. These methods predicted 39 residue sites responsible for thermostability of the enzyme, and a mixture of single mutations and combination mutations were tested for stability. Each enzyme was screened for its thermostability through an assay that assesses the enzymatic activity at varied temperature intervals [28]. This guided mutagenesis testing resulted in an enzyme with three mutations that demonstrated an eleven-fold increase in catalytic activity at increased temperatures.

Although these combinations of experimental and computational methods provided promising results, this is not always the case. The complexity of many enzymatic systems can lead to inaccurate predictions which are unable to accomplish their design goal. Furthermore, iterations between computational predictions and experimental testing can be costly and time consuming. Nonetheless, each enzyme system, design goal, and approach requires its own set of restraints and challenges that must be overcome to reach a successful outcome.

2.1.3 The complexity of the CYP76AD1 system poses a challenge to understand and engineer

CYP76AD1, an enzyme native to the *B. vulgaris* plant, is a member of the cytochrome P450 superfamily [26]. Cytochrome P450 enzymes in general are found in all life forms from humans to bacteria and preform catalysis on a wide variety of compounds [32]. I examined the CYP76AD1 enzyme within the *S. cerevisiae* microsomal system. This enzyme is a class II P450 enzyme, meaning that it is found in eukaryotic organisms within the membrane of the endoplasmic reticulum and requires the pairing of a cytochrome P450 reductase (CPR) enzyme for its function [32]. Each CPR enzyme include important small molecules responsible for electron transfer (FAD, FMN). This class of P450s functions to accomplish metabolite synthesis, making them an area of interest for industrial applications [33].

The challenge engineering the CYP76AD1 system for increased catalytic activity is that it encompasses a coupled reaction, a lipid bilayer, and small molecules. CYP76AD1 is a mono-spanning membrane protein and the oxidase enzyme in a redox pair. In order for the CYP76AD1 to perform its catalytic activity, it requires a reductase enzyme capable of transferring electrons to it [34]. NCP1, a natural reductase enzyme within the host *S. cerevisiae,* satisfies this criterion, resulting in a system that functions with low catalytic activity. To understand how these enzymes function, and to engineer CYP76AD1 for increased activity, I must first determine the protein-protein interface between them.

Traditional computational methods for determining interfaces between proteins include global docking, often followed by local docking. Global docking methods, such as ClusPro or ZDock, utilize a Fast Fourier Transform method to find a rough interface that increases shape commentary [35,36], while local docking methods, such as Rosetta Dock [37], rely on more complex algorithms and score functions to locate residue-residue contacts within the interface.

While soluble protein–protein interaction prediction is well studied [35–37], NCP1 and CYP76AD1 are membrane-anchored proteins with large soluble head regions, which presents a challenge to dock, as most tools, such as those mentioned previously, are not designed to predict the membrane protein-protein interface. A different approach to determining the interface is the use of molecular dynamics, which allows for inclusion of a lipid bilayer and is applied to understand the dynamics of a system [38]. However, this method is not feasible for predicting interfaces of large proteins such as our case, because it requires large time scales unachievable by most computational resources.

Furthermore, additional challenges result from the membrane as it interacts with the proteins in multiple possible orientations, depending on how the soluble head regions lay atop the surface [39]. This interaction can result in conformational changes and limit residues available to interact in the protein-protein interface. As a result of these many challenges, there currently is no protocol to determine the interface between membrane proteins, such as CYP76AD1 and NCP1, highlighting the need for a prediction method capable of capturing systems of this type.

2.1.4 A new approach for the prediction of membrane protein-protein interfaces

The goal of my project is to develop an approach for predicting the interface between membrane-anchored proteins such as CYP76AD1 and NCP1. Due to the intricacies of these types of systems, I will leverage multiple computational methods to tackle this problem. Utilizing each method, I will understand a different part of the system, with the goal of generating a docked complex model. This goal is important because the next step is to design mutations that improve the catalytic efficiency.

To develop this complex structure model, I will first generate the structure of each enzyme using homology and helix modeling techniques. This step will provide me with structural models required to further understand the interaction and determine the interface region. Next, to account for conformational changes due to the membrane, I will use molecular dynamics to equilibrate each structure within its natural lipid bilayer.

From the MD trajectory, I can investigate the effect of the membrane on each protein through 50 nanoseconds. Moreover, through the trajectory, I can examine whether the protein has equilibrated within the membrane environment in the length of the simulation time chosen. After each protein is equilibrated within its membrane environment, I will dock these proteins together using a combination of global docking and local docking tools.

To predict the interface between membrane-anchored proteins, I apply different docking algorithms. For global docking, the methods I apply Fast Fourier Transformations (FFT) to provide general regions in which a set of proteins may interact, based on their shape complementary. For local docking, I apply several Rosetta methods ideal for sampling complex model orientations and providing more precise interface regions of my system. This results in model predictions of the interaction between the proteins, which I will validate experimentally.

My combined approach has not yet been attempted on membrane anchored protein-protein systems. This model must be combined with experimental methods to validate and guide the predictions. Being able to accurately predict the interface region of these types of systems is a step in understanding function and designing enzymes for increased activity.

2.2 Methods

2.2.2 Protein structure generation and orientation

By homology modeling, I generated the structure for each enzyme's soluble head region. I applied Swiss Model [40] to discover template structures based on the sequence identity, the global model quality estimation (GMQE), the quaternary structure quality estimation (QSQE), the coverage, and the resolution of each template. Through analysis of the sequence alignment, I identified large gaps in candidate template sequences and poor templates could be discarded. Top template structures identifications chosen for CYP76AD1 were PDBs 1og5A, 6b82 and 3e43. The top template structure chosen for NCP1 was 2bn4. Utilizing the top template structure(s), I generated models for CYP76AD1 and NCP1 through Modeller [41]. Due to the low sequence identity, below 30 percent, of available template models for CYP76AD1, I chose three top identity template structures to generate its structural model. At the end, each soluble head region was ready for the addition of its native ligands.

To accurately represent each protein, it was necessary to include all ligands involved in the system. To do so, I added each small molecule to its respective enzyme through the protein-relative position within the template model. Small molecules nicotinamide adenine dinucleotide phosphate hydrogen (NADPH), flavin adenine dinucleotide (FAD), and flavin mononucleotide (FMN) were added to NCP1 utilizing the top template crystal structure chosen for homology modeling. Heme was added to CYP76AD1 utilizing the top template crystal structure identified through Swiss Model. To avoid clashes due to small molecule placement and to provide unbiased structural models, I applied Rosetta Fast Relax [42] for each crystal based structure. For the relax run, 500 models were generated, and I chose the lowest energy model to continue.

To assemble the completed structure for each protein, I attached the transmembrane domain portions to the soluble head regions. To generate the transmembrane helical domain of each protein, I used Rosetta, with the known transmembrane sequences as the input. Utilizing Pymol [43], I attached each transmembrane helical structure domain to its respective soluble head region. Next, using PyRosetta MP [44], I oriented the protein correctly for placement within the membrane. This step required span file inputs, which described the region of residues within the membrane, to optimally place residues within the correct location. I generated each span file based on the membrane predicted residues for each protein through the use of Octopus [45]. Once I completed these steps, I had prepared each protein for placement within the membrane.

2.2.3 Molecular dynamics simulations

I equilibrated CYP76AD1 and NCP1 computationally within their natural lipid bilayer using all-atom molecular dynamics (MD) simulations. Simulations were setup through CHARMM GUI [46], utilizing the bilayer builder option with the oriented PDB coordinates for the input. First, utilizing the default options, I applied the CHARMM36 forcefield as well as the TIP3 model for water. Second, I chose the composition of each system to provide adequate room for the enzyme to fit within the system, and offering a 15 Å buffer region of water above the protein head. I built the CYP76AD1 system with a composition of 324 DLPE molecules, and 9999 water molecules. Next, I built NCP1 system with a composition of 611 DLPE molecules, 9999 water molecules, and 26 KCl neutralizing ions. Once I generated these starting systems, I had prepared the proteins for the next equilibration steps through molecular dynamics simulations.

To simulate each protein's trajectory, I used NAMD [47], with the input files generated from CHARM GUI. For stability, six simulation equilibration runs ranging from 100 to 400 picoseconds were performed utilizing collective variable restraints to slowly release the system. After the systems were stable, I executed a production run of 50 nanoseconds at a temperature of 303.15 K to equilibrate each protein within its system. I chose the temperature based on the optimum temperature that S. cerevisiae grows. From each output trajectory, I pulled a single structure that most represented the final equilibrated protein. These two structural models are now ready to be used as inputs to generate docked models.

2.2.4 Prediction of the protein-protein interface

To prepare the proteins for global docking, I removed each enzyme from its lipid bilayer and generated masks for the transmembrane helical domains. Each mask specified the residues involved in the transmembrane region within each enzyme and I applied these files as repulsion residues within ClusPro [36]. Using ClusPro, along with the input of each enzyme's PDB file, I globally docked NCP1 to CYP76AD1. ClusPro outputted 30 best ranked models. A favorable complex model resulted in 5 to 7 residue-residue contact sites, and structures with fewer than 3 sites were unfavorable. Complex model shapes which minimized the distance for electron transfer between heme and

FMN, were considered favorable. Of these models, I chose the top four structures, based on the number of contacts within the interface and the shape complementary.

Once globally docked structures were chosen, Rosetta was applied to locally dock the top four complex models. Three different docking protocols were tested, Local Dock [48], MP Dock [44] and Ensemble Dock [49]. Rosetta Dock does not take into account protein flexibility or membrane constraints, whereas Rosetta MP Dock models the membrane and the constraints that these membrane anchored proteins could have. Rosetta's Ensemble Dock allows for flexibility through small conformational changes, which may occur through docking. By combining these methods, I collected likely protein-protein contact residues.

For each protocol, I generated a thousand structures and the structure with the lowest interface energy score was chosen to move forward. To run Ensemble Dock, I generated an ensemble consisting of 50 structures utilizing Rosetta Relax. Outputs from these protocols were then refined further utilizing the Rosetta local refinement protocol.

2.2.5 Experimental validation and structure prediction guidance

To test the docked models, I used experimental results. For each top complex model, mutations were proposed that would disrupt the interface. I picked mutation sites based on polar interface residues within the CYP76AD1 enzyme. For each mutation site, I chose both a conservative and non-conservative mutation depending on the current residue type. Mutations were picked to alter the residue from large to small, polar to non-polar, charged to no-charge, and vice versa. Experimental testing of these mutations was performed by the Dueber at UC Berkeley with a biosensor enzyme [26] to quantify the catalytic activity. I applied Rosetta Flex *ΔΔG* [50] in combination with these experimental results to analyze each complex prediction for accurate mutation sites within the interface. I ran Flex *ΔΔG* on each mutation suggested for each structure, with the default settings. This outputted quantitative values representing whether a mutation stabilized or destabilized the interface.

2.3 Results

2.3.1 Protein structure generation and orientation within the membrane

Homology modeling is a powerful tool that exploits sequence similarity to generate structural models. I applied homology modeling tools because the structure of CYP76AD1 was not available and the structure of NCP1 was not complete. This method allowed me to generate predictive structural models for each protein based on its sequence solely. To model CYP76AD1, I chose three top identity template structures, 1og5A, 6b82 and 3e43. These template structures had sequence identities of 27%, 29% and 27% respectively to CYP76AD1. Within each template, 9-12 gaps were present, and there were 113 conserved residue positions between the three templates.

To model NCP1, I used template structure 2bn4 with an identity of 99%. The template structure resulted in 24 gaps arising from the transmembrane domain region, which was missing in the template structure. However, all residues that were present within the template were conserved within NCP1. The generated models, template structure IDs, and percent identities are shown in **Table 1**.

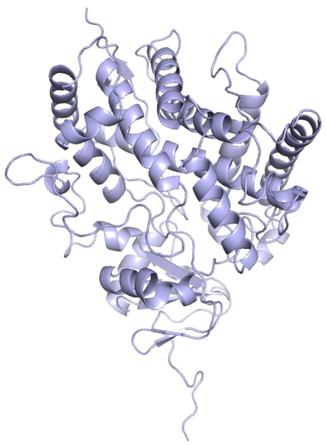
STRUCTURAL MODEL		
PROTEIN	CYP76AD1	NCP1
PDB ID	6b82, 1og5A, 3e43	2bn4
IDENTITY (%)	29, 27, 27	99

Table 1: Structural model generated for CYP76AD1 (blue), NCP1 (tan), the percent identity for each model and template PDB ID(s) for NCP1 and CYP76AD1.

Protein topology prediction methods are beneficial for predicting environmental conditions of each residue because I did not have environmental information for each residue and I did not know which residues were located within the membrane. I applied OCTOPUS, a topology prediction method, to both proteins. This method predicted the transmembrane span for each protein and these outcomes are shown in **Figure 3**. For both NCP1 and CYP76AD1, there were two regions predicted to be within the membrane, and the bulk of each protein located inside the membrane.

Outside Membrane Inside Membrane Within Membrane

Sequence:CYP76AD1
MDHATLAMILAILFISFHFIKLLFSQQTTKLLPPGPKPLPIIGNILEVGKKPHRSFANLAKIHGPLISLRLGSVTTIVVSSA
DVAKEMFLKKDHPLSNRTIPNSVTAGDHHKLTMSWLPVSPKWRNFRKITAVHLLSPQRLDACQTFRHAKVQQLYE
YVQECAQKGQAVDIGKAAFTTSLNLLSKLFFSVELAHHKSHTSQEFKELIWNIMEDIGKPNYADYFPILGCVDPSGIR
RRLACSFDKLIAVFQGIICERLAPDSSTTTTTTTDDVLDVLLQLFKQNELTMGEINHLLVDIFDAGTDTTSSTLEWVMT
ELIRNPEMMEKAQEEIKQVLGKDKQIQESDIINLPYLQAIIKETLRLHPPTVFLLPRKADTDVELYGYIVPKDAQILVNL
WAIGRDPNAWQNADIFSPERFIGCEIDVKGRDFGLLPFGAGRRICPGMNLAIRMLTLMLATLLQFFNWKLEGDISPK
DLDMDEKFGIALQKTKPLKLIPIPRY

Sequence:NCP1
MPFGIDNTDFTVLAGLVLAVLLYVKRNSIKELLMSDDGDITAVSSGNRDIAQVVTENNKNYLVLYASQTGTAEDYAK
KFSKELVAKFNLNVMCADVENYDFESLNDVPVIVSIFISTYGEGDFPDGAVNFEDFICNAEAGALSNLRYNMFGLGN
STYEFFNGAAKKAEKHLSAAGAIRLGKLGEADDGAGTTDEDYMAWKDSILEVLKDELHLDEQEAKFTSQFQYTVLN
EITDSMSLGEPSAHYLPSHQLNRNADGIQLGPFDLSQPYIAPIVKSRELFSSNDRNCIHSEFDLSGSNIKYSTGDHLA
VWPSNPLEKVEQFLSIFNLDPETIFDLKPLDPTVKVPFPTPTTIGAAIKHYLEITGPVSRQLFSSLIQFAPNADVKEKLT
LLSKDKDQFAVEITSKYFNIADALKYLSDGAKWDTVPMQFLVESVPQMTPRYYSISSSSLSEKQTVHVTSIVENFPN
PELPDAPPVVGVTTNLLRNIQLAQNNVNIAETNLPVHYDLNGPRKLFANYKLPVHVRRSNFRLPSNPSTPVIMIGPG
TGVAPFRGFIRERVAFLESQKKGGNNVSLGKHILFYGSRNTDDFLYQDEWPEYAKKLDGSFEMVVAHSRLPNTKK
VYVQDKLKDYEDQVFEMINNGAFIYVCGDAKGMAKGVSTALVGILSRGKSITTDEATELIKMLKTSGRYQEDVW

Figure 3: Octopus topology prediction results for CYP76AD1 and NCP1. Residues on the outside of the membrane are labeled in blue, residues on the inside of the membrane are labeled in black and residues within the membrane are labeled in red.

2.3.2 Molecular dynamics simulations

All-atom MD simulations, in combination with experimental validation, are useful for visualizing molecular interactions on the atomic scale. This method is needed because CYP76AD1 and NCP1 structural models were generated from crystal structures in the solubilized form. I used constant pressure of 1 atm, a temperature of 30 °C, and a membrane composed of 1,2-Dilauroyl-sn-glycero-3-phosphoethanolamine (DLPE) lipids. MD allowed me to interrogate the protein conformation and interactions with the membrane. To this end, I simulated both enzyme models in this membrane system.

To analyze the simulations, each trajectory was judged as equilibrated using two quantities: (i) root mean square deviation (RMSD) of the backbone atoms referenced to the starting structure, and (ii) distance dependent measurements from a the lipid bilayer center of mass to set residues throughout the protein. Selected residues to track over the duration of the trajectory are highlighted in **Figure 4.** residue distance measurement plots for each protein are shown in **Figure 5**, which illustrates small deviations with maximum magnitudes of 2.2 angstroms for CYP76AD1 and 2.8 angstroms for NCP1. Moreover, RMSD plots of each protein's trajectory are shown in **Figure 6**, which illustrates small deviations with maximum magnitudes of 0.56 Å for CYP76AD1 and 0.50 Å for NCP1.

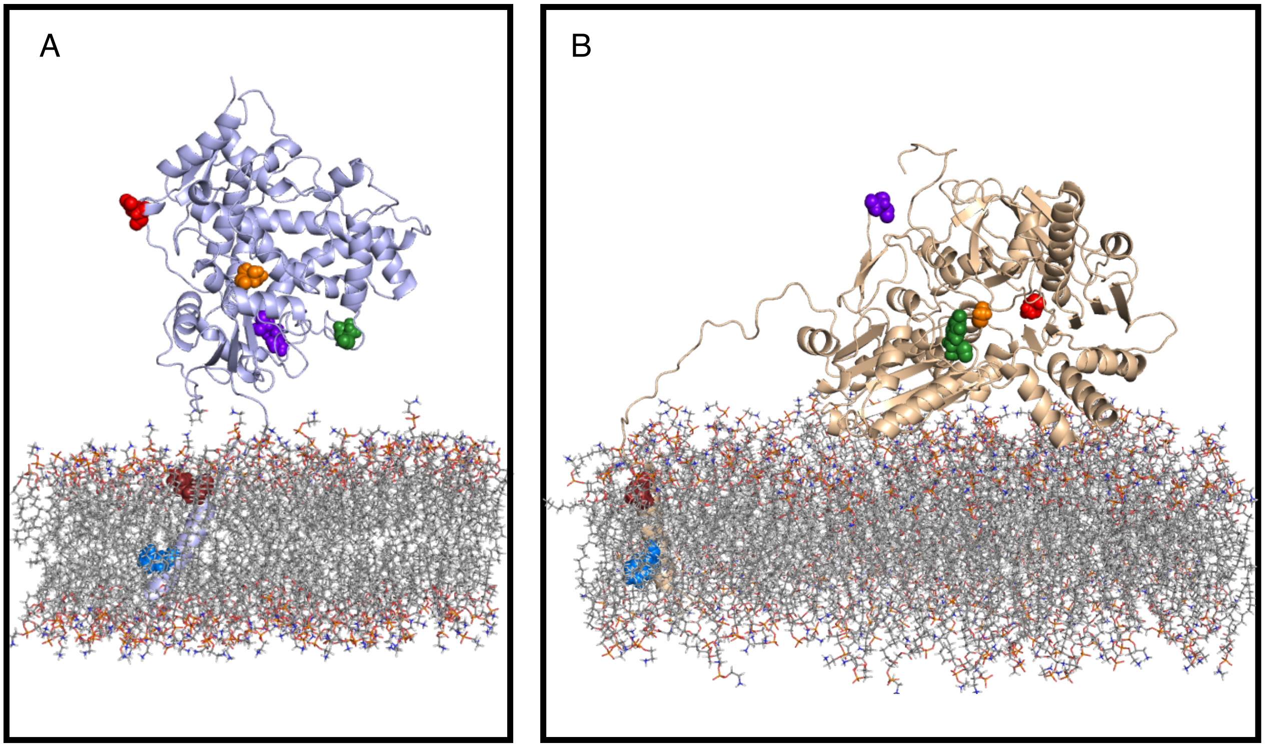


Figure 4: (A) CYP76AD1 (purple) and (B) NCP1 (tan), with tracked residues represented as spheres. The colors correspond to the residue being tracked. Residue 10 in blue, residue 21 in brown, residue 118 in green, residue 235 in purple, residue 364 in orange, and residue 471 in red.

