Improving Membrane Protein Modeling and Design using Empirical Data

By

Amanda Marie Duran

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
Chemistry
May 11, 2018
Nashville, Tennessee

Approved:

Jens Meiler, Ph.D.

Terry P. Lybrand, Ph.D.

Michael P. Stone, Ph.D.

Terunaga Nakagawa, M.D. Ph.D.

To my grandparents, (Elida Ornelas, Augustine Ornelas, Maria Elena Duran, Manuel Duran), and parents (Oscar Duran Sr. and Sulema Duran) whose generations of hard work and dedication to our family made this work possible. You are my inspiration.

ACKNOWLEDGEMENTS

First, I would like to thank my PI, Jens Meiler, for accepting me into his lab and challenging me. I had originally agreed to do wet lab 75% of the time and computational work 25%, but before I knew it, I was 100% computational. Jens gave me a lot of room for creativity and always heard my ideas, and, in many cases, we pursued them. My experience in your lab really forced me to become an independent scientist, and for that I am very grateful.

My rotation was only in the wet-lab, so I had very little experience in Rosetta when I first started in the lab. Thank you, David Nannemann and Darwin Fu for helping me get started in Rosetta, and for putting up with my humming. Thank you to the wonderful ladies who welcomed me into the lab and are still so encouraging to this day- Stephanie Hirst DeLuca, Brittany Allison, Liz Dong Nyugen, and Julia Koehler. Thank you to Alyssa Lokits for helping me through the final years- you are one of the most positive people I have ever met, and I will never forget when we wrote unit tests together. Thank you, Heather Darling for going out of your way to make things go smoothly!

I would also like to thank Brian Bender for all of our introspective chats and hiking adventures. Thank you to Yan Xia, Soumya Ganguly, Sam DeLuca, Steven Combs, Rocco Moretti, Alex Sevy, Alberto Cisneros, Ben Mueller, Georg Kuenze, and everyone else in the lab who has answered my questions or provided insightful discussion. Thank you to my committee, Drs. Jens Meiler, Terry Lybrand, Mike Stone, and Teru Nakagawa for your guidance and feedback over the years. Thank you to my collaborators of several years, Drs. Chuck Sanders and Brett Kroncke.

To my music buddies- for several years I was a member of the Vanderbilt Community band alongside my friends Jennifer Stancill and Dan Sprague, thank you for understanding my band jokes and putting up with my puns. Thank you to Austin Oleskie for teaching me my first song on guitar, and for always providing me with solid jams when I needed them. Thank you to Kevin Monroe for swapping music, attending concerts with me, and just your general encouragement over the years. Even though we are far apart, music brings us together.

Later I was involved in the Vanderbilt University Women in Science and Engineering organization where I chaired the outreach committee for multiple years. I interacted with several wonderful ladies while in this group including Chrystal Starbird, Abigail Searfoss, Heather McCartney, and Dr. Greene. These women encouraged me and were truly great examples of amazing women in science.

I was lucky to have met a few of my best friends in my very first week at Vanderbilt, and we were pretty much inseparable from that point on. I first met Gwynne at Suzy's which would later be our morning huddle place for the first few years. Shortly after meeting, she mentioned how much she liked hiking and agreed to attend a Pearl Jam movie with me-I knew I had made a friend for life. I met Nicole soon after that, and because Nicole is only a quarter-inch shorter than I am, she made for an ideal hiking partner. Gwynne, Nicole and I hiked and backpacked in many places all over Tennessee. They forced me to take some weekends off and were endless sources of support. I truly could not have finished without you. I am still sane only because of you.

I was fortunate to have so many friends supporting me from far away. To Laura Dzugan- thank you for your endless, daily support! You are my moral compass and our lives will always be parallel. I honestly don't know what I would do without you. To Amanda Natter-not only do you have a great name, but you are a great person. Your crazy work and school schedule was truly inspiring and encouraging. To Amanda “Panda” Conde – we will always be soul sisters! Thanks for all of the encouragement for staying healthy, fit and in grad school. It only took 10 weeks! Thank you to my friends back home in Texas, in particular Dana Nelson, Tabatha Jurak, and Kayla Martinez for helping me find my way in high school and college, and for continuing to be supportive from a distance.

A special thank you to the many people who helped me during the most difficult time in my life. Thanks to all of the doctors and nurses who were able to diagnose my vestibular disorders and provide options for treatment. Thank you to my many friends who went out of their way to help me and support me, including many members of my lab. I would like to especially thank Nicole Diggins who drove me to and from work, doctor visits, and the grocery store for at least six months while I was still trying to find a treatment. You were my guide in a world that was constantly rocking. A big thank you to Steve Schrider and Peter Griffin for helping socialize me on the weekends, and all of your empathy and support during this time-you guys are great friends. A special thank you to Kevin, Gwynne, and Kate Mittendorf whose words of encouragement at the right time were just what I needed to push forward.

Finally, a big thank you to my big family. I grew up hearing stories about the lives and struggles of the generations that have come before me. When I struggled during my time at Vanderbilt, I reminded myself of those stories and was able to push through. Not a day went by when I didn’t recognize what a privilege it was to have this opportunity for higher education. I have had many strong women in my life, and my family isn’t short of them. Thank you to my Aunt Donna for traveling with me and providing me with any advice I ever needed. My sister Valerie was my very first teacher to give me homework well before I was even in school. Her exceptional grades and accomplishments in school were both inspiring and challenging to follow-thank you for setting a great example. My nieces and nephews have all been endless sources of joy. My sister Elizabeth was only seven when I moved to Nashville. I have missed her terribly but she is so strong and encouraging. Elizabeth, your silliness and spunk has brought so much joy to my life. My brother Oscar always had a love for biology and is largely responsible for sparking my interest in science. Finally, I have been blessed with the greatest parents in the world. My parents always told me that I would go to college, but I needed to find a way to afford it. They taught me how to value an education. For a while I wanted to be a musician, but after all of the years that my parents dealt with me performing experiments with whatever I could find in the house, to drawing up schematics on Microsoft paint, to making 3-D models of my room using graph paper, they knew where I belonged and gave me the push to where I needed to go. My mom is the strongest woman I know and I am constantly learning from her. She taught me how to cope with a chronic illness diagnosis and has been my greatest support and advocate from day one. I call my mom every night on my walk to my car. I will never forget one night I stayed in the lab until after 2:30 AM. She texted me to make sure I was alright and I said I would be done in about 30 minutes. She responded with “Okay, call me when you are done. I am just crocheting”. I don’t think anyone is that into crocheting, but I do know that my Mom makes me feel like she is right along with me every step of the way even if we are hundreds of miles apart. I couldn’t have asked for a better ear to listen to me and shoulder to lean on in my most difficult times. Anyone who is familiar with my work ethic would understand if they knew my Dad. My Dad is incredibly talented at so many different trades and passionate about every single one of them. He’s been a successful manager, small business owner, and great father – how he manages to balance it all is beyond me. Thank you, Mom and Dad for believing in me and inspiring me.

TABLE OF CONTENTS

Appendix

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

SUMMARY

Membrane proteins are a challenging but important class of proteins, and computational methods for membrane proteins are lagging far behind those of soluble proteins. The overall focus of this dissertation was to improve computational approaches for membrane protein modeling and design. Herein, empirical data was used to understand and improve the process of computational modeling and design.

In Chapter 1, computational structural biology is introduced along with Rosetta protocols pertinent to the studies in this dissertation. In a Rosetta review entitled “Protocols for Molecular Modeling with Rosetta3 and RosettaScripts” by Bender et al. in 2016, I described the protocols relevant to my dissertation, thus I have included sections for Rosetta Design, RosettaMembrane, and Rosetta Symmetry. This chapter also includes a brief explanation of homology modeling and RosettaCM. Finally, the significance of studying membrane proteins, in particular pseudo-symmetry and biomedical relevance, is described along with how machine learnings can play a role in pushing the field forward.

In Chapter 2, the Rosetta Design algorithm is benchmarked for membrane proteins through the use of the RosettaMembrane and Rosetta Symmetry. I establish an ideal strategy for preparing membrane protein structures for design calculations regardless of the resolution, and I demonstrate the strengths and shortcomings of the RosettaMembrane energy function. This chapter is almost an entire reproduction of the manuscript “Computational Design of Membrane Proteins using RosettaMembrane” by Duran and Meiler 2017. I have provided additional commentary for figures published in the supplement. I have also added a number of figures not included in time for the publication. I developed these figures as a means of describing exactly

which native residues were changing to which designed residues and propose that these plots be used in design benchmarks in the future.

In Chapter 3, several programs were evaluated for their ability to predict mutation-induced stability changes in membrane proteins. Existing programs were trained on soluble data sets, so this was the first study to detail the need for a membrane protein specific program. This chapter is nearly a reproduction of the original publication of the article entitled “Documentation of an Imperative to Improve Methods for Predicting Membrane Protein Stability” by Kroncke et al. 2016. I contributed a substantial amount of data and data analysis. We found that all programs poorly predicted experimental values. This was a call to the community to develop clever methods to circumvent the sparse amount of thermostability data, as well as stress the importance of continuing to produce thermostability data for membrane proteins.

In Chapter 4, machine learning methods were used to refit the score terms in the RosettaMembrane energy function. Empirical data, specifically the $\Delta\Delta G$ of unfolding from Chapter 3, was used to determine new coefficients towards accurate predictions of mutation-induced stability changes, as well as which score terms were contributing to noise. A simplified energy function was created which outperformed RosettaMembrane. However, many caveats and cautions are also discussed.

In Chapter 5, a model of the resting VSD, and closed pore of KCNQ1 is created using RosettaCM, RosettaMembrane, and Rosetta Symmetry. It describes the challenges of the model building process for state of a protein that is difficult to capture. Many of the sequences of templates proposed are near 20% sequence identity. To overcome a low homology, empirical data from the literature was utilized for model filtering and selection. Finally, complicated relaxation studies are described in addition to model validation through the use of external servers including MolProbity, PDBSum/ProCheck, and PoreWalker.

In Chapter 6, the biological implications of pseudo-symmetry in membrane proteins is reviewed. This chapter is a complete reproduction of the article entitled “Inverted Topologies in Membrane Proteins: A Mini-Review” by Duran and Meiler in 2013. This mini-review proposes an evolutionary pathway for pseudo-symmetry seen in membrane proteins and focuses on the implications such symmetry has on function. This mini-review provides the necessary background to understand some of the motivation for the project described in Chapter 7.

In Chapter 7, a set of 13 aquaporins, pseudo-symmetric membrane proteins, are engineered to be symmetric in sequence and in structure. Computational techniques involve the strategy of circular permutation as well as repair, relaxation, and scoring of models in Rosetta. Genes were synthesized for the top 20 models and experimental studies have not identified expression for any of the engineered proteins. The experimental conditions tested for each construct are detailed in tables here.

Finally, in Chapter 8, the results of the experiments conducted in the previous chapters are summarized and discussed. Much cross-over is seen in the projects and these observations are discussed such as the use of RosettaMembrane. Many future directions are proposed relating to the further improvement of the RosettaMembrane energy function, most specifically, through leveraging additional empirical data.

Appendix A is the protocol capture that I created for past Rosetta Workshops relating to modeling membrane proteins, symmetric proteins, and design calculations. This protocol was included in the Rosetta review supplemental materials.

Appendix B contains the tabular supplemental data for the “Computational design of Membrane Proteins using RosettaMembrane” publication in Chapter 2.

Appendix C contains the protocol capture for the membrane protein relaxation studies, membrane protein design studies, symmetric membrane protein studies as well as analysis from “Computational design of Membrane Proteins using RosettaMembrane” publication in Chapter 2. This protocol capture is included in the supplemental materials provided with the publication.

Appendix D contains the protocol capture for Chapter 3. This was not included in the supplemental materials but is detailed for both the standard Rosetta ddg_monomer protocol as well as with RosettaMembrane.

Appendix E contains the dataset that was used to evaluated mutation-induced stability programs in Chapter 3 and was used to train and evaluate the machine learning algorithms in Chapter 4.

Appendix F contains the raw tables from the regression analyses performed in Chapter 4. Results from approaches including Ridge and Elastic Net regressions along with cross validation by protein backbone and leave-one-out are shown.

Appendix G contains the protocol capture for Chapter 5 which involves the modeling of a resting VSD, closed pore KCNQ1 from multiple templates. This includes the sequence alignment of the multiple templates, the XML file that contains the detailed controls and order of templates used, as well as details for the complex relaxation protocol to restrict movement away from the cryo-EM structure of frog KCNQ1.

Appendix H contains the supplemental figures for Chapter 6. Due to the large amount of symmetric backbones evaluated, not all figures are included in the main text. Individual energetic contribution to various numbers of mutation pairs are shown here.

Appendix I contains the protocol capture for Chapter 6 and details the different types of relaxation and scoring strategies used.

Appendix J contains information on previous studies for the project in Chapter 6. Prior to the approach in Chapter 6, extensive computational and experimental studies were done for symmetric variants of the glyceroaquaporin, GlpF.

Appendix K contains information for a project in collaboration with a laboratory performing trafficking assays on the A2a receptor. This project involved computational design of six positions, specifically away from Cysteine. These proposed mutations were tested experimentally to evaluate Rosetta's performance. An experimental screen of mutants revealed a few contenders and these were also modeled in Rosetta. This is a great example of the feedback between experimental and computational studies.

Appendix L contains the protocol capture for the computational experiments performed in Appendix K. The protocols span full sampling of design at several specific regions, modeling of single point mutations, and modeling of multiple mutations. This includes a description of the approach for modeling the wild-type protein in such a way that is comparable in proper energetic analysis.

CHAPTER 1

INTRODUCTION

Part of this chapter includes published work from.

Bender*, Cisneros*, Duran*, Finn*, Fu*, Lokits*, Mueller*, Sangha*, Sauer*, Sevy*, Sliwoski, Sheehan, DiMaio, Meiler, and Moretti, 2016 (*Authors contributed equally)

Author contributions: I was the sole contributor for sections of Rosetta Design, Rosetta Membrane, and Rosetta Symmetry for the publication entitled "Protocols for Molecular Modeling with Rosetta3 and RosettaScripts" in the Biochemistry Journal, published as an ACS AuthorChoice open access article (Bender et al., 2016). I also developed the protocol capture for the design of a symmetric membrane proteins using Rosetta, available in the supporting information, in its entirety.

1.1 Introduction and implications for structural biology

Proteins have many different roles ranging from architectural, to signaling and response in organisms. Proteins are encoded by deoxyribonucleic acid (DNA) sequences, which are transcribed into messenger ribonucleic acid (mRNA), and eventually translated to an amino acid sequence. Protein structures are influenced by their sequence, and at the same time, protein structures have been optimized by nature for a particular function. Across species, proteins with a similar DNA sequence, thus amino acid sequence, tend to have related roles in their respective organisms. Understanding the relationship between protein sequence, structure, and function is essential to driving forward our knowledge of genetic diseases, drug discovery, and novel biomaterials.

Recent technologies have enabled extensive genomic sequencing in an effort to understand the impacts of gene variants on patients. While sequencing can identify variants to help diagnose patients, there are thousands of gene variants with unknown clinical significance termed variants of unknown significance (VUS) (Kroncke, Vanoye, Meiler, George, & Sanders, 2015). These gene variants are often different by just one base pair, referred to as single-nucleotide polymorphisms (SNPs). Structural and functional studies of these proteins can help bridge the gap between genes and patients. This not only provides a much better understanding regarding the mechanism of disease and stability of proteins, but it provides opportunities for developing personalized treatments.

1.2 Computational modeling of proteins

The number of known protein structures will always lag far behind that of known protein sequences. Structural modeling of proteins is crucial to filling in the gap of knowledge between protein sequence and structure. Computational structural biology methods rely heavily on experimental methods for developing and improving predictions regarding protein structure. Methods such as X-ray crystallography, nuclear magnetic resonance spectroscopy (NMR), cryo-election microscopy (cryo-EM), and electron paramagnetic resonance (EPR) provide valuable structural characterization of proteins. However, for proteins that are not amenable to these such approaches, even mutational data can provide insight that can aid computational methods. Therefore, continued experimental efforts act as an iterative feedback for improving computational methods. Additionally, computational predictions can provide feedback for which experimental approaches are worth seeking. By filling in the knowledge gap, computational methods are the best way to accelerate the understanding of the impact of sequence variation on protein structure and even protein function.

For protein sequences with unknown structures, *de novo* modeling approaches involve libraries of fragments from known structures that are used along with perturbations and Monte Carlo based sampling to rapidly sample and calculate optimal physical interactions towards an energetically favorable state. Because of the large conformational space sampled during *de novo* modeling, these calculations are exhaustive of time and resources. Sparse experimental restraints can greatly reduce the conformational sampling space. However, proteins that have related proteins of known structure can leverage the backbone of the known structure as a template for modeling. This is called comparative modeling, or in the case of homologous proteins, homology modeling.

In comparative modeling, the sequence of the target protein is threaded onto the backbone of the protein of known structure through the use of a sequence alignment. In Rosetta, fragment libraries are used to sample the regions that are less certain such as flexible loops, unstructured regions, or regions of low sequence identity or coverage. These models are then clustered to identify the largest populations with high structural similarity. Because Rosetta uses a Monte Carlo approach, with enough sampling, the models that are energetically favorable and seen frequently are ideal.

Recently, Rosetta has enabled the use of multiple templates for comparative modeling. RosettaCM allows the user to input additional templates in an effort to increase the conformational search space beyond that which would exist from a single template alone. Additionally, the inclusion of multiple templates has been shown to generate more accurate loops (Song et al., 2013). Because previous methods of loop modeling involved two anchor points and increasing degrees of freedom with each addition of an amino acid (Combs et al., 2013), the use of multiple loops from multiple templates could restrict the sampling space into a more reasonable range of conformations.

1.3 Computational modeling of membrane proteins

Membrane proteins represent nearly 30% of open reading frames; however, they are a particularly challenging class of proteins to study due to their complicated lipid environment, limited number of structures available, and, often, the low resolution of available structural models. Membrane proteins are particularly difficult to structurally characterize because of their inherent flexibility and resulting conformational dynamics requiring stabilization prior to structural characterization (J U Bowie, 2001). Additionally, other considerable challenges are that over-expression of membrane proteins can be toxic to cells (Wagner, Bader, Drew, & de Gier, 2006) and that membrane proteins require the use of membrane memetics. These pose challenges both experimentally as well as computationally due to the resulting limited structural knowledge-base for membrane proteins from which to derive accurate energy functions. Nevertheless, membrane proteins remain a very important class of proteins to study as they are approximately 60% of drug targets (Arinaminpathy, Khurana, Engelman, & Gerstein, 2009) and diseases such as Alzheimer’s disease, long-QT syndrome (Bokil, Baisden, Radford, & Summers, 2010; J. Wu, Ding, & Horie, 2016), Charcot-Marie-Tooth disease, and cystic fibrosis can be associated with membrane proteins (C R Sanders & Myers, 2004).

RosettaMembrane has been the method used to model helical transmembrane proteins for several years. RosettaMembrane consists of both low-resolution (Yarov-Yarovoy, Schonbrun, & Baker, 2006) and high-resolution (Barth, Schonbrun, & Baker, 2007) scoring functions that were developed to describe how the protein interacts with the membrane environment. In addition to score terms that describe the membrane environment, the RosettaMembrane energy function contains the same score terms as Score12, the default soluble scoring function at the time of the study. Recently, RosettaMP, a new framework for modeling membrane proteins in Rosetta, was developed to facilitate communication between model sampling and scoring (Alford et al., 2015). Work is ongoing to adapt existing protocols to be compatible with RosettaMP.

The studies herein were completed using RosettaMembrane. RosettaMembrane implicitly models the membrane bilayer (Figure 1.1) meaning that instead of full atom representations of the atoms in the lipids, the energy function contains information regarding which amino acids are likely to interact with the lipid based on the hydrophobicity. This implicit modeling approach is ideal because it reduces the computational resources, including time and memory usage, required to perform such a simulation of a membrane protein in a native-like environment. However, it should also be noted that there are limitations for this approach. The most obvious of these limitations are the rigid definitions for the distances of the hydrophobic layers, thus the predefined thickness of the bilayer. The statistics for RosettaMembrane were derived with these definitions in mind; however, each structure in the dataset was not generated using lipids of the same length. Additionally, the native membrane that membrane proteins exist in can be very different, meaning membrane proteins have been designed for various membrane environments. For example, the lipid composition of bacteria versus humans is very different, and sometimes there are inner and outer membranes. Outer membrane proteins tend to have a narrower membrane than inner membrane proteins.

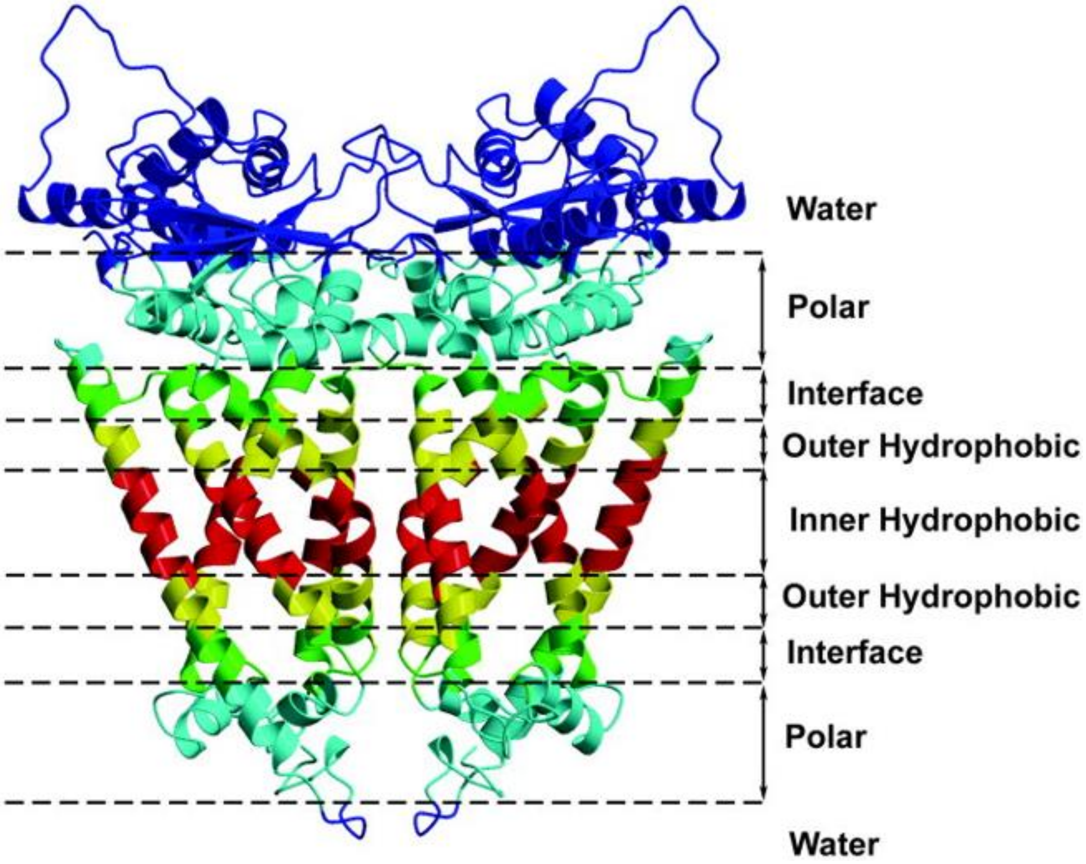


Figure 1.1. Hydrophobic layers defined by RosettaMembrane. The inner and outer hydrophobic regions span 24 Angstroms while the interface spans 6 Angstroms on either side, and the polar region spans 12 Angstroms on either side. Reprinted from, Proteins: Structure, Function and Bioinformatics, Vol 62, Vladimir Yarov-Yarovoy, Jack Schonbrun, and David Baker, Multipass membrane protein structure prediction using Rosetta, 4, 2005, with permission from John Wiley and Sons (Yarov-Yarovoy et al., 2006).

Many membrane proteins, such as ion channels, are homo-oligomeric. As the number of subunits increases, it becomes more computationally exhaustive of time and resources to model even the simplest of interactions because of the large size of the complex. Previously, Rosetta2 was limited in its ability to model large symmetric complexes (André, Strauss, Kaplan, Bradley, & Baker, 2008). In 2011, DiMaio et al. introduced a new mode in Rosetta to model symmetric proteins called Rosetta Symmetry (Dimaio, Leaver-fay, Bradley, Baker, & Andre, 2011). This allowed protocols to sample and score large, symmetric complexes much more quickly and with less memory usage as this approach samples only symmetric degrees of freedom, greatly reducing the search space. The underlying assumption, however, is that the interactions between all subunits are symmetric. The current implementation of Rosetta Symmetry can create complex symmetric assemblies through the use of a symmetry definition file for a symmetric or nearly symmetric structure from the PDB. In the case of *de novo* folding, a symmetry definition file must be generated from scratch.

Rosetta Symmetry works well for homo-oligomeric systems, and hetero-oligomeric systems can sometimes be models using Rosetta Symmetry if the correct asymmetric unit is identified. However, another type of symmetry exhibited in membrane proteins is internal symmetry. This is symmetry within the same chain of the protein that is likely the result of gene duplication, fusion, and diversification events (Duran & Meiler, 2013). Rosetta Symmetry does not handle internal symmetry; however, for design applications, the Favor Symmetric Sequence mover was developed to force constraints on N number of symmetric fragments within the sequence.

1.4 Computational design of proteins

Computational protein design has the potential to contribute to various fields including protein-ligand design (Allison et al., 2014; Tinberg et al., 2013), protein therapeutics, and materials science (King et al., 2014; King & Lai, 2013; King et al., 2012). Protein design is a unique protocol in that instead of finding the optimal conformation of a particular sequence, it aims to determine an optimal sequence for a given conformation. For this reason, it is often termed the “inverse protein folding problem” (Kaufmann, Lemmon, Deluca, Sheehan, & Meiler, 2010).

Protein scaffolds have been defined as a frame-work that can handle mutations to make variants with different functions (Binz, Amstutz, & Plückthun, 2005). Generally, there are two main design strategies utilizing a scaffold: design for stability and design for function. The stability protocol considers the entire protein for design, and the score terms of interest are generally focused on improved packing. The design for function protocol is usually a localized design, centered on a specific region, domain, pocket, etc., of a protein with a focused energy function that governs precise interactions, such as electrostatics or hydrogen bonding.

Protein design involves iterative optimization of sequence and structure. During the fixed backbone side-chain optimization step, sequence space is sampled simultaneously with side-chain conformational space using Monte Carlo-simulated annealing by exchanging all possible amino acids at user-specified designable positions while evaluating the predicted energy (Brian Kuhlman et al., 2003). This is followed by flexible backbone minimization to optimize the model. The first successful use of *de novo* Rosetta Design produced a sequence for a fold not seen in the PDB (Brian Kuhlman et al., 2003). The experimentally determined structure had an RMSD of 1.1 Å from the computationally design model. An example tutorial for protein design, protein_design, is provided in the Appendix A.

Design for stability

Protein stability can be affected by a single-point mutation. Kellogg et al. evaluated several protocols with varying levels of flexibility and sampling and determined one method in particular to be useful for single-point mutations (Kellogg, Leaver-Fay, & Baker, 2011). This method was made into the application ddg_monomer. When ddg_monomer was tested on a set of 1210 single-point mutants from the ProTherm database, the correlation of predicted ddGs to experimental ddGs was 0.69 while the stability classification accuracy was 0.72.

While ddg_monomer is a tool for predicting how a single-point mutation affects the stability of a protein, RosettaVIP (void in packing) is a design strategy that has been developed to identify single-point mutations that could improve the stability of a protein (Borgo & Havranek, 2012). When Borgo et al. fully designed proteins, they found that the hydrophobic cores of the designed models were poorly packed when compared to their respective native proteins. RosettaVIP was able to identify packing deficiencies and sample a much smaller sequence space to fill the void in packing, resulting in a more stable design.

Design for functionality

In addition to stabilizing monomeric proteins, Rosetta Design can be used to design interfaces between proteins. Fleishman et al. established a dock design protocol that optimizes the sequence of a protein to bind a surface patch of a target protein during design. Docking was used to optimize the positioning of the interacting proteins at the interface. Experimentally determined structures had an interface very similar to those of the designed models (Fleishman, Whitehead, Ekiert, Dreyfus, & Jacob, 2012).

Other types of interfaces of interest for design applications are protein–small molecule interfaces. Tinberg et al. (Tinberg et al., 2013) provided a great example of using Rosetta Design to design for affinity as well as stability. First, RosettaMatch (Zanghellini et al., 2006) was used to find a stable scaffold for design for binding a particular small molecule. Next, Rosetta Design was used to maximize the binding affinity between the protein and small molecule. Finally, a second round of design was used to minimize destabilization due to mutagenesis in the first round. To ensure these mutations were meaningful, design was guided by a multiple-sequence alignment. The resulting most energetically favorable model was the highest-affinity binder in experimental studies and had a cocrystal structure that agreed with the computational model.

Most design algorithms in Rosetta are performed while considering a single fixed backbone structure. Recently, efforts to consider several structures during the design process have been undertaken to tackle more difficult design problems. A generalized multistate design protocol was introduced in 2011 (A. Leaver-Fay, Jacak, Stranges, & Kuhlman, 2011) to help in cases in which design should occur to satisfy multiple conformations or to design specificity toward one state and negative design against other states. Willis et al. showed that RosettaMultistateDesign was capable of predicting residues that were important for polyspecificity when designing the heavy-chain variable region of an antibody (Willis, Briney, DeLuca, Crowe, & Meiler, 2013). Sevy et al. introduced a new approach to multistate design that accelerates the process of multistate design by reducing the sequence search space (Sevy, Jacobs, Crowe, & Meiler, 2015), allowing more complex backbone movements to be incorporated into a design protocol.

1.5 Machine learning methods to aid optimization of energy functions

Rosetta uses a combination of terms based off statistics that physical interactions as well as internal energies associated with probabilities. As our knowledge of proteins continues to grow, these methods should be evaluated for continued accuracy. Empirical data can be leveraged to improve the performance of energy functions (Guerois, Nielsen, & Serrano, 2002). Furthermore, machine learning methods can aid in energy function optimization (Lise, Archambeau, Pontil, & Jones, 2009). Multiple linear regression (MLR), as the name implies, aims to determine a linear relationship between two or more variables and a continuous outcome. Variable selection in MLR should not be completed automatically, but rather manually in identifying the most pertinent predictors (Eberly, 2007). Ridge regression, however, uses L2-regularization which means it aims to reduce the residual sum of squares from a linear regression. It is considered to be a shrinkage method, but is not parsimonious (Zou & Hastie, 2005) meaning that it does not simplify the model to only the relevant predictors (Eberly, 2007). Lasso (Tibshirani, 1996) is an approach that uses L1-regularization and acts similarly to penalized least squares. Like ridge regression, lasso is a shrinkage method; however, unlike ridge lasso does employ automatic variable selection to create a parsimonious model (Zou & Hastie, 2005). Finally, elastic net regression (Zou & Hastie, 2005) is also a shrinkage method that employs automatic variable selection, but it is actually a regularization strategy that combines ridge and lasso regression penalties and can be tuned for the specific needs of the user.

CHAPTER 2

COMPUTATIONAL DESIGN OF MEMBRANE PROTEINS USING ROSETTAMEMBRANE

This chapter includes published work from:

Duran and Meiler, 2017

"Computational Design of Membrane Proteins using RosettaMembrane" in the journal Protein Science (Duran and Meiler, 2017). Reprinted from, Protein Science, Vol 27, Amanda M. Duran and Jens Meiler, Computational Design of Membrane Proteins using RosettaMembrane, 1, 2017, with permission from John Wiley and Sons. Commentary and figures have been added in addition to the original full article and supplementary material.