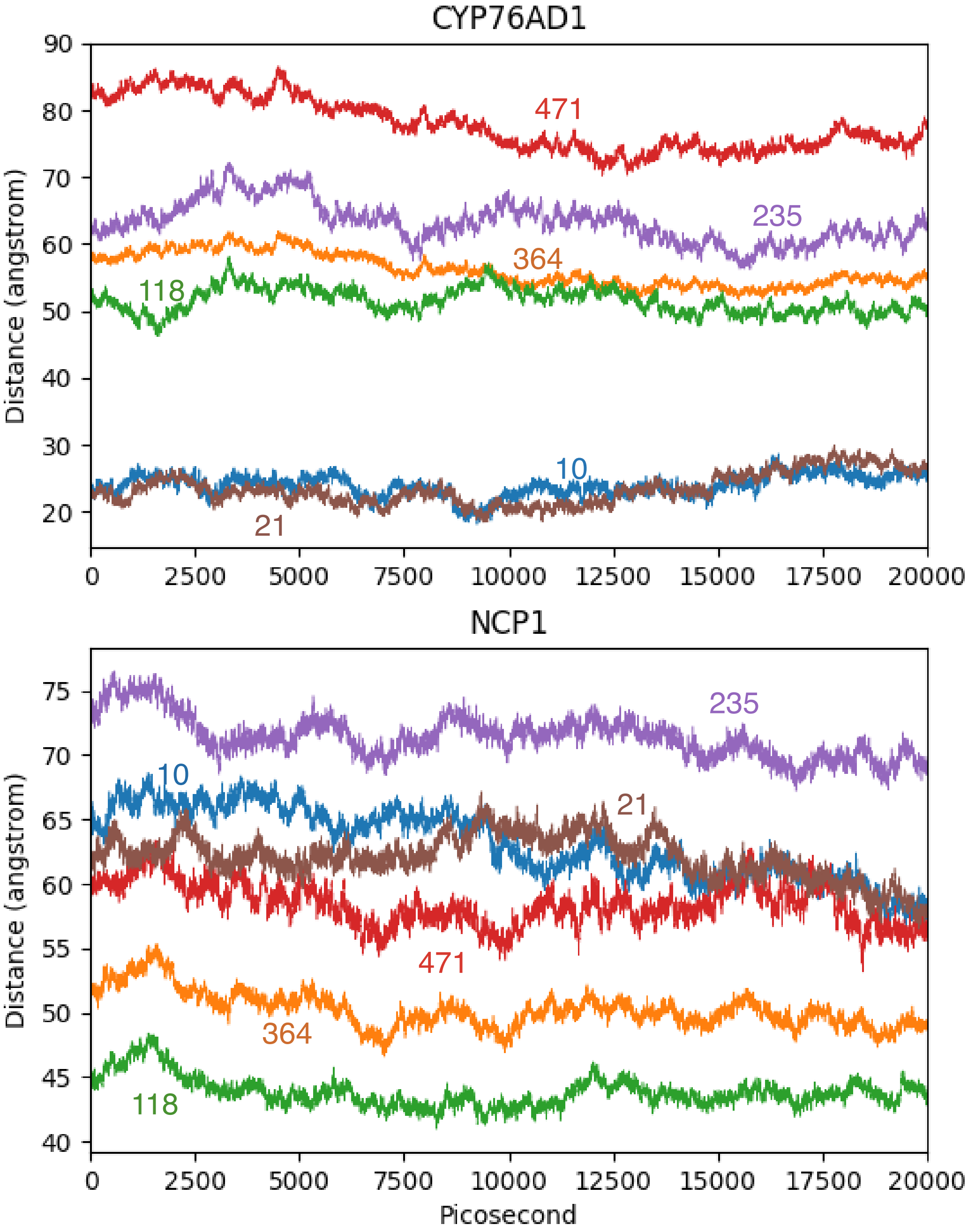


Figure 5: Distance measurement from specified residues to the lipid bilayer center of mass over the last 20 nanoseconds of each protein's trajectory. The colors correspond to the residue being tracked. Residue 10 in blue, residue 21 in brown, residue 118 in green, residue 235 in purple, residue 364 in orange, and residue 471 in red.

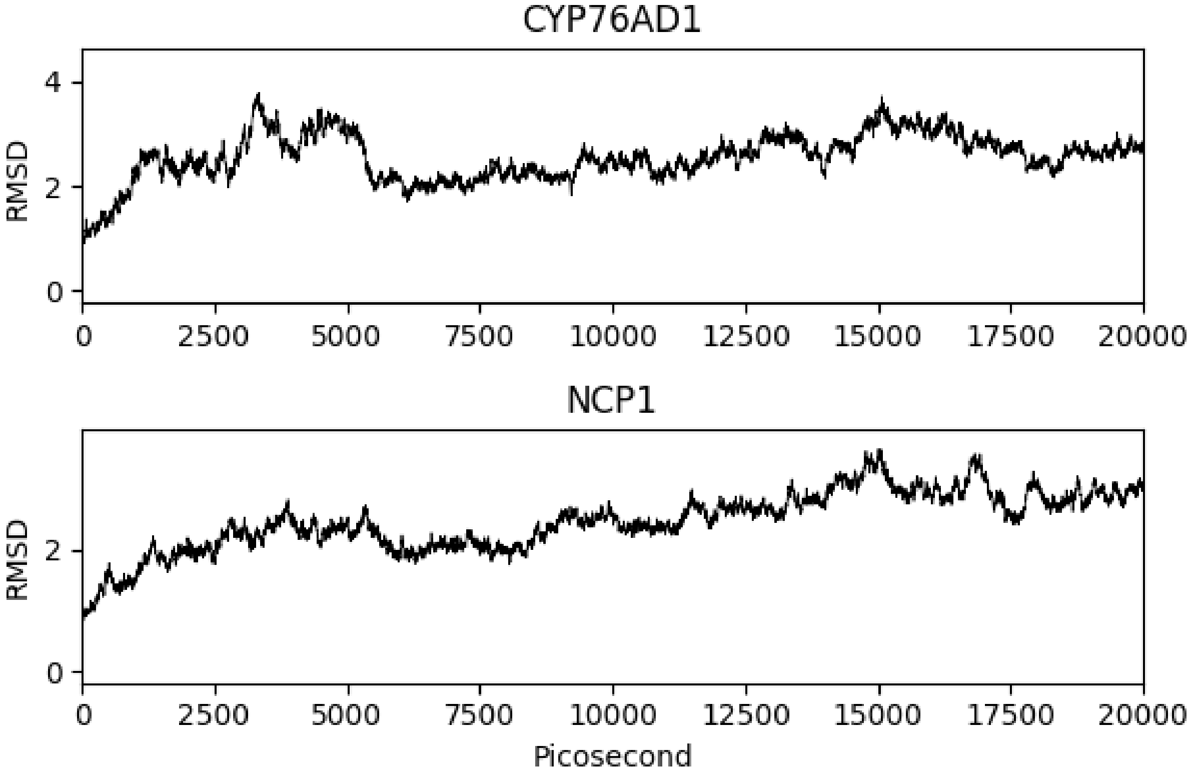


Figure 6: Backbone RMSD of CYP76AD1 and NCP1, relative to the starting conformation, over last 20 nanoseconds of the trajectory.

2.3.3 Prediction of the protein-protein interface

Global docking predicts the interface interactions between two proteins based highly on their shape complementary. I used docking because there was no previous data about how these two proteins interacted with each other. With ClusPro, I predicted candidate interface regions between CYP76AD1 and NCP1. Through visual examination of these candidate complexes, I discovered that all models involved CYP76AD1 interacting with a single side of NCP1, with the top 4 models shown in **Figure 7B**. Interaction of this side of NCP1 with CYP76AD1 is advantageous because it minimizes the electron transfer distance from FMN to heme (**Figure 7A**). This result suggests the favorability for this region of NCP1 to be involved in the interface. Furthermore, CYP76AD1 preferred to orient itself so that it interacted with NCP1 on the side furthest from the heme small molecule. In fact, 17 out of the 30 complex models produced resulted in this orientation of CYP76AD1.

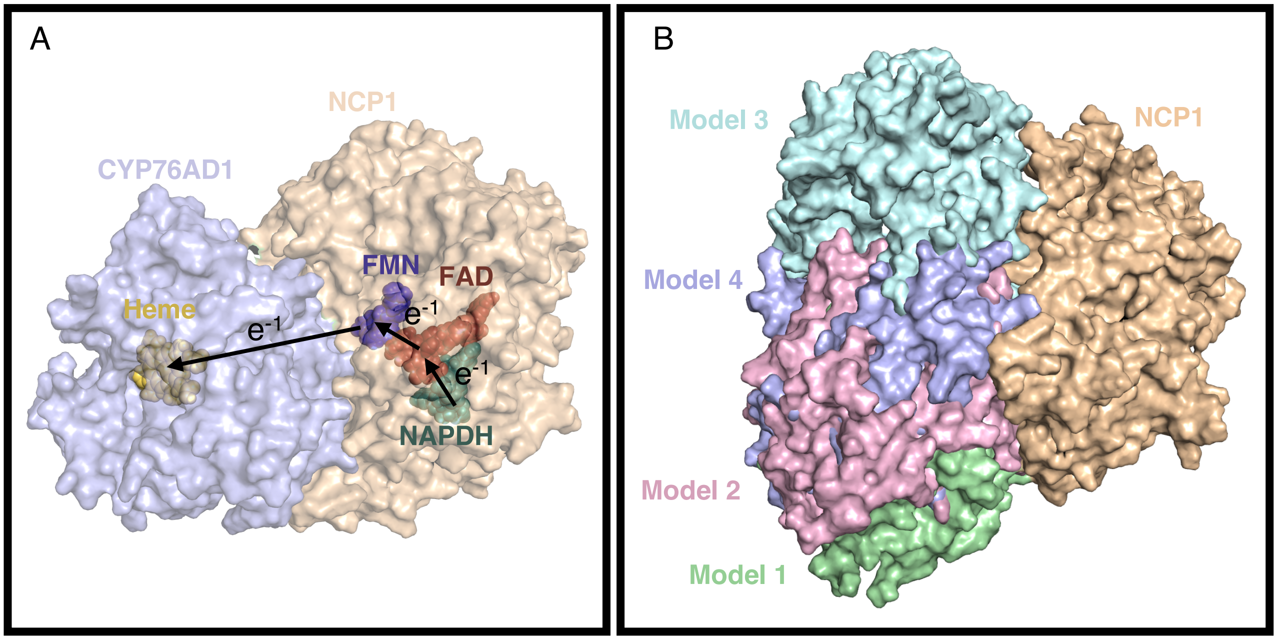


Figure 7: (A) Electron transfer pathway for the NCP1-CYP76AD1 complex model 4. (B) Top 4 complex model predictions through docking. NCP1 is aligned for each complex model and shown in tan. CYP76AD1 is shown in teal, purple, pink, and green, depending on the complex model.

I performed local docking through Rosetta Dock, Rosetta MP Dock, and Rosetta Ensemble Dock to gain final complex models, which resulted in interface contact residues between CYP76AD1 and NCP1. The number of polar interface contact sites for the top 4 models was 4-6 sites, providing a large interface surface area. Each of the top complexes demonstrated an orientation of NCP1 which minimized the distance for electron transport. Coevolutionary methods apply the use of multisequence alignment and evolutionary information from homologous proteins to predict the residues of each protein expected to be involved in the interface. I applied InterEvDock2 [51] to determine five residues within each CYP76AD1 and NCP1 that were predicted to be involved in the interface. These residues are labeled and shown in cyan in **Figure 8A-B**. Further analysis of the structure revealed that the evolutionarily predicted residues within NCP1 fell in the globally docked prediction interface region for the top models, as shown in **Figure 8C**.

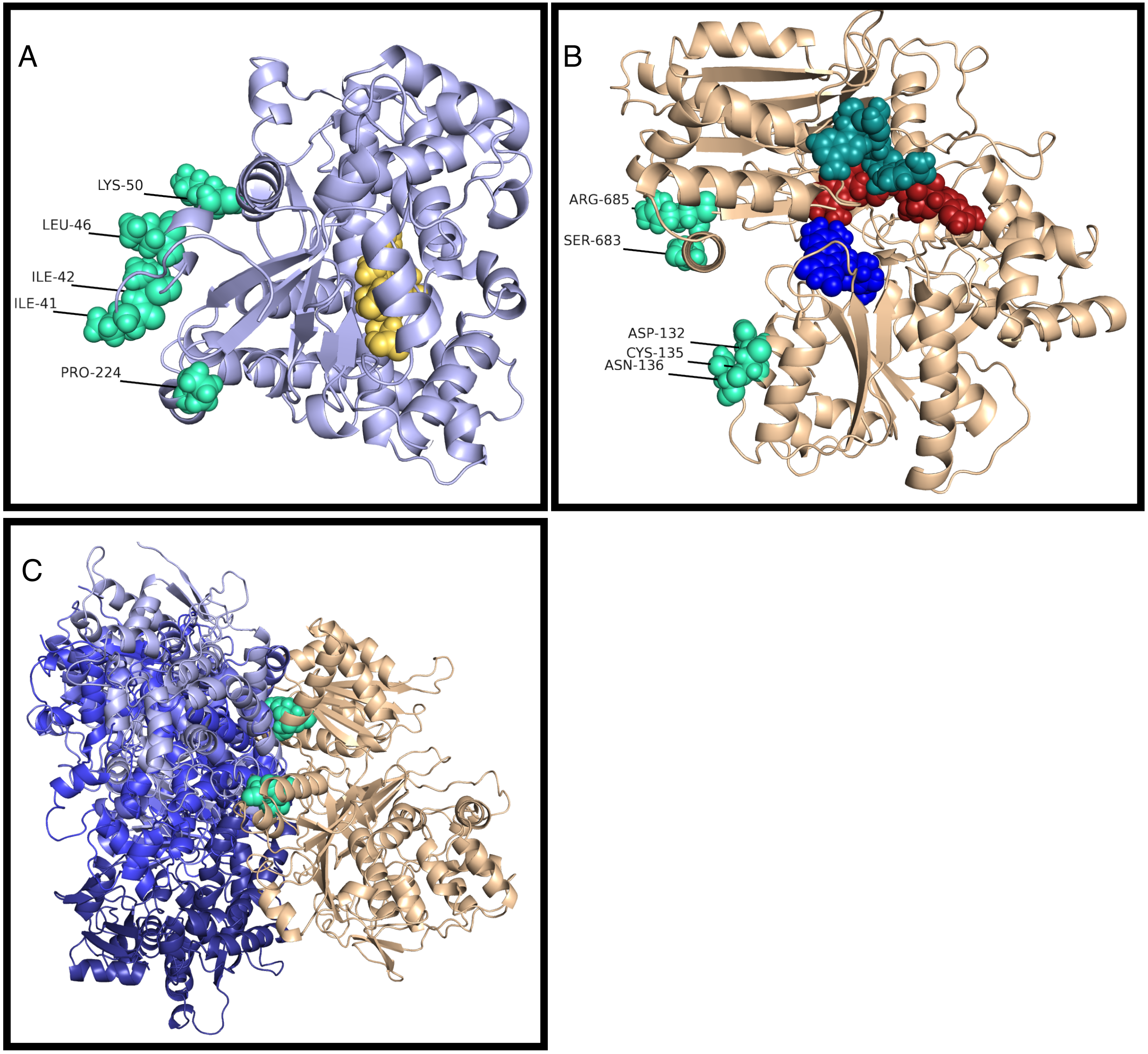


Figure 8: (A) Top 5 residue predictions of InterEVDock2 for CYP76AD1, represented in spheres and colored in cyan. (B) Top 5 residue predictions of INterEVDock2 for NCP1, represented in spheres and colored in cyan. (C) Involvement of coevolutionary interface residue predictions for NCP1, within the docked top 4 models. CYP76AD1 is shown in different shades of blue, depending on the complex model and NCP1 aligned for each complex model and shown in tan.

2.3.4 Experimental validation and structure prediction guidance

Our collaborators at UC Berkeley experimentally tested the predicted complex models that I generated through computational modeling. The experimental technique required the use of a biosensor enzyme to convert L-DOPA to fluorescent betaxanthin, which could be used to quantify the catalytic activity. By experimentally testing 30 unique interface disruptive mutations, shown in Table 2, which I determined for each complex model, I gained quantitative measurements of the percent activity for each mutant oxidase.

		WT	WT		
#	Origin	Donor	Acceptor	Suggested Mutation 1	Suggested Mutation 2
1	CYP76AD1	ARG 685	ASN 208	ASN 208 → GLY 208	ASN 208 → TRP 208
		THR 106	GLU 131	THR 106 → ASP 106	THR 106 → TRP 106
		ARG 234	GLU 120	ARG 234 → THR 234	ARG 234 → GLU 234
		ARG 235	ASP 122	ARG 235 → THR 235	ARG 235 → ASP 235
		ARG 236	GLU 120	ARG 236 → SER 236	ARG 236 → ASP 236
		HIS 111	HIS 170	HIS 111 → GLY 111	HIS 111 → TRP 111
2	CYP76AD1	THR 285	THR 682	THR 285 → GLY 285	THR 285 → TRP 285
		ARG 235	GLU 554	ARG 235 → SER 235	ARG 235 → TRP 235
		ARG 685	GLU 254	GLU 254 → GLY 254	GLU 254 → TRP 254
		SER 533	ASP 242	ASP 242 → GLY 242	ASP 242 → HIS 242
3	CYP76AD1	ARG 255	ASN 298	ARG 255 → SER 255	ARG 48 → ASP 255
		LYS 300	GLU 190	GLU 190 → GLY 190	GLU 190 → TRP 190
		ARG 523	GLU 254	GLU 254 → SER 254	GLU 254 → ARG 254
		ARG 48	ASP 242	ASP 242 → GLY 242	ASP 242 → ARG 242
		ARG 235	GLU 56	ARG 235 → GLY 235	ARG 235 → GLU 235
		ASN 136	HIS 111	HIS 111 → ALA 111	His 111 → ASP 111
4	CYP76AD1	ARG 523	ASP 229	ASP 229 → GLY 229	ASP 229 → TRP 229
		ARG 70	SER 561	ARG 70 → SER 70	ARG 70 → ASP 70
		ARG 235	ASP 125	ARG 235 → SER 235	ARG 235 → GLU 235
		HIS 63	ASN 567	HIS 63 → GLY 63	HIS 63 → ASP 63
		SER 533	ASP 220	ASP 220 → GLY 220	ASP 220 → TRP 220

Table 2: Mutations suggested for each predicted complex model. Residues within NCP1 are shown in red and residues within CYP76AD1 are shown in blue. For each mutation site, there is two proposed mutations, a conservative choice and a less conservative choice.

Out of the 15 unique mutation sites I chose, only 4 were experimentally possible. The percent activity obtained from each oxidase mutation is shown in **Figure 9**. Out of these results, two mutations decreased the catalytic activity significantly. Both mutations were located at site T106 within the oxidase enzyme, which is a surface residue site located on an outer loop. I can use these experimental results further, in combinations with binding affinity calculations to investigate the likelihood of each complex model.

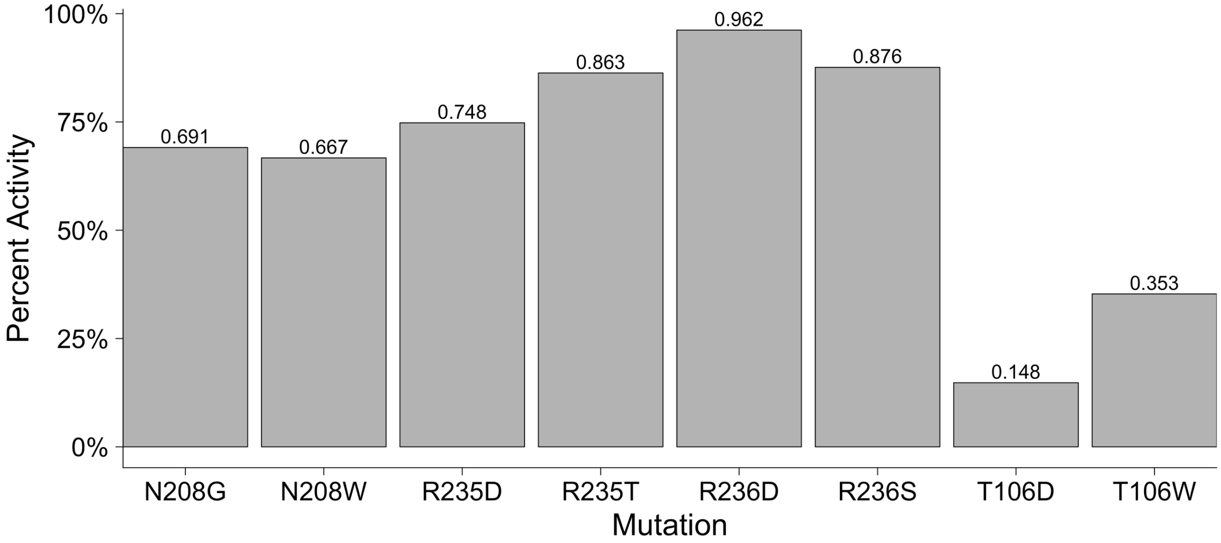


Figure 9: Percent activity, compared to the wildtype, for each mutant oxidase variant obtained by John Dueber's Lab.

To explore the likelihood that these mutations may be involved in the interface region and not just affecting catalytic activity through other means, I performed computational binding energy calculations. Using Rosetta's Flex $\Delta\Delta G$ protocol [50], I calculated the change in binding energy due to each mutation within each complex model. These results are shown in **Figure 10**. These data show the extent to which each mutation should disrupt or stabilize each complex model. Mutations predicted to be within the interface for each complex model demonstrated disruptive effects in most cases. One case to highlight is mutation T106W, which presented stabilizing effects on each complex model proposed.

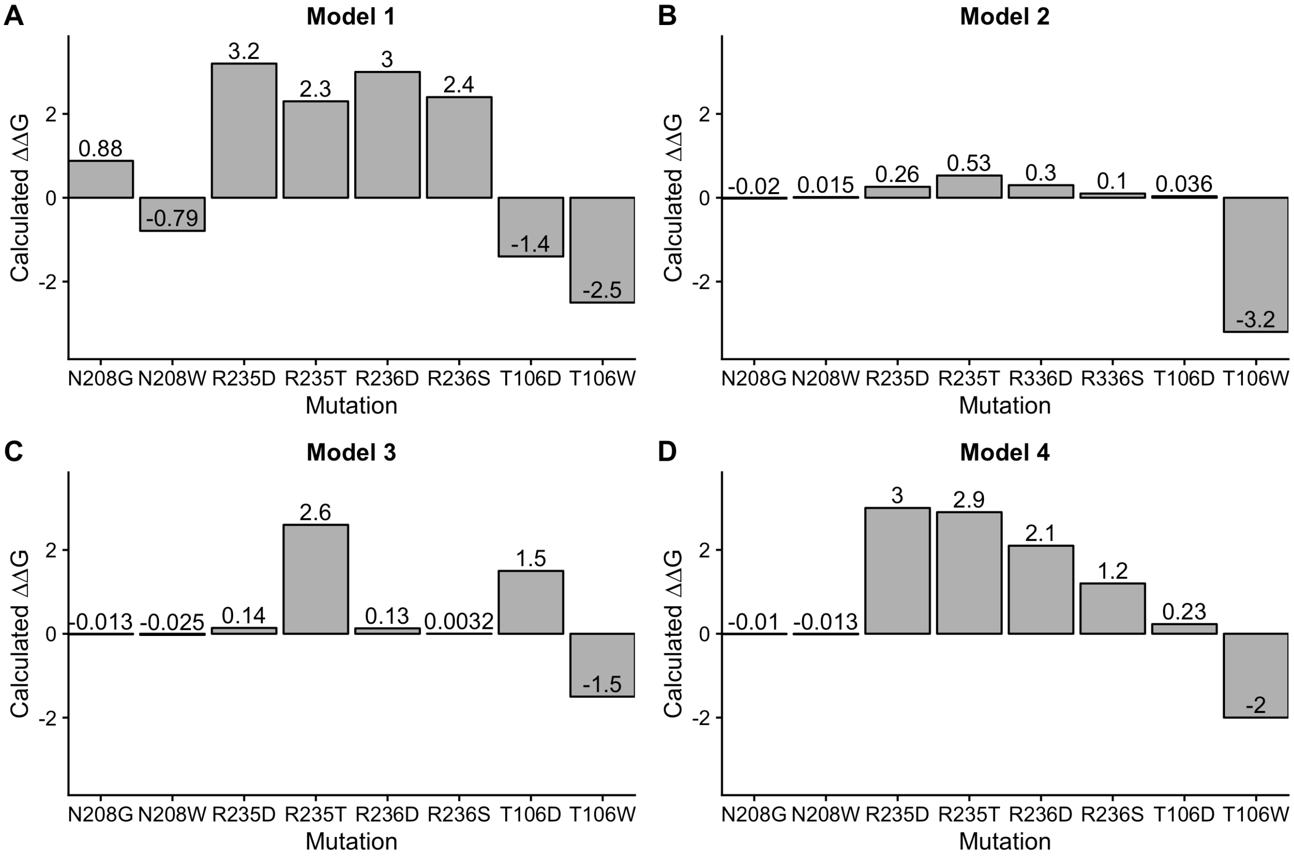


Figure 10: Flex ΔΔG calculated results representing the change in binding energy due to each mutation. Panels A-D illustrate results for the top models, respectively 1-4.

By comparing the experimental catalytic activity results with the calculated changes in binding energy, I could validate or discredit each complex model. To do this, I plotted the percent catalytic activity vs. the change in binding energy for each model (**Figure 11**).

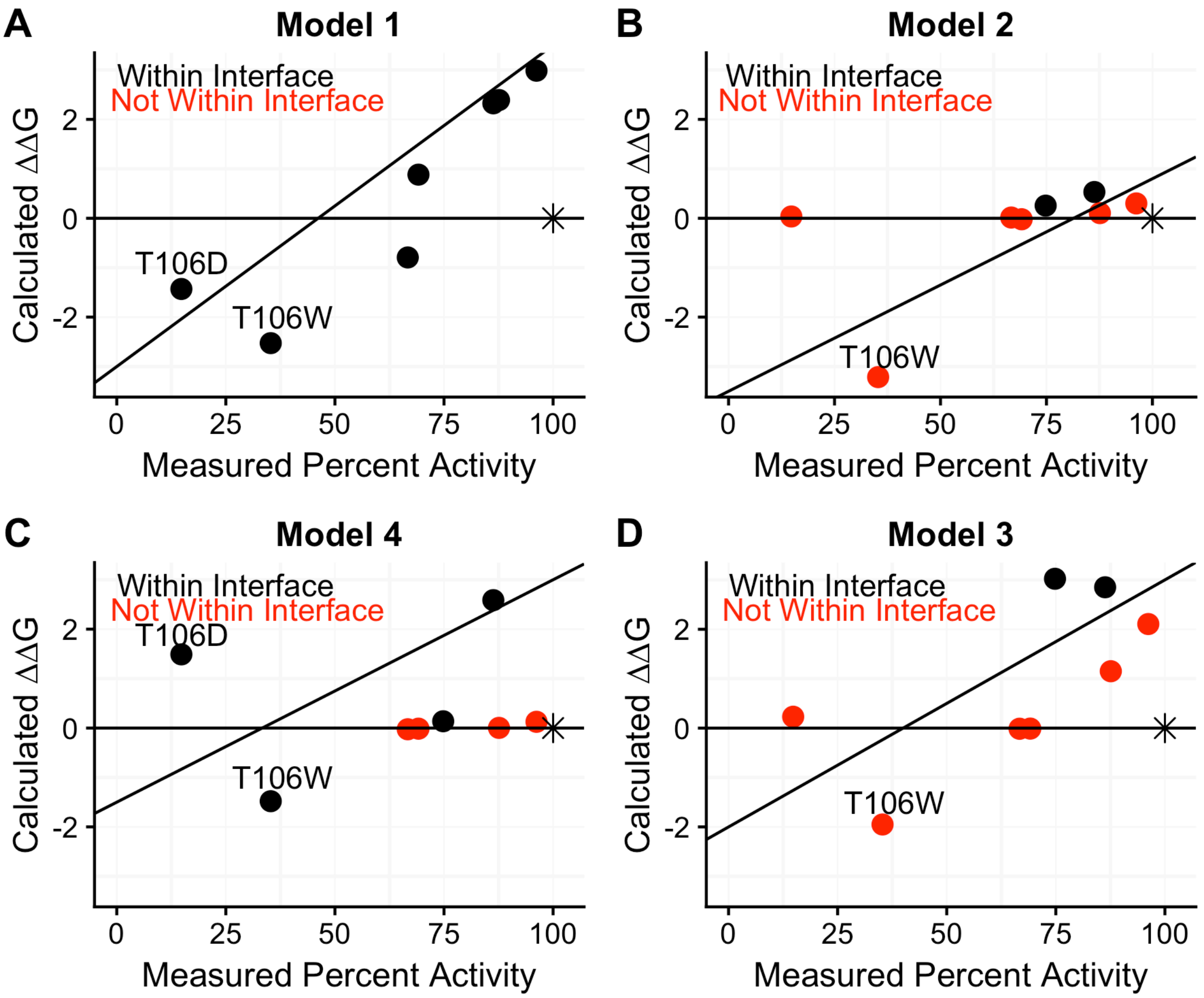


Figure 11: Plots comparing the change in calculated binding energy due to each mutation and the measured catalytic activity of each CYP76AD1 Variant. The top 4 complex models 1-4 are shown in panels A-D, respectively.

Results of the correlation plots do not show a clear trend. In Figure 11, each model is predicted to stabilize results for mutation T106W, even though experiments show significant decreased catalytic activity. Model 1, predicts disruptive mutation sites (235, 236, 208) within the interface, but experiments demonstrate a minimal decrease in catalytic activity. Model 2 predicts one mutation site (235) to be within the interface. Experimentally, this model mutation site shows little decrease in catalytic activity and a neutral effect on the binding energy. Model 3 show mutation T206D to be within the interface. For this structural mutation, experimental decreased catalytic activity is applied with the predicted disruptive calculations. Model 4 shows two mutation sites predicted to be disruptive (235, 236). Of these sites, one was located within the interface (235) and the other was not (236), but both showed little deceases in catalytic activity.

2.4 Discussion

2.4.1 Protein structure generation and orientation within the membrane

The accuracy of homology modeling depends on the percent sequence identity of the template to the structure and the number/location of gaps in the template structure. The reductase template structure had a high identity (99%) and a region of 24 gaps at the N-terminus, which encompassed the section located within the membrane only. Thus, the prediction structure for the soluble head region of the reductase is expected to be reliable. Conversely, the top oxidase template structures chosen did not have high identities (< 30%), and demonstrated gaps throughout the alignment. However, prior research has shown that this superfamily of proteins is structurally conserved [52], and studies show identity templates of approximately 30 percent that are capable of yielding reliable structures [53]. As a result, these studies provide evidence that our oxidase prediction structure has the likelihood to generate a reliable representation of the true structure.

Topology prediction methods are useful, but not always accurate. Thorough examination of the results, along with the knowledge of each protein, must be used to determine if the predictions are credible. The output from OCTOPUS predicted two regions from each protein to be located in the membrane. Examination of the cytochrome P450 superfamily of proteins suggested that these microsomal P450 enzymes contain a single N-terminal membrane spanning region [54]. Furthermore, by examining these regions manually, I could see that the second transmembrane predicted region for each protein was located within the soluble head portion of the enzyme, a place that may contact the membrane but is unlikely to be located within the membrane. As a result, I concluded that only the first transmembrane segment predicted was located within the membrane, for each protein, aligning with previous studies on cytochrome P450 enzymes [54,55].

2.4.2 Molecular dynamics simulations

I used all-atom MD simulations to explore small structural changes occurring through the equilibration of each protein within its natural lipid environment. Previous studies have used MD to focus on identifying the multiple orientations that cytochrome P450 enzymes can have with the lipid bilayer [39]. For my case, I observed that each protein had the potential to interact with the membrane and preferred to lay on the membrane with the possibility of multiple different protein-membrane interface orientations. Although my project does not go deeply into these possible orientations, it does support that the enzymes do interact with the membrane, as seen in previous work [39,56]. Through this equilibration analysis, I discovered only small variations in each protein's structure as a result of the membrane environment. These small variations suggests that each protein has had time to equilibrate within the environment, and the resulting conformations chosen are representative of the proteins within the membrane.

2.4.3 Prediction of the protein-protein interface

Global and local docking tools provided me with four top model predictions of the interface region between CYP76AD1 and NCP1. For membrane anchored proteins, global docking falls short, as it does not take into account the membrane and the constraints/effects it could have on the complex model. However, it does provide likely starting points that I can down based on my knowledge of the system. From pruning the given structures resulting from global and local docking, I was able to narrow the complex models down to predictions that demonstrated 4-6 polar contact sites, and complex models that minimized the distance for electron transport. The final structures possessed a high count of interface contact residues, the interface did not block any active sites, and each complex had the ability for the helical transmembrane domains to be oriented parallel to each other within the membrane.

Coevolutionary methods suggested five residues within each protein to be involved in the interface. I investigated the residue locations within the top models produced through local docking. These results were used to either support and strengthen or weaken the complex model predictions. Although the results predicted residues involved in the interface, there is little prior information about the interaction contact residues between the reductase and oxidase enzymes of the cytochrome P450 superfamily. As a consequence, these results were not weighted heavily in choosing complex model predictions, but did help to validate some of the top prediction models.

2.4.4 Experimental validation and structure prediction guidance

Experimental measurements located a noteworthy site, residue 106, which when mutated to tryptophan or aspartate, decreased the catalytic activity significantly. Although the percent activity values acquired through this method can correlate to mutations within the interface, they may also correlate to mutations near the active site, or mutations altering the stability of the oxidase. Site 106 is located on the surface of the protein and ample distance away from the active site. Therefore, it is likely that the decrease in catalytic activity was not due to alterations in protein stability or effects in the active site, but instead due to a destabilizing mutation within the interface.

I investigated the validity of each predicted complex model by comparing experimental catalytic activity and the computationally calculated changes in binding energy. If the complex model was accurate, I would have expected to see a trend where the calculated disruptive mutations decrease the catalytic activity experimentally. However, a clear trend was not seen for any complex model, and predictively disruptive mutations did not decrease the catalytic activity significantly. These results suggest that the complex models predicted are incorrect and do not provide the interface structure. Therefore, I was able to use the calculated change in binding free energy to discredit the current prediction models, a step that can be repeated for future prediction models.

Although this approach found a residue site likely involved in the interface, none of the predicted complex models are likely to be correct. For models 1, 2 and 4, the mutation 106W was not predicted to be disruptive through Flex $\Delta\Delta G$, which is required for the complex prediction models to be accurate. For models 1, 3, and 4, Flex $\Delta\Delta G$

predicted several mutations within sites 235, 236 and 208 to be disruptive. However, these mutations showed little decrease in the measured catalytic activity, suggesting the complex models are inaccurate. If this approach was applied again, it would suggest alterations that could improve model predictions.

In the future, there are several steps I would change that may yield better models. First, since initially finding homology template models for CYP76AD1, a new structure has been characterized which presents a higher sequence identity of 46%. This structure better represents CYP76AD1, and this template structure could produce more favorable results. Second, more MD simulations of each enzyme could be implemented to determine more possible membrane-protein orientations of each CYP soluble head region. Not only would this provide a better understanding of the system and its function, but it could also limit the interface search space, resulting in better complex model predictions.

Through the entirety of this process, I was not able to determine the exact interface regions between the two membrane-anchored enzymes, demonstrating how challenging membrane protein systems are to model. However, I did find an important residue at site 106, which is likely involved in the interface. This discovery is important, and can be used in the future as a constraint for predicting new complex models. Additional rounds of model prediction and validation through experimental methods is required, with the potential to successfully yield the interface region between CYP76AD1 and NCP1.

Chapter 3: Dynamic helical hinges in membrane proteins

3.1 Background

3.1.1 Helical kinks are important for membrane protein function and are responsible for conformational changes

Protein structure prediction at atomic-level resolution is important to accurately capture inter- and intra-molecular interactions important for function. A unique feature of α-helical membrane proteins is kinked helices: a bend or distortion in the primary axis of the helix, often breaking or altering the hydrogen bonding pattern. interestingly, 64% of the transmembrane helices contain kinks or non-idealities [57]. These kinks or distortions are an attribute of long α-helices and have been identified as playing an important role in the function of many proteins. By acting as sites for flexibility, kinks can be responsible for conformational changes within membrane proteins [58–61]. For example, opening and closing of the KcsA, a potassium channel protein is facilitated by flexible helix kinks [59,61]. Similarly, other membrane proteins functioning in signaling and enzymatic activity involve conformational changes resulting from helical kinks [58,62]. Thus, describing the importance of understanding helical kinks and what causes them.

3.1.2 The origin of helix kinks

There have been two studies to investigate the origin of helix kinks [60,63], resulting in different hypotheses about their origin. Initially, helix kinks were thought to originate from proline residues at or near the hinge-point of the helix. Even though proline is present in many alpha helices [64], only 20 percent of prolines result in a kinked helix [60]. An alternative hypothesis is the formation of kinks due to vestigial prolines [63].

Vestigial proline kinks are a result of past proline residues that have been mutated over the course of evolution. A recent study demonstrated that vestigial proline kinks accounted for 16 percent of transmembrane protein helical kinks [57], composing a large portion of kinked helices. A third hypothesis is the importance of non-canonical hydrogen bonding between side-chain and main-chain. Common residues that lead to these types of kinks are serine and glycine [60], but it is also possible for other residues to be responsible for such kinks. The percentage resulting from this type of kinking is currently unknown, and there are still other factors that likely come into play, which have not yet been identified.

3.1.3 Classification Methods available to identify helical kinks in structures

Currently there are several methods used to identify helical kinks, and each uses its own definition of what extent of bend or distortion represents a kink and how to calculate the angle [57,65–67]. For example, the method ProKink [65] relies on a proline residue as the hinge-point within the helix. Based-on the hinge-point, it generates a pre-proline axis and a post-proline axis, which define the kink angle. Although this method is commonly used, it can only account for proline kinks, which leaves 65-80 percent of helical kinks undefined. Another popular method is KinkFinder [66], which fits a cylinder over six residues of a helix. Using the location of each cylinder, the method finds adjacent cylinder segments. To find the angle between the two cylinders, the axis of each connecting segment is calculated. Other methods, MC-Helan [57] and Helanal [67], are available and use similar strategies of defining two axes from the helix and finding the angle between them.

Studies have applied popular methods [68,69] to predict helical kinks. MD has been applied to predict helical kinks in transmembrane proteins. A study by Hall et al. demonstrates the use of MD to reproduce membrane protein helical kinks based on the local sequence alone [60]. Out of 405 helices tested, 79% of proline kinks and 59% of vestigial proline kinks were detected. Of the remaining kinks, only 18% were reproduceable through MD. This study concluded that helical kinking likely depends on the topology of the entire protein, along with the supporting interactions between the helix and the protein with the lipid bilayer.

Machine learning techniques have also been used for the prediction of helical kinks [68,69]. Such methods have commonly relied on sequence as an input for the predictions. A study by B. Kneissl et al. utilized a support vector machine and sequence input to predict transmembrane helical kinks [68]. The data set included 132 membrane proteins, with 1,014 helices. The method resulted in 80% prediction accuracy for non-kinked helices, when training on straight helices and all kinked variations. However, the method was only able to predict non-proline kinks with 55% accuracy, when trained on straight and non-proline kinked helices. These results are an advancement in comparison to past studies, but still show room for improvement. The outcome demonstrates the need for a method capable of predicting helical kinks with higher accuracy.

3.1.4 Refinement methods are not proven to sample kinked helices in membrane proteins

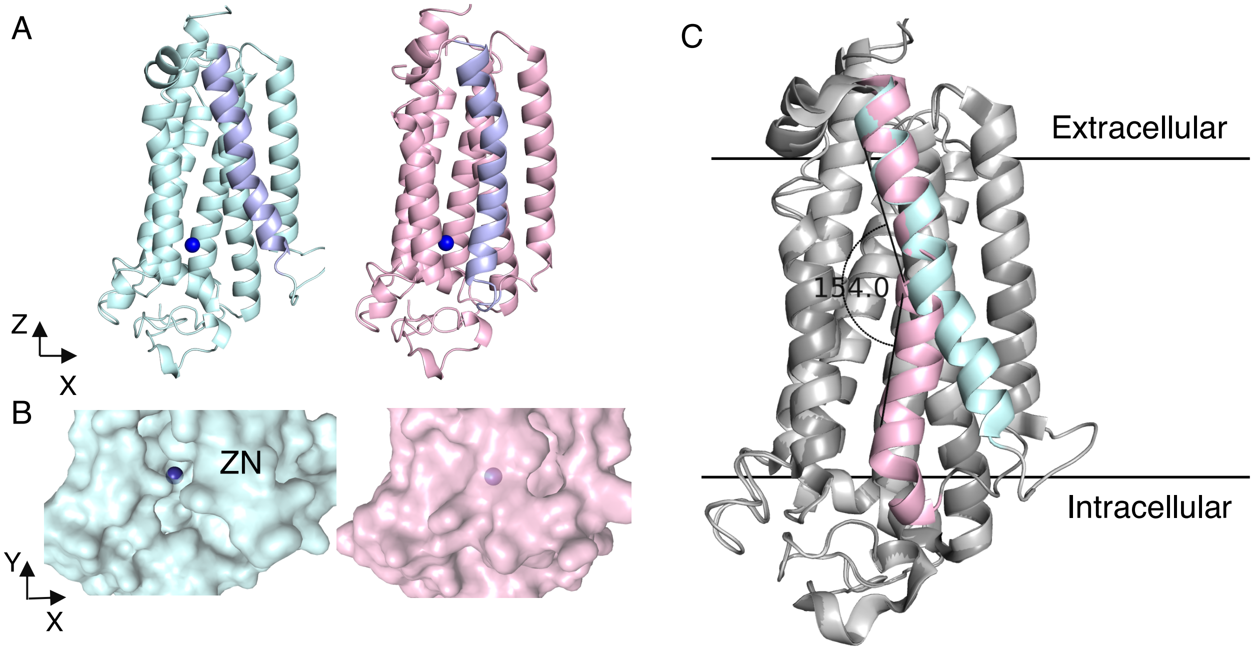
To the best of my knowledge, no refinement method has been developed to account for sampling of kinked helices within membrane proteins. Within Rosetta, current methods do not capture the possible conformations that membrane proteins may have as a result of flexibility. Rosetta's Relax Protocol [70] is a refinement method that locally samples the conformational space through packing of side chains and minimization of torsional degrees of freedom. This protocol has not been proven to sample between the kinked and non-kinked conformational changes present in nature, demonstrating the need for further exploration into current refinement methods or a new technique capable of capturing both possible protein states.

The goal of my second project is to identify features that are predictive of membrane helical kinks, and to use this information to develop a refinement method capable of sampling these conformational changes effectively. Using membrane proteins that exhibit kinked and non-kinked helix conformations as testcases, I demonstrate that current Rosetta refinement protocols are not successful in sampling between the multi-conformational helix states. Furthermore, I examined past and present Rosetta score functions to determine if there was a biased prediction for straight or kinked helices. To gain a better understanding of these helical kinks and their features, I developed a classification method capable of identifying them. Using this classification method, I culled a new fragment library consisting of kinked helical segments. I aim to use this new library to sample kinked conformations within membrane proteins. This would enable accurate prediction of multi-conformational membrane proteins resulting from kinked helices.