Author contributions: I contributed the vast majority of the text and designed experiments under the mentorship of Jens Meiler. I designed experiments and analysis, generated all of the data, designed and created all of the figures, as well as created all of the tables, and protocol captures.

Abstract

Computational membrane protein design is challenging due to the small number of high-resolution structures available to elucidate the physical basis of membrane protein structure, multiple functionally important conformational states, and a limited number of high-throughput biophysical assays to monitor function. However, structural determination of membrane proteins has made tremendous progress in the past years. Concurrently the field of soluble computational design has made impressive inroads. These developments allow us to tackle the formidable challenge of designing functional membrane proteins. Herein, Rosetta is benchmarked for membrane protein design. We evaluate strategies to cope with the, often, reduced quality of

experimental membrane protein structures. Further, we test the usage of symmetry in design protocols, which is particularly important as many membrane proteins exist as homo-oligomers. We compare a soluble scoring function with a scoring function optimized for membrane proteins, RosettaMembrane. Both scoring functions recovered around half of the native sequence when completely redesigning membrane proteins. However, RosettaMembrane recovered the most native-like amino acid property composition. While Leucine was overrepresented in the inner and outer-hydrophobic regions of RosettaMembrane designs, it resulted in a native-like surface hydrophobicity indicating that it is currently the best option for designing membrane proteins with Rosetta.

2.1 Introduction

Membrane proteins comprise approximately 30% of all open reading frames of known genomes (Tan, Hwee, & Chung, 2008). However, in the Protein Data Bank (PDB) (Berman et al., 2000) membrane proteins continue to be underrepresented. Membrane proteins, many of which are alpha-helical, include classes of proteins that are responsible for functions such as channel and transporter proteins, or signal transduction in receptors. Additionally, more than 60% of drugs target membrane proteins (Arinaminpathy et al., 2009), therefore insight to the structure and function of membrane proteins is valuable for the development of treatment strategies for diseases such as cancer (Jura et al., 2009; Mark A Lemmon & Schlessinger, 2010), cardiac arrhythmia (Moss & Kass, 2005; Q. Wang et al., 1996), schizophrenia (Conn, Lindsley, & Jones, 2008; Meisenzahl, Schmitt, Scheuerecker, & Möller, 2007), and many more.

Membrane proteins are difficult to structurally characterize because over-expression of the protein is typically toxic to bacterial cells (Arinaminpathy et al., 2009; Wagner et al., 2006), resulting in low protein yields. Additionally, membrane proteins must be reconstituted into micelles, bicelles, nanodisks, or liposomes to provide a native-like environment. Often an extensive screening for the optimal detergents and lipids is needed for maximal solubility and stability (Arinaminpathy et al., 2009). However, membrane mimetics can have a destabilizing effect on the structure of the membrane protein. Finally, membrane proteins have inherent conformational dynamics (J U Bowie, 2001), which often requires engineering of a thermodynamically stabilized mutant for structural studies.

Challenges in membrane proteins structure determination have resulted in limited available structural information for membrane proteins. In the PDB less than 3% of structures are membrane proteins. Approximately 700 unique membrane proteins structures have been deposited in the PDB (Berman et al., 2000; White) to date, which is a vast improvement to the structural information that was available nearly a decade ago, but far away from complete coverage of membrane protein folds. Computational modeling by *de novo* and comparative modeling can provide structural insights to membrane proteins without experimentally determined structures. However, in order to obtain more accurate models of membrane proteins, more high-resolution structures are needed to understand the physical basis of membrane protein folding and derive more accurate scoring functions.

The PDB is a depository of structure files which provides the knowledge-base for proteins of known structure to drive the development of accurate scoring functions and for rigorous testing of newly developed computational methods. As a result, methods for computational membrane protein structure prediction lag behind considerably, and computational design of function – an area of great success for soluble proteins in the past ten years – is largely absent for membrane proteins. However, the structures of many important membrane proteins have been determined at a stunning rate over the past ten years (Loll, 2003;

Charles R. Sanders & Sönnichsen, 2006; White, 2004; Wiener, 2004; H. Wu et al., 2014) increasing the knowledge-base for scoring function development, providing higher-resolution structures for benchmarking, and yielding templates of important membrane protein classes to begin engineering.

Computational protein design is a difficult problem due to the large number of possible sequences for a particular protein backbone. Computational design tools aim to rapidly evaluate possible interactions between side-chains to determine likely sequences of low-energy. Some methods have an emphasis on calculations that evaluate electrostatics and solvation of a side-chain in its environment (Marshall, Vizcarra, & Mayo, 2005; Pokala & Handel, 2004; Vizcarra et al., 2007). However the environment for membrane proteins is complicated and consideration for differences in membrane protein folding should be taken into account (Senes, 2011). Additionally, these methods fail to consider features that many membrane proteins have that are important for function and membrane solubility (Perez-aguilar & Saven, 2012). Tools have been developed empirically to overcome the shortcomings of these calculations for membrane proteins. Walters and co-workers developed idealized geometries and position-specific sequence propensities for helix-packing motifs most commonly seen in membrane proteins (Walters & Degrado, 2006). Senes and co-workers developed a potential based on the membrane depth dependent propensities of amino acids to predict if sequences would insert in the membrane (Senes et al., 2007).

The Rosetta software suite for biomolecular modeling and design has an impressive track record in the design of soluble proteins including the design of a *de novo* protein fold (Brian Kuhlman et al., 2003), enzymes (Jiang et al., 2008; Korkegian, Black, Baker, & Stoddard, 2012; Röthlisberger et al., 2008; Siegel et al., 2010), protein-protein interactions (Fleishman et al., 2011; Joachimiak, Kortemme, Stoddard, & Baker, 2006; Kortemme et al., 2004; Strauch, Fleishman, & Baker, 2014), protein-small molecule interfaces (Tinberg et al., 2013), and self-assembling materials (Eisenbeis et al., 2012; Fortenberry et al., 2011; King et al., 2014; King et al., 2012). The Monte-Carlo search strategy that allows changes to amino acid identities during sampling combined with a multi-scale knowledge-based scoring function that is optimized to capture structural features at the protein fold level as well as at atomic detail create a unique ability to engineer proteins that set Rosetta apart from other computational strategies. The scoring function and sampling methods used by Rosetta, however, are tailored for the needs of soluble-protein modelers; despite some progress in adapting it for membrane proteins, modeling abilities in membrane proteins lag behind those of soluble proteins.

Rosetta's knowledge-base has been derived in large part using statistical analysis of geometric arrangements within structures reported in the PDB. For protocols involving minimization, backbone torsion angles are randomly perturbed and rotational side-chain conformations are optimized for interactions including van der Waals, electrostatics, and hydrogen-bonding (Rohl, Strauss, Misura, & Baker, 2004; Schueler-furman, 2005). Interactions with the solvent are modeled implicitly by determining the likelihood of a certain amino acid type being in a particular burial state. Monte Carlo sampling combined with knowledge-based scoring functions are parameterized so that resulting models exhibit properties of proteins of known structure (Kaufmann et al., 2010). The membrane protein scoring function, RosettaMembrane, additionally considers the likelihood of an amino acid being in a particular membrane environment and burial state (Barth et al., 2007; Yarov-Yarovoy et al., 2006).

Previously, Rosetta was used to completely redesign 108 soluble proteins. Designs recovered 51% of the native sequence in the protein core. The terms involving the Lennard-Jones potential and Lazaridis solvation drove the scoring function to design sequences that were native-like (B. Kuhlman & Baker, 2000). In the current study, complete redesign of membrane proteins was benchmarked using RosettaMembrane, (Barth et al., 2007; Yarov-Yarovoy et al., 2006) and for comparison, the Rosetta scoring function for soluble proteins "Talaris" (Andrew Leaver-Fay et al., 2013; O'Meara et al., 2015). Many membrane proteins like channels and transporters are functional homo-oligomers. In order to model membrane proteins in their native states and obtain correct representation of the surfaces and interfaces, one must consider how such a protein might symmetrically assemble. Therefore, homo-oligomeric membrane proteins were modeled with Rosetta Symmetry (Dimaio et al., 2011) which is able to sample and rapidly score these larger assemblies while considering interface interactions between subunits.

One important application of membrane protein design is thermostabilization to facilitate structural characterization. Membrane proteins often require flexibility in order to perform their function (J U Bowie, 2001; K.-Y. M. Chen, Zhou, Fryszczyn, & Barth, 2012). By stabilizing a single conformation, one can reduce the flexibility, thus yielding a more ideal protein for experimental structure determination. Computational methods like Rosetta Design can propose an optimal sequence for a particular conformation by using information from known membrane protein structures. The proposed mutations in the optimized sequence could presumably lead to a thermostabilized membrane protein.

This study evaluates how well Rosetta recovers native sequences for membrane proteins when fully redesigned. We find that the methods for minimizing the structure prior to design play a role in native sequence recovery. Additionally, total sequence recovery was similar among different scoring functions; however, unsurprisingly, RosettaMembrane performed best in designing membrane proteins with native-like properties.

2.2 Results and discussion

Initial energy minimization improves membrane protein design for low-resolution experimental structures. When benchmarking protein design algorithms, the question arises whether or not to minimize the starting experimental structure with the respective scoring function. The argument against minimization is that adjustment of backbone and side-chain coordinates to minimize energy will imprint a ‘memory’ for the correct amino acid into the backbone coordinates. The native amino acid will score better as the backbone is positioned in such a way that the native amino acid can be placed in an energy minimum for the scoring function used. As a result, artificially inflated sequence recovery values might be reported. The counter argument is that energetic frustrations such as clashes in the starting structures that could be relieved with energy minimization might cause the design algorithm to prefer smaller, non-native amino acids in these locations. This is a particular concern for membrane proteins where many structures of reduced resolution are deposited in the PDB. For soluble proteins the latter problem can be easily circumvented by benchmarking only on highest-quality protein structures with resolutions better than 2Å (B. Kuhlman & Baker, 2000). However, the sparseness of membrane proteins in the PDB requires usage of lower-quality structures. Accordingly, we developed a protocol that applies an initial moderate energy minimization to resolve frustrations but avoids an aggressive optimization that might result in inflated sequence recovery values.

Without initial energy minimization, the sequence recovery of fully redesigned membrane proteins correlates with the resolution of the input structure such that low-resolution structures tend to have reduced sequence recovery (Fig. 2.1). For monomeric membrane proteins, the Pearson’s correlation coefficient is strongly negative at -0.75 (R^2=0.56). For homo-oligomeric membrane proteins, the Pearson’s correlation coefficient is -0.47 (R^2=0.22). When extrapolated, sequence recovery for a structure with 0 Å resolution is approximately 57% and 45% for monomeric and homo-oligomeric membrane proteins, respectively. Upon energy minimization, the correlation is absent independent of the Rosetta minimization protocol employed (Fig. 2.1). At the same time, we observe that average sequence recovery for monomeric membrane proteins improves from 31% without backbone energy minimization to 38%, 49%, 48%, and 54% with the four Rosetta minimization protocols minimization with constraints (MWC), constrained to start coordinate relax(CSC), FastRelax, and Dualspace. For homo-oligomeric membrane proteins average sequence recovery starts at 36% and results in 35%, 48%, 48%, and 55%, respectively.

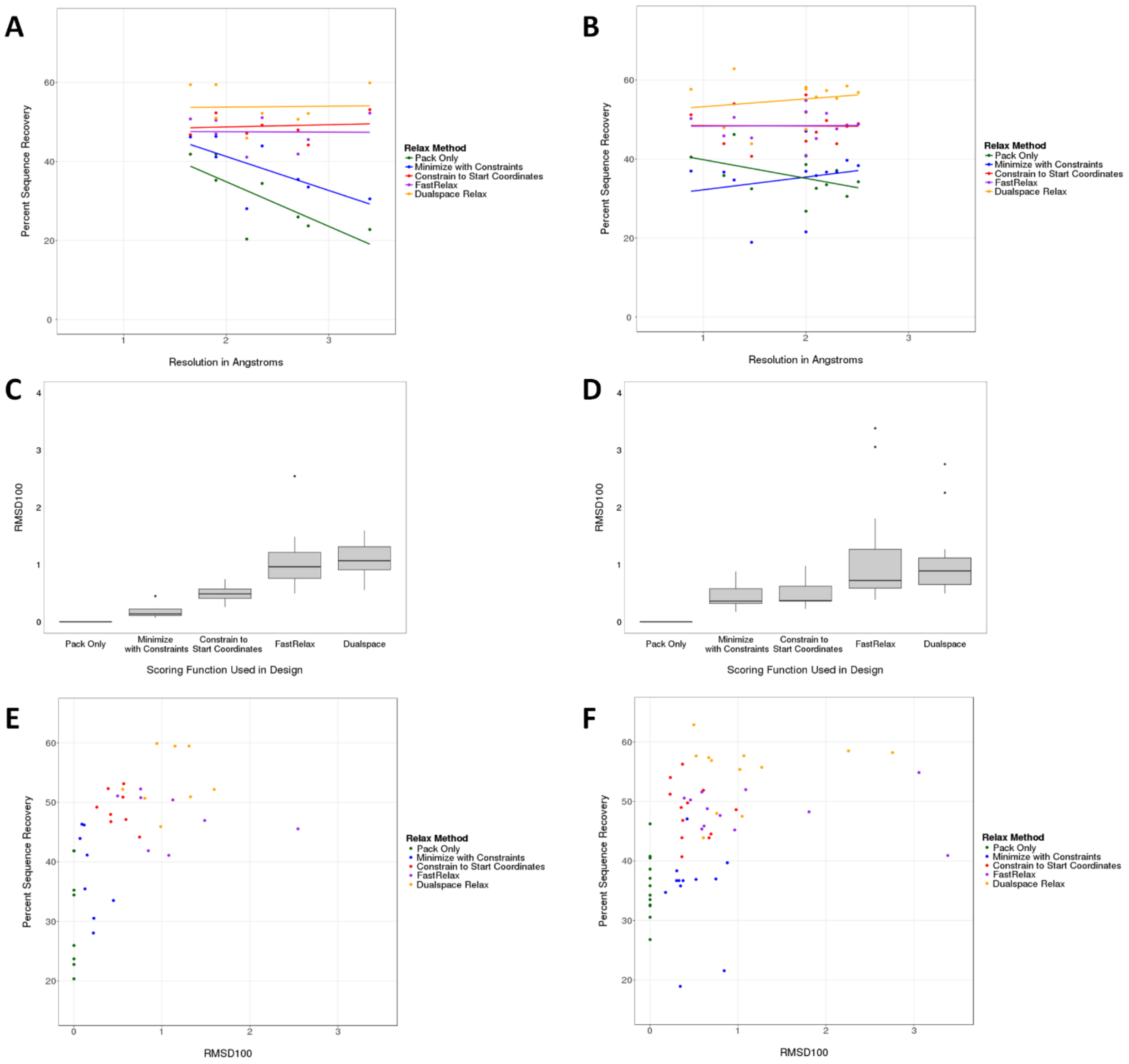


Figure 2.1. Sequence recovery for monomeric (A,C,E) and homo-oligomeric (B,D,F) sets. Various minimization methods were used to prepare crystal structures as input for Rosetta. When considering sequence recovery by resolution (A,B), pack-only and less stringent minimization (MWC) result in a correlation. CSC, FastRelax and Dualspace minimization resulted in a consistently high sequence recovery independent of the initial structure resolution. The normalized, average movement of minimized structures for each minimization protocol (C,D) showed that FastRelax and Dualspace tend to move the protein further away from the starting structure. When examining sequence recovery by average movement (E,F), we find that pack-only and MWC had a larger range over low sequence recovery whereas protocols that allowed more movement during minimization, CSC, FastRelax, and Dualspace, yielded more consistently high sequence recovery rates. FastRelax and Dualspace in some cases moved the backbone further than 1 Å.

Our analysis indicates that both initial concerns have merit. A clear correlation between model resolution and sequence recovery is observed. Upon energy minimization this correlation vanishes. However, aggressive minimization protocols such as Dualspace, may inflate sequence recovery beyond what would be expected from the extrapolation to a membrane protein model with 0 Å resolution. Additionally, FastRelax and Dualspace move the protein beyond 1 Å RMSD_{100}(Carugo & Pongor, 2008), whereas CSC attains similar average sequence recovery rates despite movement of less than 1 Å RMSD_{100} during minimization (Fig. 2.1 E,F). We conclude that CSC, the limited energy minimization with a constraint to starting coordinates, is a good compromise to avoid over- and under-reporting algorithm accuracy.

Interestingly, for the highest resolution monomer, 2xov, the pack-only preparation resulted in an average sequence recovery of 42%, while MWC was 46%. Using the recommended CSC protocol, the average sequence recovery is 47% (Fig. 2.2A). This indicates that any major clashes that typically lessen sequence recovery were resolved prior to minimization. Additionally, for the lowest resolution monomer, 4a2n, the pack-only and MWC preparations resulted in sequence recoveries of 23% and 31%. However, after more flexible minimization strategies, CSC, FastRelax, and Dualspace, sequence recoveries increased to 53%, 52%, and 60%, respectively, indicating that perhaps major clashes were resolved once more flexibility was introduced.

For homo-oligomers, this analysis had a different finding. While most of the homo-oligomeric structures were of high-resolution more stringent minimization-CSC, FastRelax, or Dualspace- was required in order to achieve higher sequence recovery percentages (Fig. 2.2B). This is likely due to an option used during symmetric relax which enables rigid body movement (see protocol capture in Appendix B). Whereas the pack-only preparation would only move side-chains while MWC might constrain the minimization without considering the placement of the rigid bodies with respect to each other.

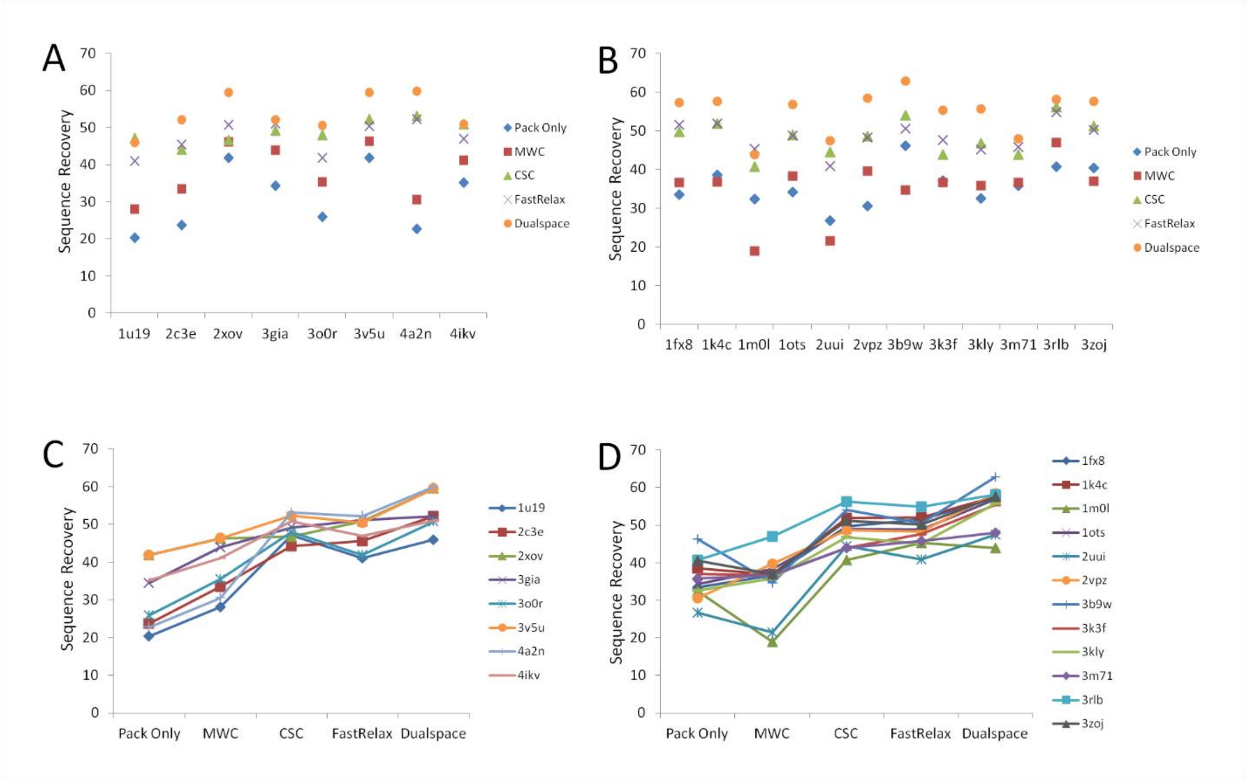


Figure 2.2. Percent sequence recovery of individual proteins and their respective minimization method. Monomeric proteins consistently had pack only as the lowest performance for sequence recovery (A). Oligomeric proteins sometimes had worse performance from MWC due to lack of rigid body movement (B). Sequence recoveries are plotted for each increase of minimization stringency for each individual monomer (C) and oligomer (D).

Sequence recovery is highest in the core of the protein

To evaluate the performance of RosettaMembrane (Barth et al., 2007; Yarov-Yarovoy et al., 2006) redesigning membrane proteins, we compared the performance of the soluble scoring function Talaris (Andrew Leaver-Fay et al., 2013; O'Meara et al., 2015). The largest differences in score terms between RosettaMembrane and Talaris are the membrane-related terms that describe the membrane-specific environment (including burial state) and differences in solubility. We used Talaris to test how well Rosetta can design native-like membrane proteins in the absence of these membrane protein specific terms.

For both monomeric and homo-oligomeric sets, average core sequence recovery was higher with the Talaris scoring function when compared to RosettaMembrane (Fig. 2.3B). Talaris had an average core sequence recovery of 63% and 65% for monomeric and homo-oligomeric datasets, respectively, compared to RosettaMembrane with 52% and 55%. A Wilcoxon signed rank test determined that the difference in percent core sequence recovery between RosettaMembrane and Talaris was significant for both monomers and homo-oligomers (z=2.49, p=0.013 ; z=3.04, p=0.002). Residues in the core are less influenced by the membrane environment than surface residues that are likely interacting with the lipid bilayer. Therefore, sampling and scoring in the core is driven by van der Waals packing interactions that are similar for membrane and soluble proteins. RosettaMembrane was derived from score12, the scoring function that preceded Talaris. Membrane specific scoring terms were added. Meanwhile, score12 evolved to Talaris through improvement of the electrostatic term, hydrogen bond terms, and reference energies (Andrew Leaver-Fay et al., 2013; O'Meara et al., 2015). These changes give rise to the improved core sequence recovery observed with the Talaris energy function (Fig. 2.3) as amino acid interactions are modeled more precisely.

Surface sequence recovery for monomers improved in designs using RosettaMembrane (40%) when compared with Talaris (34%, Fig. 2.3A). However for homo-oligomers, the average surface sequence recovery was 35% for both RosettaMembrane and Talaris. A Wilcoxon signed rank test determined that the difference in percent surface sequence recovery between RosettaMembrane and Talaris was significant for monomers (z=2, p=0.046), and not significant for homo-oligomers (z=0.69, p=0.492). RosettaMembrane models a membrane of fixed thickness implicitly. The higher surface sequence recovery observed with RosettaMembrane is attributed to the membrane-specific score terms that adjust the polarity of the environment (Fig. 2.3). However, the improvement in sequence recovery on the surface within RosettaMembrane when compared to Talaris is only moderate. We attribute this to the absence of specific interactions on the surface of the proteins that allow for the presence of only one specific amino acid.

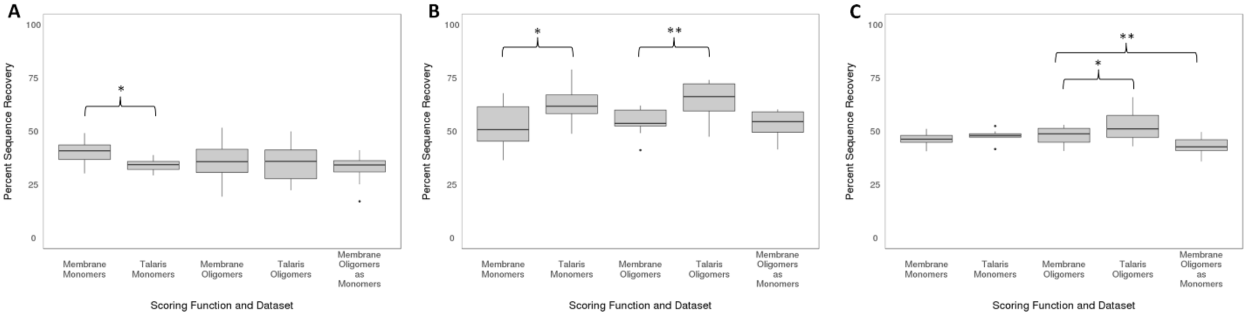


Figure 2.3. Percent of native sequence recovery for design of membrane proteins using various scoring functions. Boxplots show recovery of native sequence on the surface (A) and core (B) of the protein. RosettaMembrane (Membrane) designed monomeric proteins have a higher average surface recovery than Talaris. The total sequence recovery (C) shows that both scoring functions evaluated appear to have similar native sequence recovery percentages; however, core recovery is higher in Talaris which likely contributes to the total sequence recovery. When homo-oligomers were modeled as monomers, the total average sequence recovery rate was approximately 5% lower than the sequence recovery rate for design considering homo-oligomeric interfaces.

Finally, when evaluating the total sequence recovery in monomers, RosettaMembrane had an average of 46% while Talaris had an average of 48%. In homo-oligomers, the average total sequence recovery was calculated to be 48% for RosettaMembrane and 53% for Talaris. A Wilcoxon signed rank test revealed that the difference in percent total sequence recovery between RosettaMembrane and Talaris was not significant for monomers (z=0.81,p=0.421) while it was significant for homo-oligomers (z=2.1, p=0.036). When homo-oligomers were designed as monomers, the average percent native sequence recovery for surface (Fig. 2.3 A) and core (Fig. 2.3 B) were similar to that of homo-oligomers designed in a homo-oligomeric state. A Wilcoxon signed rank test confirmed there was no significant difference (z=1.24, p=0.217; z=0.33, p=0.739). However, the difference in percent total sequence recovery was found to be significant (z=2.77, p=0.006). This is likely due to a subset of residues not classified as either surface (less than or equal to16 neighbors within a c-beta distance of 10Å) or core residues (more than 24 neighbors within a c-beta distance of 10Å) contributing to the difference in percent total sequence recovery differences.

Additionally, we calculated the sequence recovery based on secondary structure element. Over 4,000 residues were annotated as helical and over 1,000 residues were annotated as coil, while nearly 30 residues were annotated as strand (Fig 2.4). Talaris appeared to have a higher sequence recover in helical regions compared to membrane proteins. But this analysis of sequence recovery by secondary structural annotations is too crude of an assessment to draw conclusions.

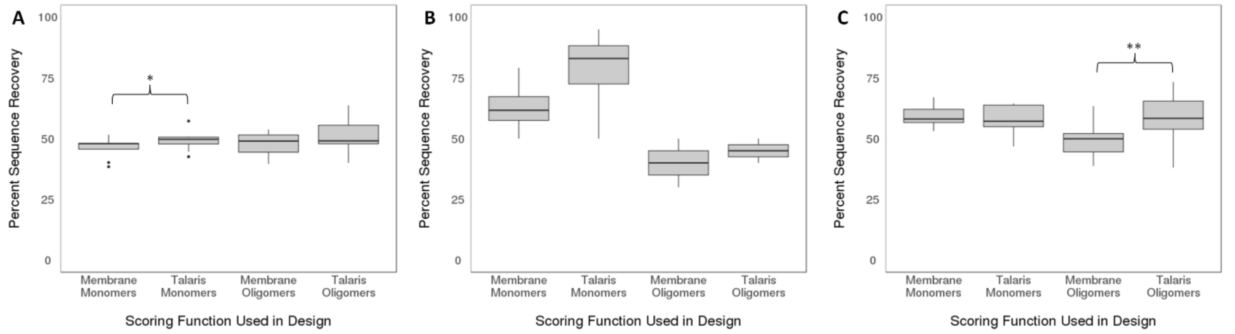


Figure 2.4. Percent sequence recovery reported by secondary structure element. Sequence positions were annotated as helix (A), strand (B), or coil (C). The average sequence recovery is reported for the appropriate members of the dataset. Because the datasets are of alpha helical proteins, the amount of residues considered for sequence recovery of helical regions is over 4,000 for each dataset, while only approximately 30 residues represent strands. Therefore, significance of the difference in average sequence recoveries for strands cannot be determined. Coils are represented by over 1,000 residues for each dataset.

We selected top models as representatives to better understand which residues were designed by mapping those residues on the structure. For both scoring functions, designed residues tended to be on the surface where residues would be lipid-exposed (Fig. 2.5), in monomers (Fig. 2.6), and homo-oligomers (Fig. 2.7). Residues at the interface of subunits (Fig. 2.5 C,E and Fig. 2.7 C,F) appear to be designed less frequently and result in core-like recovery indicating that design considers neighboring residues from different chains when using Rosetta Symmetry.

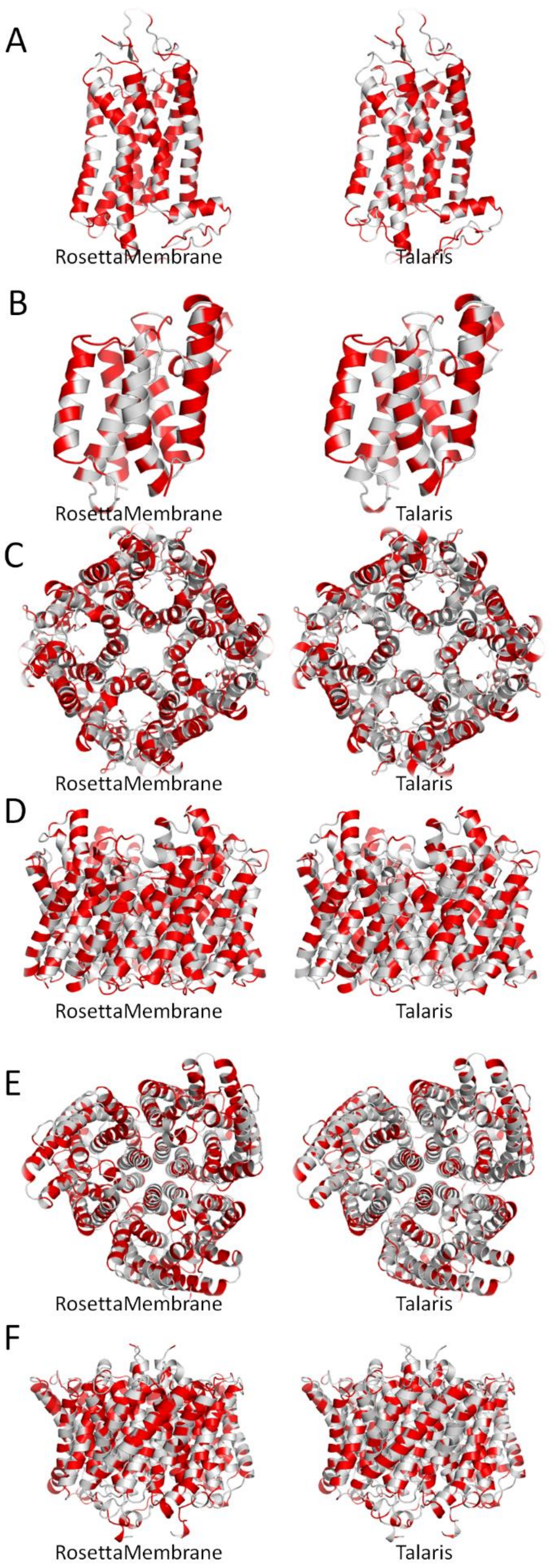

A
RosettaMembrane
Talaris
B
RosettaMembrane
Talaris
C
RosettaMembrane
Talaris
D
RosettaMembrane
Talaris
E
RosettaMembrane
Talaris
F
RosettaMembrane
Talaris

Figure 2.5. Designed residues mapped on models. Top models were selected as representative models for visualizing designed sites of 1u19(A), 2xov(B), 1fx8 (C-top down,D), and 3b9w (E-top down,F). Red indicates sites that have been designed while gray represents sites that have maintained the wild-type residue. Surface residues that would be lipid-exposed have the tendency to be designed for both scoring functions. Residues at the interface of subunits (C,E) appear to be designed less frequently.

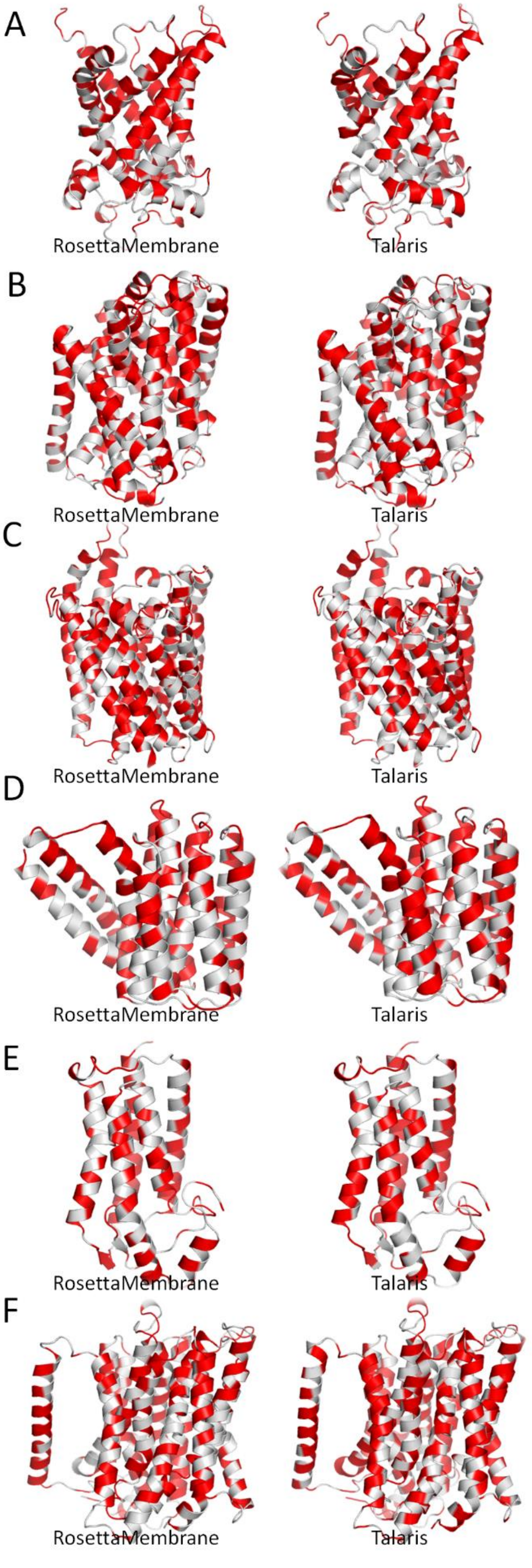

A
RosettaMembrane
Talaris
B
RosettaMembrane
Talaris
C
RosettaMembrane
Talaris
D
RosettaMembrane
Talaris
E
RosettaMembrane
Talaris
F
RosettaMembrane
Talaris

Figure 2.6. Designed residues mapped on models of monomers. Top models were selected as representative models for monomers visualizing the designed sites of 2c3e (A), 3gia (B), 3o0r (C), 3v5u (D), 4a2n I, and 4ikv (F). Red represents sites that have been designed while gray represents sites that have retained the native amino acid. Designed residues tend to be surface residues regardless of scoring function.

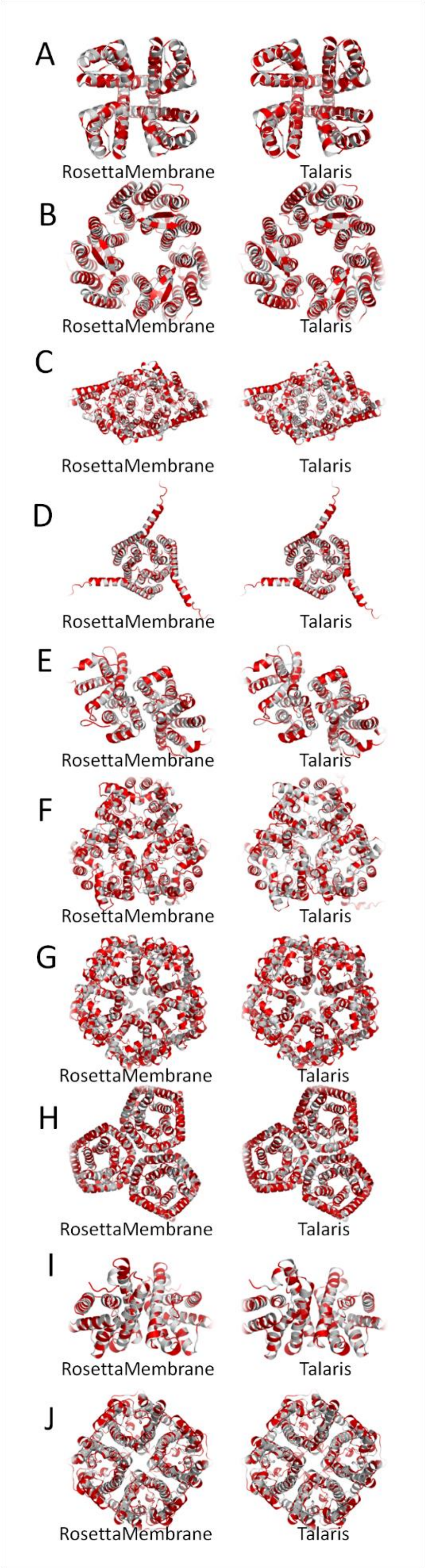
A
RosettaMembrane
Talaris
B
RosettaMembrane
Talaris
C
RosettaMembrane
Talaris
D
RosettaMembrane
Talaris
E
RosettaMembrane
Talaris
F
RosettaMembrane
Talaris
G
RosettaMembrane
Talaris
H
RosettaMembrane
Talaris
I
RosettaMembrane
Talaris
J
RosettaMembrane
Talaris

Figure 2.7. Designed residues mapped on models of oligomers. Top models were selected as representative models of assembled oligomers visualizing the designed sites of 1k4c(A), 1m0l(B), 1ots(C), 2uui(D), 2vpzI, 3k3f(F), 3kly(G), 3m71(H), 3rlb(I), 3zoj(J). All images are top down. Red represents sites that have been designed while gray represents sites that have retained the native amino acid. Designed residues tend to be surface residues regardless of scoring function. In some cases such as C and F, core residues and residues at the interface of subunits are distinctly native.

Amino acid properties are most native-like in proteins design using RosettaMembrane

Sequence recovery is a limited metric for design in that it only reports how much of the sequence changes from the native sequence. A more pronounced improvement is observed when comparing amino acid property composition between RosettaMembrane and Talaris (Fig. 2.8). The percent difference in sequence composition (design percent composition – native percent composition) was calculated to further detail how design sequences differed from native (Fig. 2.8). A negative percent difference (red) indicates that Rosetta introduces that particular amino acid less frequently than is observed in the native proteins in our dataset, while a positive percent difference (blue) indicates Rosetta introduces it more frequently. The average absolute deviation from native sequence composition for monomers was ±3.4% for RosettaMembrane, and ±2.8% for Talaris. For homo-oligomers, a similar trend was seen with ±2.5% for RosettaMembrane, ±1.6% for and Talaris.

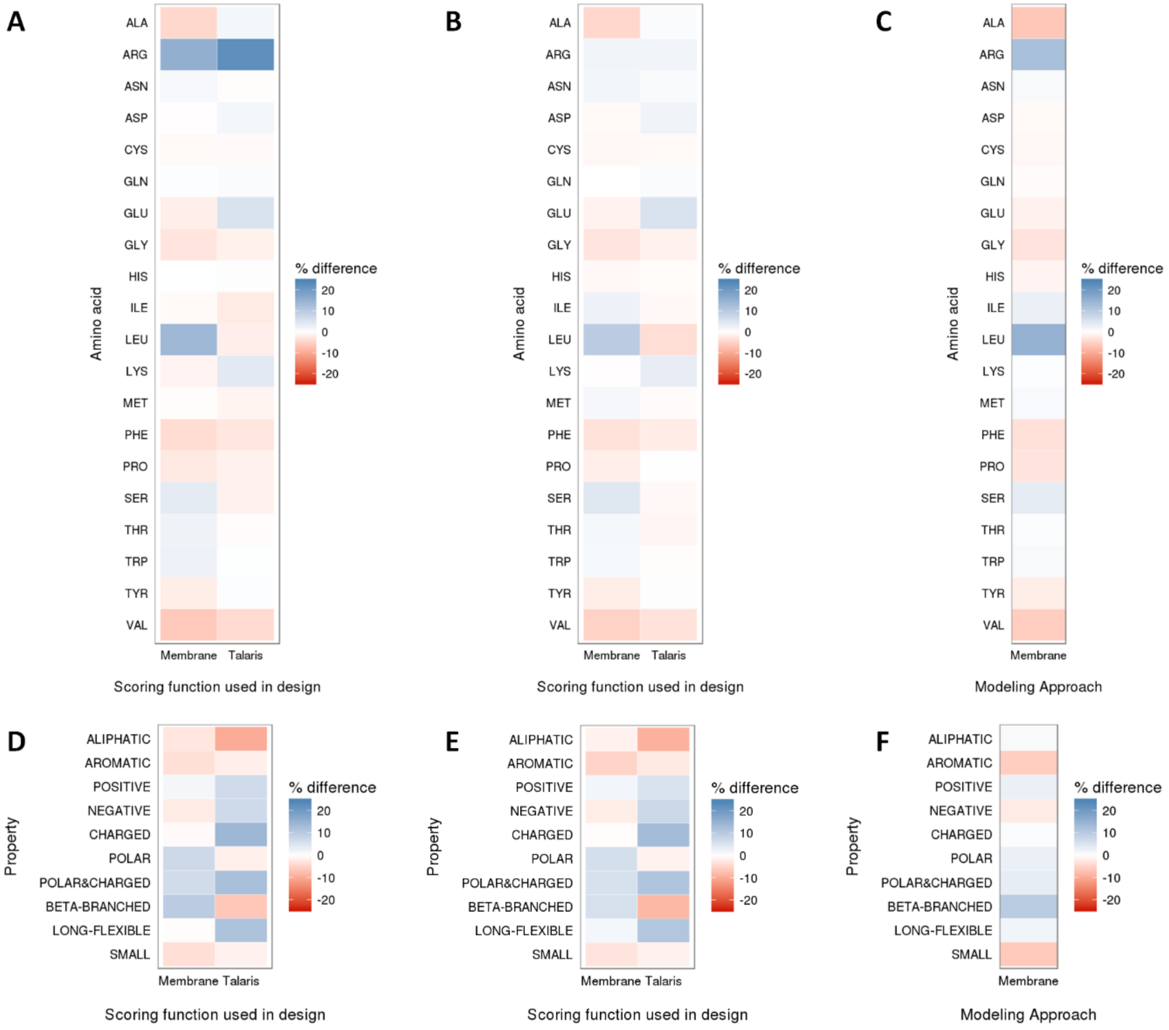


Figure 2.8. Heatmaps for composition of sequence (4A-C) and amino acid properties (4D-F) by percent difference of wild-type from design. Datasets evaluated were monomers (4A,D), homo-oligomers (4B,E), and homo-oligomers as monomers (4C,F). Both RosettaMembrane (Membrane) and Talaris scoring functions have strong and weak amino acid recovery for different amino acids in the monomeric set (4A,D). The homo-oligomeric set (4B,E) performs similarly to the monomeric set for each respective scoring function. Finally, when homo-oligomers are designed as monomers using RosettaMembrane (4C,F), the design is less native-like, but has a similar sequence composition as the homo-oligomeric design.

For each amino acid, I have plotted the fraction recovered for the whole (Fig. 2.9), monomeric (Fig. 2.10), and homo-oligomeric (Fig. 2.11) datasets. Also plotted for each amino acid is the number of occurrences in all native, best-scoring RosettaMembrane designs, and best-scoring Talaris designs with respect to their position in the membrane layer in the whole (Fig.

2.12), monomeric (Fig. 2.13), and homo-oligomeric (Fig. 2.14). Arginine was found more frequently in designs than in native membrane proteins. In figure 2.9, the fraction recovered drops in the inner hydrophobic layer for RosettaMembrane designs. In figure 2.12, it is clear that Talaris is solubilizing the designs as an increase in occurrence of Arginine is seen in the inner and outer hydrophobic regions.

Table 2.1 Layers of the membrane represented by bins. Calculated distances from the membrane center have been binned to aid in visualization of data. Bins have been defined by the layers described by Yarov Yarovoy and co-workers(Yarov-Yarovoy, Schonbrun, and Baker 2006). A negative distance indicates it is on the intracellular side of the membrane whereas a positive distance indicates it is on the extracellular side.

Bin	Distance from the membrane center (Å)	Hydrophobic layer
1	-40 to -30	Water
2	-30 to -24	Polar
3	-24 to -18	Interface
4	-18 to -12	Outer Hydrophobic
5	-12 to 0	Inner Hydrophobic
6	0 to 12	Inner Hydrophobic
7	12 to 18	Outer Hydrophobic
8	18 to 24	Interface
9	24 to 30	Polar
10	30 to 40	Water

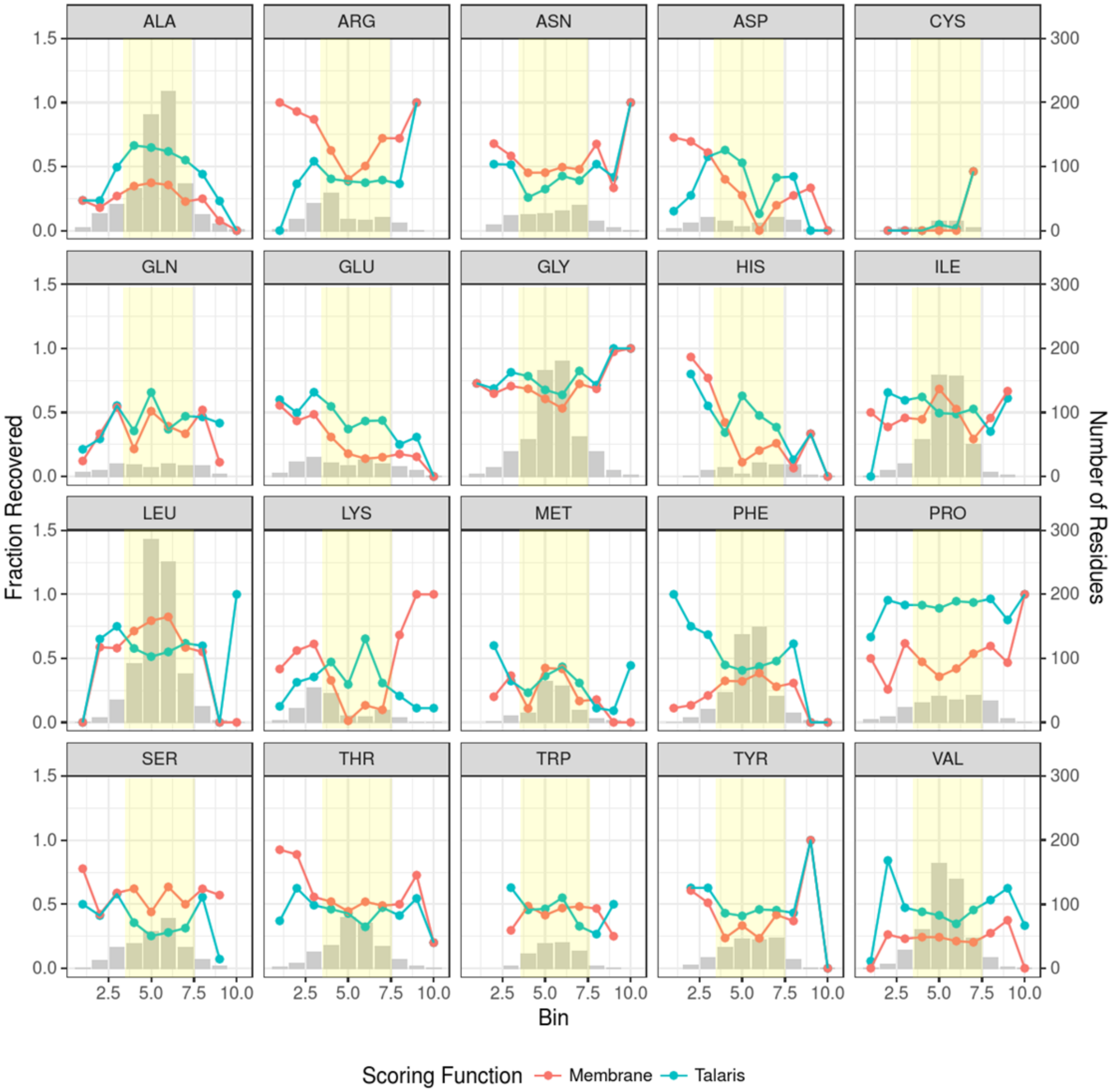


Figure 2.9. Fraction of sequence recovery for each amino acid with respect to distance from the membrane center. Bars indicate the raw number of residues observed at a particular distance bin (see Table 2.1) from the membrane center. Dots indicate the fraction recovered at that particular distance bin and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein.

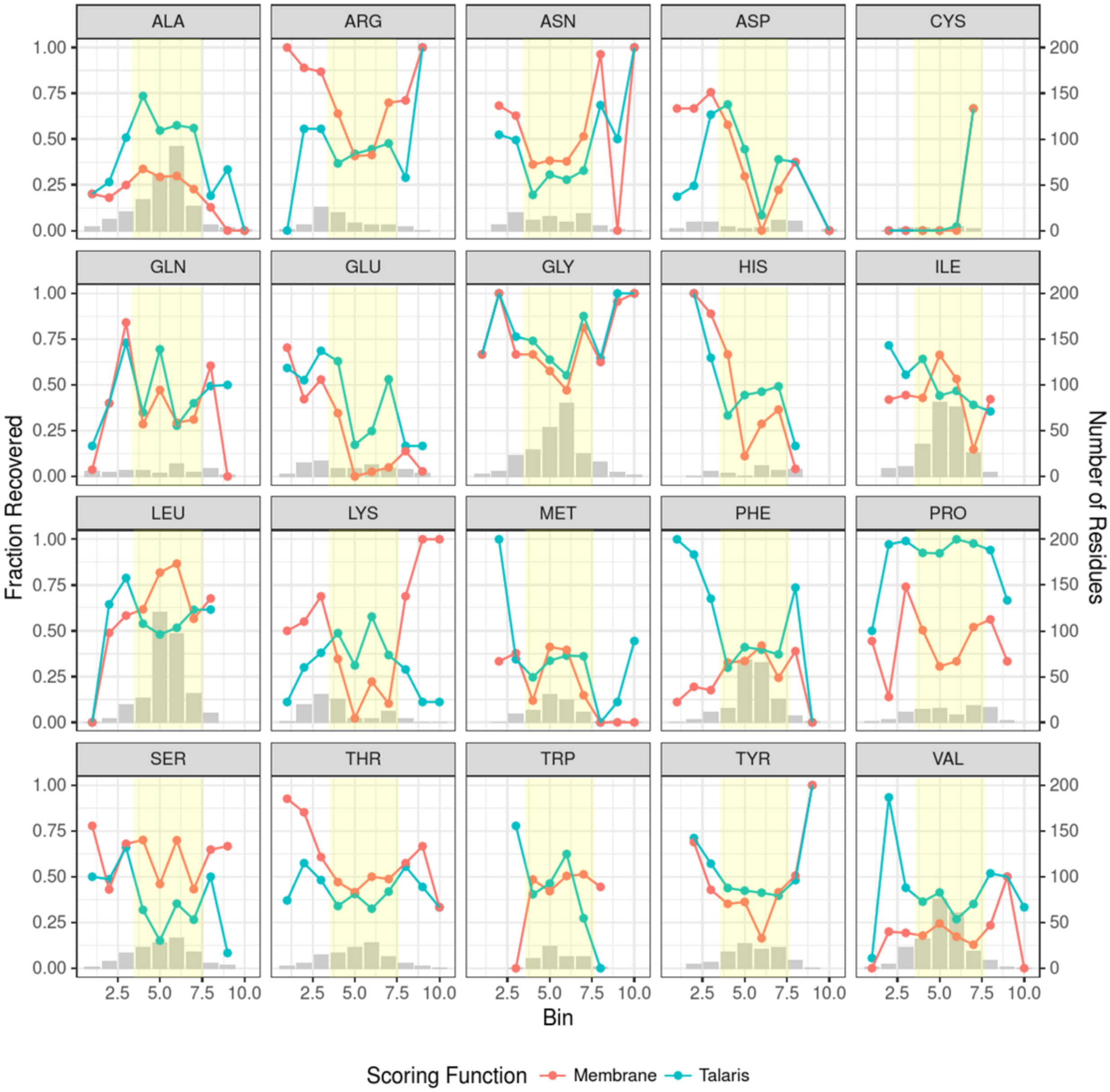


Figure 2.10. Fraction of sequence recovery for each amino acid with respect to distance from the membrane center for the monomeric dataset. Bars indicate the raw number of residues observed at a particular distance bin (see Table 2.1) from the membrane center. Dots indicate the fraction recovered at that particular distance bin and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein.

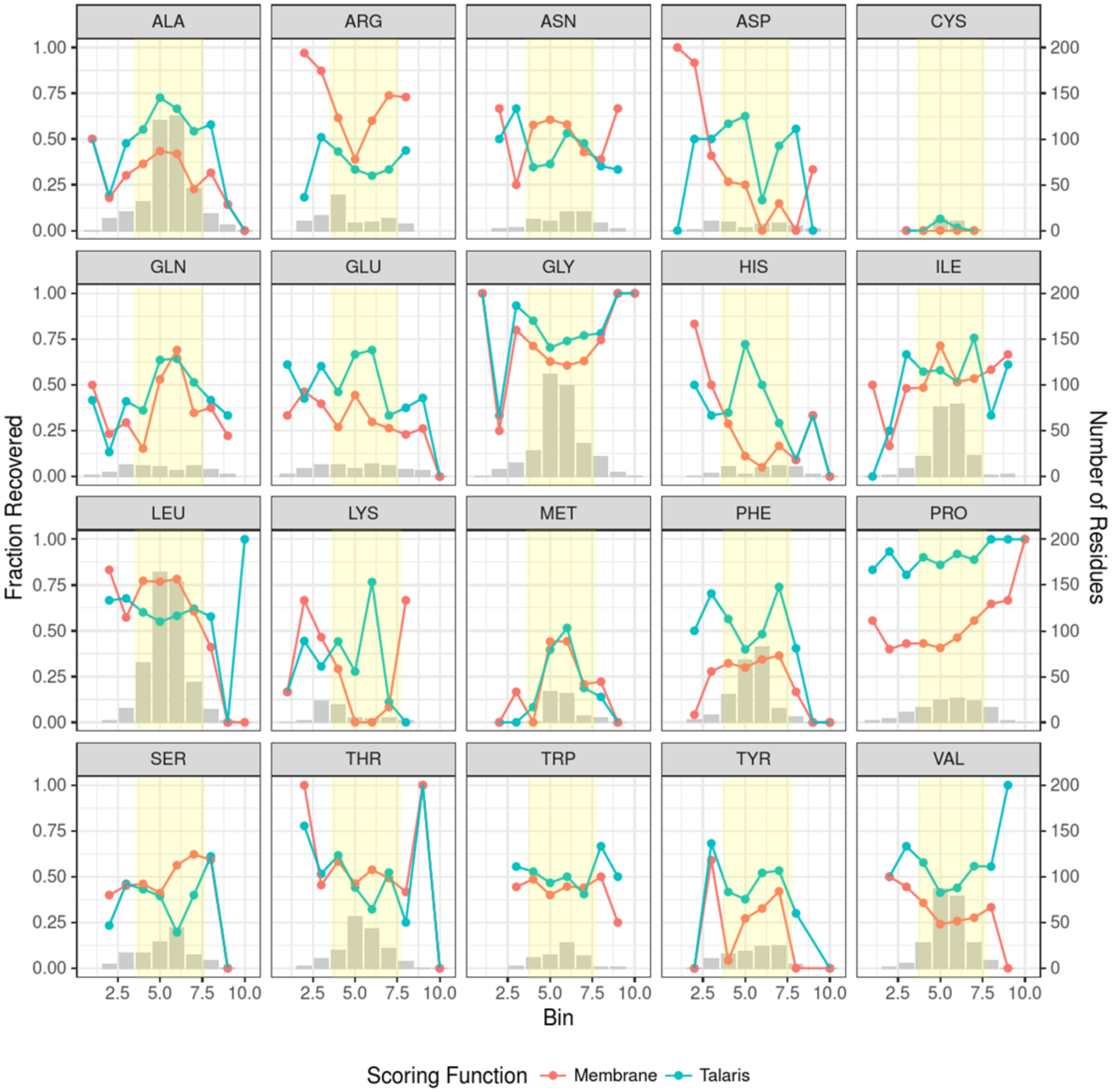


Figure 2.11. Fraction of sequence recovery for each amino acid with respect to distance from the membrane center for the homo-oligomeric dataset. Bars indicate the raw number of residues observed at a particular distance bin (see Table 2.1) from the membrane center. Dots indicate the fraction recovered at that particular distance bin and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein.

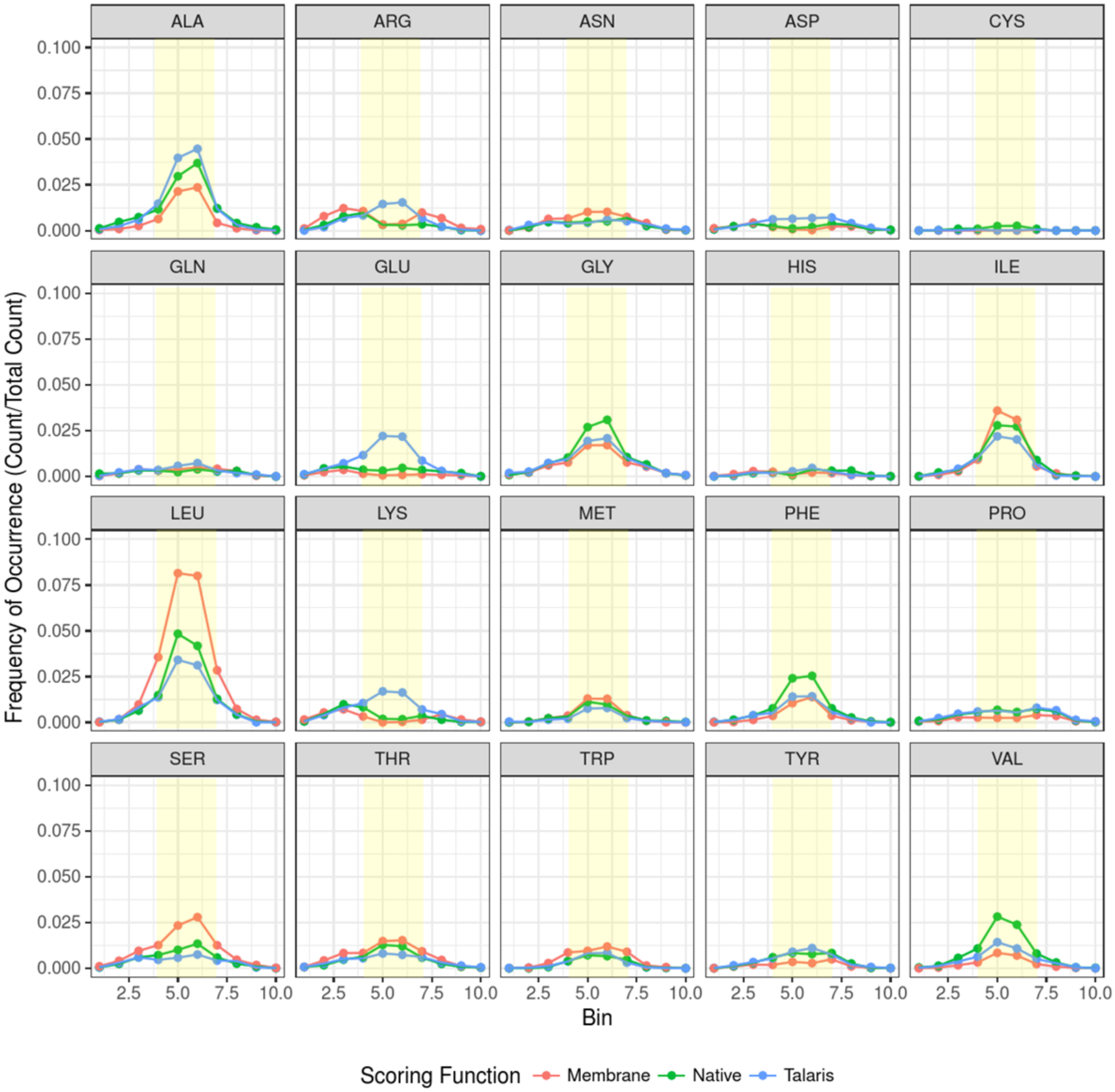


Figure 2.12. Frequency of occurrence for each amino acid by membrane layer. Bins are a range of distances from the membrane center (see Table 2.1). Dots indicate the frequency of occurrence of an amino acid seen at a particular distance from the membrane center, and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein.