3.2 Methods part 1

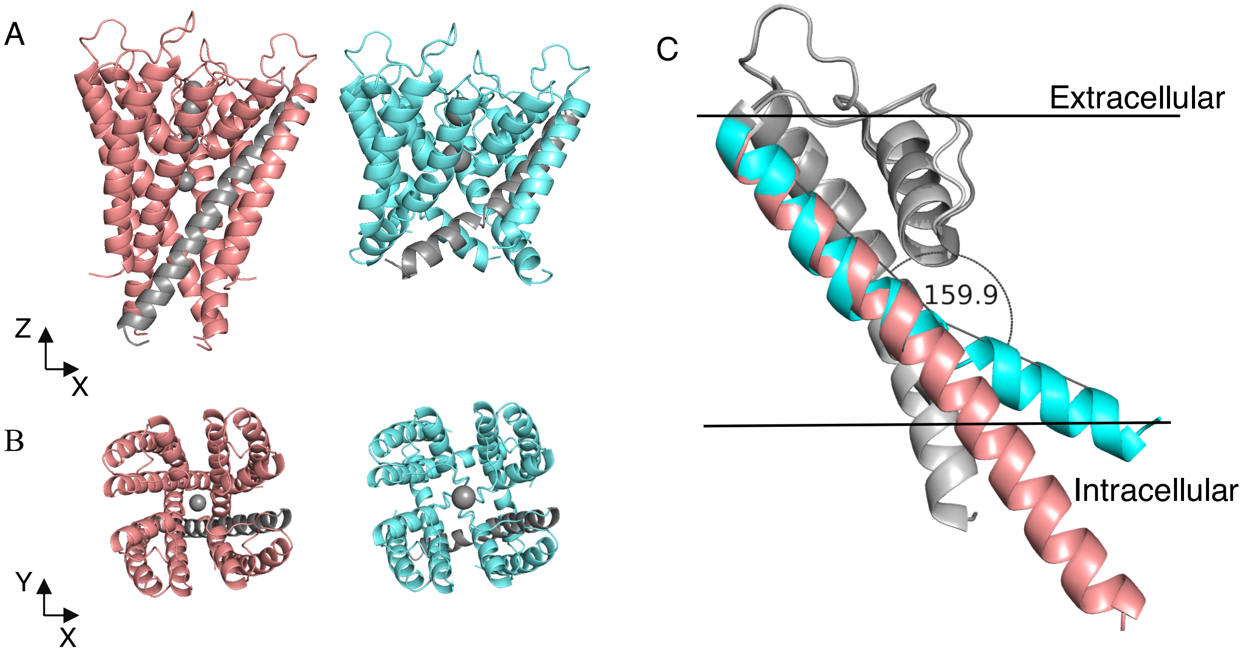
3.2.1 Testing Rosetta refinement on systems with helical hinge dynamics

To examine whether refinement methods within Rosetta were able to sample between kinked and straight helices found in nature, I ran Rosetta's Relax protocol on three different case study proteins. Each protein was chosen based on the following criteria: (i) Protein flexibility resulting in multiple conformations, (ii) availability of one crystal structure with a straight helix and one with a kinked helix, (iii) other than the kinked helix, there is no major change in the conformation of each structure. The three proteins I found that meet these conditions are human adiponectin receptor 1 (**Figure 12**), KcsA potassium channel (**Figure 13**) and human platelet-activating factor receptor (**Figure 14**).



5lxg (open) 3wxv (closed) Aligned structures showing changes in helix

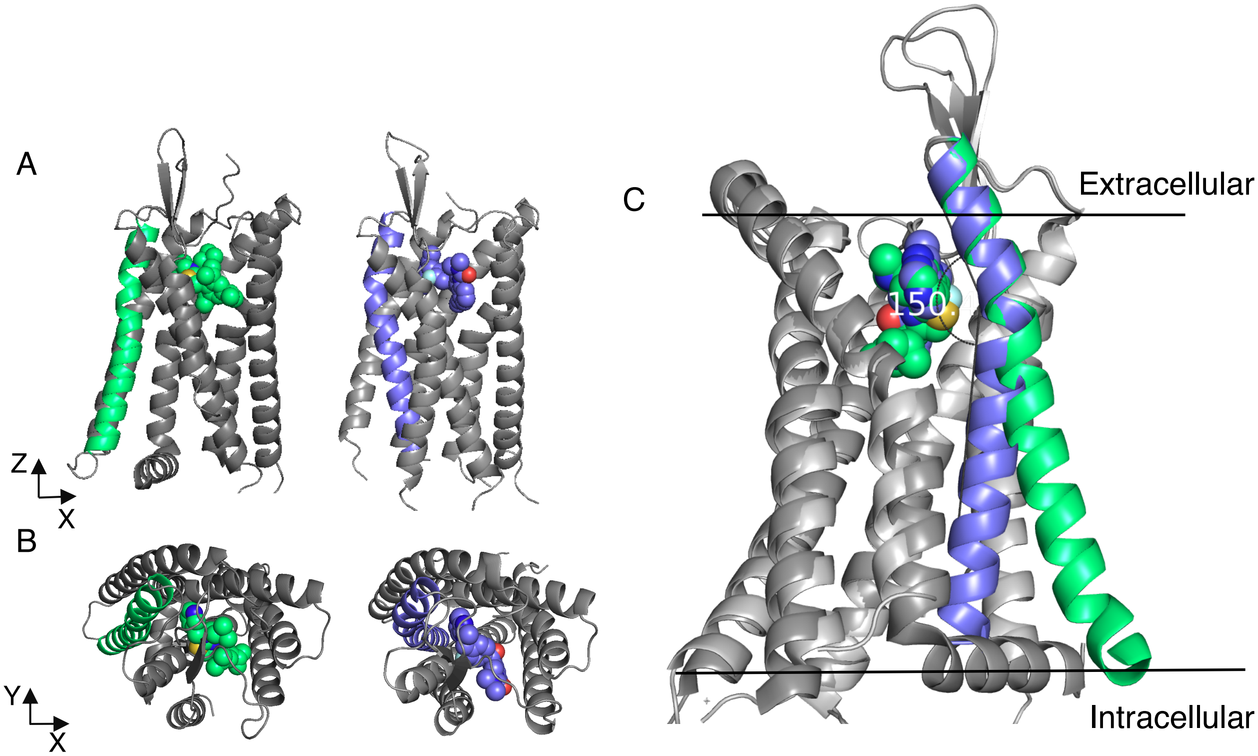
Figure 12: The structural conformation of both the kinked and straight conformation of human adiponectin receptor 1 is shown in panel A, with the helical portion highlighted in purple, the straight protein conformation in teal and the kinked protein conformation in pink. Panel B illustrates the closed and open conformational structures from a top down view, with zinc shown in blue. Panel C shows the alignment of both conformational variants with the kinked helix shown in pink and the straight helix shown in teal.



1r3j (closed variation) 3f5w (Open variation)

Aligned structures showing changes in helix

Figure 13: The structural conformation of both the kinked and straight conformation of KcsA potassium channel is shown in panel A, with the helical portion highlighted in gray, the straight protein conformation in teal and the kinked protein conformation in pink. Panel B illustrates the closed and open conformational structures from a top down view, with ions shown in gray. Panel C shows the alignment of both conformational variants for a single subunit, with the kinked helix shown in teal and the straight helix shown in pink.



5zkp (Non-Kinked) 5zkq (Kinked) Aligned structures showing changes in helix

Figure 14: : The structural conformation of both the kinked and straight conformation of human platelet activating factor receptor is shown in panel A, with the straight helical portion highlighted in green and the straight helical portion shown in purple. Panel B illustrates the closed and open conformational structures from a top down view, with supporting cofactors shown in sphere representation. Panel C shows the alignment of both conformational variants, with the kinked helix shown in purple and the straight helix shown in green.

With these crystal structures, I applied Rosetta's Relax protocol to each conformational variant of each protein. Using default settings (see Appendix) I generated 1000 decoys for each protein variant. Through analysis of the total score calculated for each decoy model, and the RMSD of the kinked helix within each decoy to the kinked helix in the native structure, I determined whether Relax was capable of sampling between the two conformational states. This allowed me to determine the need for a method with the ability to sample kinked conformational forms within membrane proteins.

3.2.2 Testing Rosetta score functions for biasing kinked or straight helical conformations

To investigate if Rosetta's past and present score functions favored straight or kinked helical conformations, I applied Rosetta's Relax protocol to a second set of proteins. Test case proteins were chosen from a diverse set of protein classifications (voltage gated channel, rhomboid, cation channel, and signaling receptor) and each protein contained at least one helical kink. Because this set does not require crystal structures of both conformational states, I found a larger diversity of proteins. The test case protein identifications chosen were PDBs 2irv, 4h33, 5h35, and 5tud (**Figure 15**).

3.2.3 Manual straightening of kinked helices

The following steps show how I used PyRosetta [71] to straighten a kinked helix within each testcase protein:

Step 1. As a result of downstream residue effects caused by changes in the helix residue angles, it was necessary to alter the fold tree [72]. The fold tree denotes how the residues within each structure are connected together and how movements will propagate throughout the protein structure. By adding a cut-point within the loop region, downstream of the helix, I altered the fold tree to prevent residue alterations past the loop.

Step 2. Set ideal helix ϕ and ψ angles of residues involved within the helix.

Step3. To correct the loop after this change, I applied PyRosetta's loop modeling movers to connect and remodel the broken region. This resulted in a new conformational variation for each test case protein (**Figure 15**).

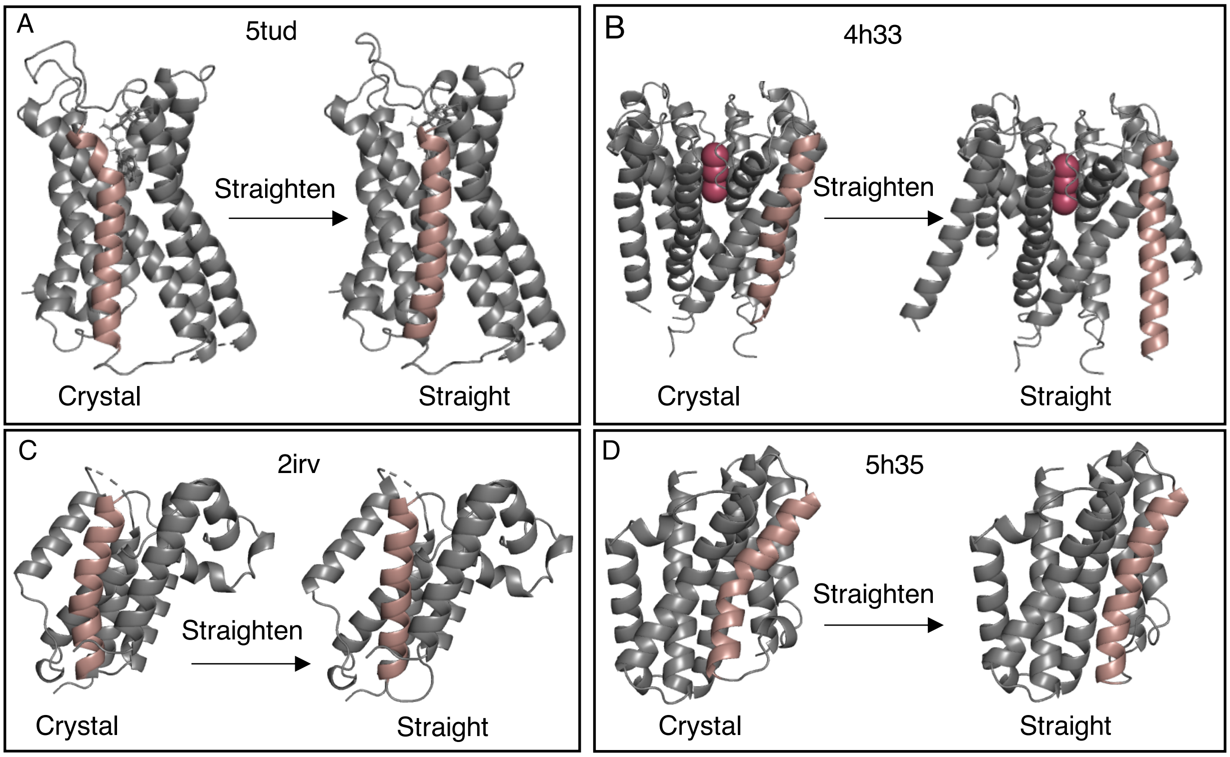


Figure 15: Kinked and straightened helical conformations of four test proteins. (A) 5tud, a GPCR protein, (B) 4h33 a voltage gated potassium channel, (C) 2irv, a rhomboid protease protein, and (D) 5h35, a TRIC trimer cation channel.

Each protein now had a natural kinked variant and a manually altered variant with a straight helix and a remodeled loop region. I applied Rosetta's Relax protocol to generate 500 decoys for each conformational variant of the four test case structures. I repeated this step with three recent score functions, Ref2015, MP07, and MP12. Using this strategy, I could determine if each score function favored either the kinked or straight conformational variant.

3.3 Results Part 1

3.3.1 Need for sampling between kinked and non-kinked helical conformational forms in membrane proteins

Using the output from Rosetta’s Relax protocol for each conformational variation of each of the three testcase structures, I investigated whether sampling between multiple conformations was possible. To evaluate the results of these trajectories, I utilized three different plots: (i) Density vs. RMSD from the kinked ensemble to the non-kinked wildtype or kinked wildtype, and the density vs. RMSD of the non-kinked ensemble to the kinked wildtype or the non-kinked wildtype (**Figure 16**), (ii) RMSD to non-kinked native vs. RMSD to kinked native for each ensemble generated (**Figure 17**), and (iii) total score vs. kinked native for each ensemble (**Figure 18**). I saw several trends through these results. For each structure shown in Figure 16, there was a single peak for each density line, with some peaks being broader than others. Furthermore, for structures 1-3, shown in Figures 17-18, I saw a single region of points and there was no clustering of the data into multiple regions. The trends determined through this data suggest that Relax does not sample between the multi-conformational states.

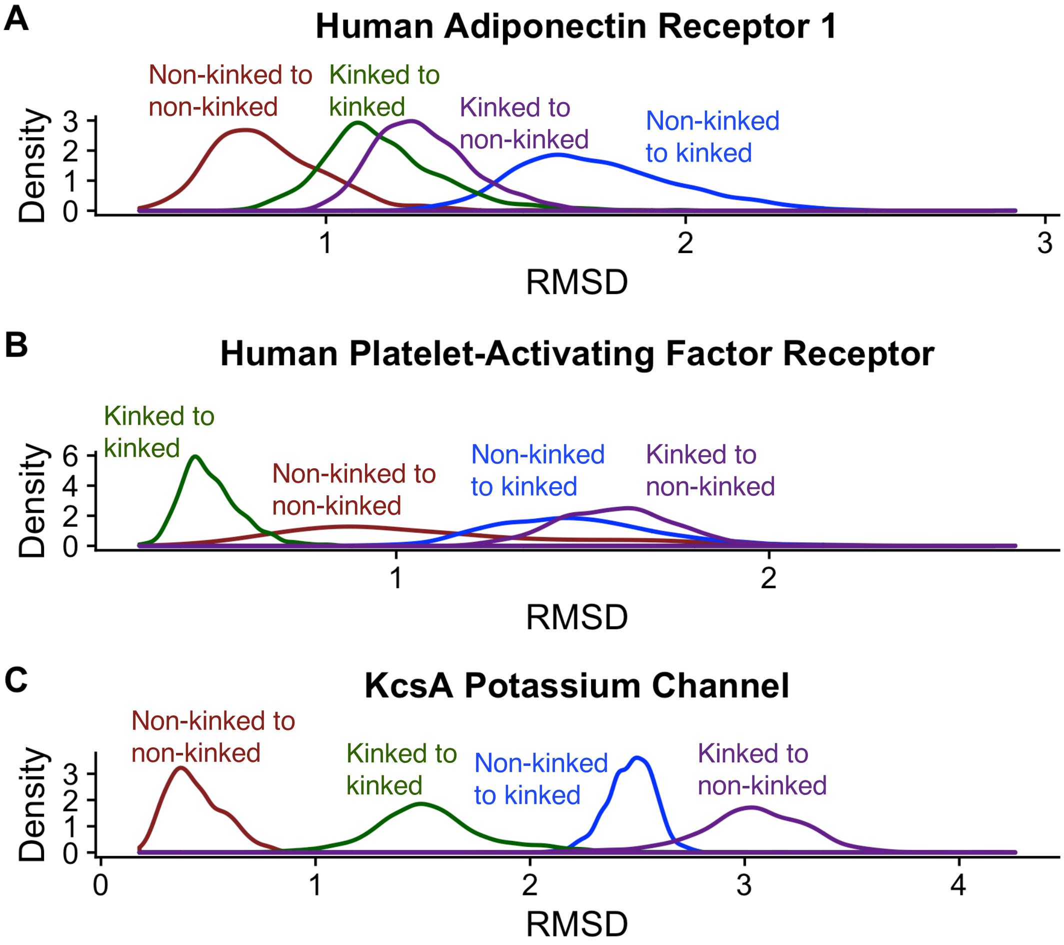


Figure 16: Panels A-C show plots of the density of each ensemble vs. RMSD to each native state. The line shown in red is the density of the non-kinked ensemble vs. the RMSD to the non-kinked native. The line shown in blue is the density of the non-kinked ensemble vs. the RMSD to the kinked native. The line shown in green is the density of the kinked ensemble vs. the RMSD to the kinked native. The line shown in purple is the density of the kinked ensemble vs. the RMSD to the non-kinked native.

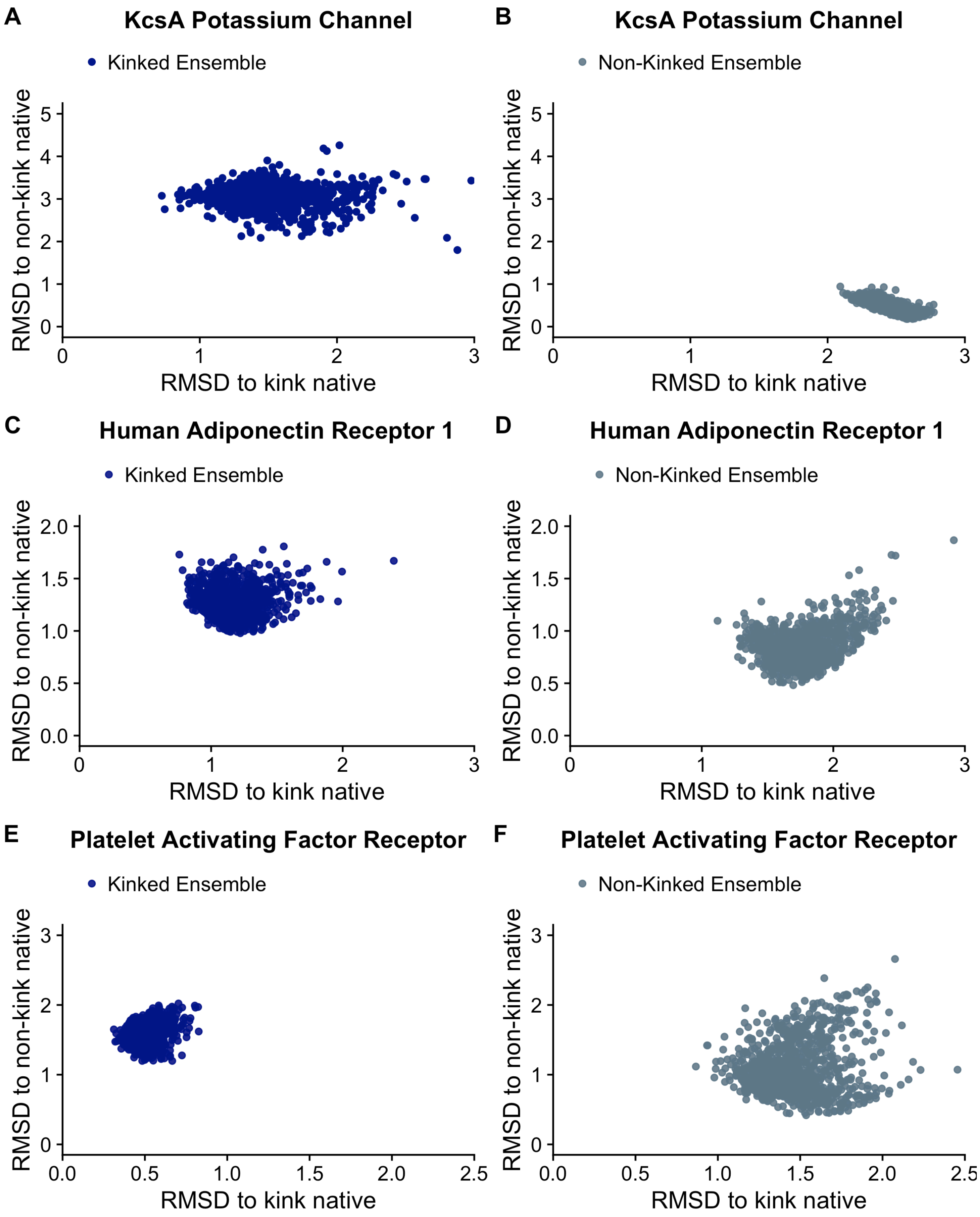


Figure 17: Panels A-C illustrate the plot of the RMSD of each ensemble vs. RMSD to the kink native for each protein. For each protein, the kinked ensemble is shown in blue and the non-kinked ensemble is shown in gray.