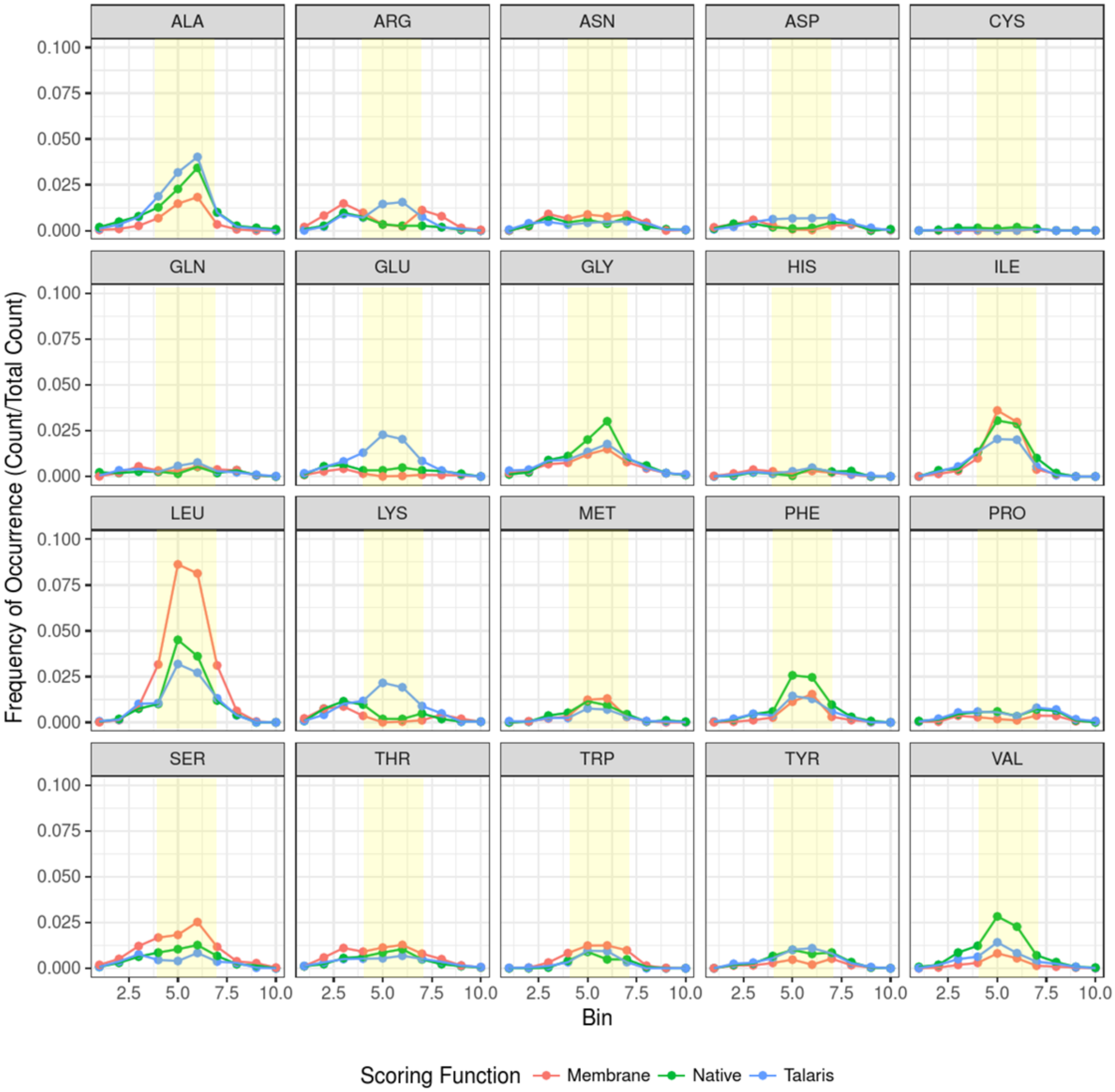


Figure 2.13. Frequency of occurrence for each amino acid by membrane layer in the monomeric set. Bins are a range of distances from the membrane center (see Table 2.1). Dots indicate the frequency of occurrence of an amino acid seen at a particular distance from the membrane center, and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein.

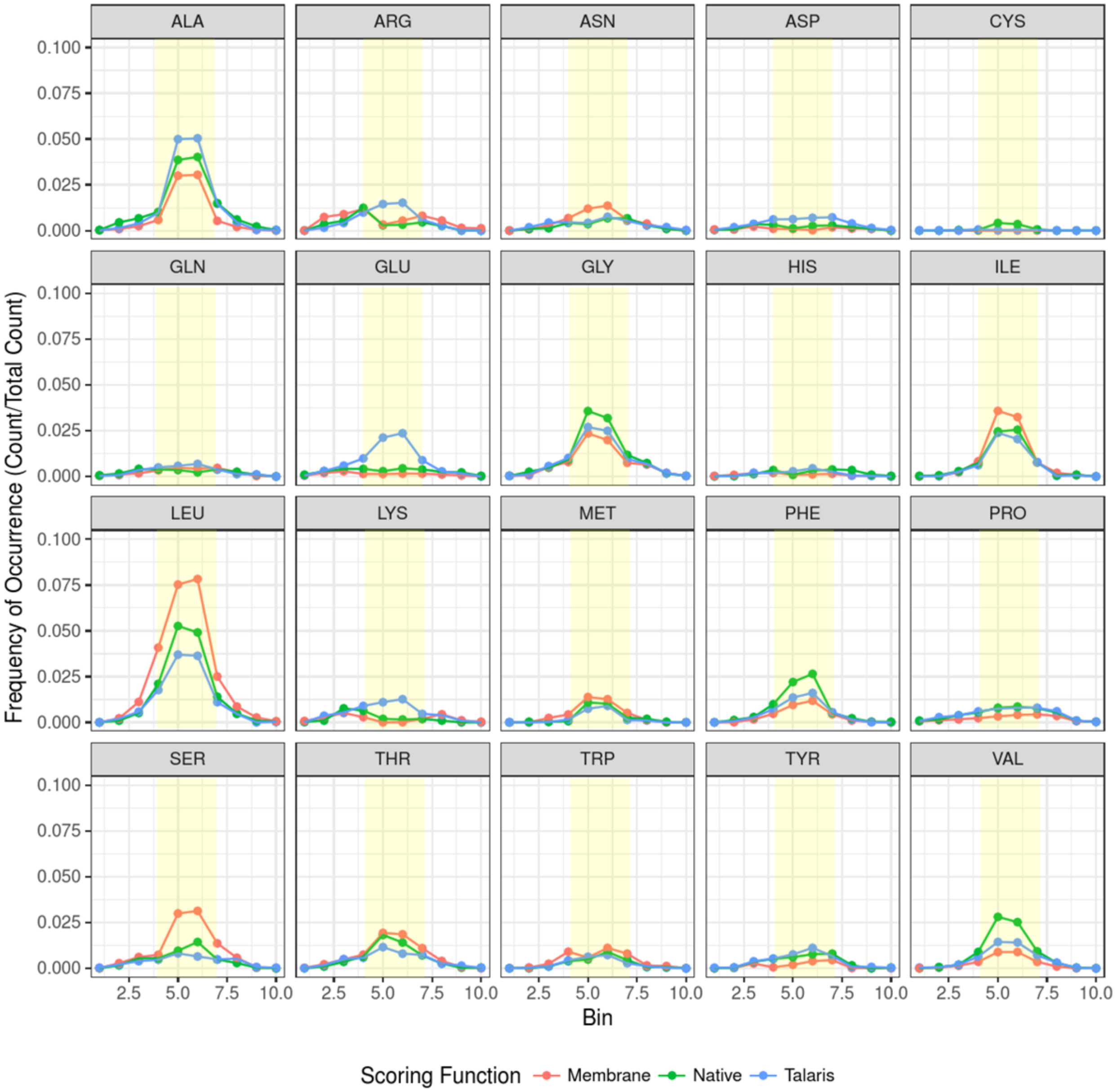


Figure 2.14. Frequency of occurrence for each amino acid by membrane layer in the homo-oligomeric set. Bins are a range of distances from the membrane center (see Table 2.1). Dots indicate the frequency of occurrence of residues of an amino acid seen at a particular distance from the membrane center, and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein.

However, for RosettaMembrane, only the outer hydrophobic and interface regions have an increase of occurrence. Additionally, this is more pronounced in the monomeric dataset (Fig.

2.13), perhaps indicating that there is an additional cost of designing in a bulky residue at a protein-protein interface region (Fig. 2.14). Talaris adds charged residues such as Arginine, Aspartate, Glutamate, and Lysine on the surface and in the inner and outer hydrophobic regions, as expected, to solubilize the protein.

The most striking difference for RosettaMembrane designs when compared with native membrane protein sequences was that the amino acid composition is shifted towards Leucine residues (Fig. 2.9) while other hydrophobic amino acids such as Phenylalanine, Valine and Alanine, have a lower than native probability. This indicates that RosettaMembrane has a bias towards Leucine at the cost of other hydrophobic amino acids. The fraction recovered for Leucine in the inner and outer hydrophobic regions ranged from 58% to 82% while Valine and Alanine had recoveries in the ranges of 20-24% and 23-37%, respectively (Fig. 2.9). When the number of occurrences of Leucine in native proteins and designed proteins was plotted with respect to their position in the membrane layer, Leucine was found to be overrepresented by 1.9 fold in the inner and outer hydrophobic regions for RosettaMembrane designs (Fig. 2.12). An increase is also seen in both datasets with a 2.2 fold increase for monomers (Fig. 2.13), and a 1.6 fold increase for homo-oligomers (Fig. 2.14). Additionally, RosettaMembrane designs Valine and Alanine less frequently than what is seen in native proteins in the inner and outer hydrophobic regions by 3.4 fold and 1.6 fold, respectively. This further supports that in the hydrophobic regions, Valine and Alanine are replaced by Leucine in RosettaMembrane designs. Sequence recovery may be too crude of an analysis to determine the extent of which designed proteins have changed. In addition to calculating recovery of native amino acid identities, we calculated the percent difference in the composition of amino acids grouped by properties such as polarity, charge, etc. (design percent composition – native percent composition). Here, the average absolute deviation from native amino acid property composition in monomers was 3.9% for RosettaMembrane, and 7.4% for Talaris, while in homo-oligomers, it was 3.4% for RosettaMembrane, and 7.3% for Talaris. When considering the composition of all amino acid properties, RosettaMembrane resulted in proteins with more native-like properties in both monomeric and homo-oligomeric sets (Fig. 2.8 D, E). The differences in sequence composition between native and designed proteins is primarily caused by mutations on the protein surface as core sequence recovery is high for both, Talaris and RosettaMembrane. Recall that surface sequence recovery rates of monomers averaged at 40% for RosettaMembrane designs, whereas Talaris had lower averages of 34% and 38%, respectively (Fig. 2.3 A). However, when comparing the difference in amino acids that are aliphatic (Fig. 2.8 D,E), RosettaMembrane is near native with a percent difference of nearly -3% in monomers and -1% in homo-oligomers whereas Talaris had a percent difference near -10% for both monomers and homo-oligomers.

Additionally, I looked at the frequency of occurrence of each amino acid identity with respect to the distance from the central axis of the protein for the full (Fig. 2.15), monomeric (Fig. 2.16), and homo-oligomeric (Fig. 2.17) datasets. The distances are defined by Table 2.2. RosettaMembrane designed over 1.7 fold as many Leucines near the core than what is seen in native membrane proteins. Near the core of the protein, RosettaMembrane designs fewer Alanine and Valine than what is seen in native membrane proteins. This may be due to the small size of Alanine and Valine side-chains because the Rosetta energy function is driven by packing interactions (B. Kuhlman & Baker, 2000).

Table 2.2. Bins for distance from central axis of the protein. These bins are used in visualization of sequence recovery by amino acid identity in Figures 2.15, 2.16, and 2.17.

Bin	Distance from central axis of protein (Å)
1	0 < 5
2	5 < 10
3	10 < 15
4	15 < 20
5	20 < 30
6	30 < 45

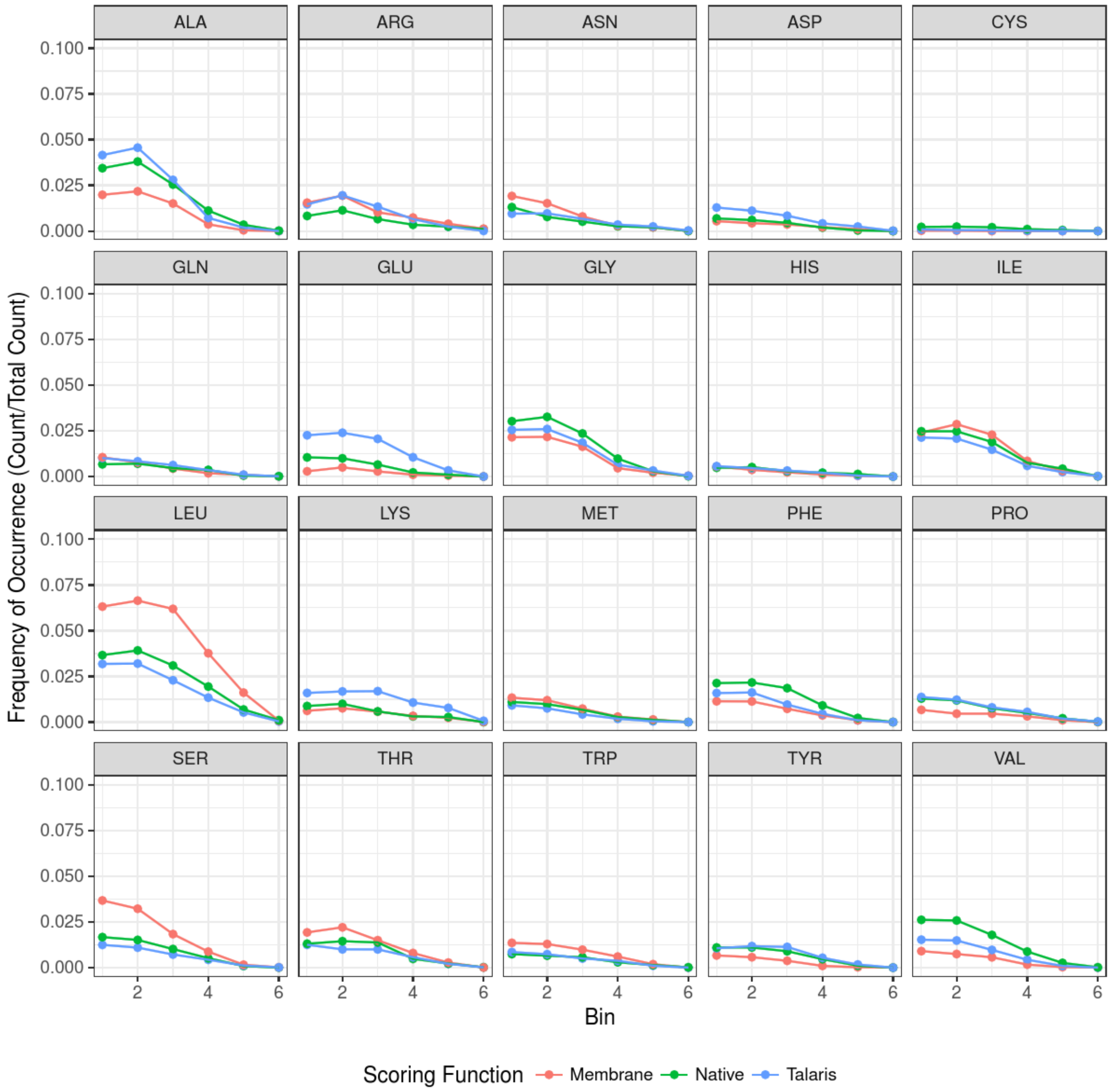


Figure 2.15. Frequency of occurrence of residues with respect to distance from the central axis of the protein for the entire dataset. Distances are binned (see Table 2.2) and non-directional. In RosettaMembrane designs, leucine and serine are designed more frequently than native near the central axis whereas alanine, phenylalanine and valine are designed less frequently near the central axis.

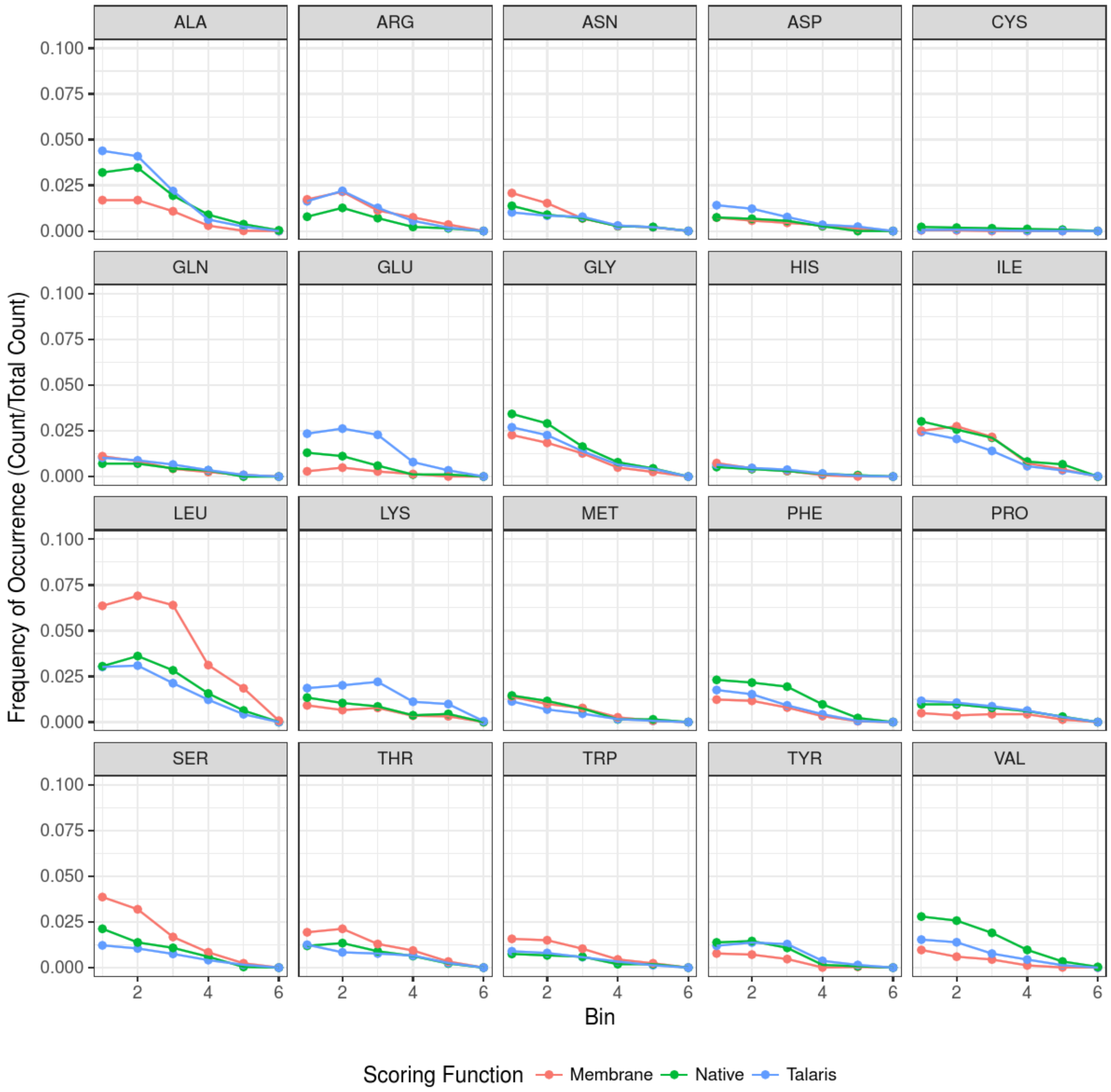


Figure 2.16. Frequency of occurrence of residues with respect to distance from the central axis of the protein for the monomeric dataset. Distances are binned (see Table 2.2) and non-directional. In RosettaMembrane designs, leucine and serine are designed more frequently than native near the central axis whereas alanine, phenylalanine and valine are designed less frequently near the central axis.

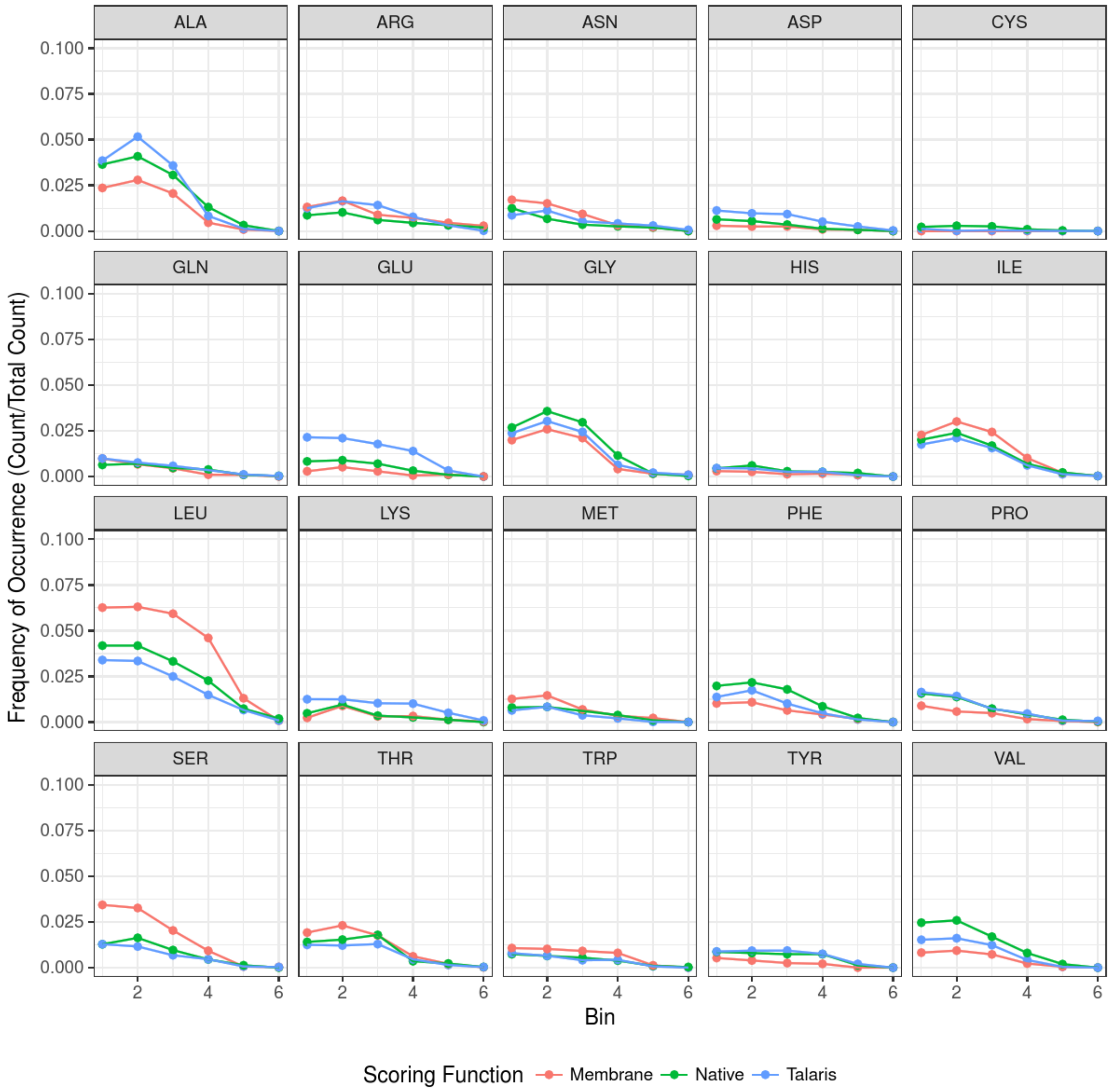


Figure 2.17. Frequency of occurrence of residues with respect to distance from the central axis of the protein for the homo-oligomeric dataset. Distances are binned (see Table 2.2) and non-directional. In RosettaMembrane designs, leucine and serine are designed more frequently than native near the central axis whereas alanine, phenylalanine and valine are designed less frequently near the central axis.

To further investigate which amino acid mutations would be tolerated by evolution, Position Specific Scoring Matrix (PSSM) Recovery (Deluca, Dorr, & Meiler, 2011) was calculated using the uniref50membrane database. Because PSSM recovery is considering all tolerated amino acids that have been seen in known sequences, PSSM recovery will be higher than sequence recovery alone (Allison et al., 2014). In monomers, RosettaMembrane had an average PSSM recovery of 73% while Talaris had a recovery of 72% (Fig. 2.18 A). In homo-oligomers, RosettaMembrane had an average PSSM recovery of 69% while Talaris was at 70% (Fig. 2.18 B). Despite using a membrane specific database, the PSSM recovery did not favor RosettaMembrane designs.

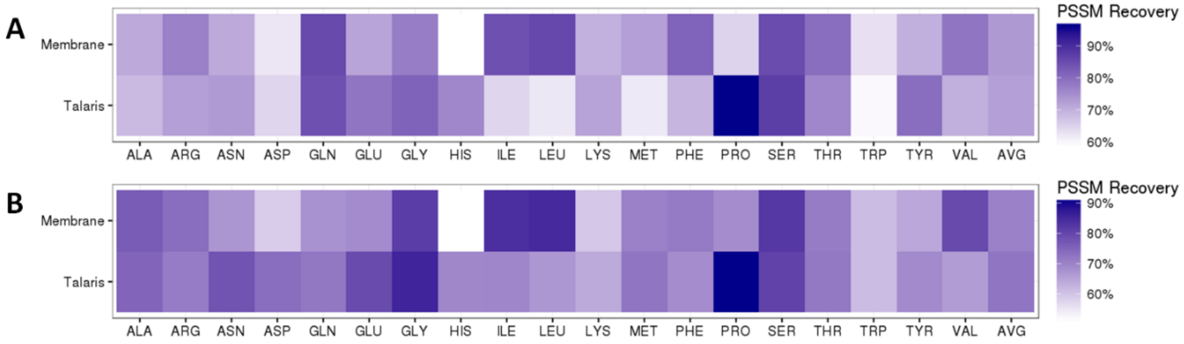


Figure 2.18. Heatmaps for position specific scoring matrix (PSSM) recovery for the monomeric set (A) and homo-oligomeric set (B). The PSSM recovery for each scoring function is similar when comparing the monomeric set to the homo-oligomeric set. RosettaMembrane (Membrane) has limitations for recovering Histidine and Proline, but shows improved recovery for Isoleucine, Leucine, Valine, and Phenylalanine.

RosettaMembrane designs a native-like hydrophobicity gradient and predicted ΔG$_{transfer}$

The HotPatch server (Pettit, Bare, Tsai, & Bowie, 2007) was used to visualize the relative hydrophobicity on the surface of proteins (Fig. 2.18). For Talaris, despite having a similar sequence composition as native structures (Fig. 2.8 A,B), the resulting designs had a noticeably different surface composition. This is supported by the sequence recovery analysis where core sequence recovery is typically much higher than the surface sequence recovery (Fig. 2.3 A,B). Representative design models selected for monomers show that both scoring functions resulted in a large amount of surface residues being redesigned (Fig. 2.5 A,B). Design models of assembled homo-oligomers highlight a similar feature; however, design at the interface of subunits is typically more restricted and thus more core-like (Fig. 2.5 C-F). For Talaris, the surfaces of the majority of the protein designs were covered in hydrophilic residues (Fig. 2.19) as the scoring function attempted to solubilize the surface of the protein. However, RosettaMembrane resulted in a designed protein with a native-like hydrophobicity gradient on the surface. These models had more strongly hydrophobic and hydrophilic areas whereas native surfaces had moderate hydrophobic and hydrophilic regions.

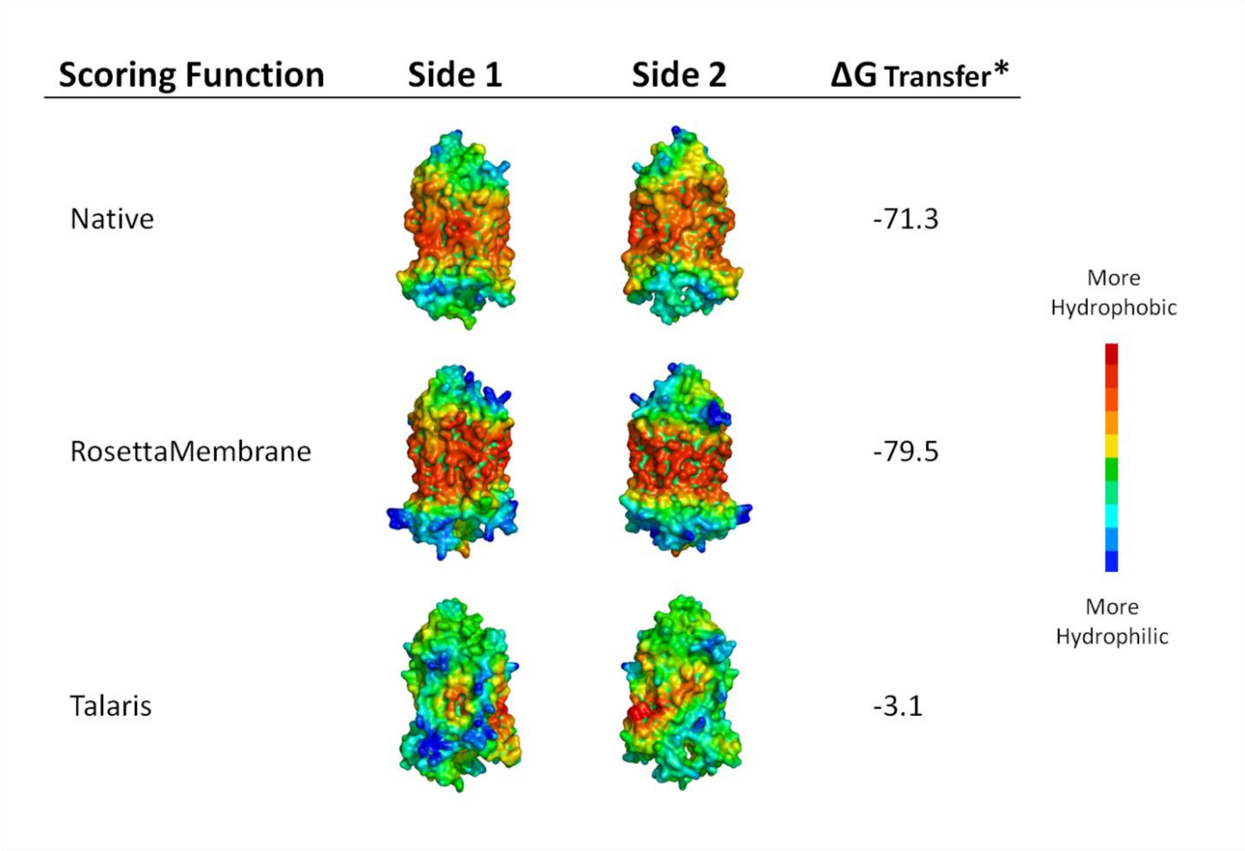


Figure 2.19. Surface hydrophobicity of proteins designed by various scoring functions in Rosetta. The native protein (1u19) has a clear hydrophobic region where the membrane is present. Overall, the surface has a hydrophobic gradient so that it is more hydrophobic in the middle and extends to be polar on the edges. When designed, the RosettaMembrane optimized the sequence so that the surface closely resembles, and even idealizes the hydrophobic gradient. The predicted $\Delta G_{transfer}$ is close to that of the native protein. However, when designed with the soluble scoring function, Talaris, the surface is mostly covered in polar, hydrophilic residues which gives a $\Delta G_{transfer}$ that has decreased significantly.

The OPM server (A. L. Lomize, Pogozheva, Lomize, & Mosberg, 2006; M. a. Lomize, Pogozheva, Joo, Mosberg, & Lomize, 2012) was used to predict the ΔG_{transfer} for both monomeric and homo-oligomeric sets (Fig. 2.20). The server tends to predict that integral membrane proteins and peptides have a $\Delta G_{transfer}$ between -400 and -10 kcal/mol (M. a. Lomize et al., 2012). For our datasets, the native proteins were in the range of -44 to -164. Designs by the RosettaMembrane scoring function were near and above native in a range of -71 to -275 whereas designs by Talaris were near zero indicating that the designed protein would not be membrane soluble.

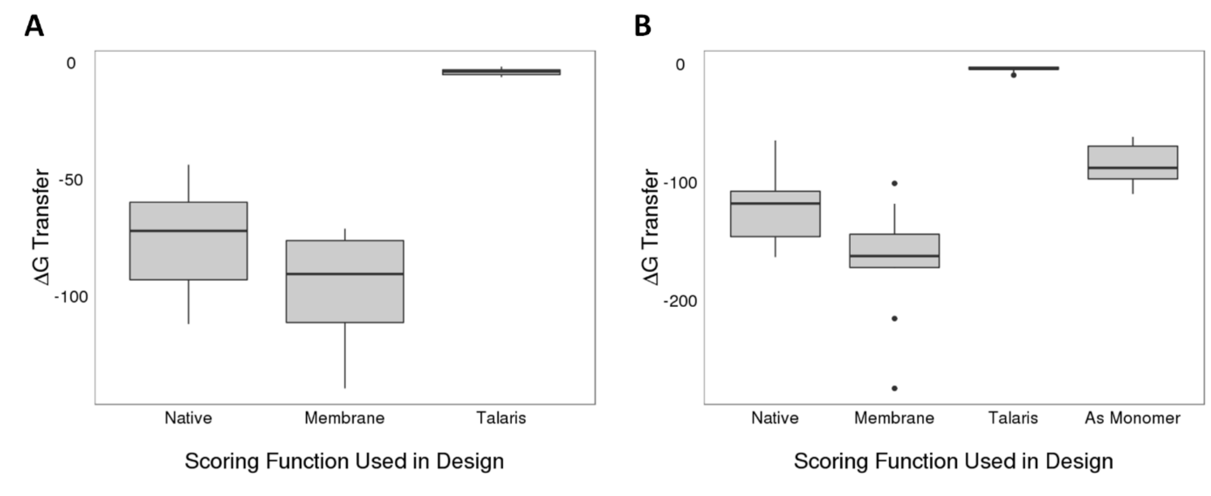


Figure 2.20. Predicted $\Delta G_{transfer}$ for designs from membrane and soluble scoring functions. For both monomeric (A) and homo-oligomeric (B) sets, the membrane scoring function resulted in more native-like $\Delta G_{transfer}$ values in comparison to the soluble scoring functions. For the soluble scoring function, the value was nearly zero indicating it would likely not partition into the membrane. Finally, the homo-oligomeric design took into account surfaces when assembled as an homo-oligomer, resulting in more native-like values.

RosettaMembrane replaces other hydrophobics with Leucine

RosettaMembrane chooses Leucine over other hydrophobic amino acids. Although Leucine may be ideal for the particular membrane environment modeled in Rosetta, this may not be ideal biologically as it does not account for asymmetry and heterogeneity of the membrane. A previous study showed Leucine to be the most frequent amino acid in the inner hydrophobic and outer hydrophobic layers of the membrane (Yarov-Yarovoy et al., 2006). Because Leucine has such a high frequency compared to other amino acids, it scores quite favorably in RosettaMembrane and is overrepresented in designs often replacing native, hydrophobic amino acids (Fig. 2.8 A, Fig. 2.12).

To further investigate how Leucine might replace hydrophobic amino acids like Alanine, Valine and Phenylalanine, we mapped their occurrences onto the structures to understand where each scoring function would typically place them compared to where they are found on the native membrane protein. For both monomers and homo-oligomers, native membrane proteins have Alanine in the core as well as on the surface (Fig. 2.21). Both scoring functions typically placed Alanine in the core of the protein and RosettaMembrane had a lower Alanine sequence composition than native membrane proteins. In homo-oligomers, very few Alanine occur on the surface of the protein that would be lipid-exposed, and very few are seen in the interface between subunits, likely due to Alanine's small size.

Designs from both scoring functions resulted in fewer Valine and Phenylalanine. Both residues are hydrophobic and, in the case of RosettaMembrane, were likely replaced by Leucine. Valine was typically designed in the core of the protein regardless of scoring function; however, in homo-oligomers, Talaris does place Valine in the core-like interface between sub-units more frequently than RosettaMembrane (Fig. 2.22). Despite Phenylalanine typically occurring in the interface and inner and outer hydrophobic layers, fewer Phenylalanines are seen on the surface of designs from both scoring functions (Fig. 2.23). This suggests that Leucine's abundance in these layers overshadows the presence of Phenylalanine in native membrane proteins. As a comparison, Arginine, was also highlighted onto structures (Fig. 2.24). Although the percent difference in composition was like that of Leucine, the number of occurrences (Fig. 2.9) was much lower, so the effect was pronounced.

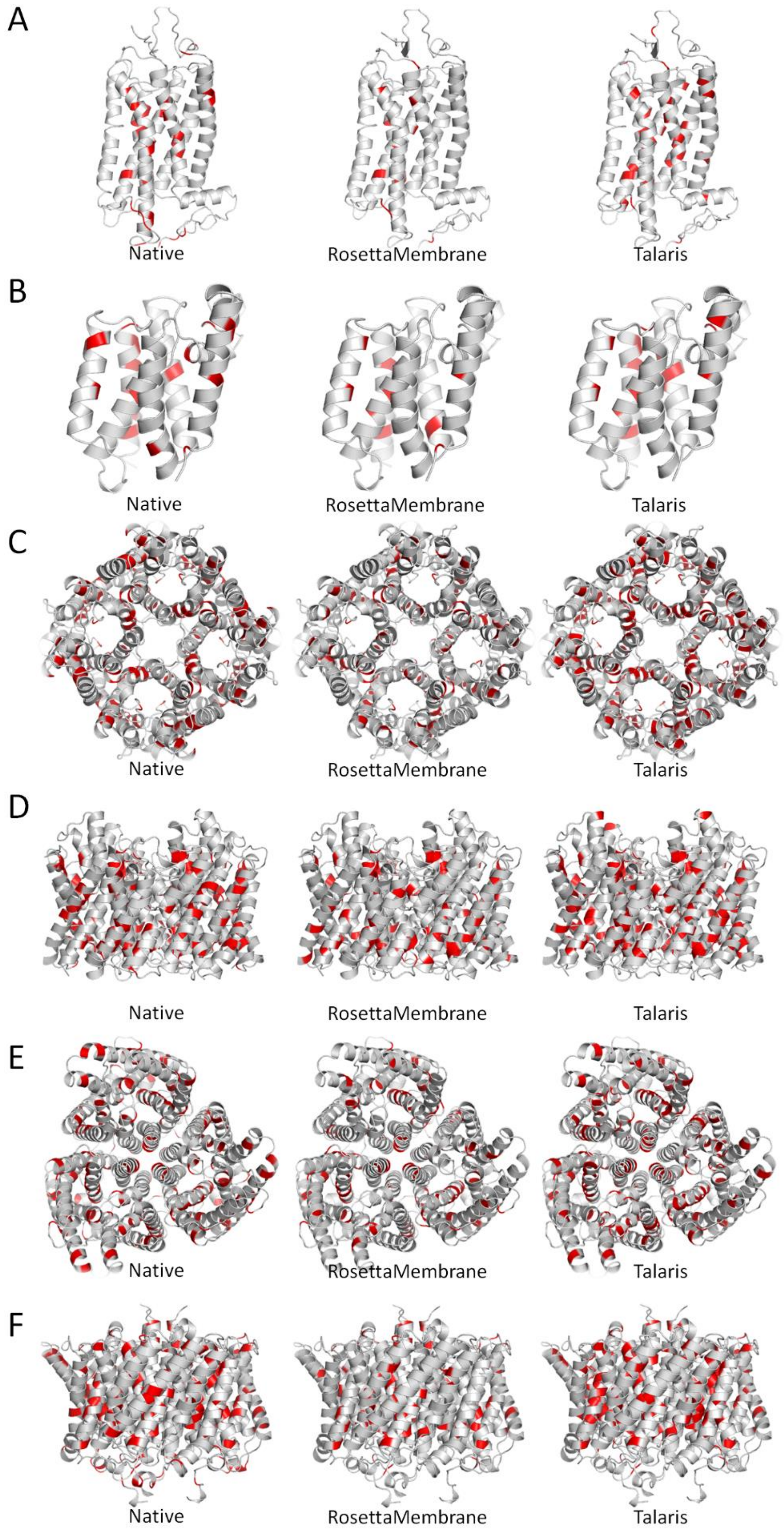

A
Native
RosettaMembrane
Talaris
B
Native
RosettaMembrane
Talaris
C
Native
RosettaMembrane
Talaris
D
Native
RosettaMembrane
Talaris
E
Native
RosettaMembrane
Talaris
F
Native
RosettaMembrane
Talaris

Figure 2.21. Visualization of Alanine on models. Top models were selected to visualize where Alanines occur in monomers 1u19 (A), 2xov(B) and in oligomers 1fx8 (C-top down,D) and 3b9w (E-top down,F). Native structures (left) were compared to representative models of proteins designed using RosettaMembrane and Talaris. RosettaMembrane designs in fewer Alanines as compared to native membrane proteins. Both scoring functions tend to place Alanines in the core of the protein. Oligomers show very few Alanines on the lipid-facing, surface of the protein as well as few in the interface between subunits likely due to Alanine being small.

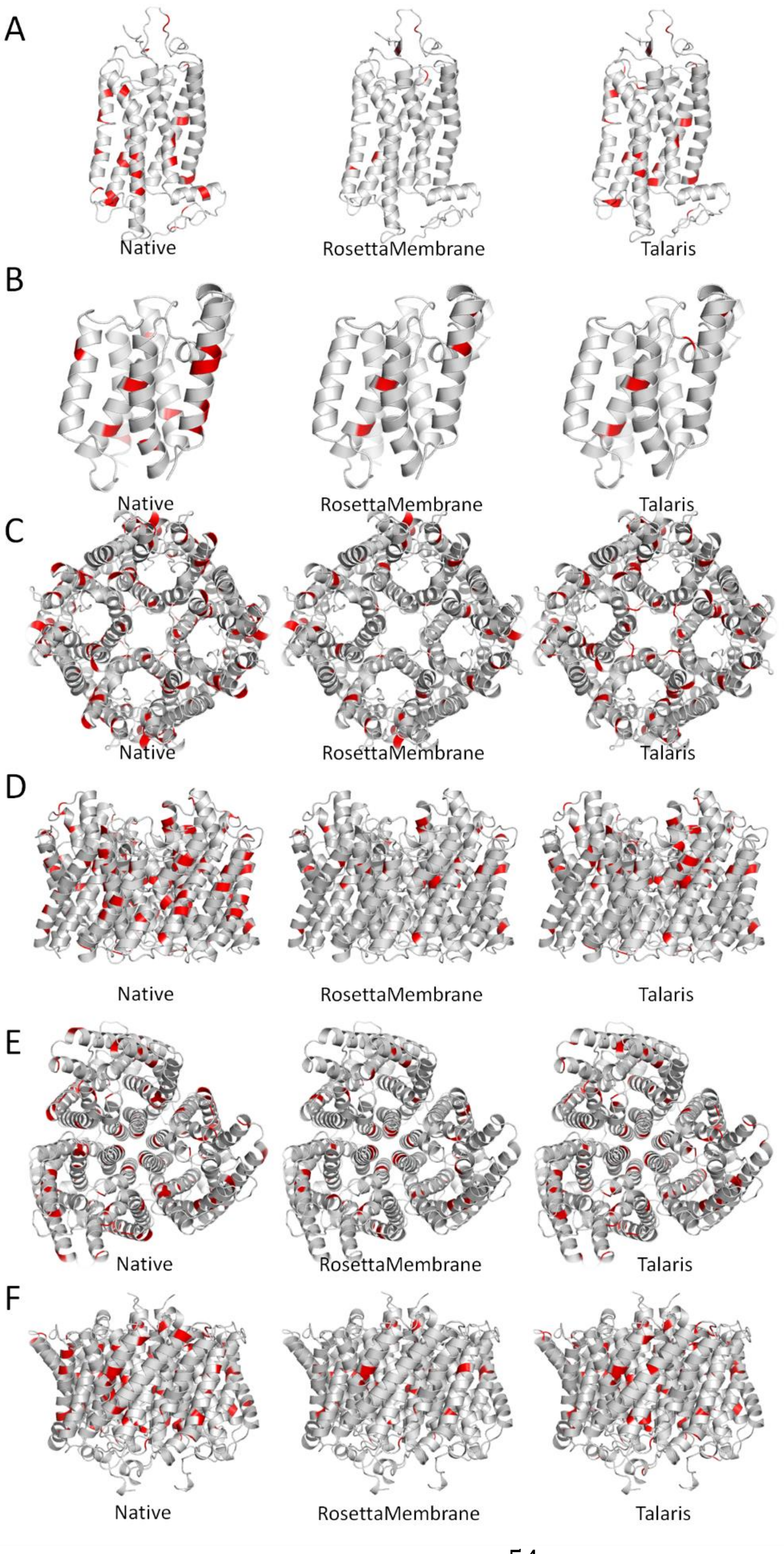

A
Native
RosettaMembrane
Talaris
B
Native
RosettaMembrane
Talaris
C
Native
RosettaMembrane
Talaris
D
Native
RosettaMembrane
Talaris
E
Native
RosettaMembrane
Talaris
F
Native
RosettaMembrane
Talaris

Figure 2.22. Visualization of Valine on models. Top models were selected to visualize where Valines occur in monomers 1u19 (A), 2xov (B) and in oligomers 1fx8 (C-top down,D) and 3b9w (E-top down,F). Native structures (left) were compared to representative models of proteins designed using RosettaMembrane and Talaris. Both scoring functions, most noticeably RosettaMembrane, design in fewer Valines as compared to native membrane proteins. Both scoring functions tend to place Valine in the core, and designed oligomers show very few Valines on the lipid-facing, surface of the protein. However, Talaris place Valines in the interface between subunits more frequently than RosettaMembrane.

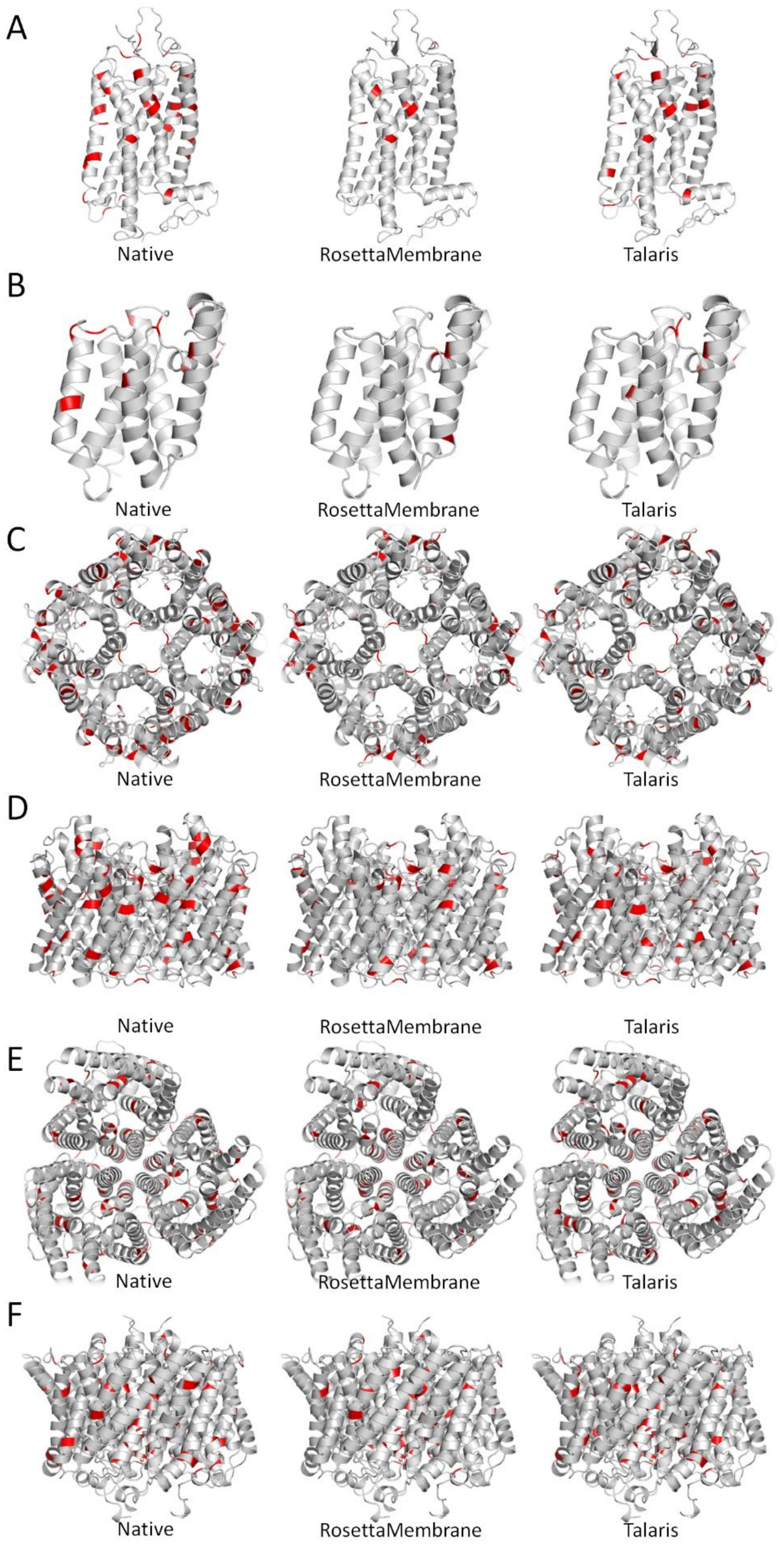

A
Native
RosettaMembrane
Talaris
B
Native
RosettaMembrane
Talaris
C
Native
RosettaMembrane
Talaris
D
Native
RosettaMembrane
Talaris
E
Native
RosettaMembrane
Talaris
F
Native
RosettaMembrane
Talaris

Figure 2.23. Visualization of Phenylalanine on models. Top models were selected to visualize where Phenylalanines occur in monomers 1u19 (A), 2xov (B) and in oligomers 1fx8 (C-top down,D) and 3b9w (E-top down,F). Native structures (left) were compared to representative models of proteins designed using RosettaMembrane and Talaris. Phenylalanine can be seen in native proteins in the inner hydrophobic, outer hydrophobic, and interface regions. Both scoring functions result in designs with fewer Pheylalanines than native, especially on the surface.

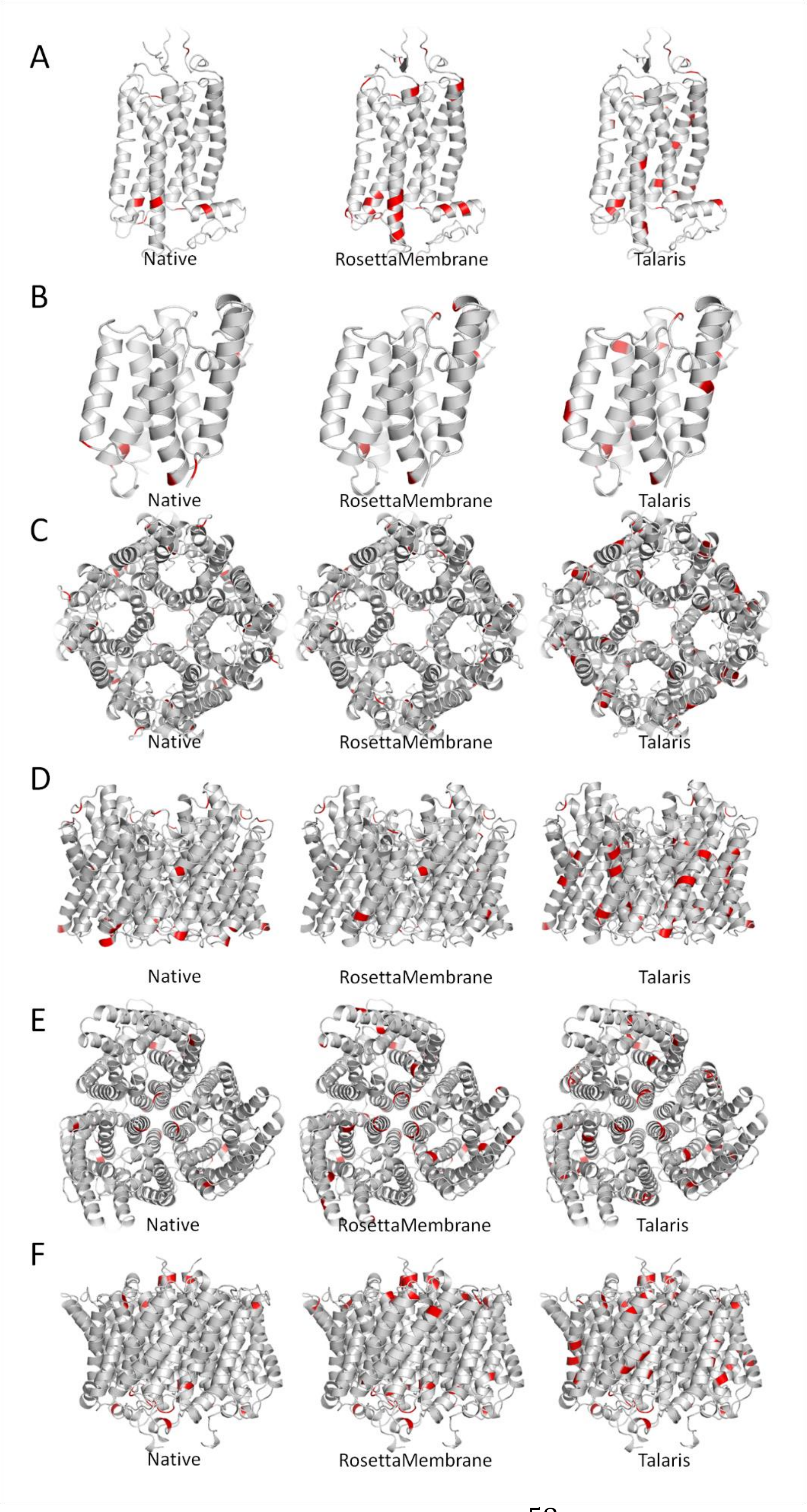

A
Native
RosettaMembrane
Talaris
B
Native
RosettaMembrane
Talaris
C
Native
RosettaMembrane
Talaris
D
Native
RosettaMembrane
Talaris
E
Native
RosettaMembrane
Talaris
F
Native
RosettaMembrane
Talaris

Figure 2.24. Visualization of Arginine on models. Top models were selected to visualize where Arginines occur in monomers 1u19 (A), 2xov (B) and in oligomers 1fx8 (C-top down,D) and 3b9w (E-top down,F). Native structures (left) were compared to representative models of proteins designed using RosettaMembrane and Talaris. For native membrane proteins, Arginine can usually be found outside of the hydrophobic layers near the interface, polar, and solvent exposed environments. RosettaMembrane tends to design Arginines in these layers. Talaris places Arginines on the surface along helices as an attempt to help solubilize the protein.

To visualize which native identities were designed to Leucine, I created a correlation plot that shows a percentage of native residues that were designed to all other residue identities. Figure 2.25 shows the native sequence recovery on the diagonal for the whole dataset while Figures 2.26 and 2.27 show the monomeric and homo-oligomer datasets, respectively. Additionally, it appears that all 20 native residue identities have some percentage that were designed to Leucine. This indicates that the cost of designing in a Leucine is low compared to all residues. In Rosetta, reference energies are used to represent the costs of designing in residues of a particular energy. Reference energies are usually given a weight of 1 in a scoring function and are only used during design. They are supposed to represent the energy of the unfolded state but have been tuned over time to optimize native sequence recovery in design experiments (Andrew Leaver-Fay et al., 2013).

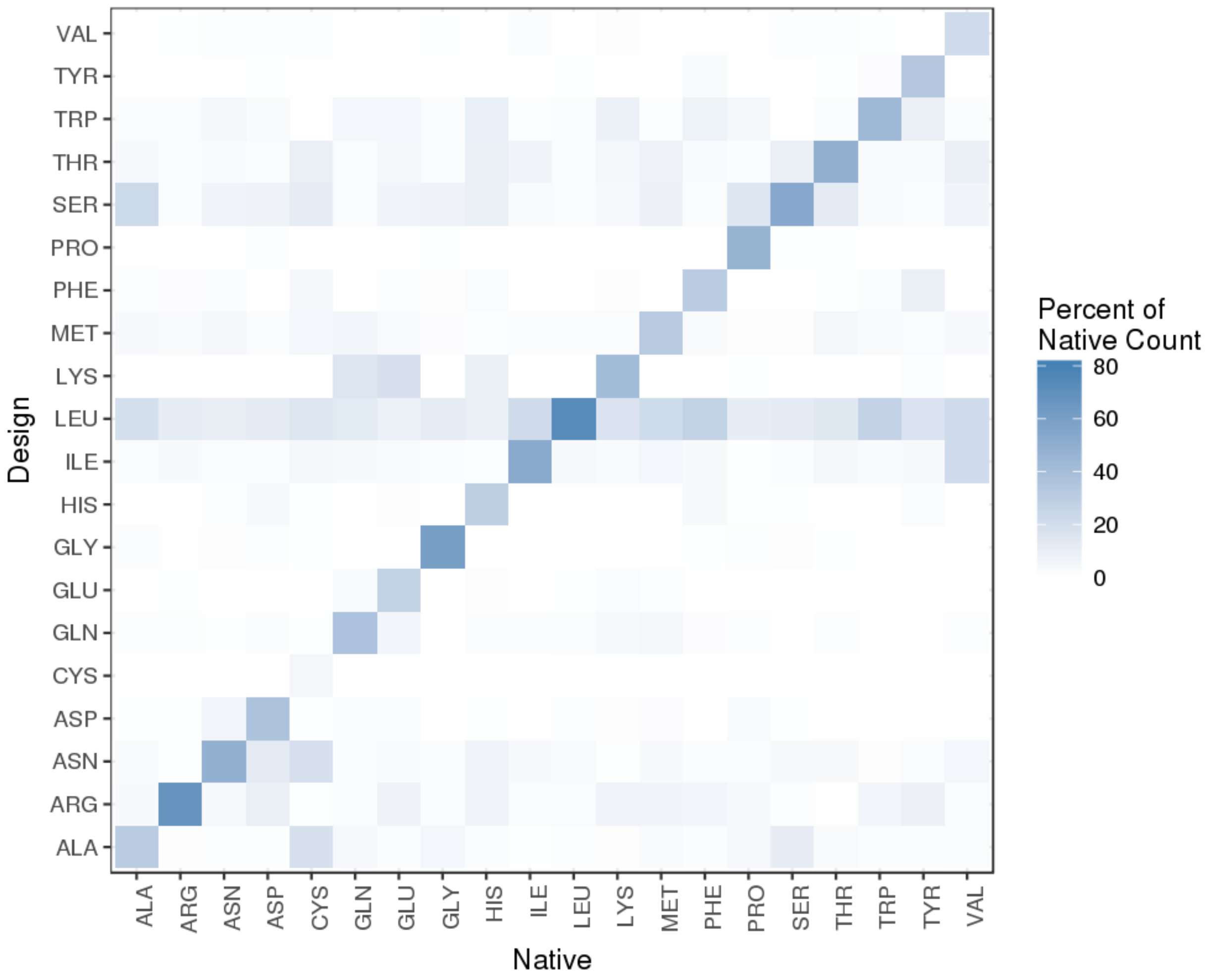


Figure 2.25. Correlation plot of residues designed in the place of native residues for the full dataset. Each column adds up to 100%, the diagonal represents when the residue is recovered during design. Leucine constitutes X percent of each native residue count. Alanine is often replaced by leucine and serine whereas valine is replaced by isoleucine, leucine, serine, and threonine.

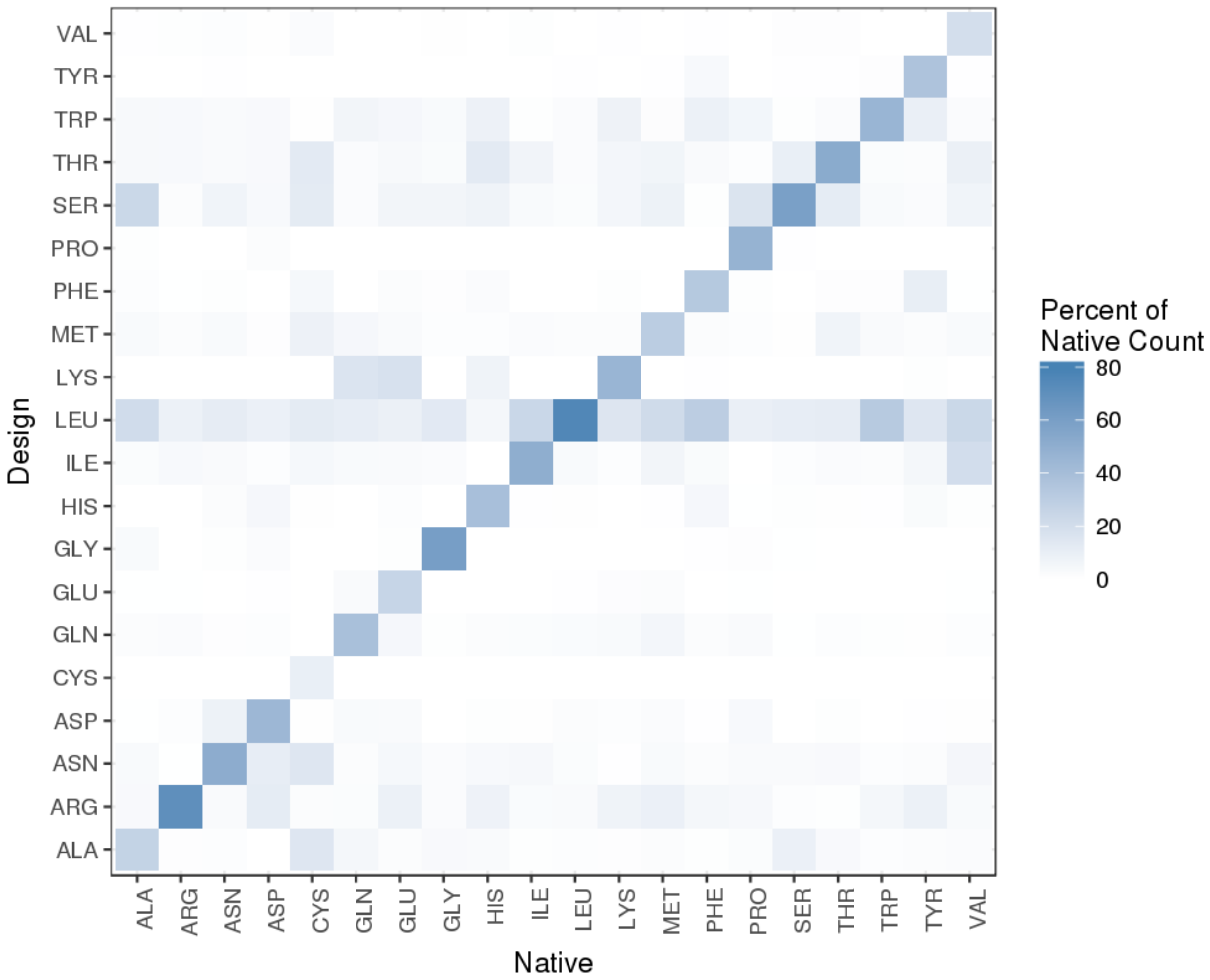


Figure 2.26. Correlation plot of residues designed in the place of native residues for the monomeric dataset. Each column adds up to 100%, the diagonal represents when the residue is recovered during design. Leucine constitutes X percent of each native residue count. Alanine is often replaced by leucine and serine whereas valine is replaced by isoleucine, leucine, serine, and threonine.