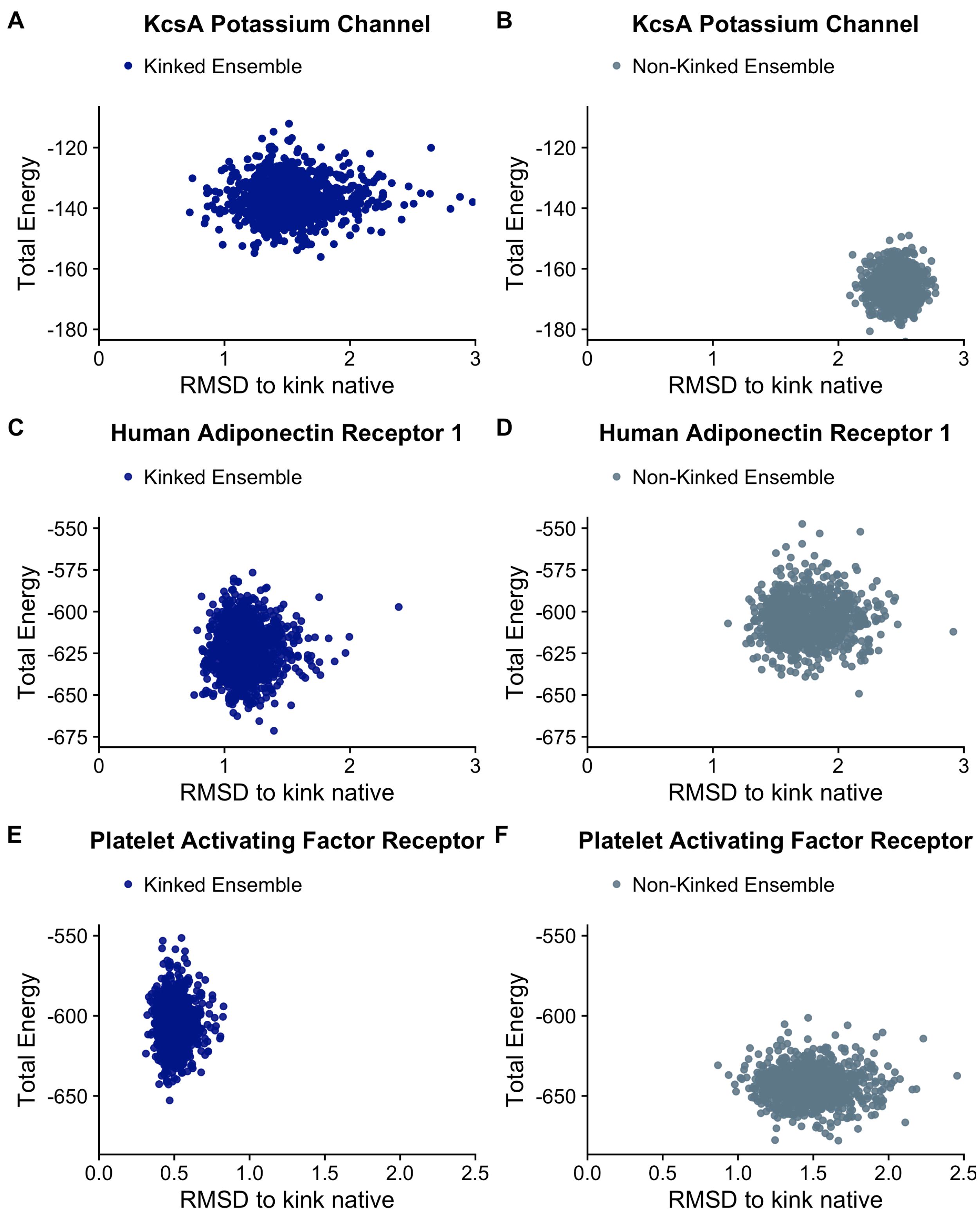


Figure 18: Panels A-C illustrate the plot of the total energy vs. RMSD to the kink native. For each protein, the kinked ensemble is shown in blue and the non-kinked ensemble is shown in gray.

3.3.2 Do score functions favor kinked or straight helices?

To investigate whether Rosetta refinement favored straight helices over kinked helices, I applied Rosetta's Relax protocol to a second test case set of proteins. Each protein had both an altered conformational variation of an unnatural straight helix and a variation with a natural kinked helix. Using the 500 decoys generated, I examined the distribution of the scores due to the total-score and due to each hydrogen bonding term. For each structural ensemble, I compared the kinked ensemble vs. the straight ensemble. Hydrogen bonding terms present in each score function of Rosetta are backbone side-chain hydrogen bonding (hbond_bb_sc), long-range backbone hydrogen bonding (hbond_lr_bb), side-chain-side-chain hydrogen bonding (hbond_sc) and short-range backbone hydrogen bonding (hbond_sr_bb). Because the score of helices rely heavily on hydrogen bonding, these terms can be used to determine if a specific conformation is favored. The density plots of each structure for each energy function are shown in Figures 19-21.

For the MP07 (**Figure 19**) score function, the kinked conformation total-score term resulted in lower scores for three of the four proteins (2irv, 4h33, 5h35). For protein 5tud, the straight conformation scored lower in the total-score by 10 Rosetta energy units (REU). The long-range backbone hydrogen bonding term for this protein is the only hydrogen bonding term that scored the straight conformation lower, with a difference of 2 REU. This separation in score, due to hydrogen bonding, is not enough to explain the total-score preference toward a straight helix in this protein. It is likely that other global effects of straightening the helix within this protein cause this preference.

For all proteins scored by MP07, the side-chain-side-chain hydrogen bonding term resulted in lower energy scores for kinked conformations and higher scores for straight conformations. In the remaining three hydrogen bonding score terms I observed mixed results where kinked conformations scored higher in some proteins and lower in others. Therefore, it is not likely that the hydrogen bonding terms are affecting the total-score to prefer kinked or straight conformations.

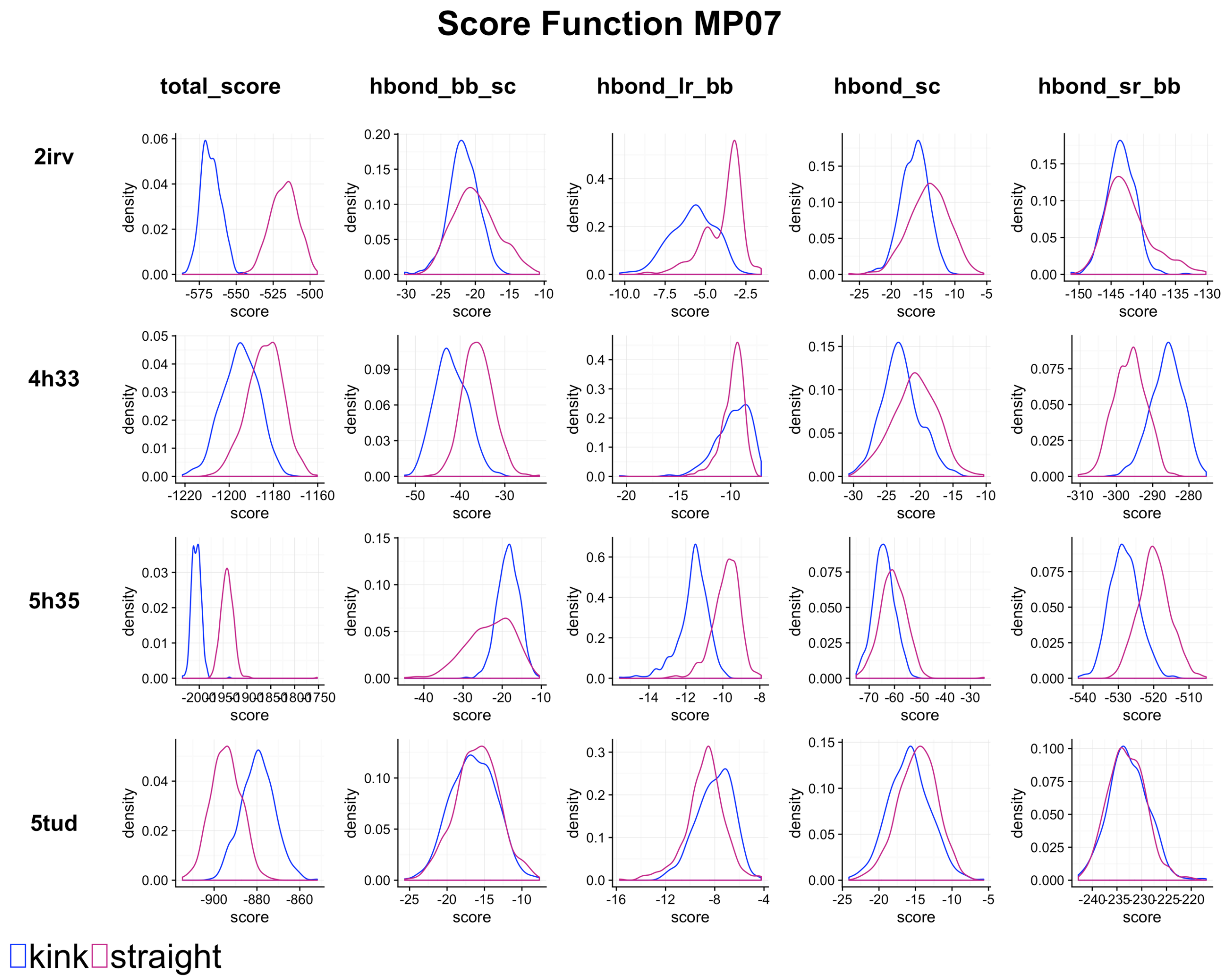


Figure 19: Plots for the MP07 score function for the four hydrogen bonding terms and the total-score within Rosetta. The pink line shows the kinked ensemble and the teal line shows the straight ensemble for each protein in the testcase set.

For the MP12 (**Figure 20**) score function, the kinked conformation total-score term resulted in lower scores for each protein. In all proteins, the hydrogen bond side-chain-side-chain and long-range backbone terms scored the kinked conformation lower overall. However, I saw mixed results for the short-range backbone hydrogen bonding term, where half the proteins (5tud, 4h33) scored the straight conformation lower. Thus, the MP12 score function works as expected, by scoring kinked conformations lower than straight conformations. The kinked conformation is in its natural form, which is expected to be at its lowest energy state. In these cases, the hydrogen bonding terms do affect the total-score, which resulted in a lower score for kinked conformations.

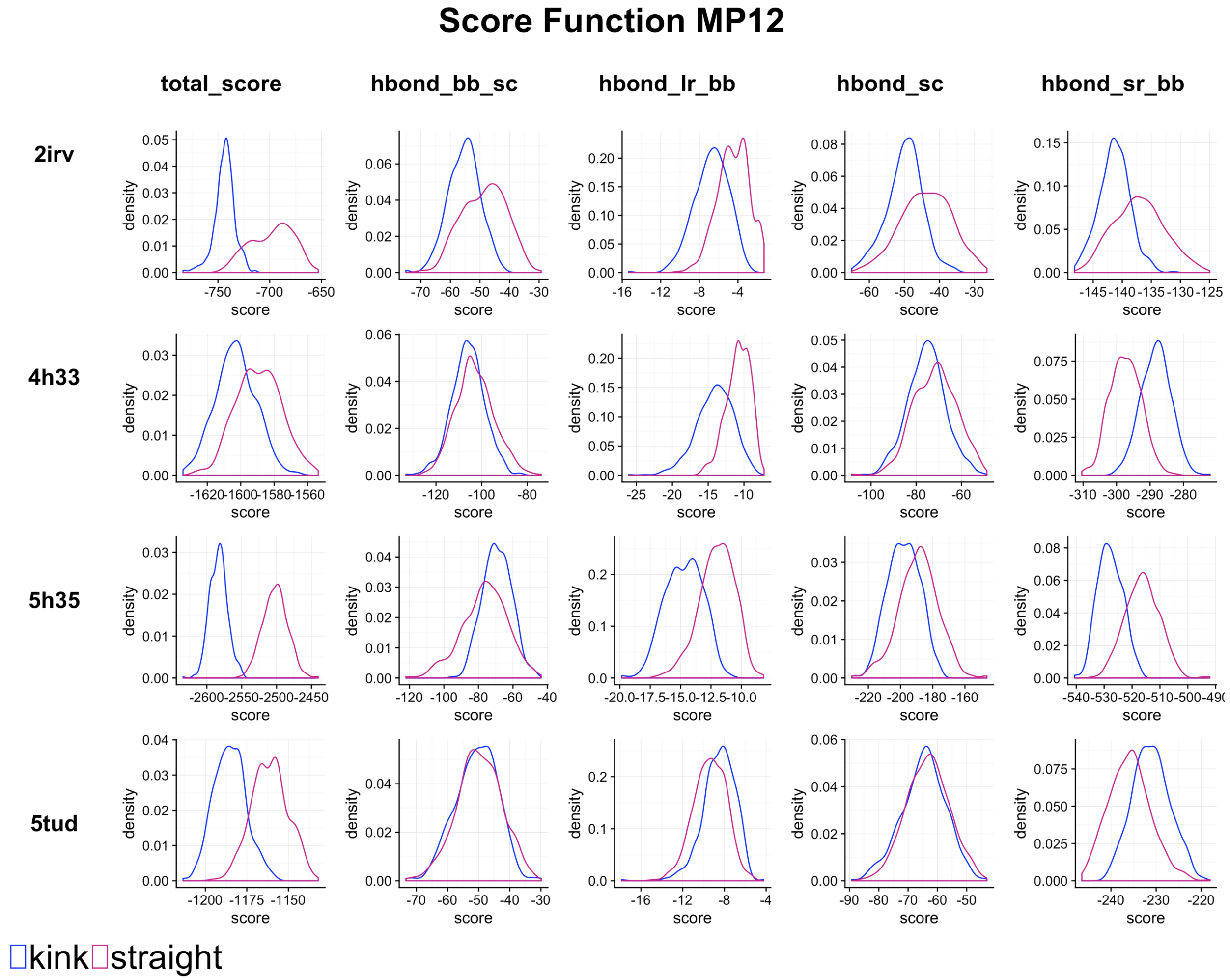


Figure 20: Plots for the MP12 score function for the four hydrogen bonding terms and the total score within Rosetta. The pink line shows the kinked ensemble and the teal line shows the straight ensemble for each protein in the testcase set.

For the Ref2015 score function (**Figure 21**), the straightened conformation total-score term resulted in lower scores for two of the proteins (2irv, 5tud). I observed that the hydrogen bonding terms for protein 2irv showed comparable or lower scores for the kinked conformation, as expected. This does not explain why the total score is lower for the straight conformation. It is probable that global changes throughout the protein resulted in favoring of the straight conformation. For protein 5tud, I visualized that the hydrogen bonding terms were comparable or scored lower for the straightened conformation. Thus, the hydrogen bonding terms directly affected the total-score term for this protein, resulting in its favoring of a straightened conformation. For protein 5h35, the hydrogen bonding terms overall scored the kinked conformation lower, as expected. The hydrogen bonding terms did have an effect on the total-score for 5h35, leading to the preference of a kinked conformation. For protein 4h33, the hydrogen bonding terms scored the straight conformation lower overall. This is not what I expected, and it is likely that global effects contributed to the lower total-score of the kinked conformation.

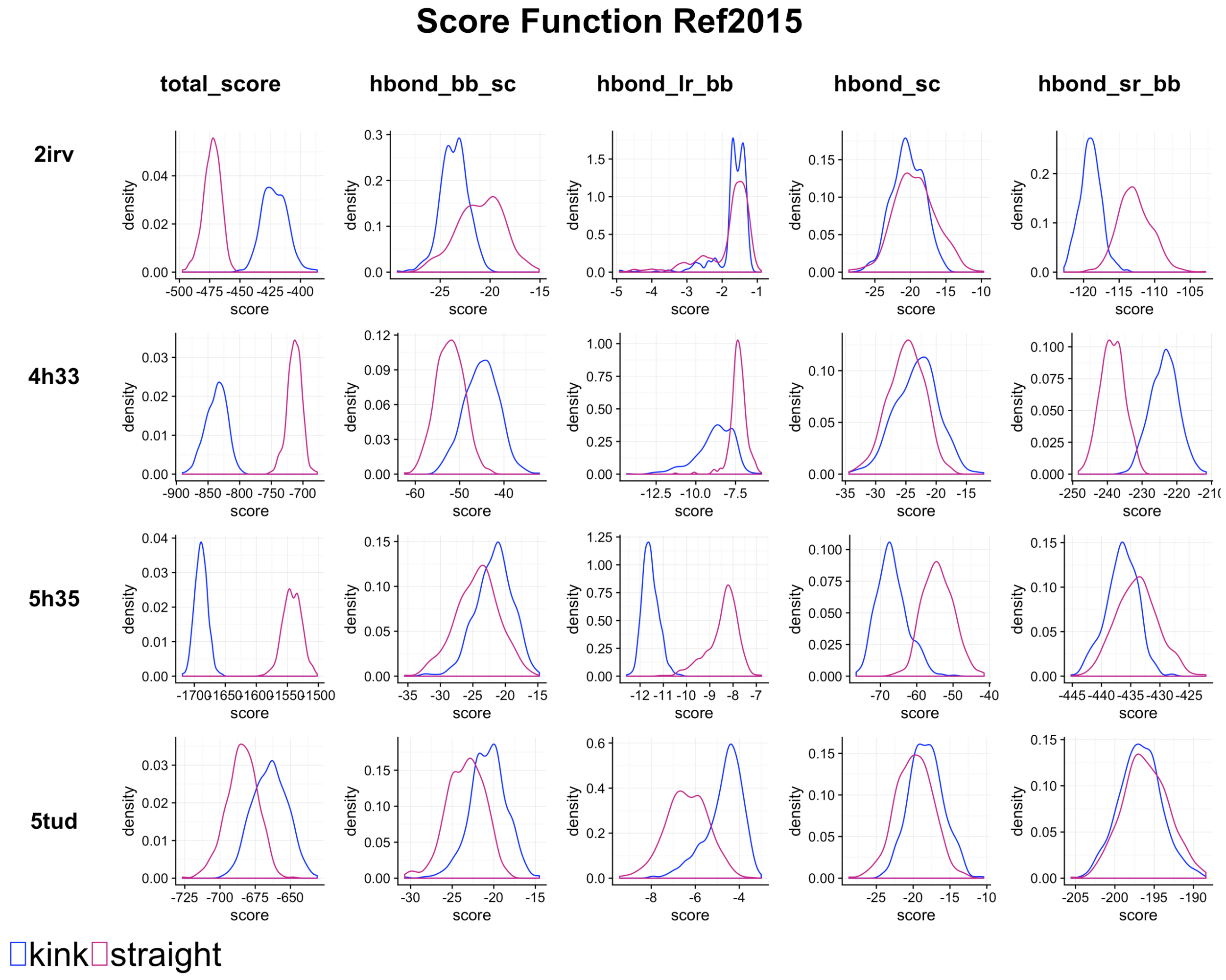


Figure 21: Plots for the Ref2015 score function for the four hydrogen bonding terms and total score within Rosetta. The pink line shows the kinked ensemble and the teal line shows the straight ensemble for each protein in the testcase set.

3.4 Discussion Part 1

3.4.1 Need for sampling between kinked and non-kinked helical conformational forms in membrane proteins

For the plot of RMSD to non-kinked native vs. RMSD to kinked native, I observed a single cluster of points. Similarly with the plot of total score vs. RMSD to kinked-native (**Figure 18**), I detected a single cluster of data-points. If Relax was able to sample both conformations, I would expect to see clustering of points into two different regions for each ensemble. I would expect this because the kinked and straight conformations would likely be at distinctive low energy wells within the protein energy landscape. Because Rosetta protocols are MC-based, results are driven towards finding structures within low energy wells. Thus, I would expect structures to fall into one of these low energy wells, generating two divergent regions of points. Since these results only demonstrated a single group of points, the results indicate no multi-conformational sampling.

Furthermore, analysis of the density of each ensemble vs. RMSD to both native variations demonstrated little overlap. If there was sampling between both conformations, I would have expected to see the density plots having a large area of overlap between the two ensembles. Because there is little to no overlap in the density areas, it suggests that there is no multi-conformational sampling. As a result of the analysis, I concluded that Rosetta's Relax protocol does not sample between conformational variants. Relax only allows for a small amount of movement within the backbone, but generating a kinked helix requires large backbone movements.

3.4.2 Do score functions favor kinked or straight helices?

Using density plots for each score function, I compared the kinked and straight conformational scores for the total-score and each hydrogen bonding term. The outcome established that each score function had different results. For score functions MP07 and Ref2015, different results were seen for different targets. For score function MP12, I observed expected results, where the kinked conformation scored lower than the straight conformation. It is possible that the mixed results stem from the location of each helix within the structure. The variation of structures result in some helices buried deep within the protein, making many contacts, and others located near the outward portion of the protein, making few contacts. Furthermore, the straight conformations were manually generated, putting this conformation at a disadvantage to be favored initially. This disadvantage is hard to account for when examining the results, but it must be mentioned that this likely plays a role in the outcome.

3.5 Methods Part 2

3.5.1 Development of a classification method for kinked helices

First, to determine whether kinked helical conformations are present in a structure, a method must label each helix as kinked or straight. I developed a method to determine the degree to which a helix is kinked. As input information, I used data present in Rosetta's culled PDB library. This consisted of each residue's C_α coordinates, the secondary structure, the PDB identification, the ϕ angle and the ψ angle of each residue. My method worked by determining the hinge-point within the helix. To locate the hinge-point I searched for the residue resulting in the greatest variation from ideal helical ϕ and ψ angles. To determine the angle, I applied principal component analysis (PCA) to define two vectors, v_1 and v_2. v_1 represented the direction of the span of residues from the beginning of the helix to the hinge-point, and v_2 represented the span of residues from the hinge-point to the end of the helix. The angle between the two vectors was determined as: $\theta = \cos^{-1}\frac{v_1 \cdot v_2}{|v_1| \cdot |v_2|}$, where θ is the helix kink angle.

Furthermore, I was able to use this classification method (**Figure 22**) to generate a new library consisting of only kinked helices.

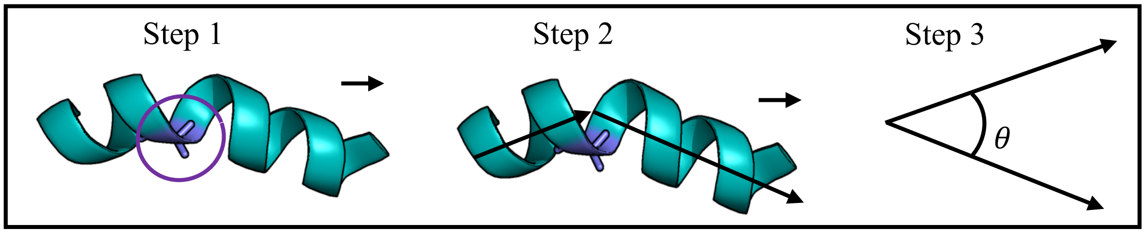


Figure 22: Steps for the helical classification method. Step 1 locate of the hinge-point, step2 use PCA to gain two vectors representing the helix, and step 3 use of the two vectors to determine the helix kink angle.

3.5.2 Generation of a kinked helical fragment library of different lengths

Applying my classification method, I generated a library consisting of kinked helical fragments. To accomplish this, I first needed to search from an all-inclusive library culled from the entire PDB. I used Rosetta's raw PDB data file, consisting of all secondary structures from all proteins within the PDB. Sifting through this library, I first pulled a segment of 24 residues for examination. To confirm that the segment consisted only of a helix, I started from each end of the segment, moving inward and trimming all residues without helical secondary structure. With the remaining segment, only two residues with secondary structure other than helical were allowed. If the segment did not meet these requirements, it was discarded and a new segment of 24 residues was pulled sequentially from the library. Once each segment was prepared, I was able to apply my classification method.