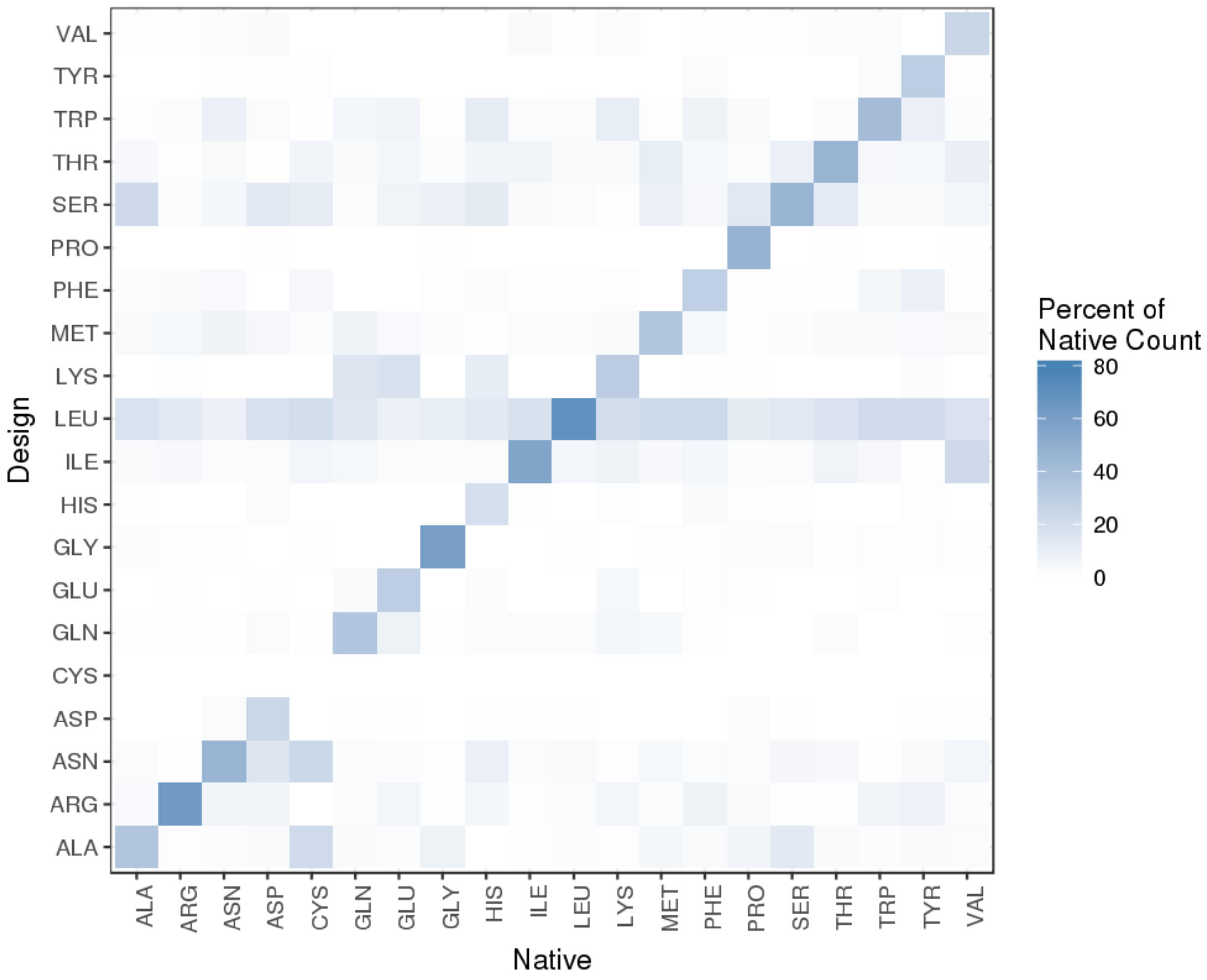


Figure 2.27. Correlation plot of residues designed in the place of native residues for the homo-oligomeric dataset. Each column adds up to 100%, the diagonal represents when the residue is recovered during design. Leucine constitutes X percent of each native residue count. Alanine is often replaced by leucine and serine whereas valine is replaced by isoleucine, leucine, serine, and threonine.

Bonuses and penalties can also be utilized by the Rosetta Design algorithm to favor or disfavor particular residues. A favor native residue bonus, as the name implies, is often used in design experiments to give a favorable energy to the native residue identity. This minimizes the amount of sequence variation from the native sequence to show the positions that are most likely to benefit from sequence optimization. Figures 2.28, 2.29, 2.30, and 2.31 show the same design experiment with favor native residue bonuses of 0.5,1,1.5, and 2, respectively. Although a higher favor native residue bonus lowered the usage of Leucine in the inner and outer hydrophobic regions, all differences improved uniformly, as would be expected from this such experiment.

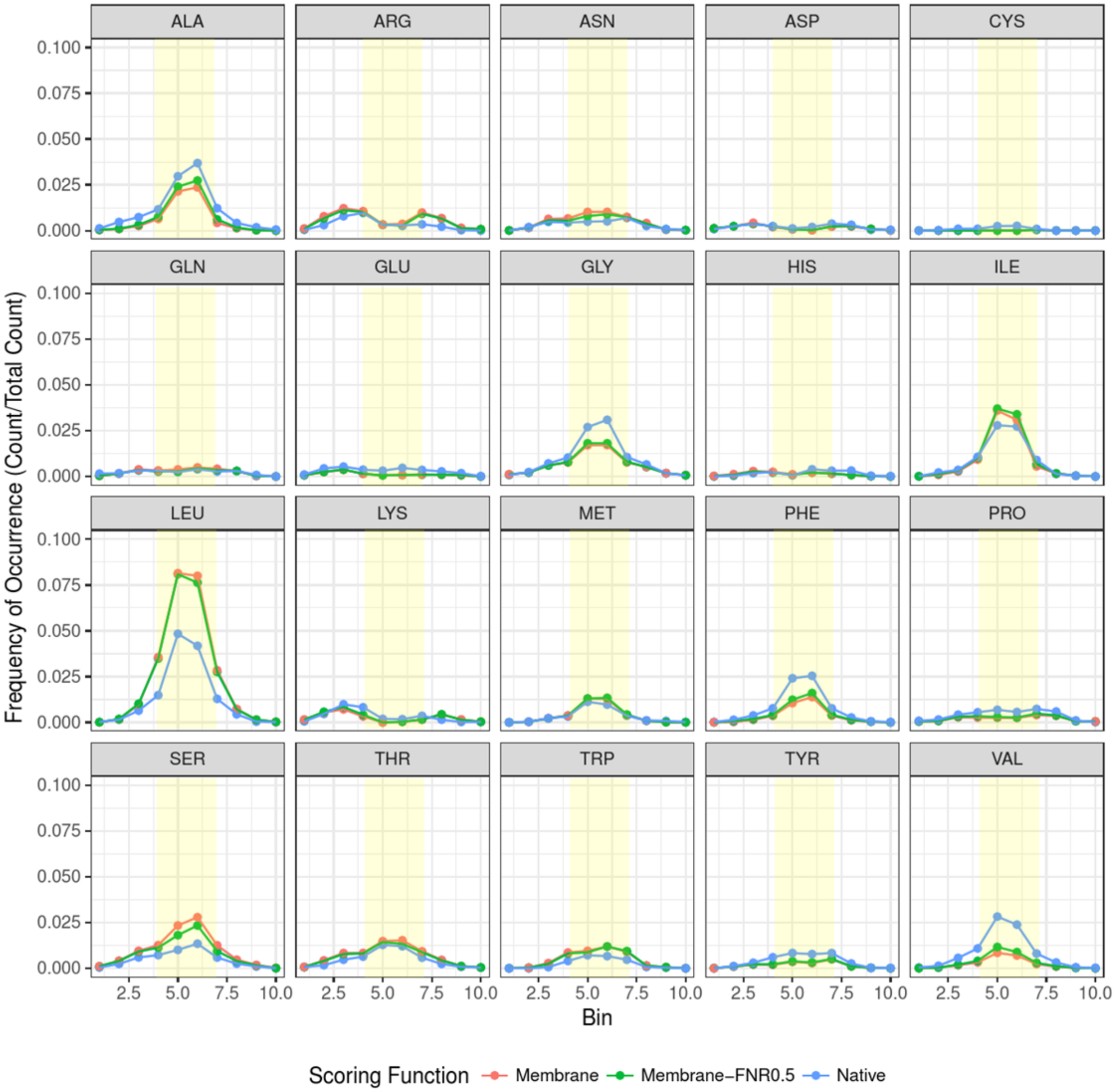


Figure 2.28. Frequency of occurrence of amino acids with respect to the distance from the membrane center for designs with a favor native residue bonus of 0.5. Dots indicate the frequency of occurrence of an amino acid seen at a particular distance from the membrane center, and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein. A bonus of 0.5

results in very little change relative to the frequency of occurrence for vanilla RosettaMembrane design.

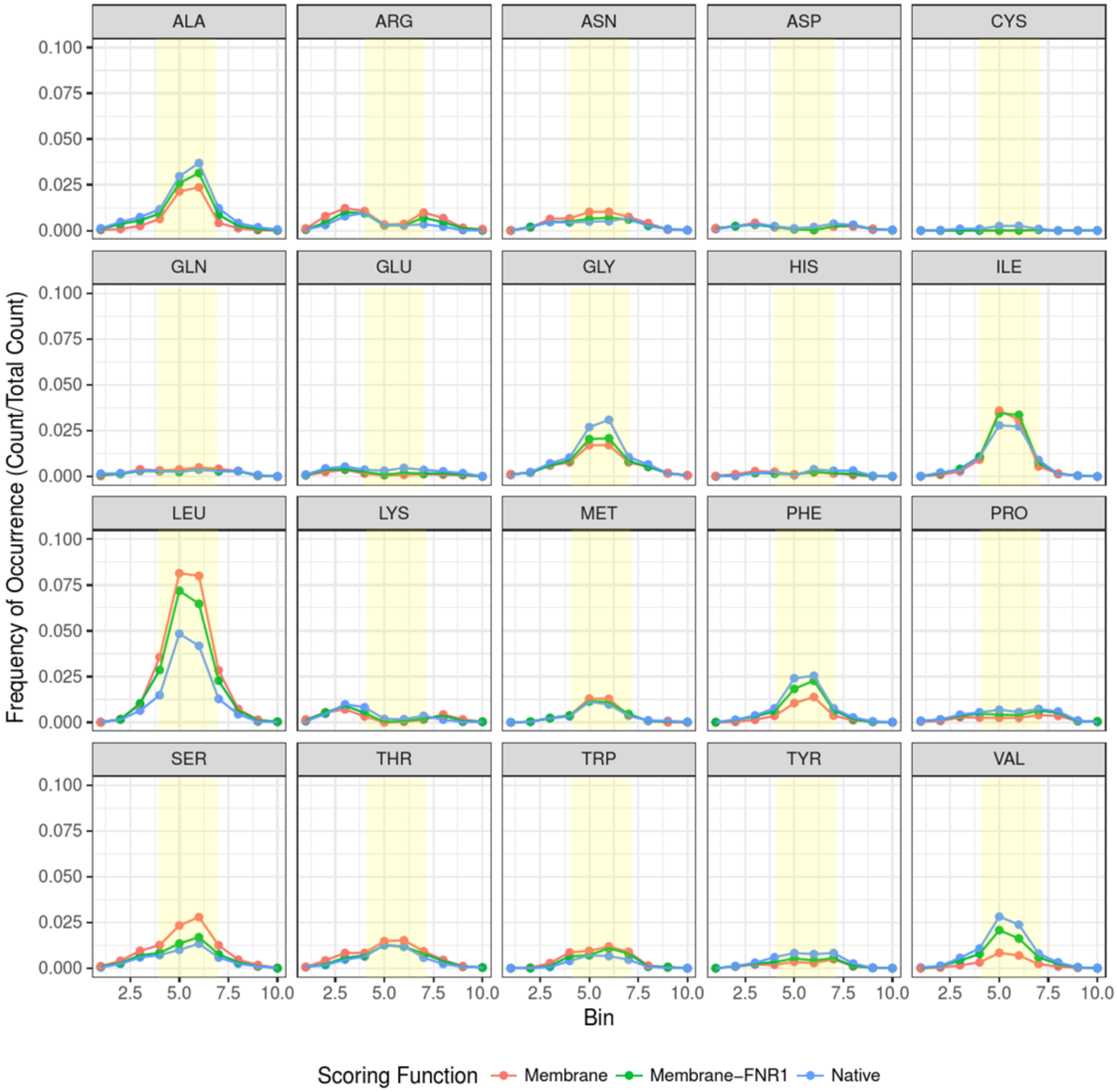


Figure 2.29. Frequency of occurrence of amino acids with respect to the distance from the membrane center for designs with a favor native residue bonus of 1. Dots indicate the frequency of occurrence of an amino acid seen at a particular distance from the membrane center, and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein. A bonus of 1 results in little change relative to the frequency of occurrence for vanilla RosettaMembrane design.

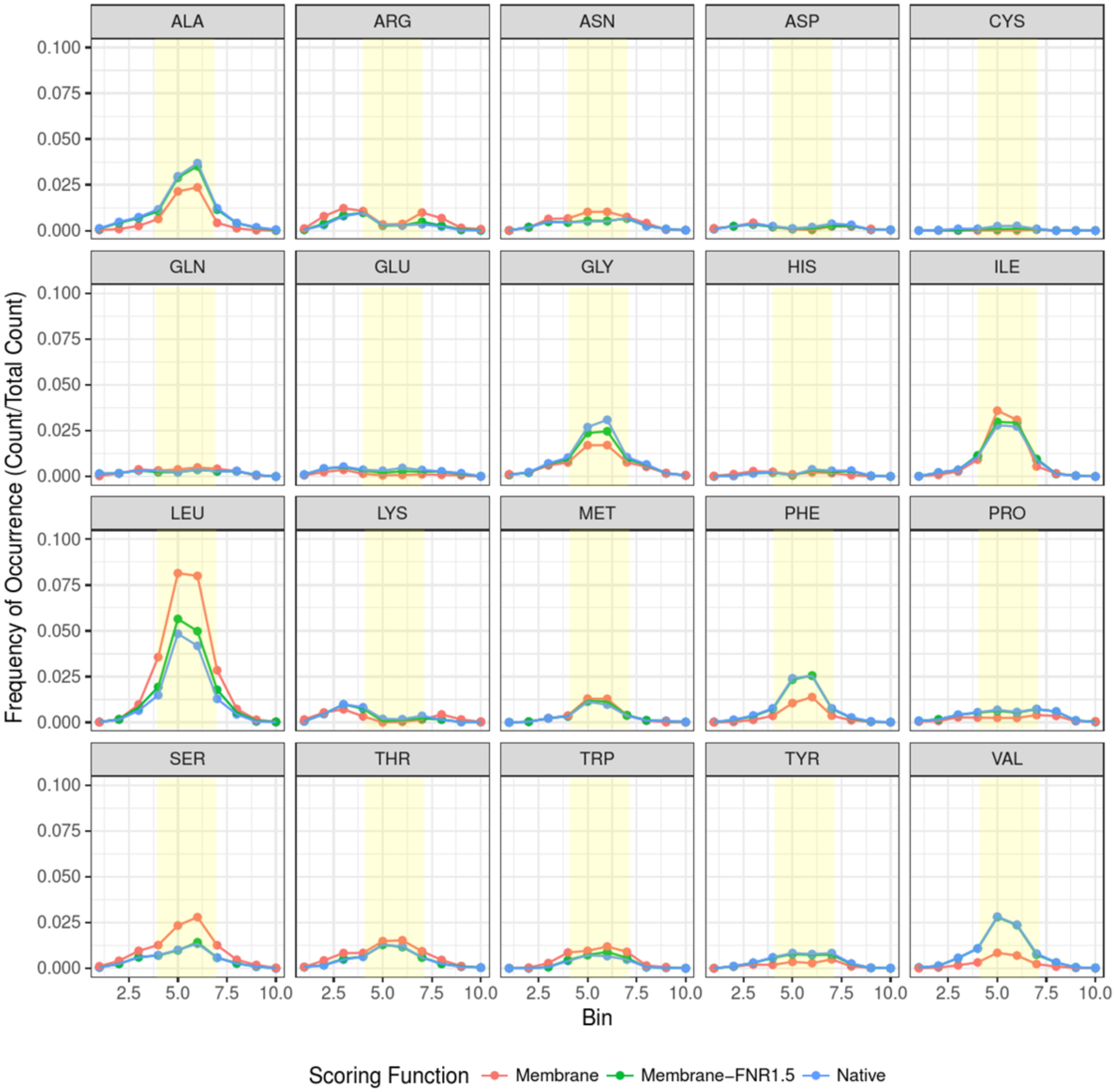


Figure 2.30. Frequency of occurrence of amino acids with respect to the distance from the membrane center for designs with a favor native residue bonus of 1.5. Dots indicate the frequency of occurrence of an amino acid seen at a particular distance from the membrane center, and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein. A bonus of 1.5 results in a closure in the gap between the frequency of occurrence for vanilla RosettaMembrane design and native membrane proteins. RosettaMembrane design with a 1.5 favor native residue bonus results in near native frequency of occurrence.

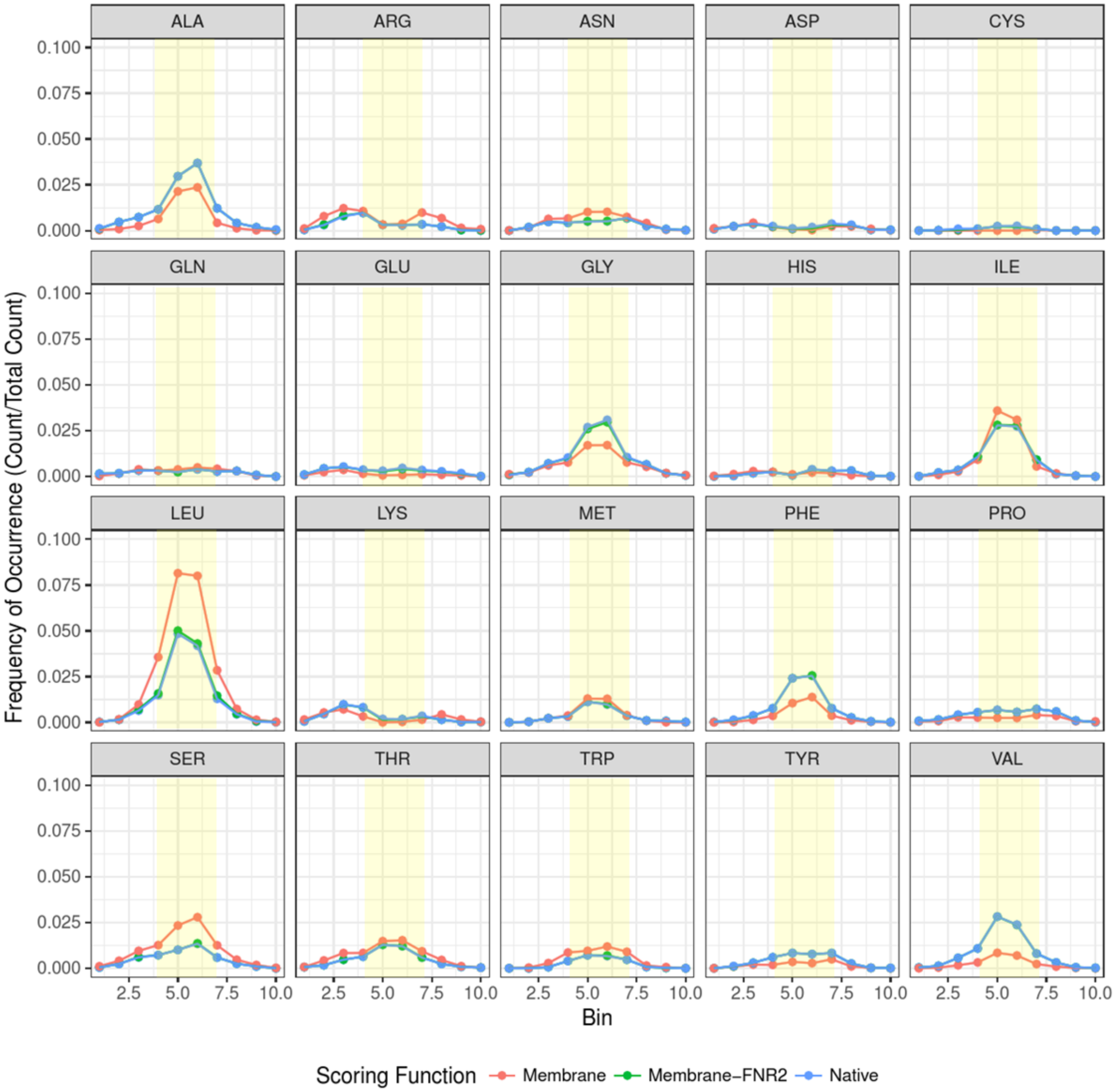


Figure 2.31. Frequency of occurrence of amino acids with respect to the distance from the membrane center for designs with a favor native residue bonus of 2. Dots indicate the frequency of occurrence of an amino acid seen at a particular distance from the membrane center, and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein. A bonus of 2 results in a complete closure in the gap between the frequency of occurrence for vanilla RosettaMembrane design and native membrane proteins.

While the favor native residue bonus did affect the energy of the individual amino acids by biasing the native identities, this did not change the energy function itself. As a proof of principal, I increased the energetic cost of designing in a Leucine from -0.1 to 0.2. The result was a dramatic decrease in the amount of Leucine designed in the inner and outer hydrophobic regions. The development of energy functions requires extensive testing and optimization. While it is clear that the cost of designing in a Leucine residue is too low in the current energy function, increasing the cost of residue only allowed for Isoleucine to be favored even more as indicated in Figure 2.32. This indicates that all reference energies in the membrane protein scoring function of Rosetta should be optimized to recapitulate native sequences.

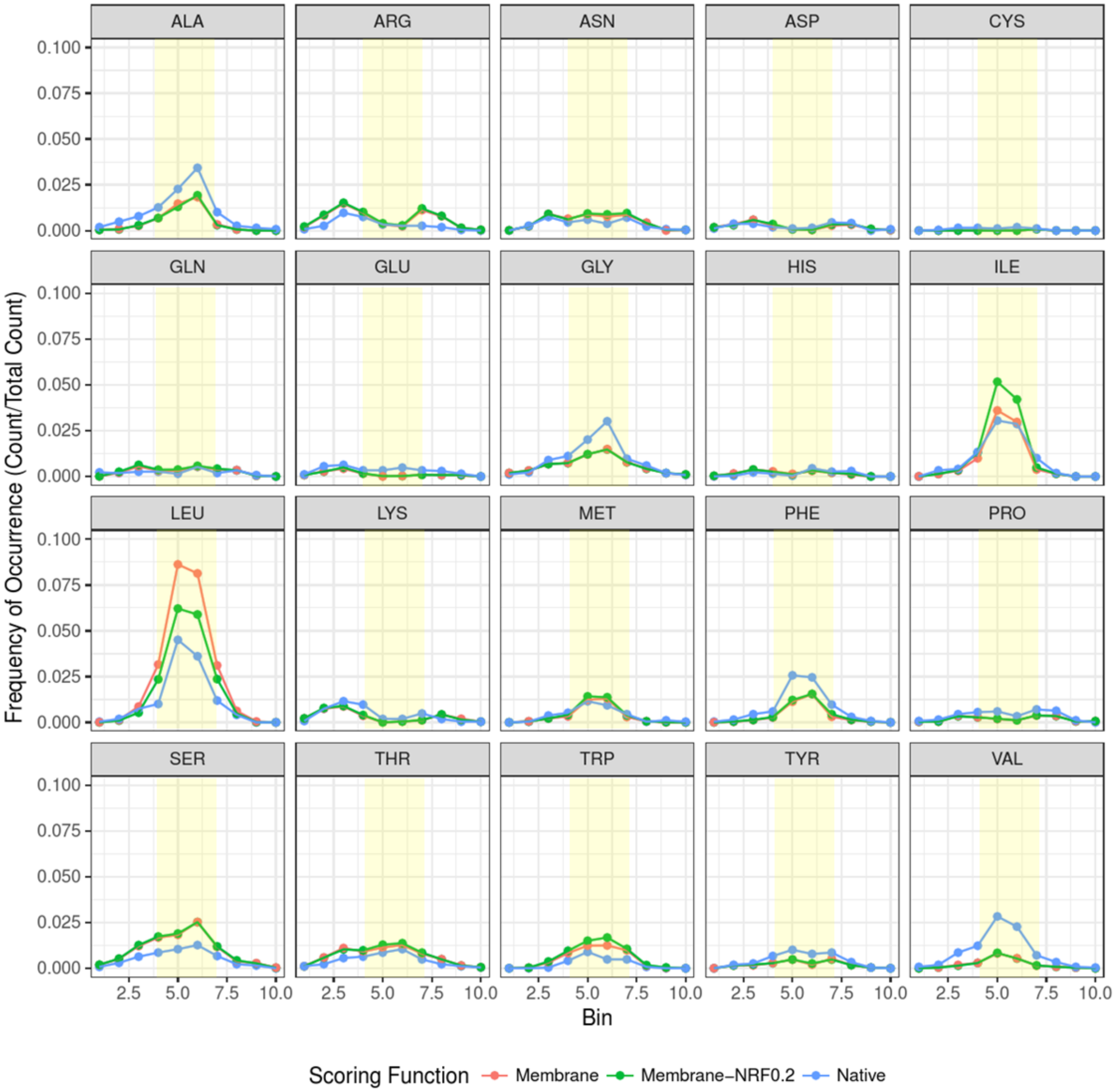


Figure 2.32. Frequency of occurrence of amino acids with respect to the distance from the membrane center for designs with an increased leucine reference energy. Dots indicate the frequency of occurrence of an amino acid seen at a particular distance from the membrane center, and lines are not to infer a continuous dataset. The distance bins are discrete and the lines are only to aid the eye in following the trend between layers. The yellow box overlays bins of distance that would contain the inner and outer hydrophobic layers of the protein. The increase of the leucine reference energy from -0.1 to 0.2 resulted in a reduction of occurrence of leucine, but an increase in isoleucine.

A closer look at trends seen in designs

Core residues have a better chance of recovering the native amino acid. For example, the core of 2xov has several residues surrounding Asparagine 64 that remain the same for both scoring functions (Fig. 2.33 A-C). The native core is likely well-packed with favorable hydrophobicity. The largest differences among designs are expected at the surface of the protein. While RosettaMembrane is designing towards an optimal hydrophobicity gradient so that the protein can partition in the membrane, Talaris is designing towards a soluble protein (Fig. 2.33 D-F). For this reason, many of the surface residues that were designed by Talaris are charged when the native protein would likely not tolerate multiple charged residues embedded in the membrane. As previously noted, an interesting finding was the abundance of Leucine on the surface of proteins designed using RosettaMembrane. In many cases, native hydrophobic residues, such as Phenylalanine at position 45 and Methionine 49 (Fig. 2.33 D-F), were replaced by Leucine. In homo-oligomers, the surface and core are similar to that in monomers; however, the homo-oligomers have interface regions between the sub-units. The interface regions should be designed similarly to the core in that they are surrounded by neighboring residues, provided that distance is close enough to be considered buried, despite those residues residing on a different chain. As expected, these regions, when well packed, will remain the native amino acid for both scoring functions (Fig. 2.33 G-I).

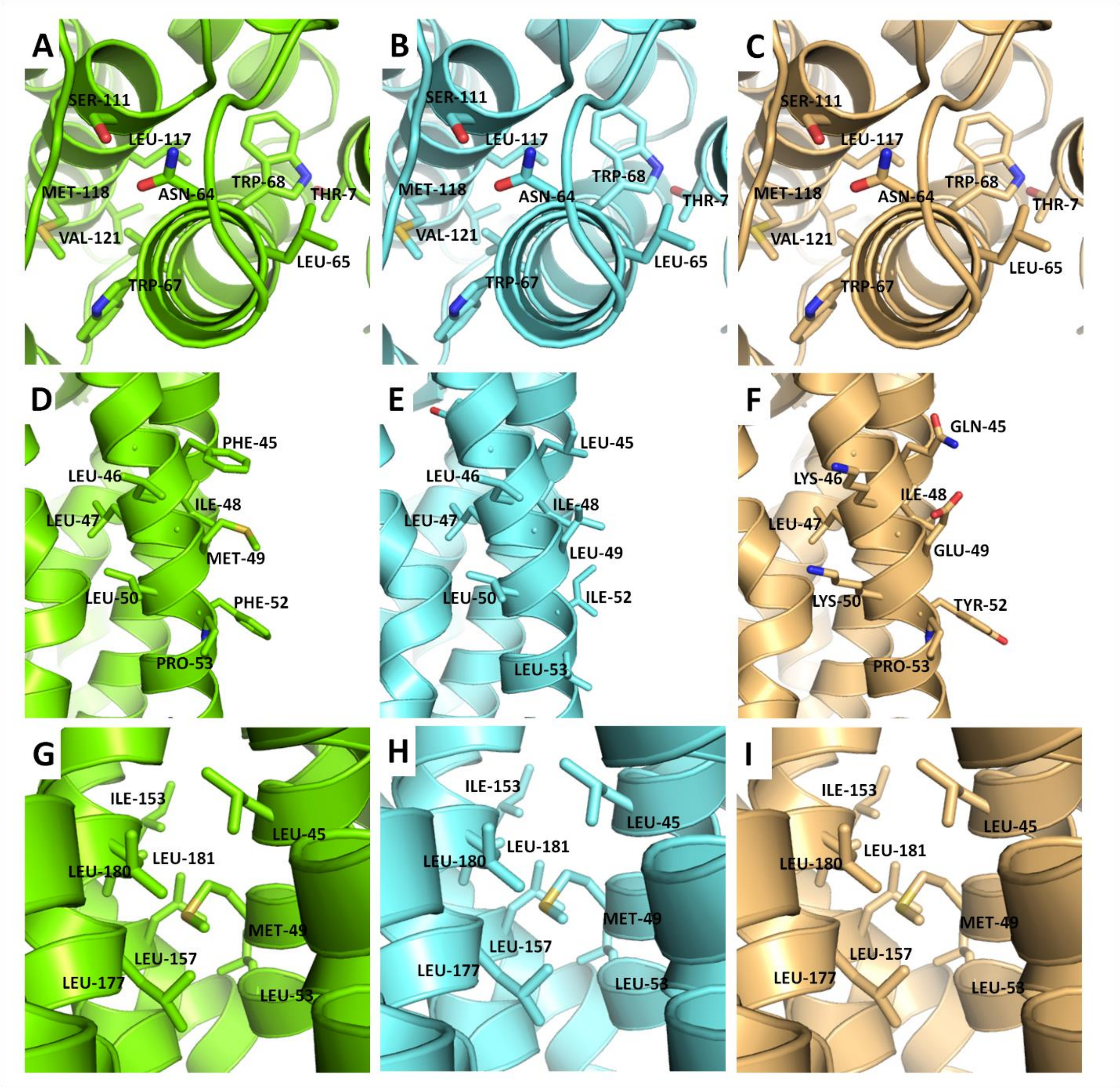


Figure 2.33. Atomic detail of designs compared to wild-type. A closer look at typical interactions at the core (A-C), surface (D-F), and homo-oligomeric interface (G-I). Representative cases were selected from 2xov (A-C), 1u19 (D-F), and 1fx8 (G-I). Green represents respective minimized native, aquamarine is RosettaMembrane, and light orange is Talaris.

RosettaMembrane designs membrane proteins that capture native-like properties. We have reported in-silico sequence redesign experiments using two different Rosetta scoring functions. Despite having similar sequence recoveries (Fig. 2.3), Talaris did not, as expected, appropriately design the surface. RosettaMembrane was developed to implicitly model an appropriate hydrophobic gradient that is often seen in native membrane proteins (Barth et al., 2007). RosettaMembrane designed a hydrophobic gradient that was native-like (Fig. 2.19). However, an artifact of designing in RosettaMembrane was the over-use of Leucine because of their high frequency at various layers in the membrane (Fig. 2.12, Fig. 2.34).

Also indicative of a native-like surface, the $\Delta G_{transfer}$ was above or near native for RosettaMembrane designs, whereas Talaris designs were near zero (Fig. 2.18, 2.20). Interestingly, although both scoring functions resulted in a similar amino acid composition (Fig. 2.8 A,B), the difference in composition of amino acid properties made it evident that RosettaMembrane designed in amino acids that were aliphatic, charged, or long and flexible more realistically (Fig. 2.8 D-F). Additionally, when evaluating position-specific scoring matrix recovery (PSSM), RosettaMembrane's strength was recovering hydrophobic residues like Isoleucine, Leucine, Valine, and Phenylalanine (Fig. 2.12). Despite both of the scoring functions resulting in similar amino acid composition, design using RosettaMembrane results in membrane protein designs with more native-like properties.

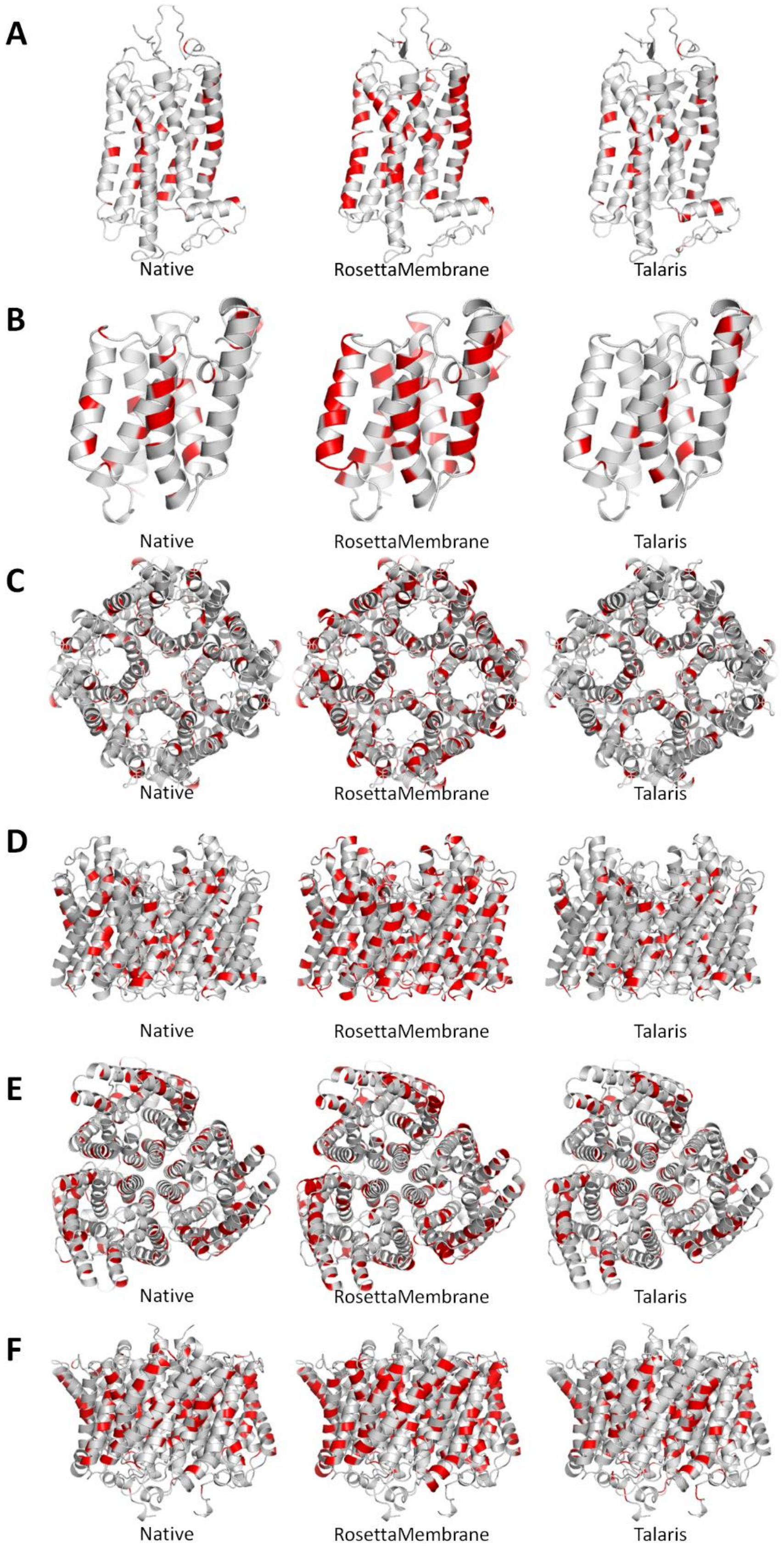

A
Native
RosettaMembrane
Talaris
B
Native
RosettaMembrane
Talaris
C
Native
RosettaMembrane
Talaris
D
Native
RosettaMembrane
Talaris
E
Native
RosettaMembrane
Talaris
F
Native
RosettaMembrane
Talaris

Figure 2.34. Visualization of Leucine on models. Top models were selected to visualize where Leucines occur in monomers 1u19 (A), 2xov(B) and in homo-oligomers 1fx8 (C-top down,D) and 3b9w (E-top down,F). Native structures (left) were compared to representative models of proteins designed using RosettaMembrane and Talaris. RosettaMembrane designs proteins with an abundance of Leucine at multiple layers of the membrane and surface residues. In homo-oligomers, Leucine is also seen in regions that are buried at the interface between subunits and in the core of the protein.

RosettaMembrane and Rosetta Symmetry can be used in conjunction to model obligate homo-oligomeric membrane proteins

Because many membrane proteins are functional as homo-oligomers, it is important the Rosetta Design algorithm works well with Rosetta Symmetry so that both the internal energy of all subunits and interface interactions are taken into account during the design process. Rosetta Symmetry is ideal for larger, symmetric systems because the subunits in homo-oligomers are moved in the same way, which enables the sampling process to rapidly occur. The homo-oligomeric set performed similarly to the monomeric set in amino acid composition and slightly better in recovering native-like properties. To ensure this comparison was not an artifact of the sets of proteins, the homo-oligomeric set was modeled as monomers in a separate design experiment. This revealed that although the patterns for amino acid composition were similar, the monomeric representation deviated further from the native (Fig. 2.8 B,C) indicating that homo-oligomeric modeling result in more native-like designs.

2.3 Conclusions

This study illustrates that with minimized structures, membrane proteins have core sequence recovery rates of 52-63% for monomeric membrane proteins and 53-65% for homo-oligomeric membrane proteins. These rates are similar to the 51% core sequence recovery rates calculated from a large soluble protein set (B. Kuhlman & Baker, 2000). The chance of designing a position with the correct amino acid identity is roughly 5% (selecting the correct amino acid out of 20), so a recovery of approximately 50% indicates the algorithm is working well. Increasing sequence recovery even further would involve extensive backbone minimization and/or an improved scoring function. We find that PSSM recovery (here averaging around 70%) is a more reliable metric because the recovery tolerates mutations that have been seen in evolution. Additionally, to avoid minimizing structures that imprint the native sequence, we recommend using CSC to prepare structures for design as this reduces backbone RMSD from native during minimization and still achieves moderately high sequence recovery for a range of starting resolutions.

While RosettaMembrane designs native-like surface hydrophobicity, it is important to note that RosettaMembrane has a tendency to favor Leucine over other hydrophobic residues at these positions. This may be due to high occurrence of Leucine for proteins in the original training set. An updated RosettaMembrane scoring function with a larger, more diverse, and higher resolution membrane protein knowledge-base may help dampen this bias. Finally, as membrane protein structures have varying membrane thicknesses, an accurate depiction of the hydrophobicity gradient during modeling and design of membrane proteins in Rosetta could improve the quality of native-like designs even further.

Table 2.3. Membrane Protein Benchmark Set. The bioassembly used for the duration of the modeling process, unless otherwise noted, is stated in the assembly column. Residues selected are listed by the chain ID and range of residue numbers; homo-oligomeric assemblies were created by applying symmetry to the chain listed. Solvent and ions were excluded for the duration of this study.

PDB ID	Protein Name	Resolution in Angstroms	Assembly	Residues from PDB	Chain Length	Protein Length	# TM Spans	Percent of Residues in Membrane

1FX8	Glycerol Facilitator	2.2	Tetramer	A 6-259	254	1016	24+8 half	61.9
1K4C	KcsA Potassium Channel, H+ with Fab	2.0	Tetramer	C 22-124	103	412	8	47.6
1M0L	Bacteriorhodopsin	1.5	Trimer	A 5-231	222	666	21	59.9
1OTS	H+/Cl- exchange transporter	2.5	Dimer	A 17-460	444	888		70.1
1U19	Rhodopsin (bovine outer segment)	2.2	Monomer	A 1-348	348	348	7	47.7
2C3E	Mitochondrial ADP/ATP Carrier	2.8	Monomer	A 1-293	293	293	6	60.6
2UUI	(Apo) Leukotriene Synthase	2.0	Trimer	A -5-149	155	465	12	46.8
2VPZ	Polysulfide Reductase	2.4	Dimer	C2-251	250	500	16	72.7
2XOV	Rhomboid-Family intramembrane protease	1.7	Monomer	A91-271	181	181	6	77.9
3B9W	Rh50 protein	1.3	Trimer	A8-369	362	1086	33	64.6
3GIA	(Apo) ApcT Na+-independent Amino Acid Transporter	2.4	Monomer	A3-435	433	433	12	65.5
3K3F	Urea Transporter	2.3	Trimer	A-1-334	332	996	30	72.4
3KLY	FocA formate transporter w/o formate	2.1	Pentamer	A22-278	257	1285	30	64.3
3M71	SLAC1 anion channel TehA homolog	1.2	Trimer	A6-313	308	924	30	68.2

3O0R	Nitric Oxide Reductase subunit B	2.7	Monomer	B10-458	449	449	12	61.3
3RLB	ThiT, S component of the Thiamin Transporter	2.0	Dimer	A7-182	176	352	12	70.3
3V5U	Sodium Calcium Exchanger (MCX)	1.9	Monomer	A1-304	297	297	10	67.8
3ZOJ	AQY1 Yeast Aquaporin	0.88	Tetramer	A11-273	263	1052	24+8 half	59.9
4A2N	Isoprenylcysteine carboxyl methyltransferase	3.4	Monomer	B1-192	192	192	5	57.7
4IKV	Proton-dependent oligopeptide transporter	1.9	Monomer	A2-493	492	492	14	60.2

2.4 Methods

A set of 20 membrane proteins with resolutions ranging from 0.88-3.4 Å was compiled. Twelve of these membrane proteins are modeled as homo-oligomers (Table 2.3). All of the coordinates were obtained from the PDB. Solvent and ions were excluded for the duration of this study. Span files that specify the trans-membrane spanning region were created using information obtained from PDBTM (Kozma, Simon, & Tusnády, 2013). The symmetry definition files were created using the non-crystallographic symmetry mode in the make_symmdef_file.pl script provided in Rosetta. This mode calculates the point symmetries using the homo-oligomers present in the PDB file, or from symmetry mates generated in Pymol from the original PDB file. The RosettaScripts XML scripting language framework (Fleishman et al., 2011) from the Rosetta week 52 build was used for all of the protocols tested. The Rosetta software suite is publicly accessible and free for non-commercial use.

Pre-minimization trials

Five minimization protocols were tested on this benchmark set: pack-only where the backbone is not perturbed and only the side-chains conformations are optimized; minimize with constraints (MWC) where harmonic constraints are used to minimize both the backbone and side-chains to within nine Å of the starting structure (used to prepare structures for thermostability calculations (Kellogg et al., 2011); FastRelax with an added constraint to the start coordinates (CSC) which only allows minimal deviations from the initial backbone; FastRelax, the standard minimization protocol ; and DualSpace relax (Conway, Tyka, DiMaio, Konerding, & Baker, 2014) which uses a combination of internal and Cartesian minimization. Three of these protocols, CSC, FastRelax, and Dualspace, were set up using the FastRelax mover in Rosetta Scripts and can also be set up using the relax application by including commandline options appropriate for each protocol. For pack-only and MWC, the appropriate applications and options were used (please see a complete, detailed protocol capture in the Appendix B)

Full redesign to assess pre-minimized structures

Full redesign, where all canonical amino acids identities are allowed to be sampled at each position, was performed on the pre-minimized membrane protein sets. For each minimization protocol, two to three top models by score and RMSD for each membrane protein were chosen as the input models for full design to introduce backbone diversity. Full design was set up using PackRotamersMover and the SymPackRotamersMover, where appropriate, to generate design models of each minimized model. The top ten percent models by score were chosen for sequence recovery analysis (protocol capture, Appendix B).

Full redesign using various scoring functions

Full redesign was performed on the top three models by score and RMSD from the CSC protocol. The scoring functions tested were the RosettaMembrane full atom smoothed potential (membrane_highres_Menv_smooth.wts) and Talaris (talaris2013.wts). Full design was set up using PackRotamersMover and SymPackRotamersMover, where appropriate, to generate design models from each selected minimized model. The top scoring ten percent models were used to calculate sequence recovery of the native protein sequence (protocol capture, supplementary materials parts 2a, 2b).

Sequence analysis of redesigned proteins

The top ten percent of designs by score were analyzed. Native sequence recovery was calculated for the full protein, core residues (a residue with at least 24 contacts within a C-β distance of ten Å), and surface residues (a residue with at most 16 contacts within a C-β distance of ten Å) using the Sequence Recovery application in Rosetta. Additionally, we determined whether the scoring functions reproduced native-like amino acid composition.

CHAPTER 3

COMPUTATIONAL PREDICTIONS FOR MUTATION-INDUCED STABILITY CHANGES IN MEMBRANE PROTEINS

This chapter includes published work from:

Kroncke, Duran, Mendenhall, Blume, and Meiler, 2017

Author contribution: I contributed a substantial amount of data and data analysis to the manuscript entitled "Documentation of an Imperative to Improve Methods for Predicting Membrane Protein Stability" published in Biochemistry as an ACS AuthorChoice open access article (Kroncke et al., 2016). I calculated the thermostabilizing effects of these mutations using the Rosetta ddg_monomer in high-resolution and low-resolution modes. Both of these methods are analyzed through the entirety of the manuscript. I was active throughout the entire process of data analysis. I also developed a way to use ddg_monomer in concert with RosettaMembrane in a high-resolution mode, and used the existing RosettaMP protocol for evaluating the low-resolution protocol. Table 1 was also of my creation.

Abstract

There is a compelling and growing need to accurately predict the impact of amino acid mutations on protein stability for problems in personalized medicine and other applications. Here the ability of 10 computational tools to accurately predict mutation-induced perturbation of folding stability (ΔΔG) for membrane proteins of known structure was assessed. All methods for predicting ΔΔG values performed significantly worse when applied to membrane proteins than when applied to soluble proteins, yielding estimated concordance, Pearson, and Spearman correlation coefficients of <0.4 for membrane proteins. Rosetta and PROVEAN showed a

modest ability to classify mutations as destabilizing ($\Delta\Delta G < -0.5$ kcal/mol), with a 7 in 10 chance of correctly discriminating a randomly chosen destabilizing variant from a randomly chosen stabilizing variant. However, even this performance is significantly worse than for soluble proteins. This study highlights the need for further development of reliable and reproducible methods for predicting thermodynamic folding stability in membrane proteins.

3.1 Introduction

Each individual's genome has, on average, 10000−20000 nonsynonymous single-nucleotide polymorphisms (nsSNPs) (Kroncke et al., 2015). Deleterious, loss-of-function nsSNPs constitute the most common cause of monogenic disorders (Stenson et al., 2012; Z. Wang & Moult, 2001; Yue, Li, & Moult, 2005). Substantial evidence suggests a majority of disease-promoting nsSNPs act, at least in part, by destabilizing the folded conformation of the encoded protein (Casadio, Vassura, Tiwari, Fariselli, & Luigi Martelli, 2011; Shi & Moult, 2011; Stefl, Nishi, Petukh, Panchenko, & Alexov, 2013; Z. Wang & Moult, 2001; Yue et al., 2005). The resulting loss of thermodynamic stability leads to a reduced population of functional protein available to cells, which in some cases is compounded by the toxicity of the misfolded protein (Calamini & Morimoto, 2012; Knowles, Vendruscolo, & Dobson, 2014; Valastyan & Lindquist, 2014). The more accurately mutation-induced changes in protein stability can be determined, the more accurately and specifically we can predict loss-of-function phenotypes for previously uncharacterized point mutations, a growing concern as more genomes are sequenced to unveil variants of unknown significance (Kroncke et al., 2015). There are many algorithms that predict changes in folded protein stability caused by single- or multiple-amino acid mutations. Some approaches rely on known protein structures using functions that predict the energetic perturbation introduced by the mutation (Guerois et al., 2002). Other methods train machine learning methods on large data sets to combine selected physical, statistical, and empirical features for stability predictions (Berliner, Teyra, Çolak, Lopez, & Kim, 2014; Yang, Chen, Tan, Vihinen, & Shen, 2013). For water-soluble proteins, several algorithms are able to predict mutation-induced change in stability with a Pearson correlation coefficient near or above 0.7 (Table 3.1; Figure 3.1); however, the performance of these methods on membrane proteins is an open question. Membrane proteins fold and reside in a heterogeneous environment – a lipid bilayer bounded on both sides by water – with distinct forces driving folding and unfolding compared to soluble proteins, and therefore may require treatment separate from that of soluble proteins (Cymer, von Heijne, & White, 2015; Hong, Park, Flores Jimenez, Rinehart, & Tamm, 2007; Neumann, Klein, Otzen, & Schneider, 2014; Popot & Engelman, 2000).

Membrane protein structures comprise only ~1% of the protein structure database (http://www.rcsb.org/pdb/home/ and http://blanco.biomol.uci.edu/mpstruc/), and thermodynamic stability measurements of membrane proteins are grossly underrepresented. This paucity of data dictates that all currently available ΔΔG calculators have been trained and refined from data sets strongly biased toward soluble proteins. Here we evaluate the ability of current methods to predict amino acid mutation-induced free energy changes in membrane protein stability in cases both for which an atomic-resolution structure is available and for which stabilities of wild-type and mutant forms have been measured.

Table 3.1. Reported performance of programs evaluated.

Method	**Reported Correlation Coefficient**
Rosetta High	0.69
Rosetta Low	0.68
iMutant3	0.69
FoldX	0.8
Mcsm	0.824
Sdm	0.58
Duet	0.71
Ddgm8	0.65
Ddgm47	0.82
Provean[a]	0.74
Elaspic	0.77
Easemm	0.56

[a] indicates that the value is derived from an activity assay

3.2 Methods

We used all available (as of January 2016) experimental ΔΔG data sets for mutant forms of membrane proteins of known structure. The relevant Protein Data Bank (PDB) codes are as follows: 1PY6 for bacteriorhodopsin (Faham et al., 2004), 1AFO for glycophorin A (MacKenzie, Prestegard, & Engleman, 1997), 2XOV for the *Escherichia coli* rhomboid protease (GlpG) (Vinothkumar et al., 2010), 2K73 for disulfide formation protein B (DsbB) (Y. Zhou et al., 2008), 1QD6 for outer membrane phospholipase A1 (OmpLA) (Snijder et al., 1999), 1QJP for outer membrane protein A (OmpA) (Pautsch & Schulz, 2000), and 3GP6 for the lipid A palmitoyltransferase (PagP) (Cuesta-Seijo et al., 2010). The 224 rigorously determined ΔΔG measurements originated from the following studies: bacteriorhodopsin (Cao, Schlebach, Park, & Bowie, 2012; Faham et al., 2004; N. H. Joh et al., 2008; N H Joh, Oberai, Yang, Whitelegge, & Bowie, 2009; Schlebach, Woodall, Bowie, & Park, 2014; Yohannan et al., 2004), glycophorin A (Fleming, Ackerman, & Engelman, 1997; Fleming & Engleman, 2001), GlpG (Baker &

Urban, 2012; Paslawski et al., 2015), DsbB (Otzen, 2011), OmpLA (Moon & Fleming, 2011), OmpA (Hong et al., 2007), and PagP (Huysmans, Baldwin, Brockwell, & Radford, 2010) (Table 3.2). We ensured that each of the studies reported ΔΔG values and did not extrapolate these values from thermal unfolding experiments. The type of experiment that the ΔΔG value was derived from can be found in table 3.2. Additionally, the ΔΔG values were collected so that a negative ΔΔG indicates that it is the result of a destabilizing mutation.

Table 3.2. Summary of the dataset

Protein name	**PDB Code**	Type	**Method**	**Number of mutations in dataset**
Bacteriorhodopsin	1PY6	Helical	SDS titration	67
Glpg	2XOV	Helical	SDS titration	71
DsbB	2K73	Helical	SDS titration	12
Glycophorin	1AFO	Helical	Dimerization AUC	12
OMPA	1QJP	Barrel	Urea titration	12
OMPLA	1QD6	Barrel	Urea titration	31
PagP	3GP6	Barrel	Urea titration	19
	Total			224

3.3 Results and discussion

We tested available methods for which servers or software were available online and functional as of January 2014 or for which the authors of published algorithms were responsive to our request for software (Table 3.3).

Protein stability programs

The following programs were used to predict ΔΔG values for each membrane protein mutation in the experimental database mentioned above: Rosetta (revision 58019) with both low-resolution (Rosetta-low) and high-resolution (Rosetta-high) protocols (Kellogg et al., 2011), I

Mutant (3.0; http://gpcr2.biocomp.unibo.it/cgi/predictors/I-Mutant3.0/I-Mutant3.0.cgi) (Capriotti, Fariselli, & Casadio, 2005), FoldX (3.0, beta 6.1) (Guerois et al., 2002), mCSM (Pires, Ascher, & Blundell, 2014b), SDM (Worth, Preissner, & Blundell, 2011), DUET (http://bleoberis.bioc.cam.ac.uk/duet/stability) (Pires, Ascher, & Blundell, 2014a), PPSC (Prediction of Protein Stability, version 1.0) with the 8 (M8) and 47 (M47) feature sets (Yang et al., 2013), PROVEAN (http://provean.jcvi.org/seq_submit.php) (Y. Choi, Sims, Murphy, Miller, & Chan, 2012), ELASPIC (http://elaspic.kimlab.org/) (Berliner et al., 2014), and EASE-MM (Folkman, Stantic, Sattar, & Zhou, 2016). We also tested the standard Rosetta ddg_monomer application replacing the minimization score function score12 with membrane_highres_Menv_smooth (RosettaMembrane). In addition, we tested the RosettaMP ΔΔG calculating framework, RosettaMPddG. Both attempts failed to improve performance (Figure 3.2). The membrane protein scoring functions as they add nothing in accuracy and discrimination for calculating ΔΔG values in Rosetta.

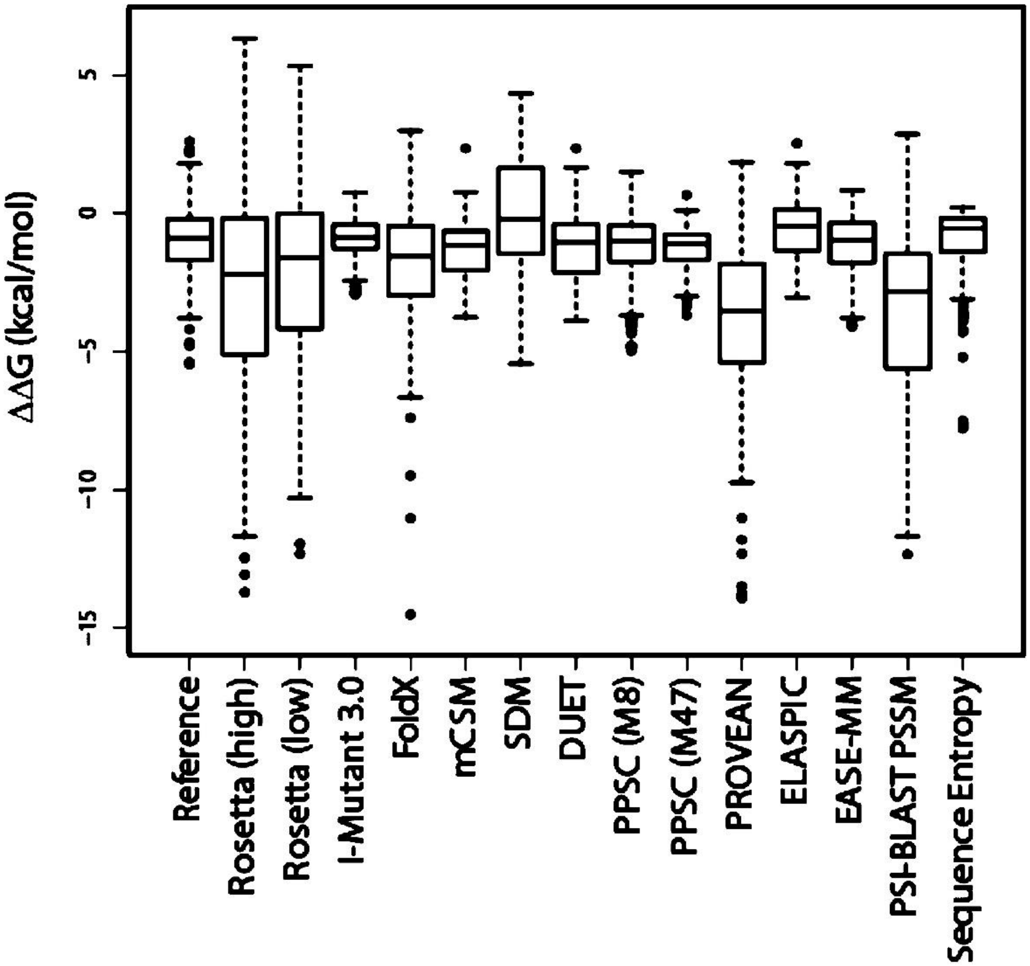


Figure 3.1. Boxplot of experimental (reference) and predicted value distributions. The middle line in the box is the median, and upper and lower bounds to the boxes are the upper and lower quartiles, respectively. Nonoutlier extrema are bracketed with dashed lines above and below the upper and lower quartiles, respectively. Dots are outliers beyond 1.5 times the upper or lower quartile.

Table 3.3. Summary of Methods Evaluated

Name	Brief Description	Method[a]	Calibrated[b]	Sequence	Pearson[c]	Stability Data Set[d]
Rosetta	Structure knowledge-based potential. Score terms considered: van der Waals, electrostatics, solvation, hydrogen bond, rotamer probability. ddG_monomer application	N/A			0.69 (high) 0.68 (low)	ProTherm (test set)
I Mutant 3.0	Support vector machine (SVM)-based predictor; can use sequence information and structure information to predict destabilizing, neutral, and stabilized	SVM	X	X	0.69	Thermodynamic Database for Proteins and Mutants ProTherm (September 2005)
FoldX	Empirical force field calibrated with experimental ddG values. Score terms considered: van der Waals, solvation, hydrogen bonding, water bridges, electrostatic, entropy of backbone and side chain, and atomic clashes	grid search	X		0.8	Derived from ProTherm
mCSM	Graph-based structural signatures: distance patterns between atoms to represent the environment. Also considers pharmacophore changes and experimental conditions. Supervised learning machine learning methods trained on regression and classification	ANN	X		0.82	Derived from ProTherm
SDM	Statistical potential energy function (structure): evaluates amino acid structural propensities in homologous protein families	N/A		X	0.58	Derived from ProTherm
DUET	SVM that combines mCSM and SDM methods	SVM	X	X	0.71	ProTherm (low-redundancy set)

PPSC (M8)	SVM with eight attributes: hydropathy, isotropic surface area, electronic charge, volume, contact energy	SVM	X		0.65	Derived from ProTherm
PPSC (M47)	SVM trained with 8 + 40 additional protein features from (I Mutant 2)	SVM	X		0.82	Derived from ProTherm
PROVEAN	Pairwise sequence alignment scores to predict effects of a mutation, including deletions, insertions, and multiple substitutions	N/A		X	0.71[e]	Derived from UniProtKB and Swiss-Prot databases
ELASPIC	Machine learning approach that combines semiempirical force fields, sequence conservation scores, and structural information through stochastic gradient boosting of decision trees	SGBT-DT	X	X	0.77	ProTherm
EASE-MM	Sequence-based SVM model that evaluates the predicted secondary structure and accessible surface area of the region of interest	SVM	X	X	0.56	Derived from ProTherm

[a]Type of machine learning method used: artificial neural network (ANN), support vector machine (SVM), and stochastic gradient boosting of decision trees (SGBT-DT). [b]The predictive method is calibrated to experimental ΔΔG values. [c]Reported Pearson correlation coefficient. [d]Used to derive both training and testing sets unless otherwise noted. [e]Activity correlation

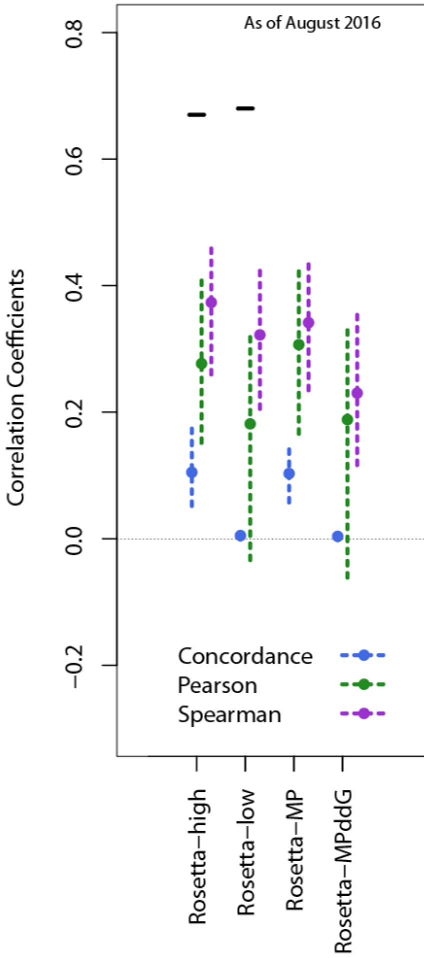


Figure 3.2. Comparison of Concordance, Pearson, and Spearman correlation coefficients for Rosetta based prediction methods that include membrane-specific score terms, ddg_monomer high-resolution protocol using RosettaMembrane, and low-resolution RosettaMPddG. As expected, this resulted in very similar correlations for RosettaMembrane (CC: 0.103, PC: 0.307, SC: 0.342) and Rosetta-High (CC: 0.11, PC: 0.28, SC: 0.37) as well as for RosettaMPddG (CC: 0.00, PC: 0.19, SC: 0.23) and Rosetta-Low (CC: 0.01, PC: 0.18, SC: 0.32).

To compare the performance of each ΔΔG calculation method with what could be obtained from sequence information alone, we calculated two parameters. First, the likelihood of a specified amino acid mutation being observed among the wild-type (WT) sequences comprising a particular protein family was assessed according to the position-specific iterative basic local alignment search tool-derived position specific scoring matrix (PSI-BLAST PSSM). PSI-BLAST PSSM values were calculated, as follows. The PSI-BLAST position specific scoring matrix value for a given mutant residue amino acid type was subtracted from the value for the native residue (PSI-BLAST employed the UniRef50, nonredundant sequence database, 5-iterations, e-value cutoff of 0.01). This metric gives an estimation of the evolutionary penalty for substituting the WT residue with the specified mutant amino acid. Second, the Shannon (or "sequence entropy") entropy was determined from PSI-BLAST results. Sequence entropy is a description of how often the identity of a particular residue in a protein changes from family member to family member. Shannon/sequence entropy is the PSSM value for amino acids located at a particular position. This parameter is agnostic with regard to the amino acid type of both the mutated-in and native residue. Instead, the Shannon/sequence entropy reports the likelihood that a change in residue identity is evolutionarily tolerated. All numbers were formatted so that negative values indicate destabilization.