To prepare a segment to input into the classification method, not only did it need to be trimmed and contain a limited amount of non-helical secondary structure residues, but it also had to meet a length criteria. For each vector obtained from PCA to result in an accurate direction, it is necessary that each vector be based off of no less than 6 residues. After a segment matching this criteria is discovered, the classification method is applied to it and an angle is calculated. With the helical kink fragments determined, I was able collect their relevant information to generate a new library.

With the kinked fragments identified, I applied a degree cutoff which only accepted helices containing kinks of 20° or more for collection. Using this cutoff, I pulled the information of each accepted segment and input it into a new library. With this algorithm (**Figure 23**), I was able to set a required fragment length to pull. Starting from the hinge-point and moving outward in each direction, I pulled lengths of 5-mers, 7-mers and 9-mers. By sequentially looping through Rosetta's entire PDB library, I collected all kinked segments into a new library labeled by fragment size.

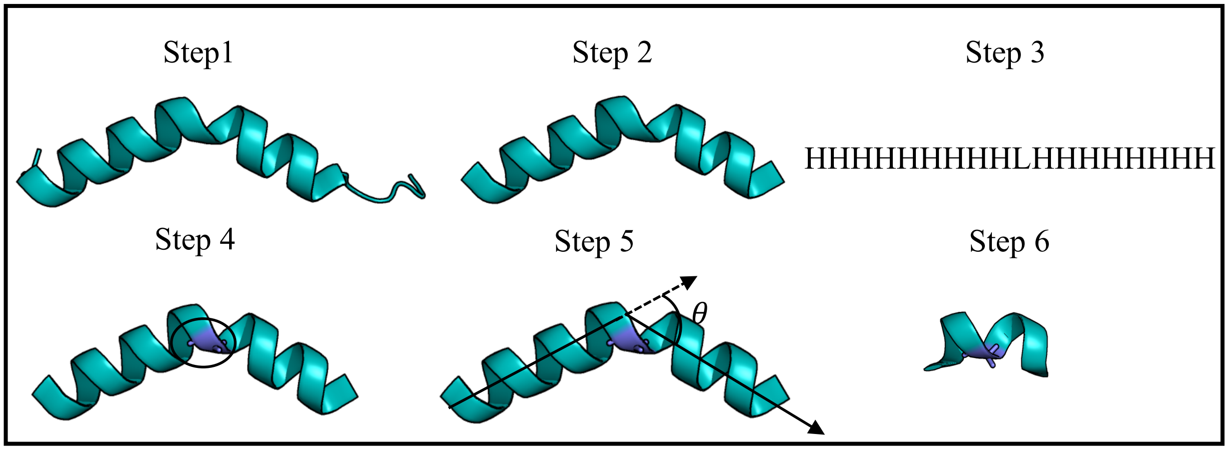


Figure 23: The steps involved in adding kinked fragments to the library. Step 1, pull a fragment of 24 residues from the full PDB library. Step 2, eliminate ends that are not helical secondary structure. Step 3, count secondary structure and allow only two non-helical residues. Step 4-5, utilize classification method to determine the angle of the kinked helix. Step 6, remove edges of the segment to reach the desired length.

3.6 Results Part 2

3.6.1 A new classification method for kinked helices

To determine the degree to which a helical segment is kinked, I developed a kink measurement method using PCA. This kink measurement method utilized vectors, representative of the backbone coordinates of helical segments, to determine the kinked angle. This method is different from other methods because it takes inputs directly from Rosetta's raw PDB data file. I used PyMol to manually examine the helix angle for 50 structures and compared the results to the output from the kink classification method. The PyMol measurements matched closely to kink classification measurements.

3.6.2 Examination of a new kinked helix fragment library

To generate a fragment library of helical kinked segments, I wrote an algorithm that sifted through a culled library consisting of all protein residues. The new library I generated contained helical segments with a kinked measurement of 20° or greater. The kinked library consists of 1500 helical fragments and demonstrates many different degrees of kinking and hydrogen bonding patterns. I calculated the number of hydrogen bonds made throughout the kinked 5-mer library per kink angle (**Figure 24**). Higher hydrogen bond numbers occur at lower degree angles. As seen in Figure 24, when there are the maximum number of 10 backbone hydrogen bonds, the degree angle is approximately 32°. Whereas, when there are no backbone hydrogen bonds, the average angle is approximately 112°.

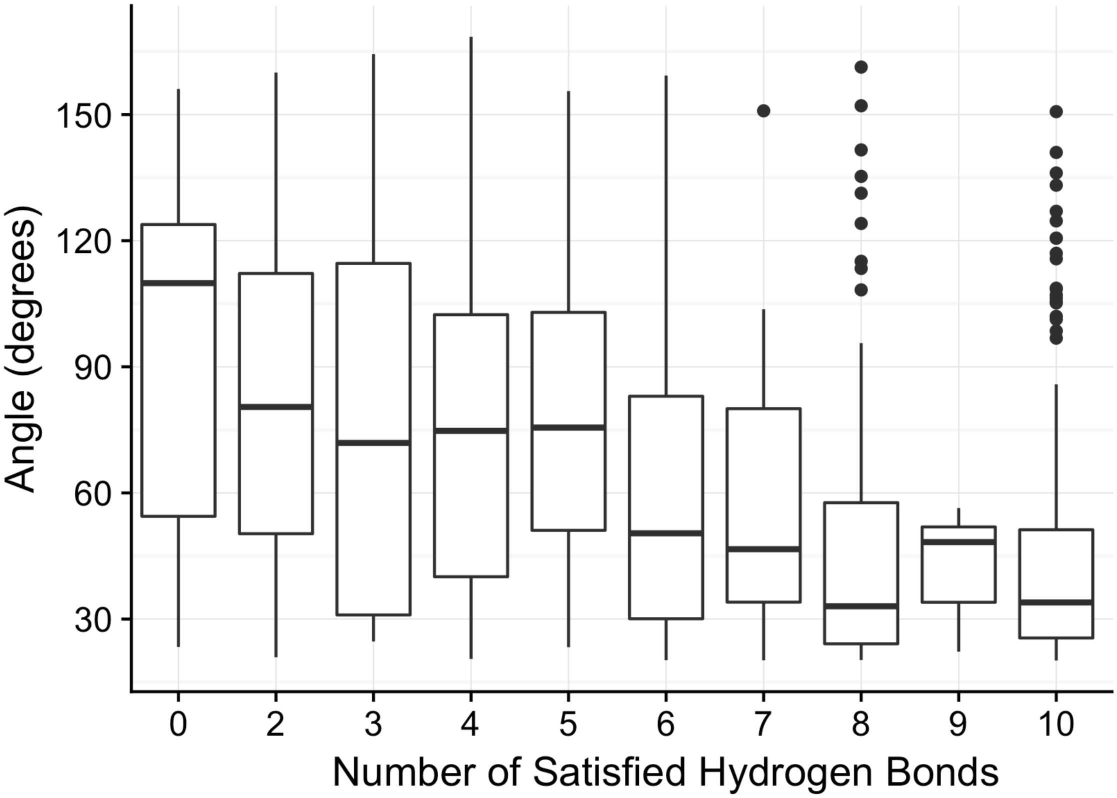


Figure 24: Number of hydrogen bonds changes as the angle of helical kinks increases from 20 to 160°. The total number of hydrogen backbone bonds capable of being satisfied is 10 and the minimum number is 0.

Within the generated library, I discovered five different types of helical kinks (**Figure 25**) resulting from the number of hydrogen bonds made within the helix. These types of kinks are: normal curved kink, normal tight kink, elongated kink, U-turn kink and U-turn gap kink. Furthermore, to analyze the distribution of angles throughout the kinked library, I generated a density vs. angle plot that is shown in Figure 26. The results from this plot showed three peak regions with the highest density falling at 33°. As the angle increased the density decreased, with the other peak points seen at 70 and 110°.

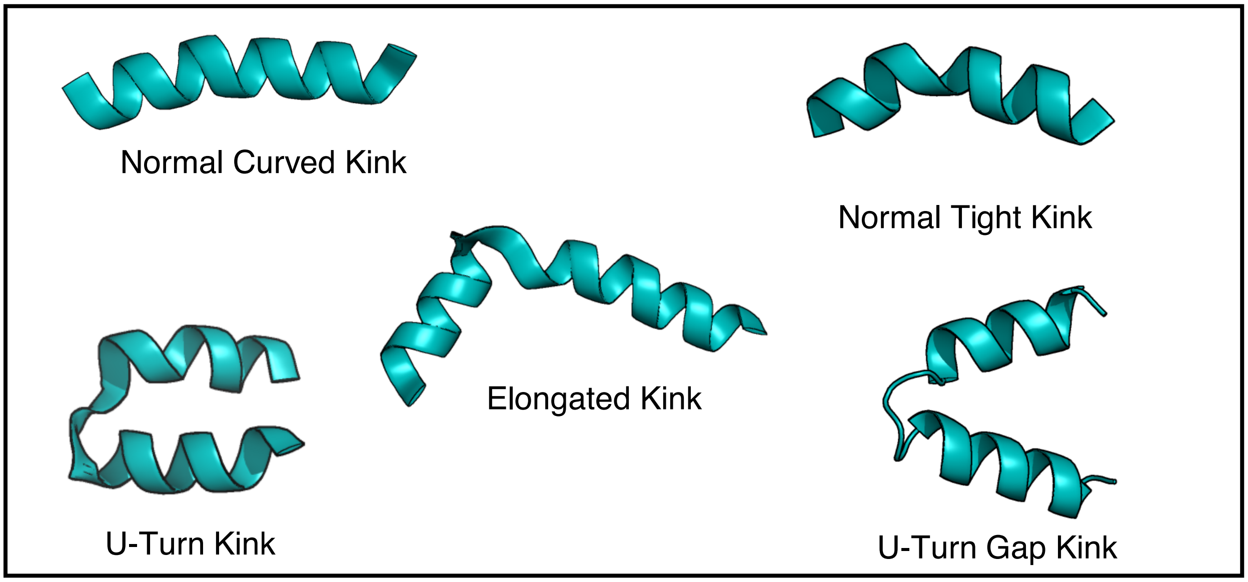


Figure 25: Types of helical kinks found within the kink fragment library.

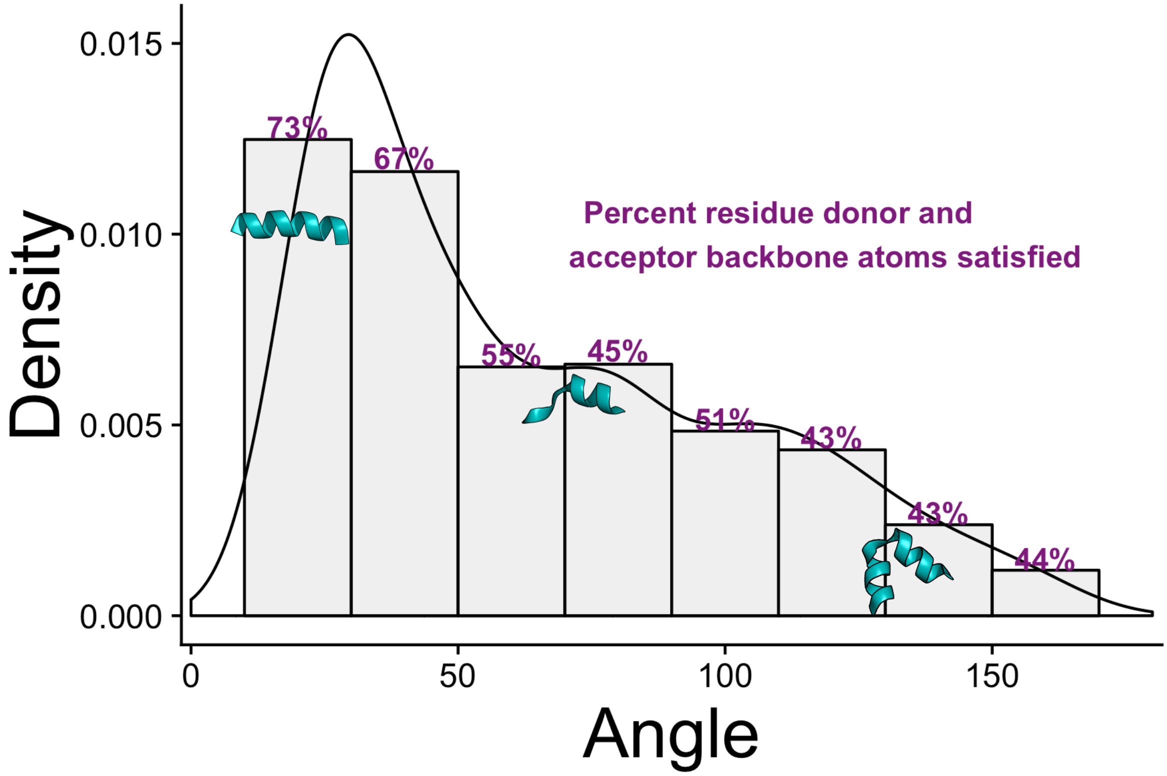


Figure 26: Density of angles for the kinked fragment library. The percent of possible backbone hydrogen bonds satisfied is shown in purple and the general shape of helices at three angles across the range of angles is shown in teal.

3.7 Discussion Part 2

3.7.1 A novel classification method for kinked helices

There have been several previous attempts to identify and measure curving and kinking of helices. Each approach is unique and defines a kink slightly different based on the calculation method and the minimum angle described as a kink. My method is novel because it utilizes information already collected within a data file. As a result, I do not have to search each structural PDB file to determine kinked segments within the entire Protein Data Bank. Instead, I can sift through a data file to determine and classify all kinked helices. This method allows for analysis and classification of large numbers of helices more efficiently. This is something that has not been done before and has a unique application towards generating a new library consisting only of kinked helical information.

3.7.2 A fragment library of kinked helices

Currently, there is not a library of data that contains helical kinked fragments. This is important for ease of analysis of helical features, and to use in fragment insertion methods. Fragment insertion is commonly used in de-novo protein prediction methods, where it can sample conformations efficiently. This new library can be incorporated into various protein prediction, design and refinement protocols.

3.7.3 Future steps

In the future, a next step is to develop a refinement algorithm which utilizes this library to insert kinked fragments into helical membrane proteins, with the goal of sampling possible kinked conformations. This type of method has not yet been attempted before and has the possibility of improving membrane protein refinement results and allowing for sampling of other low energy conformational states possible but not seen in current refinement methods. Furthermore, other applications of this library include its involvement in protein design, where a protein could be designed to have a helical kink located within a specific region. As a result this could generate novel proteins with new functions.

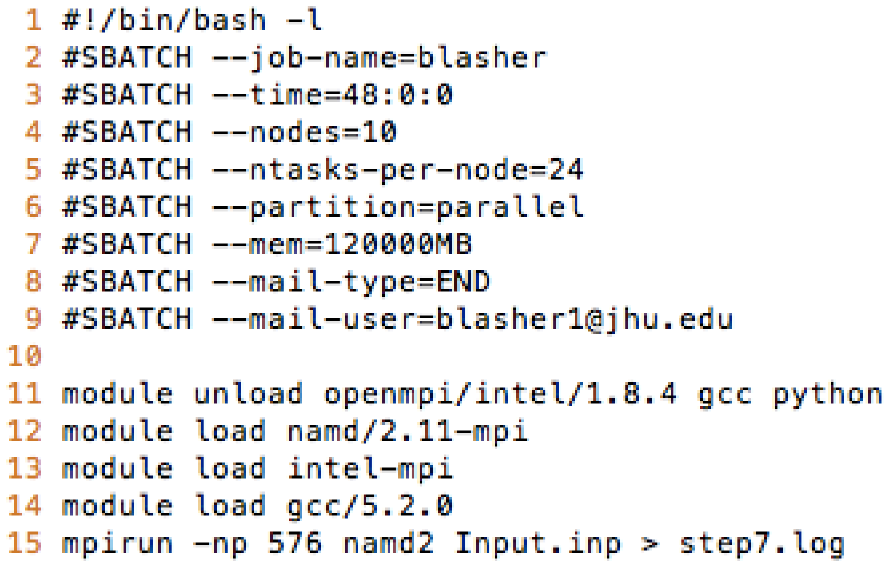
Chapter 4: Conclusion

Membrane protein prediction and design has proven to be challenging in the past, with the results of this study also resounding the difficulty. I have shown through this study that the combination of current approaches is not yet capable of predicting accurate membrane protein models through a single round of predictions. However, one round did locate a single residue within the interface, suggesting that with repeated rounds of predictions, followed by experimental validation, accurate models can be reached. This complex model prediction attempt emphasizes the importance of method development towards accurate prediction and design approaches for membrane proteins.

Kinks are an important feature within membrane proteins providing the flexibility for conformational changes. To advance the development of accurate prediction of kinked helices within membrane proteins, I generated a new kinked fragment library to enable sampling of kinked protein conformations. This library has the potential to be applied in fragment insertion methods to preform refinement on membrane proteins. Current refinement methods do not allow for conformational changes resulting from helix kinks. Furthermore, fragment insertion methods are not currently used within refinement protocols, as they can lead to protein unfolding. However, the combination of refinement with insertion of kinked fragments could provide improved sampling results.

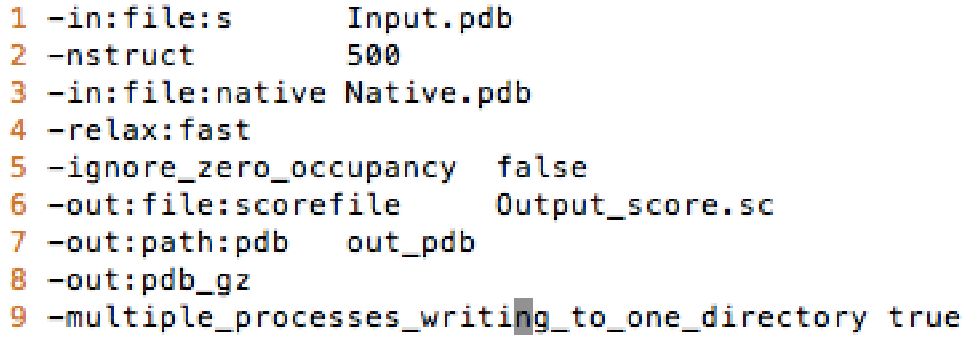
Appendix

MARCC script



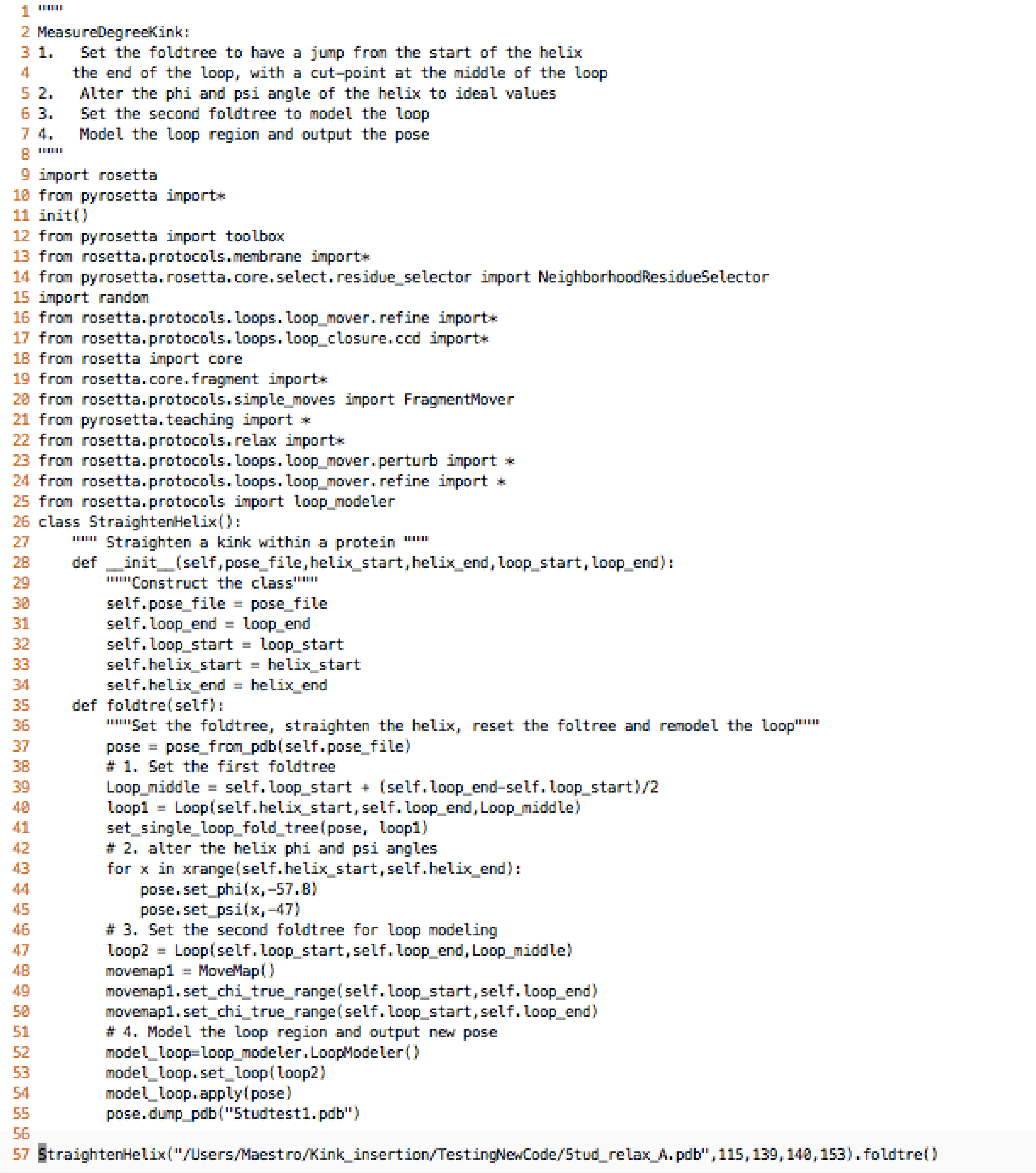
```
#!/bin/bash -l
#SBATCH --job-name=blasher
#SBATCH --time=48:0:0
#SBATCH --nodes=10
#SBATCH --ntasks-per-node=24
#SBATCH --partition=parallel
#SBATCH --mem=120000MB
#SBATCH --mail-type=END
#SBATCH --mail-user=blasher1@jhu.edu

module unload openmpi/intel/1.8.4 gcc python
module load namd/2.11-mpi
module load intel-mpi
module load gcc/5.2.0
mpirun -np 576 namd2 Input.inp > step7.log
```

Relax refinement script



```
-in:file:s      Input.pdb
-nstruct        500
-in:file:native Native.pdb
-relax:fast
-ignore_zero_occupancy  false
-out:file:scorefile     Output_score.sc
-out:path:pdb   out_pdb
-out:pdb_gz
-multiple_processes_writing_to_one_directory true
```

Straighten kink script



```
"""
MeasureDegreeKink:
1.   Set the foldtree to have a jump from the start of the helix
    the end of the loop, with a cut-point at the middle of the loop
2.   Alter the phi and psi angle of the helix to ideal values
3.   Set the second foldtree to model the loop
4.   Model the loop region and output the pose
"""
import rosetta
from pyrosetta import*
init()
from pyrosetta import toolbox
from rosetta.protocols.membrane import*
from pyrosetta.rosetta.core.select.residue_selector import NeighborhoodResidueSelector
import random
from rosetta.protocols.loops.loop_mover.refine import*
from rosetta.protocols.loops.loop_closure.ccd import*
from rosetta import core
from rosetta.core.fragment import*
from rosetta.protocols.simple_moves import FragmentMover
from pyrosetta.teaching import *
from rosetta.protocols.relax import*
from rosetta.protocols.loops.loop_mover.perturb import *
from rosetta.protocols.loops.loop_mover.refine import *
from rosetta.protocols import loop_modeler
class StraightenHelix():
    """ Straighten a kink within a protein """
    def __init__(self,pose_file,helix_start,helix_end,loop_start,loop_end):
        """Construct the class"""
        self.pose_file = pose_file
        self.loop_end = loop_end
        self.loop_start = loop_start
        self.helix_start = helix_start
        self.helix_end = helix_end
    def foldtre(self):
        """Set the foldtree, straighten the helix, reset the foltree and remodel the loop"""
        pose = pose_from_pdb(self.pose_file)
        # 1. Set the first foldtree
        Loop_middle = self.loop_start + (self.loop_end-self.loop_start)/2
        loop1 = Loop(self.helix_start,self.loop_end,Loop_middle)
        set_single_loop_fold_tree(pose, loop1)
        # 2. alter the helix phi and psi angles
        for x in xrange(self.helix_start,self.helix_end):
            pose.set_phi(x,-57.8)
            pose.set_psi(x,-47)
        # 3. Set the second foldtree for loop modeling
        loop2 = Loop(self.loop_start,self.loop_end,Loop_middle)
        movemap1 = MoveMap()
        movemap1.set_chi_true_range(self.loop_start,self.loop_end)
        movemap1.set_chi_true_range(self.loop_start,self.loop_end)
        # 4. Model the loop region and output new pose
        model_loop=loop_modeler.LoopModeler()
        model_loop.set_loop(loop2)
        model_loop.apply(pose)
        pose.dump_pdb("Studtest1.pdb")

StraightenHelix("/Users/Maestro/Kink_insertion/TestingNewCode/Stud_relax_A.pdb",115,139,140,153).foldtre()
```

Classification script

```
"""
    MeasureDegreeKink:
    1.    Method to find the dotproduct of vectors
    2.    Method to find the length of a vector
    3.    Method to find the angle between two vectors
    4.    Method to find the kink point or hinge residue within a helix
```

```
5.    Method to find the backbone coordinates of the helix
6.    Method to find the angle of the kink



"""
import rosetta
from pyrosetta import*
init()
from pyrosetta import toolbox
from pyrosetta.rosetta.protocols.membrane import*
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import numpy as np
import math
class MeasureDegreeKink():
"""Measures the degree to which a residue kinks"""
def __init__(self, file):
      """Construct the class"""
      self.file=file
      self.pose=pose_from_file(self.file)
def dotproduct(self,v1,v2):
      """ Dot product of two vectors"""
      # 1. Input vectors to find the dot product
      return sum((a*b) for a, b in zip(v1, v2))
def length(self,v):
      """ Length of a vector """
      # 2. Input a vector and return its length
      return math.sqrt(MeasureDegreeKink(self.file).dotproduct(v,
v))
def angle(self,v1, v2):
      """ Angle between two vectors """
      # 3. Take two vectors and output the angle between them
      return math.acos(MeasureDegreeKink(self.file).dotproduct(v1,
v2) / (MeasureDegreeKink(self.file).length(v1) *
MeasureDegreeKink(self.file).length(v2)))
def kink_point(self):
      """ Determines the kink point within a helix """
      # 4. Determine the hinge residue for kinking within a helix
        #     by finding which residue has the greatest phi and psi




        #     change from the ideal dihedral angles of an alpha
helix
        phi_ave = -64.0
        psi_ave = -41.0
        set=0
        res_num=1
        end=self.pose.total_residue()+1
        for x in xrange(1,end):



           phi=self.pose.phi(x)
           psi=self.pose.psi(x)
           dif_phi = math.fabs(phi_ave-phi)
           dif_psi = math.fabs(psi_ave-psi)
           tot=dif_phi+dif_psi
           #print tot
           if tot > set and x >= 5 and x <=
self.pose.total_residue()-4:
                res_num = x
                set = dif_phi+dif_psi
           else:

set=set

                res_num=res_num
     return res_num
     #print res_num
def direction(self):
     """ Find the direction of PCA vectors """
     # 9. Using vectors from the start to kink coords or the kink
to end coords
        #     the angle can be found to determine if the PCA
vectors 1
and 2 are



        #     pointing in the right direction
        coord =
MeasureDegreeKink(self.file).count(0,self.pose.total_residue())
        kink_point = MeasureDegreeKink(self.file).kink_point()
        point1 = np.array(coord[0])
        point2 = np.array(coord[kink_point-1])
        point3 = np.array(coord[-1])
        vector12 = point2-point1
        vector23 = point3-point2
        coord1 = coord[0:kink_point+1]
        coord2 = coord[kink_point:len(coord)+1]
        pca1 = PCA(n_components=1)
        pca2 = PCA(n_components=1)
        pca1.fit(coord1)
        pca2.fit(coord2)
        Vpca1 = pca1.components_[0]
        Vpca2 = pca2.components_[0]
        diff1 =
MeasureDegreeKink(self.fragfile).angle(vector12,Vpca1)*180/3.14
        diff2 =
MeasureDegreeKink(self.fragfile).angle(vector23,Vpca2)*180/3.14
        diffs=[diff1,diff2]
        return diffs
    def count(self, start, end):
        """ Find the coordinates for the backbone atoms of the
helix"""
        # 5. Find the backbone coordinates of the helix by looping
through



     #    each residue and finding the coordinates of the Calpha,
the N
     #    and C of each residue
     coord=[]
     for x in xrange(start,end):
          CA=self.pose.residue(x).xyz("CA")
          N=self.pose.residue(x).xyz("N")
          C=self.pose.residue(x).xyz("C")
          x1=[CA[0],CA[1],CA[2]]
          x2=[N[0],N[1],N[2]]
          x3=[C[0],C[1],C[2]]
          coord.append(x1)
          coord.append(x2)
          coord.append(x3)
     coord=np.array(coord)
     return coord
def kink_angle(self):
     """ Determines the kink angle of a helix by using PCA"""
     # 6. Use Principle Component Analysis (PCA)to find a vector
        #     that points in the direction of the helix before the



        #     hinge point, and a vector that points in the
direction



        #     of the helix after the hinge point. Using the two
vectors



        #     find the angle between the two vectors
        #print "this"
        start = MeasureDegreeKink(self.file).kink_point()
        #print start
        end = MeasureDegreeKink(self.file).kink_point()+1
        #print end
        coord1 = MeasureDegreeKink(self.file).count(1,end)
        #print coord1
        coord2 =
MeasureDegreeKink(self.file).count(start,self.pose.total_residue())
        pca1 = PCA(n_components=1)
        pca2 = PCA(n_components=1)
        pca1.fit(coord1)
        pca2.fit(coord2)
        vector1 = pca1.components_[0]
        vector2 = pca2.components_[0]
        #print vector1
        #print vector2
angle=MeasureDegreeKink(self.file).angle(vector1,vector2)*180/3.14
        #print angle
        if angle >= 90:
            angle=math.fabs(180-angle)
        else:
            angle=angle
        print "Angle (degree): "+ str(angle)
MeasureDegreeKink("/Users/Maestro/apps/
PyRosetta4/3f5wEndHelix.pdb").kink_angle()
```

Kinked fragment collection script

```
"""
FindHelixKinkFrags:
1.    Method to find the dotproduct of vectors
2.    Method to find the length of a vector
3.    Method to find the angle between two vectors
4.    Method to find the number of secondary structure that is
not helical
5.    Method to find the kink point or hinge residue within a
fragment
6.    Method to trim the edges of the fragment that are not
helical
secondary structure
7.    Method to find the angle of the kink
8.    Method to collect the Calpah coordinates, phi, psi, and
secondary structure
of the fragment from the current fragment library
9.   Method to determine the direction of the Principle
Component vector
10.  Method to pull only the lines +/- 4 from the kink point
```

```
11.  Method to iterate through the fragments inspecting
fragments of 24 residues at a time
"""
# -*- coding: utf-8 -*-
import rosetta
from pyrosetta import*
init()
import gzip
from decimal import Decimal
from pyrosetta import toolbox
from pyrosetta.rosetta.protocols.membrane import*
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import numpy as np
import math
import linecache
from itertools import islice
class FindHelixKinkFrags():
""" Generates a new kinked helix fragment library"""
def __init__(self,fragfile):
     """Constructs the class"""
     self.fragfile=fragfile
def dotproduct(self,v1,v2):
     """ Dot product of two vectors"""
     # 1. Input vectors to find the dot product
     return sum((a*b) for a, b in zip(v1, v2))
def length(self,v):
     """ Length of a vector """
     # 2. Input a vector and return its length
     return
math.sqrt(FindHelixKinkFrags(self.fragfile).dotproduct(v, v))
def angle(self,v1, v2):
     """ Angle between two vectors """
     # 3. Take two vectors and output the angle between them
     return
math.acos(FindHelixKinkFrags(self.fragfile).dotproduct(v1, v2)
/
(FindHelixKinkFrags(self.fragfile).length(v1) *
FindHelixKinkFrags(self.fragfile).length(v2)))
def count_ss(self,ss):
     """ Number of non helical residues """
     #  4. Takes the trimmed fragment and finds how many none
helical
     #     residues are present
     helix=0
     total=0
     for word in ss:
          total+=1
          if word=="H":

helix+=1

     return total-helix
def kink_point(self,startline,endline):
     """ Determines the kink point within a helix """
     # 5. Determine the hinge residue for kinking within a
     helix
        #     by finding which residue has the greatest phi
and psi



        #     change from the ideal dihedral angles of an
alpha helix
        phi_ave = -64.0
        psi_ave = -41.0
        set=0
        res_num=0
lists=FindHelixKinkFrags(self.fragfile).trim(startline,endline
)
        phi_list = lists[0]
        psi_list = lists[1]
        ss = lists[3]
        end=len(phi_list)
        count=FindHelixKinkFrags(self.fragfile).count_ss(ss)
        if lists != "not good frag":
            if count<=2:
                for x in xrange(1,end+1):
                    phi=phi_list[x-1]
                    psi=psi_list[x-1]
                    dif_phi = math.fabs(phi_ave-phi)
                    dif_psi = math.fabs(psi_ave-psi)
                    tot=dif_phi+dif_psi
                    if tot > set and x >= 5 and x <= end-4:

res_num = x

                        set = dif_phi+dif_psi
                    else:

     set=set
```

```
                         res_num=res_num
                    if res_num != 0:
                         return res_num

the end,

residues

          else:
               return "not good frag"
               #print "not good frag"
     else:
          return "not good frag"
          #print "not good frag"
else:
     return "not good frag"
     #print "not good frag"
def trim(self,startline,endline):
""" Trims the ends of fragments """

# 6.

     #
helica residue

#

#

Takes an input fragment and starting from the start or
removes all residues that are not helical, till a
is found. Makes sure that the fragments are at least 15
in length to move forward
     lists =
FindHelixKinkFrags(self.fragfile).splitRoute(startline,endline
)
     ss =
FindHelixKinkFrags(self.fragfile).splitRoute(startline,endline
)[3]
     ss2=lists[3]
     coord=lists[2]
     phi_list=lists[0]
     psi_list=lists[1]
     FirstNlines=lists[4]
     range = len(ss)
     for x in xrange(0,range-1):
          if x<=len(ss):
               if ss[x] != "H" and len(ss2)>=1:
                    del coord[0]
                    del ss2[0]
                    del phi_list[0]
                    del psi_list[0]
                    del FirstNlines[0]
               elif ss[x] == "H":
                    break

else: break

     ss.reverse()
     coord.reverse()
     ss2.reverse()
     phi_list.reverse()
     psi_list.reverse()
     FirstNlines.reverse()
     range=len(ss)
     for x in xrange(0,range):
          if x <= len(ss):
               if ss[x] != "H" and len(ss2)>=1:
                    del coord[0]
                    del ss2[0]
                    del phi_list[0]
                    del psi_list[0]
                    del FirstNlines[0]
               elif ss[x] == "H":
               #    """"""
                    break
               elif len(ss2)==0:
                    return "not good frag"
          else:
               break
     ss.reverse()
     coord.reverse()
     ss2.reverse()
     phi_list.reverse()
     psi_list.reverse()
     FirstNlines.reverse()
     #print ss2
     np.array(coord)
     list2=[phi_list,psi_list,coord,ss2,FirstNlines]
     #list2=list[0:2]
     #print list2
     if len(FirstNlines) >= 15:
          count = 0
          #print len(list2[0c])
          for x in xrange(0,len(list2[0])):
               phi=list2[0]
               psi=list2[1]
               #print phi[x]
               #print psi[x]
               if phi[x] <= -54 and phi[x] >= -74 and psi[x]
               <= -31
and psi[x] >= -51:
                    count+=1
          if count == len(list2[0]):
               return "not good frag"
          else:
               return list2
     else:
          return "not good frag"
def direction(self,startline,endline):
     """ Find the direction of PCA vectors """
     # 9. Using vectors from the start to kink coords or the
     kink
to end coords
        #     the angle can be found to determine if the PCA
vectors 1
and 2 are



        #     pointing in the right direction
        coord =
FindHelixKinkFrags(self.fragfile).trim(startline,endline)[2]

kink_point =

FindHelixKinkFrags(self.fragfile).kink_point(startline,endline
)
     point1 = np.array(coord[0])
     point2 = np.array(coord[kink_point-1])
     point3 = np.array(coord[-1])
     vector12 = point2-point1
     vector23 = point3-point2
     coord1 = coord[0:kink_point+1]
     coord2 = coord[kink_point:len(coord)+1]
     pca1 = PCA(n_components=1)
     pca2 = PCA(n_components=1)
     pca1.fit(coord1)
     pca2.fit(coord2)
     Vpca1 = pca1.components_[0]
     Vpca2 = pca2.components_[0]
     diff1 =
FindHelixKinkFrags(self.fragfile).angle(vector12,Vpca1)*180/3.
14
     diff2 =
FindHelixKinkFrags(self.fragfile).angle(vector23,Vpca2)*180/3.
14
     diffs=[diff1,diff2]

return diffs

     #print diffs
def pull_lines(self,startline,endline):
     """ Pull lines +/- 4 from the kink point"""
     # 10. Collect lines only +/- 4 from the kink point for
creating new
     #     fragments in the kinked fragment library
     kink_point =
FindHelixKinkFrags(self.fragfile).kink_point(startline,endline
)
     if kink_point != "not good frag":
          lines =
FindHelixKinkFrags(self.fragfile).trim(startline,endline)
          FirstNlines = lines[4]
          remove=[]
          for x in xrange(1,len(FirstNlines)+1):
               if x < kink_point-2 or x > kink_point+2:
                    remove.append(int(x-1))
          count=0
          for y in remove:
               del FirstNlines[y-count]
               count+=1
          return FirstNlines
          #print FirstNlines
     else:
          return "not good frag"
          #print "not good frag"
def kink_angle(self,startline,endline):
     """ Determines the kink angle of a helix by using PCA"""
     # 7. Use Principle Component Analysis (PCA)to find a
     vector
        #     that points in the direction of the helix before
the
```

```
        #     hinge point, and a vector that points in the
direction



        #     of the helix after the hinge point. Using the
two vectors



        #     find the angle between the two vectors
kink_point=FindHelixKinkFrags(self.fragfile).kink_point(startl
ine,endl
ine)
        if kink_point == "not good frag":
            return "not good frag"
            #print "not good frag"
        else:
            start = kink_point
            end = kink_point
            coord =
FindHelixKinkFrags(self.fragfile).trim(startline,endline)[2]
            coord2 = coord[kink_point:len(coord)+1]
            coord1 = coord[0:kink_point+1]
            pca1 = PCA(n_components=1)
            pca2 = PCA(n_components=1)
            pca1.fit(coord1)
            pca2.fit(coord2)
            vector1 = pca1.components_[0]
            vector2 = pca2.components_[0]
            #angle =
FindHelixKinkFrags(self.fragfile).angle(vector1,vector2)*180/3
.14
            diffs =
FindHelixKinkFrags(self.fragfile).direction(startline,endline)
            if abs(diffs[0]) >= 90:
                vector1 = -vector1
            if abs(diffs[1]) >= 90:
                vector2 = -vector2
            angle =
FindHelixKinkFrags(self.fragfile).angle(vector1,vector2)*180/3
.14
            #print angle
            return angle
```

```
    def kink_fragfile(self):
        """ Iterates through fragment file to find new kinked
fragments"""
        # 11. Iterate through fragments of 24 residues at a
time and



             #      determine if the angle is great enough to
     keep the
     fragment



             #      or if the fragment is not kinked enough to
     keep
             start = 30
             range = sum(1 for line in
     gzip.open(self.fragfile))
             endtot=(range-start)/9.
             end = start+24
             while end <= range:
     angle=FindHelixKinkFrags(self.fragfile).kink_angle(start,
     end)
                 #print angle
                 if angle >= 20 and angle != "not good frag":
                     """"""




#print angle
               #print 180-angle
               #lines2=
FindHelixKinkFrags(self.fragfile).trim(start,end)
               lines =
FindHelixKinkFrags(self.fragfile).pull_lines(start,end)
               #print lines2

+ '\n')

#print lines
for x in xrange(0,len(lines)):
     lines[x]=lines[x].replace('\n', "  " + str(angle)
#print lines
with open("kinkfileTest5mer.txt","a") as f:
     f.writelines(lines)
```

```
else:
""""""
start=end
          end=start+24
def splitRoute(self,start,end):
     """ Collect fragments of 24 residues from the file """
     # 8. Opens the fragfile and reads the lines, pulling the
     phi,

psi

        #     Calpha coordinates, secondary structure, and the
full
lines



        #     from the file
        with gzip.open(self.fragfile,"rb") as myfile:
            firstNlines=[]
            phi_list=[]
            psi_list=[]
            coord=[]
            ss=[]
            for line in islice(myfile,start,end):
                firstNlines.append(line)
                phi=line[104:113]
                psi=line[113:122]
Ca=[float(line[25:33]),float(line[33:42]),float(line[42:51])]
                coord.append(Ca)
                phi_list.append(float(phi))
                psi_list.append(float(psi))
                ss.append(str(line[8:9]))
            list=[phi_list,psi_list,coord,ss,firstNlines]
            #print list
            return list
FindHelixKinkFrags("/Users/Maestro/Research/ReadFragFile/
vall.jul19.2011.gz").kink_fragfile()
#FindHelixKinkFrags("/Users/Maestro/Research/ReadFragFile/
test.txt").kink_point()
```

References

1. Arinaminpathy Y, Khurana E, Engelman DM, Gerstein MB. Computational analysis of membrane proteins: the largest class of drug targets. 2009; doi:10.1016/j.drudis.2009.08.006
2. Liu Y, Engelman DM, Gerstein M. Genomic analysis of membrane protein families: abundance and conserved motifs. Genome Biol. BioMed Central; 2002;3: research0054.1. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC134483/#B4
3. Carpenter EP, Beis K, Cameron AD, Iwata S. Overcoming the challenges of membrane protein crystallography. Curr Opin Struct Biol. Elsevier Current Trends; 2008;18: 581–586. doi:10.1016/J.SBI.2008.07.001
4. Li S, Wu B, Han W. Parametrization of MARTINI for Modeling Hinging Motions in Membrane Proteins. J Phys Chem B. American Chemical Society; 2019;123: 2254–2269. doi:10.1021/acs.jpcb.8b11244
5. Hansson T, Oostenbrink C, van Gunsteren W. Molecular dynamics simulations. Curr Opin Struct Biol. Elsevier Current Trends; 2002;12: 190–196. doi:10.1016/S0959-440X(02)00308-1
6. Nelson MT, Humphrey W, Gursoy A, Dalke A, Kalé L V., Skeel RD, et al. NAMD: a Parallel, Object-Oriented Molecular Dynamics Program. Int J Supercomput Appl High Perform Comput. Sage PublicationsSage CA: Thousand Oaks, CA; 1996;10: 251–268. doi:10.1177/109434209601000401
7. Li Z, Scheraga HA. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. Proc Natl Acad Sci U S A. National Academy of Sciences; 1987;84: 6611–5. Available: http://www.ncbi.nlm.nih.gov/pubmed/3477791
8. Youngshang Pak †,‡, Istvan J. Enyedy †, Judith Varady †, Justin W. Kung §, Patricia S. Lorenzo §, Peter M. Blumberg § and, et al. Structural Basis of Binding of High-Affinity Ligands to Protein Kinase C: Prediction of the Binding Modes through a New Molecular Dynamics Method and Evaluation by Site-Directed Mutagenesis. American Chemical Society ; 2001; doi:10.1021/JM000488E
9. Monk BC, Tomasiak TM, Keniya M V, Huschmann FU, Tyndall JDA, O'connell Iii JD, et al. Architecture of a single membrane spanning cytochrome P450 suggests constraints that orient the catalytic domain relative to a bilayer. doi:10.1073/pnas.1324245111
10. Kandt C, Ash WL, Peter Tieleman D. Setting up and running molecular dynamics simulations of membrane proteins. Methods. Academic Press; 2007;41: 475–488. doi:10.1016/J.YMETH.2006.08.006
11. Metropolis N, Ulam S. The Monte Carlo Method. J Am Stat Assoc. 1949;44: 335. doi:10.2307/2280232
12. Beichl I, Sullivan F. The Metropolis Algorithm. Comput Sci Eng. 2000;2: 65–69. doi:10.1109/5992.814660
13. Hagel JM, Facchini PJ. Benzylisoquinoline Alkaloid Metabolism: A Century of Discovery and a Brave New World. Plant Cell Physiol. 2013;54: 647–672.

doi:10.1093/pcp/pct020

14. Fossati E, Ekins A, Narcross L, Zhu Y, Falgueyret J-P, Beaudoin GAW, et al. Reconstitution of a 10-gene pathway for synthesis of the plant alkaloid dihydrosanguinarine in Saccharomyces cerevisiae. Nat Commun. Nature Publishing Group; 2014;5: 3283. doi:10.1038/ncomms4283
15. Facchini PJ, Hagel JM, Liscombe DK, Loukanina N, MacLeod BP, Samanani N, et al. Opium poppy: blueprint for an alkaloid factory. Phytochem Rev. 2007;6: 97–124. doi:10.1007/s11101-006-9042-0
16. Fumihiko Sato, Takayuki Inui, Tomoya Takemura. Metabolic Engineering in Isoquinoline Alkaloid Biosynthesis. Curr Pharm Biotechnol. 2007;8: 211–218. doi:10.2174/138920107781387438
17. Galanie S, Thodey K, Trenchard IJ, Filsinger Interrante M, Smolke CD. Complete biosynthesis of opioids in yeast. Science. 2015;349: 1095–100. doi:10.1126/science.aac9373
18. Minami H, Kim J-S, Ikezawa N, Takemura T, Katayama T, Kumagai H, et al. Microbial production of plant benzylisoquinoline alkaloids. Proc Natl Acad Sci U S A. 2008;105: 7393–8. doi:10.1073/pnas.0802981105
19. Reed JW, Hudlicky T. The Quest for a Practical Synthesis of Morphine Alkaloids and Their Derivatives by Chemoenzymatic Methods. Acc Chem Res. American Chemical Society; 2015;48: 674–687. doi:10.1021/ar500427k
20. Ensley BD, Ratzkin BJ, Osslund TD, Simon MJ, Wackett LP, Gibson DT. Expression of naphthalene oxidation genes in Escherichia coli results in the biosynthesis of indigo. Science. American Association for the Advancement of Science; 1983;222: 167–9. doi:10.1126/SCIENCE.6353574
21. Mermod N, Harayama S, Timmis KN. New Route to Bacterial Production of Indigo. Nat Biotechnol. Nature Publishing Group; 1986;4: 321–324. doi:10.1038/nbt0486-321
22. Chang MCY, Keasling JD. Production of isoprenoid pharmaceuticals by engineered microbes. Nat Chem Biol. 2006;2: 674–681. doi:10.1038/nchembio836
23. Nakamura CE, Whited GM. Metabolic engineering for the microbial production of 1,3-propanediol. Curr Opin Biotechnol. Elsevier Current Trends; 2003;14: 454–459. doi:10.1016/J.COPBIO.2003.08.005
24. Hawkins KM, Smolke CD. Production of benzylisoquinoline alkaloids in Saccharomyces cerevisiae. Nat Chem Biol. Nature Publishing Group; 2008;4: 564–573. doi:10.1038/nchembio.105
25. Wang J, Guleria S, Koffas MA, Yan Y. Microbial production of value-added nutraceuticals. Curr Opin Biotechnol. 2016;37: 97–104. doi:10.1016/j.copbio.2015.11.003
26. DeLoache WC, Russ ZN, Narcross L, Gonzales AM, Martin VJJ, Dueber JE. An enzyme-coupled biosensor enables (S)-reticuline production in yeast from glucose. Nat Chem Biol. 2015;11: 465–471. doi:10.1038/nchembio.1816
27. Siegel JB, Smith AL, Poust S, Wargacki AJ, Bar-Even A, Louw C, et al. Computational protein design enables a novel one-carbon assimilation pathway Data deposition: The atomic coordinates and structure factors have been deposited

in the Protein Data Bank, www.pdb.org [PDB ID codes 4QPZ (Des1) and 4QQ8 (FLS)]. Biophys Comput Biol. doi:10.1073/pnas.1500545112

28. Chen A, Li Y, Nie J, McNeil B, Jeffrey L, Yang Y, et al. Protein engineering of Bacillus acidopullulyticus pullulanase for enhanced thermostability using in silico data driven rational design methods. Enzyme Microb Technol. Elsevier; 2015;78: 74–83. doi:10.1016/J.ENZMICTEC.2015.06.013
29. Reetz MT, Carballeira JD. Iterative saturation mutagenesis (ISM) for rapid directed evolution of functional enzymes. Nat Protoc. Nature Publishing Group; 2007;2: 891–903. doi:10.1038/nprot.2007.72
30. Behera RK, Mazumdar S. Thermodynamic basis of the thermostability of CYP175A1 from Thermus thermophilus. Int J Biol Macromol. Elsevier; 2010;46: 412–418. doi:10.1016/J.IJBIOMAC.2010.01.014
31. Zhang S-B, Wu Z-L. Identification of amino acid residues responsible for increased thermostability of feruloyl esterase A from Aspergillus niger using the PoPMuSiC algorithm. Bioresour Technol. Elsevier; 2011;102: 2093–2096. doi:10.1016/J.BIORTECH.2010.08.019
32. Hanukoglu I. Electron Transfer Proteins of Cytochrome P450 Systems. Adv Mol Cell Biol. Elsevier; 1996;14: 29–56. doi:10.1016/S1569-2558(08)60339-2
33. Werck-Reichhart D, Feyereisen R. Cytochromes P450: a success story. Genome Biol. BioMed Central; 2000;1: reviews3003.1. doi:10.1186/gb-2000-1-6-reviews3003
34. Poulos TL, Johnson EF. Structures of Cytochrome P450 Enzymes. Cytochrome P450. Boston, MA: Springer US; 2005. pp. 87–114. doi:10.1007/0-387-27447-2_3
35. Pierce BG, Hourai Y, Weng Z. Accelerating Protein Docking in ZDOCK Using an Advanced 3D Convolution Library. Keskin O, editor. PLoS One. Public Library of Science; 2011;6: e24657. doi:10.1371/journal.pone.0024657
36. Kozakov D, Hall DR, Xia B, Porter KA, Padhorny D, Yueh C, et al. The ClusPro web server for protein–protein docking. Nat Protoc. Nature Publishing Group; 2017;12: 255–278. doi:10.1038/nprot.2016.169
37. Chaudhury S, Berrondo M, Weitzner BD, Muthu P, Bergman H, Gray JJ. Benchmarking and Analysis of Protein Docking Performance in Rosetta v3.2. Uversky VN, editor. PLoS One. Public Library of Science; 2011;6: e22477. doi:10.1371/journal.pone.0022477
38. Wang T, Wade RC. Implicit solvent models for flexible protein-protein docking by molecular dynamics simulation. Proteins Struct Funct Bioinforma. John Wiley & Sons, Ltd; 2002;50: 158–169. doi:10.1002/prot.10248
39. Cojocaru V, Balali-Mood K, Sansom MSP, Wade RC. Structure and Dynamics of the Membrane-Bound Cytochrome P450 2C9. Halpert JR, editor. PLoS Comput Biol. Public Library of Science; 2011;7: e1002152. doi:10.1371/journal.pcbi.1002152
40. Schwede T, Kopp J, Guex N, Peitsch MC. SWISS-MODEL: an automated protein homology-modeling server. Nucleic Acids Res. Oxford University Press; 2003;31: 3381–3385. doi:10.1093/nar/gkg520
41. Webb B, Sali A. Comparative Protein Structure Modeling Using MODELLER. Curr

Protoc Bioinforma. John Wiley & Sons, Ltd; 2014;47: 5.6.1-5.6.32. doi:10.1002/0471250953.bi0506s47

42. Tyka MD, Keedy DA, André I, DiMaio F, Song Y, Richardson DC, et al. Alternate States of Proteins Revealed by Detailed Energy Landscape Mapping. J Mol Biol. Academic Press; 2011;405: 607–618. doi:10.1016/J.JMB.2010.11.008
43. Yuan S, Chan HCS, Hu Z. Using PyMOL as a platform for computational drug design. Wiley Interdiscip Rev Comput Mol Sci. John Wiley & Sons, Ltd (10.1111); 2017;7: e1298. doi:10.1002/wcms.1298
44. Alford RF, Koehler Leman J, Weitzner BD, Duran AM, Tilley DC, Elazar A, et al. An Integrated Framework Advancing Membrane Protein Modeling and Design. Livesay DR, editor. PLOS Comput Biol. 2015;11: e1004398. doi:10.1371/journal.pcbi.1004398
45. Andrade X, Strubbe D, De Giovannini U, Larsen AH, Oliveira MJT, Alberdi-Rodriguez J, et al. Real-space grids and the Octopus code as tools for the development of new simulation approaches for electronic systems. Phys Chem Chem Phys. 2015;17: 31371–31396. doi:10.1039/C5CP00351B
46. Jo S, Kim T, Iyer VG, Im W. CHARMM-GUI: A web-based graphical user interface for CHARMM. J Comput Chem. John Wiley & Sons, Ltd; 2008;29: 1859–1865. doi:10.1002/jcc.20945
47. Acun B, Hardy D, Kale L, Li K, Phillips JC, Stone JE. Scalable Molecular Dynamics with NAMD on the Summit System. IBM J Res Dev. 2018; 1–1. doi:10.1147/JRD.2018.2888986
48. Mandell DJ, Kortemme T. Computer-aided design of functional protein interactions. Nat Chem Biol. 2009;5: 797–807. doi:10.1038/nchembio.251
49. Marze NA, Roy Burman SS, Sheffler W, Gray JJ. Structural bioinformatics Efficient flexible backbone protein-protein docking for challenging targets. doi:10.1093/bioinformatics/bty355
50. Barlow KA, Ó Conchúir S, Thompson S, Suresh P, Lucas JE, Heinonen M, et al. Flex ddG: Rosetta Ensemble-Based Estimation of Changes in Protein–Protein Binding Affinity upon Mutation. J Phys Chem B. American Chemical Society; 2018;122: 5389–5399. doi:10.1021/acs.jpcb.7b11367
51. Quignot C, Rey J, Yu J, Tufféry P, Guerois R, Andreani J. InterEvDock2: an expanded server for protein docking using evolutionary and biological information from homology models and multimeric inputs. Nucleic Acids Res. 2018;46: W408–W416. doi:10.1093/nar/gky377
52. Poulos TL, Johnson ER. 3 Structures of Cytochrome P450 Enzymes [Internet]. Available: http://eknygos.lsmuni.lt/springer/111/87-114.pdf
53. Forrest LR, Tang CL, Honig B. On the accuracy of homology modeling and sequence alignment methods applied to membrane proteins. Biophys J. The Biophysical Society; 2006;91: 508–17. doi:10.1529/biophysj.106.082313
54. Lamb DC, Waterman MR. Unusual properties of the cytochrome P450 superfamily. Philos Trans R Soc Lond B Biol Sci. The Royal Society; 2013;368: 20120434. doi:10.1098/rstb.2012.0434
55. Monk BC, Tomasiak TM, Keniya M V, Huschmann FU, Tyndall JDA, O'connell Iii

JD, et al. Architecture of a single membrane spanning cytochrome P450 suggests constraints that orient the catalytic domain relative to a bilayer. Source. 2014;111: 3865–3870. doi:10.1073/pnas.1324245111

56. Šrejber M, Navrátilová V, Paloncýová M, Bazgier V, Berka K, Anzenbacher P, et al. Membrane-attached mammalian cytochromes P450: An overview of the membrane's effects on structure, drug binding, and interactions with redox partners. J Inorg Biochem. Elsevier; 2018;183: 117–136. doi:10.1016/J.JINORGBIO.2018.03.002
57. Langelaan DN, Wieczorek M, Blouin C, Rainey JK. Improved Helix and Kink Characterization in Membrane Proteins Allows Evaluation of Kink Sequence Predictors. J Chem Inf Model. American Chemical Society; 2010;50: 2213–2220. doi:10.1021/ci100324n
58. Cao C, Tan Q, Xu C, He L, Yang L, Zhou Y, et al. Structural basis for signal recognition and transduction by platelet-activating-factor receptor. Nat Struct Mol Biol. 2018;25: 488–495. doi:10.1038/s41594-018-0068-y
59. Köpfer DA, Song C, Gruene T, Sheldrick GM, Zachariae U, de Groot BL. Ion permeation in K^+ channels occurs by direct Coulomb knock-on. Science. American Association for the Advancement of Science; 2014;346: 352–5. doi:10.1126/science.1254840
60. Hall SE, Roberts K, Vaidehi N. Position of helical kinks in membrane protein crystal structures and the accuracy of computational prediction. J Mol Graph Model. Elsevier; 2009;27: 944–950. doi:10.1016/J.JMGM.2009.02.004
61. Jiang Y, Lee A, Chen J, Cadene M, Chait BT, MacKinnon R. The open pore conformation of potassium channels. Nature. Nature Publishing Group; 2002;417: 523–526. doi:10.1038/417523a
62. Alford RF, Koehler Leman J, Weitzner BD, Duran AM, Tilley DC, Elazar A, et al. An Integrated Framework Advancing Membrane Protein Modeling and Design. Livesay DR, editor. PLOS Comput Biol. Public Library of Science; 2015;11: e1004398. doi:10.1371/journal.pcbi.1004398
63. Law EC, Wilman HR, Kelm S, Shi J, Deane CM. Examining the Conservation of Kinks in Alpha Helices. PLoS One. Public Library of Science; 2016;11: e0157553. doi:10.1371/journal.pone.0157553
64. Tieleman DP, Shrivastava IH, Ulmschneider MR, Sansom MSP. Proline-induced hinges in transmembrane helices: Possible roles in ion channel gating. Proteins Struct Funct Genet. 2001;44: 63–72. doi:10.1002/prot.1073
65. Visiers I, Braunheim BB, Weinstein H. Prokink: a protocol for numerical evaluation of helix distortions by proline. Protein Eng. 2000;13: 603–6. Available: http://www.ncbi.nlm.nih.gov/pubmed/11054453
66. Wilman HR, Shi J, Deane CM. Helix kinks are equally prevalent in soluble and membrane proteins. Proteins Struct Funct Bioinforma. 2014;82: 1960–1970. doi:10.1002/prot.24550
67. Bansal M, Kumart S, Velavan R. HELANAL: A Program to Characterize Helix Geometry in Proteins. J Biomol Struct Dyn. Taylor & Francis Group ; 2000;17: 811–819. doi:10.1080/07391102.2000.10506570

68. Kneissl B, Mueller SC, Tautermann CS, Hildebrandt A. String Kernels and High-Quality Data Set for Improved Prediction of Kinked Helices in α-Helical Membrane Proteins. J Chem Inf Model. American Chemical Society; 2011;51: 3017–3025. doi:10.1021/ci200278w

69. Rigoutsos I, Riek P, Graham RM, Novotny J. Structural details (kinks and non- conformations) in transmembrane helices are intrahelically determined and can be predicted by sequence pattern descriptors. Nucleic Acids Res. Narnia; 2003;31: 4625–4631. doi:10.1093/nar/gkg639

70. Conway P, Tyka MD, DiMaio F, Konerding DE, Baker D. Relaxation of backbone bond geometry improves protein energy landscape modeling. Protein Sci. 2014;23: 47–55. doi:10.1002/pro.2389

71. Adolf-Bryfogle J, Dunbrack Jr. RL. The PyRosetta Toolkit: A Graphical User Interface for the Rosetta Software Suite. Sticht H, editor. PLoS One. Public Library of Science; 2013;8: e66856. doi:10.1371/journal.pone.0066856

72. Wang C, Bradley P, Baker D. Protein–Protein Docking with Backbone Flexibility. J Mol Biol. Academic Press; 2007;373: 503–519. doi:10.1016/J.JMB.2007.07.050

Curriculum Vitae

BRITTANY LASHER
3303 Paine Street, Baltimore MD 21211 I blasher1@jhu.edu I 253-691-9250

EDUCATION

JOHNS HOPKINS UNIVERSITY BALTIMORE, MD
MSE in Chemical Engineering
AUGUST 2017 - MAY 2018

University of WASHINGTON SEATTLE, WA
BS in Chemical Engineering
MARCH 2015 - JUNE 2017

TACOMA COMMUNITY COLLEGE TACOMA, WA
AS in Chemical Engineering
SEPTEMBER 2011 - JUNE 2015

RESEARCH EXPERIENCE

Engineering Cytochrome P450 Redox Enzyme Pair for Increased Production of BIA Pharmaceuticals

OCTOBER 2017 - PRESENT
Johns Hopkins University
Mentors: Jeff Gray and Rebecca Alford
Developed prediction approach for mono-spanning membrane proteins, using computational tools, to allow for enzymatic designs that result in increased catalytic activity

- Modeled the equilibrium state of each enzyme in its natural lipid bilayer through Molecular Dynamics simulations to accurately capture its biological environment
- Predicted interfacial region utilizing a combination of global docking and Rosetta local docking to generate proposed models of the complex.
- Obtained accurate interfacial region utilizing experimental results to guide and validate location of the protein-protein interface
- Increased enzymatic activity will be accomplished through application of RosettaDesign to the validated complex, improving the protein-protein interface

Capturing Kinked Conformations Within Membrane Proteins
JUNE 2018 – PRESENT
Johns Hopkins University
Mentors: Jeff Gray and Rebecca Alford

- Implemented a novel method for classification of helical kinks utilizing Principal Component Analysis (PCA)
- Developed algorithms to search current protein fragment libraries for fragments involving kinked helices in order to build a kinked-helix fragment library
- Generated a python based method for kink-fragment protein insertion followed by loop modeling
- Discovery of important features representative of kinked helices will be determined through Machine Learning methods

Modeling of Protein Purification Peptide Tags
MARCH 2016 – JUNE 2017
University of Washington
Mentors: Jim Pfaendtner and Kayla Sprenger

- Accelerated the discovery pipeline for bio-inspired nanostructures utilizing Molecular Dynamics simulations
- Determined ideal environmental conditions (e.g. salt concentration or surface charge), through simulation analysis methods, for affinity purification tags to bind to silica surfaces

Cardiac Monitoring Device for Manual Measurement of Blood Flow
SEPTEMBER 2016 - JUNE 2017
University of Washington
Mentors: Jonathan Posner and Gerardo Rodriquez
Developed a non-invasive device utilizing ultrasound technology to manually measure blood flow velocity fluctuations within patients suffering from cardiac arrest
Devised a novel strategy to create an interface between the patient's skin and the transducer using a water based gel to enable blood flow sound waves to be captured

SKILLS & ABILITIES

Software (proficient): Linux/Unix environment, VMD, MD Analysis, PyMOL, SolidWorks, Excel, ASPEN

Computer Programming (proficient): Python, MATLAB, R

Molecular Modeling (proficient): High performance computing, Statistical analysis, Molecular dynamics simulations with NAMD and Gromacs, Protein structure prediction and design with Rosetta, Pyrosetta

Laboratory: Spectroscopy, Slide and Culture Preparation, Light Microscope Analysis, Chromatography

EXPERIENCE

Collaborative Chemical Engineering Industrial Study Abroad JUNE 2016 - AUGUST 2016
Zhejiang University, Hangzho China
Industrial study abroad experience between UW and ZJU to introduce students to China's industrial potential

- Gained an understanding of largescale process engineering procedures through hands-on experiences within some of China's leading industrial process plants
- Applied chemical engineering concepts through operating manufacturing and processing equipment, including: Heat exchangers, fluid transportation systems, and separation instruments

FUTURE PUBLICATIONS

- A Novel Modeling Approach to Gain Insight into the Protein-Protein Interactions within Membrane Proteins
  EXPECTED 2019
- Toward Accurate Prediction and Design of Kinked α-Helices in Membrane Proteins
  EXPECTED 2019

LEADERSHIP, ACTIVITIES, HONORS AND ORGANIZATIONS

- Phi Theta Kappa, Tacoma Community College
  OCTOBER 2013 - JUNE 2015
- Presidents Medal, Tacoma Community College
  JUNE 2015
- Dan Evans Scholarship, University of Washington
  2015 and 2016
- Woman in Chemical Engineering Member
  OCTOBER 2016 – JUNE 2017
- AICHE Member
  OCTOBER 2016 - JUNE 2017
- University of Washington, College of Engineering Peer Mentor
  SEPTEMBER 2016 - PRESENT
- RosettaCon Poster Presentation
  AUGUST 2018