For each predictive method, the experimental versus predicted ΔΔG data were processed using an in-house R script to calculate correlation coefficients and area-under-the-curve (AUC) values. To analyze the collected data set on the basis of several features, we parsed out and evaluated separately point mutations according to the following classifications: those impacting α-helical versus β-barrel proteins, those with a point mutation site in the aqueous phase, in the aliphatic phase, or at the water−membrane interface, and mutations at positions that were either buried within the protein or exposed to solvent or lipid (Figures 3.5-3.13). We analyzed the set of predictions for each protein separately and also parsed out point mutations involving proline or glycine (Figures 3.14-3.20). Concordance, Pearson, and Spearman correlations were computed, along with ROC curves (and their AUC values) for predicting a negative ΔΔG of less than −0.5 (see Table 3.4). The concordance correlation is the proper statistic for assessing agreement among continuous measurements, though the Pearson correlation is more common in the literature. The Spearman correlation is a rank-based correlation analogue of Pearson that is less reliant on linear assumptions. We used a nonparametric bootstrap (500 replications) to obtain estimates of standard errors and bias-corrected 95% confidence intervals (Cis) for estimates. We used scatter plots with nonparametric trend lines to examine the data. Bland−Altman plots were used to visually examine the agreement between predictions and actual values.

As a control for our processing, we also computed correlation coefficients using previous Rosetta ΔΔG prediction results from a large data set containing almost exclusively soluble proteins (Kellogg et al., 2011).

Table 3.4. Summary of statistical methods used to evaluate predictive methods

Method	Description
Concordance CC	The concordance correlation coefficient measures the degree to which the predicted ΔΔG value equals the actual experimental value (0 indicates no agreement and 1 perfect agreement).
Pearson CC	The Pearson correlation coefficient measures the degree to which a uniform linear transformation of the predicted ΔΔG values (i.e., a shift and scale change) would yield the actual experimental values (0 indicates no agreement after transformation, 1 perfect agreement, and −1 perfect inverse agreement).
Spearman rank CC	The Spearman rank correlation coefficient measures the degree to which the rank ordering of the predicted ΔΔG values matches the rank ordering of the actual experimental values (0 indicates no agreement after transformation, 1 perfect agreement, and −1 perfect inverse agreement).
ROC and AUC	The area-under-the-receiver operating characteristic (ROC) curve tests several cutoff values for binning mutations as neutral or destabilizing between the most negative calculated ΔΔG value and the most positive calculated ΔΔG value, with true positive rates (sensitivity) calculated at each point. As the true positive rate is calculated, the classifier is moved to less extreme values; this yields the ROC curve. The AUC curve is a summary statistic that approximates how well the predictor actually discriminates between the two classifications.

We collected all available experimental ΔΔG data sets for structurally diverse membrane proteins of known structure (which constitutes the vast majority of all ΔΔG measurements made to date for membrane proteins). We acknowledge differences in the cellular folding landscapes of α-helical and β-barrel proteins; however, given the limited number of membrane proteins with known structure and thermodynamic stability measurements, we combined all proteins for analysis and subsequently parsed potentially relevant subsets to evaluate the effect of each. As of early 2016, there were 223 single-amino acid ΔΔG destabilization measurements available for these proteins, with mutated side chains in the following categories: water-exposed, 6% (14); lipid hydrocarbon-exposed, 25% (55); exposed interfacial, 18% (41); or protein-buried, 52% (117). The distribution of experimental ΔΔG values is consistent with a random sampling of residue point mutation stabilities (Figure 3.1): 65% of point mutations resulted in ΔΔG values of less than −0.5 kcal/mol, considered destabilizing; 24% between −0.5 and 0.5 kcal/mol, considered neutral; and 11% greater than 0.5 kcal/mol, considered stabilizing, as suggested previously (Y Zhou & Bowie, 2000). All programs except Rosetta, PROVEAN, SDM, and FoldX have a narrow, slightly negative distribution of predicted ΔΔG values (Figures 3.1 and 3.3). The PSI-BLAST PSSM scores were also more dispersed than results for the majority of the programs tested. Interestingly, SDM tended to classify nearly as many mutations as stabilizing as destabilizing, which perhaps is a consequence of restricting mutant classification to neutral or destabilizing only if |ΔΔG| > 2 kcal/mol. Most methods tended to underestimate ΔΔG for destabilizing mutations and overestimate ΔΔG for neutral to stabilizing mutations.

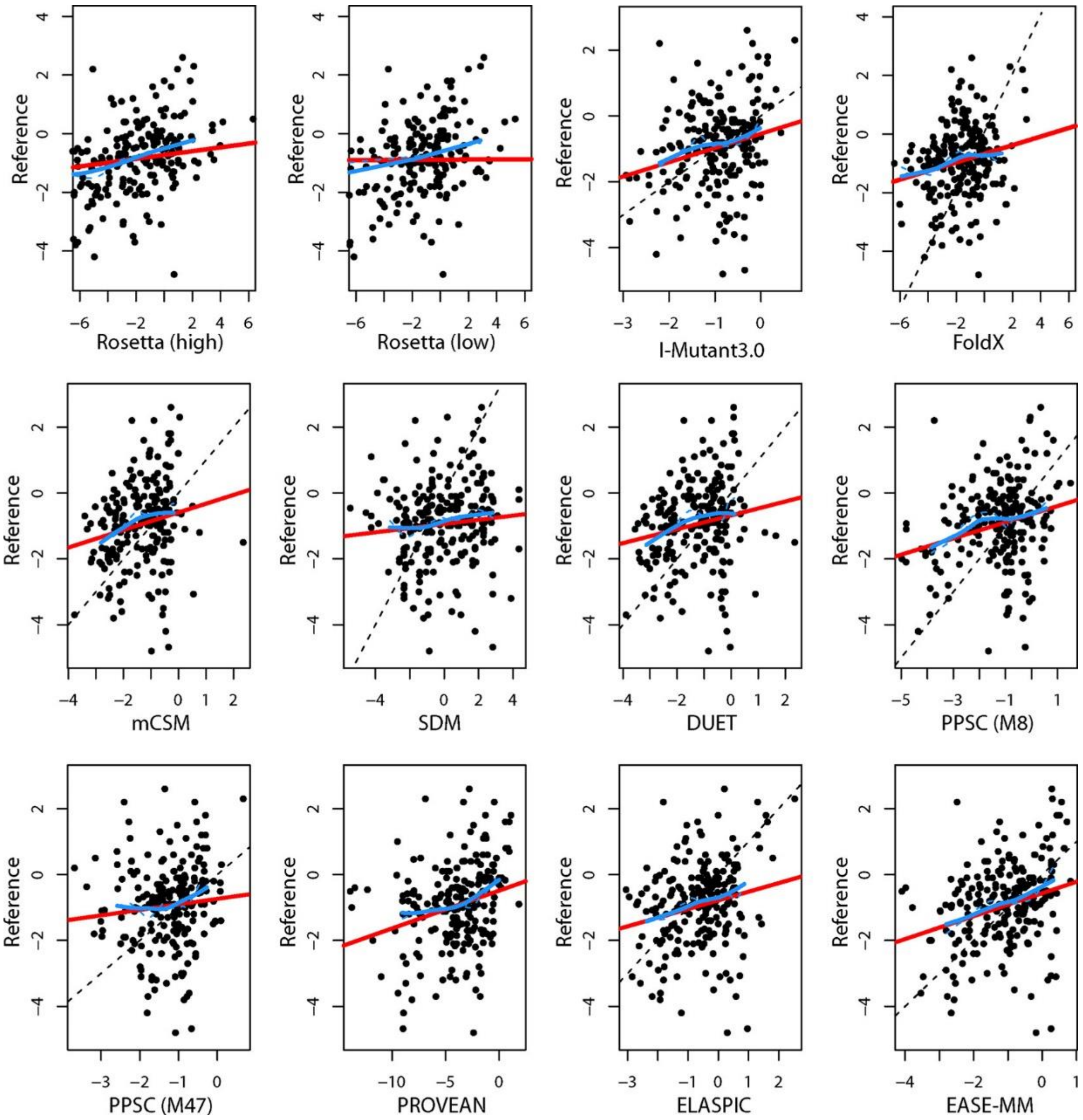


Figure 3.3. Reference (experimental) ΔΔG values vs calculated ddG values (x-axis) from each method tested (see also Table AE.1 in Appendix E). Red lines are simple linear regressions from which Pearson correlations are derived; blue lines are flexible nonparametric trend lines. For the Rosetta and FoldX plots, a few predicted points were outliers that fall outside of the plotted window. The dashed line is the y = x line measuring perfect agreement between the predicted ΔΔG and the experimental values and is plotted for methods constructed to make direct predictions.

Existing methods predict ΔΔG values that are poorly correlated to experimental ΔΔG values

To evaluate the predictive ability of each method tested, we compared concordance, Pearson, and Spearman rank correlation coefficients (Figure 3.3A; a glossary for statistical parameters is provided in Table 3.4). Note that we distinguish methods that were calibrated to predict ΔΔG values from methods that compute metrics that are expected to linearly correlate with ΔΔG values, such as ROSETTA. This distinction is important, as for optimal performance in the former group we expect a regression line that passes through the coordinate origin and has a slope of 1. In such a case, concordance, Pearson, and Spearman correlation coefficients would be equal to 1. In the latter group, for optimal performance, Pearson and Spearman correlation coefficients, but not the concordance, would be equal to 1. None of the programs tested performed well in calculating ΔΔG values for membrane proteins compared to their performance in previous studies of soluble protein data sets (Figure 3.4A). The concordance correlation coefficients for the various methods are all relatively low, the highest being ~0.2 [EASE-MM, FoldX, and PPSC (M8)]. This is compared to a concordance correlation coefficient in the range of 0.6 for the Rosetta-based method applied to an almost exclusively water-soluble protein data set. The performance of the different methods at predicting the rank order is improved compared to their ability to predict absolute ΔΔG values (Figure 3.4A), but all Spearman correlation coefficients are below 0.4, compared to 0.7 for the Rosetta-based method applied to a largely water-soluble protein data set. This means the majority of predicted rankings are still incorrect. Rosetta (high and low) and PROVEAN have the highest Spearman rank order correlation coefficients overall (0.37, 0.32, and 0.29, respectively) but still significantly underperform compared to results for soluble proteins. The general failure of these methods to reliably rank order the impact of membrane protein point mutations on stability is disappointing, as one of the anticipated applications for these methods is to aid researchers in identifying the most or least destabilizing mutations out of a hypothetical set, which then would be experimentally tested for the purpose of protein engineering.

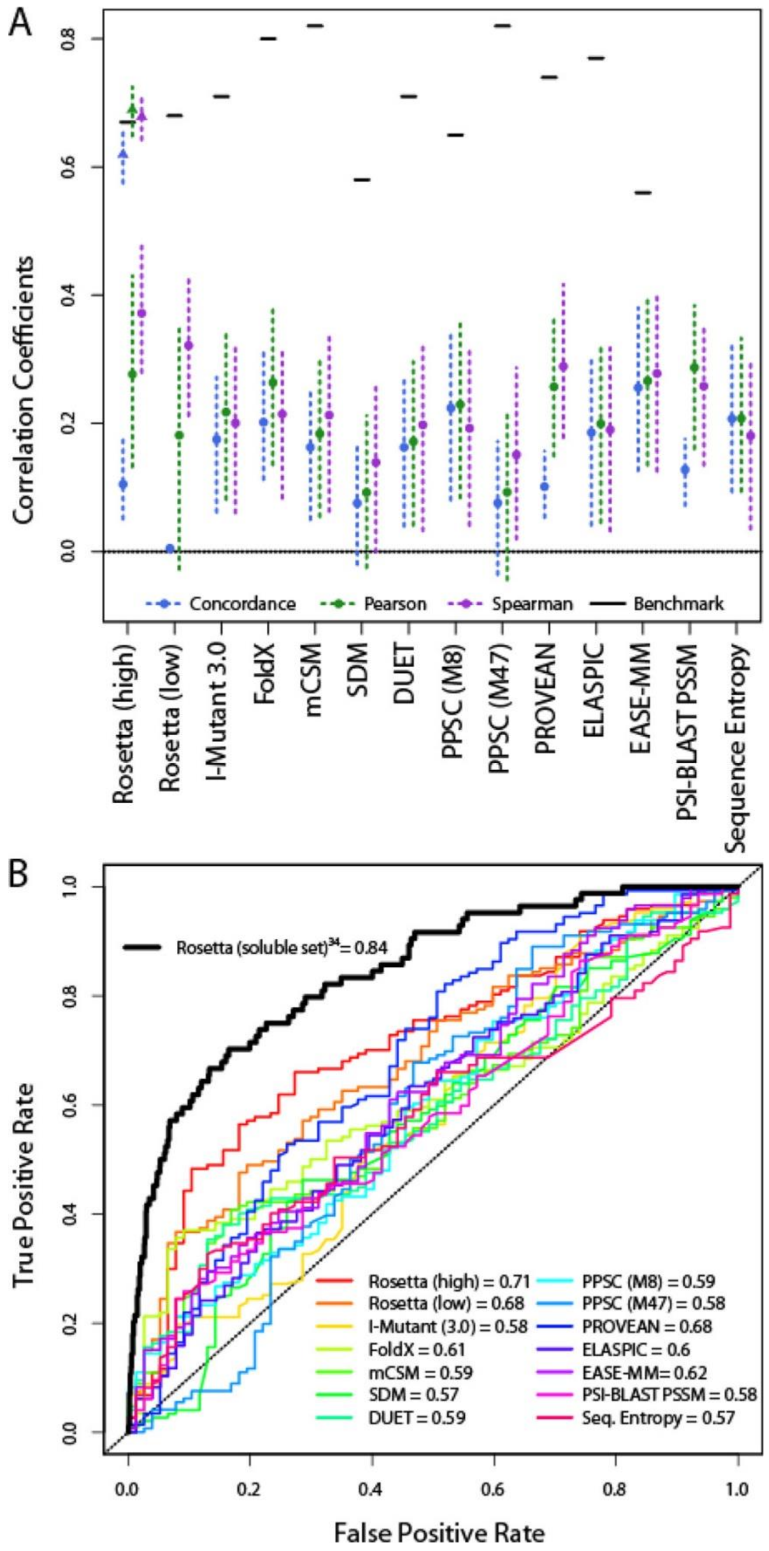


Figure 3.4. (A) Performance of each evaluated method in predicting true ΔΔG values (concordance correlation coefficient), linearly correlated ddG values (Pearson correlation coefficient), and rank order (Spearman rank order correlation coefficient). The hash marks in the upper portions of this plot indicate the published results for each method. We also evaluated the

concordance, Pearson, and Spearman correlation coefficients using the calculated and experimental data previously reported37 for a mostly water-soluble protein data set to control for processing differences, shown as triangles. (B) Receiver operating characteristic curves of the classification of variants that are more destabilized or less destabilized than 0.5 kcal/mol. We generated the black bold trace using data from a previous ΔΔG calculation effort37 involving mostly soluble proteins.

Stability classification of predicted values

Another application that can be envisioned is predicting the stability class for a given variant. For example, one might seek to identify mutants that have a ΔΔG value above or below −0.5 kcal (−0.5 is the typical uncertainty in experimentally determined stabilities (Khatun, Khare, & Dokholyan, 2004)). To compare the discriminating power of these methods, we plotted receiver operating characteristic curves [ROC (Figure 3.4B)], which show the ability to correctly classify point mutations as destabilizing (ΔΔG < −0.5) or neutral/stabilizing (ΔΔG > −0.5). ROC curves that are skewed toward a higher true positive rate (sensitivity) classify mutations more accurately, as quantified by AUC (ranging between 1.0 and 0.5 for perfect and chance classification, respectively). Rosetta and PROVEAN had the largest areas under the curve (95% Cis of 0.65−0.79 and 0.61−0.76, respectively). This is surprising because neither method was constructed or calibrated to predict ΔΔG values but is consistent with their better Spearman correlation performance. PROVEAN is designed to estimate the probability that a variant will be functionally compromised without accounting for structure, while Rosetta is optimized to incorporate protein structural features. The AUC of ∼0.8 for the soluble protein set calculated here, similar to previously reported values for these methods, further emphasizes the conclusion that the unique properties of membrane proteins require separate treatments in constructing stability prediction methods. A priori, there are several potential explanations for the observed disparity in calculating ΔΔG values for soluble versus membrane proteins. One confounding factor could be the persistence of α-helical structure in the unfolded states of helical membrane proteins, which is typically not the case for unfolded states of soluble proteins. In an effort to test this hypothesis, we separately evaluated β-barrels, expected to have no persistent secondary structure in the unfolded state, and α-helical membrane proteins. The correlation coefficients for the β-barrel protein set have considerably larger 95% confidence intervals but suggest that several programs perform somewhat better for β-barrel proteins (Spearman correlation coefficient of 0.29) than for α-helical membrane proteins (average Spearman correlation coefficient of 0.22) (Figures 3.5 and 3.6), although the poor performance for both groups of proteins proves no method is reliable at this task. Interestingly, differences in correlation and ranking ability were not uniform between the methods evaluated: FoldX performed better on α-helical proteins (second-highest Spearman correlation coefficient) than on β-barrels (lowest Spearman correlation coefficient), with estimated Spearman correlations of 0.35 and 0.01, respectively. We also evaluated the effect of parsing out the secondary structure-disrupting residues, glycine and proline.

Further analysis of the data by parsing out classes

Surprisingly, even removing proline and glycine residues did not improve Spearman correlation coefficients appreciably; 95% confidence intervals narrowed, and estimated values increased from 0.23 to 0.29 (Figure 3.4A and Figure 3.7). Another potential cause of the disparity between soluble and membrane proteins may be the unique solvent environment of the membrane. We parsed ΔΔG values based on residue position: water-exposed (Figure 3.9), at the membrane interface (Figure 3.10), membrane-exposed (Figure 3.11), solvent-facing (Figure 3.12), or buried in the protein (Figure 3.13). Given the small number of water-exposed variants assessed, the 95% confidence interval is extremely wide, precluding any real assessment. In any case, no parsing of residue position yielded significant improvements in Spearman correlations.

Indeed, to our surprise, all methods tended toward worse predictive ranking for protein-buried residues (average Spearman correlation coefficient of 0.19) than for solvent-exposed residues (Spearman correlation coefficient of 0.25). Finally, it should be acknowledged that the methods used for experimentally measuring membrane protein ddG values are not yet highly standardized, reflecting use of denaturants as different as sodium dodecyl sulfate and urea, as well as model membranes as different as micelles and bilayer vesicles. The degree to which the stability of a single membrane protein is similar when measured using different methods has yet to be extensively tested. An open question is whether more computationally intensive strategies, such as molecular dynamics-based approaches, will improve predictive power for membrane proteins. We did not investigate this kind of approach here because of the limiting throughput that can be achieved at present.

In this study, a series of diverse statistical criteria are in uniform agreement that current methods for predicting ΔΔG values of point mutations in membrane proteins will need to be improved or superseded to be reliable and useful. According to our evaluation, the predictive ability of the 10 methods assessed was not greatly improved from that of the PSI-BLAST PSSM and sequence entropy scores, i.e., what one could infer on the basis of mutated site evolutionary sequence conservation. We did not find any method to be robust at predicting either the rank order of mutations or absolute ΔΔG values. This study highlights the need to separately evaluate the performance of ΔΔG calculators on membrane proteins in the future, as well as the need for a much larger training database of experimentally measured stabilities for wild-type and mutant membrane proteins.

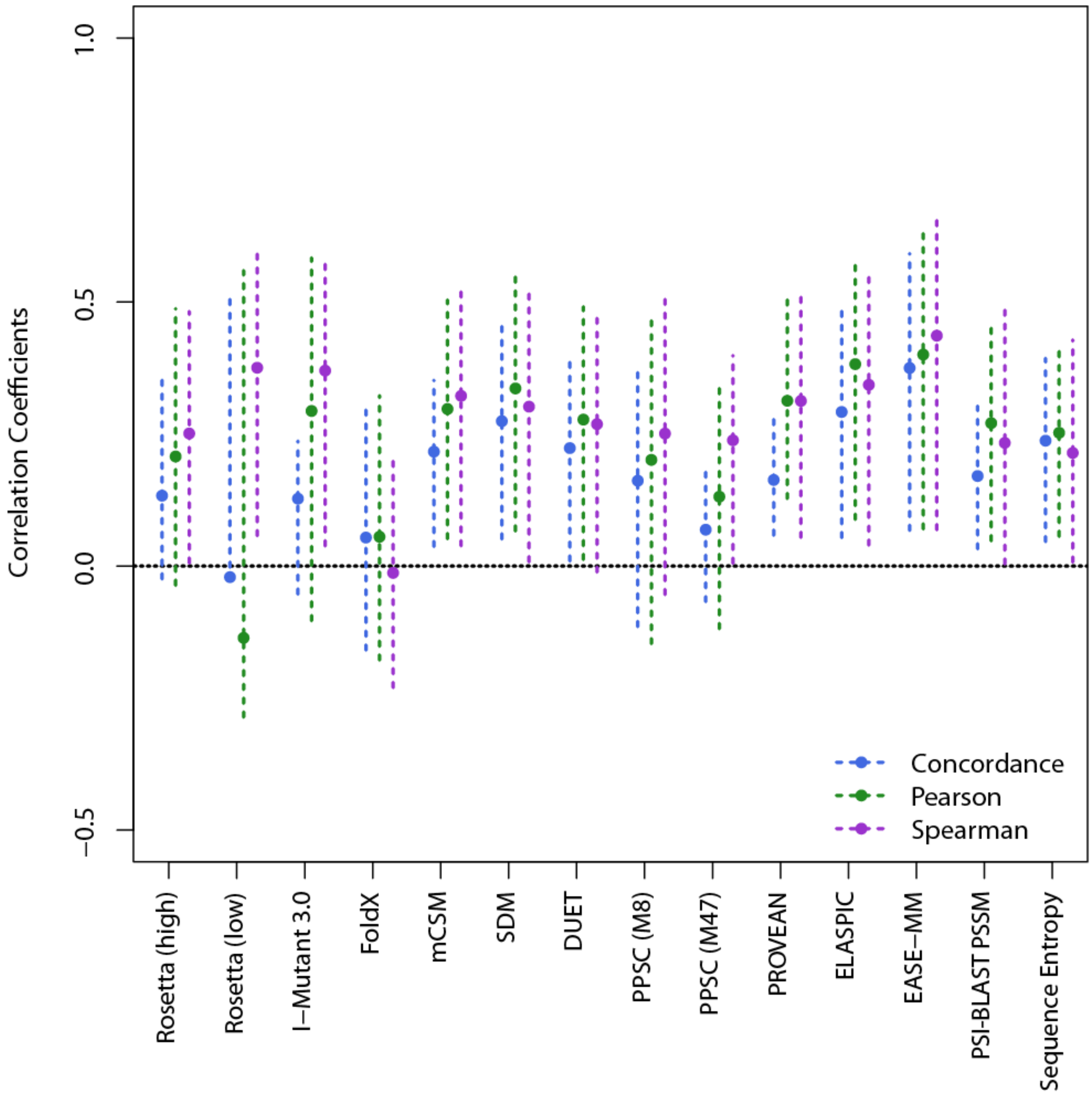


Figure 3.5. Comparison of Concordance, Pearson, and Spearman correlation coefficients for βbarrel proteins. Dashes represent a bias-corrected 95% confidence interval.

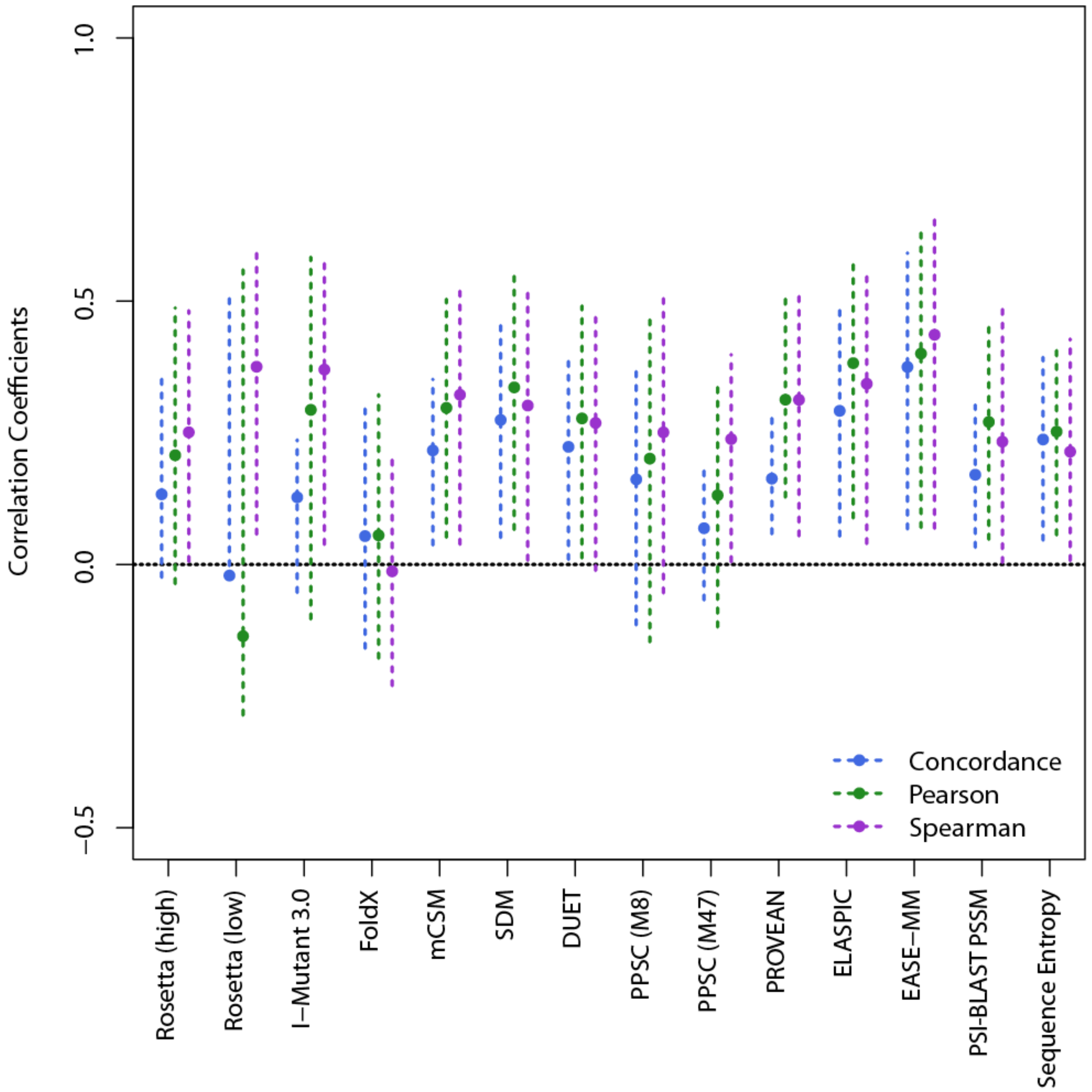


Figure 3.6. Comparison of Concordance, Pearson, and Spearman correlation coefficients for αhelical proteins. Dashes represent a bias-corrected 95% confidence interval.

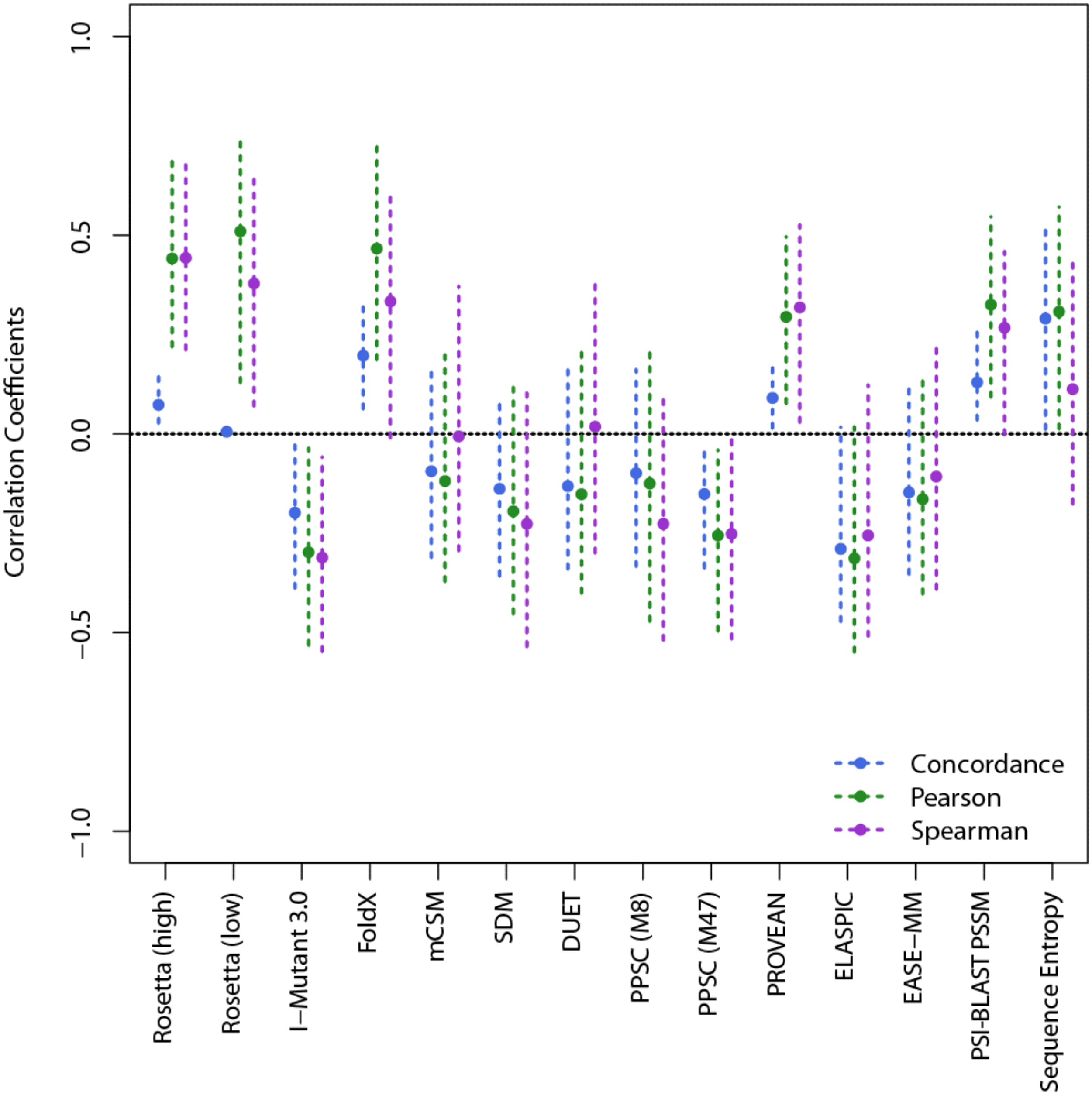


Figure 3.7. Comparison of Concordance, Pearson, and Spearman correlation coefficients of point mutations that involve a proline or glycine. Dashes represent a bias-corrected 95% confidence interval.

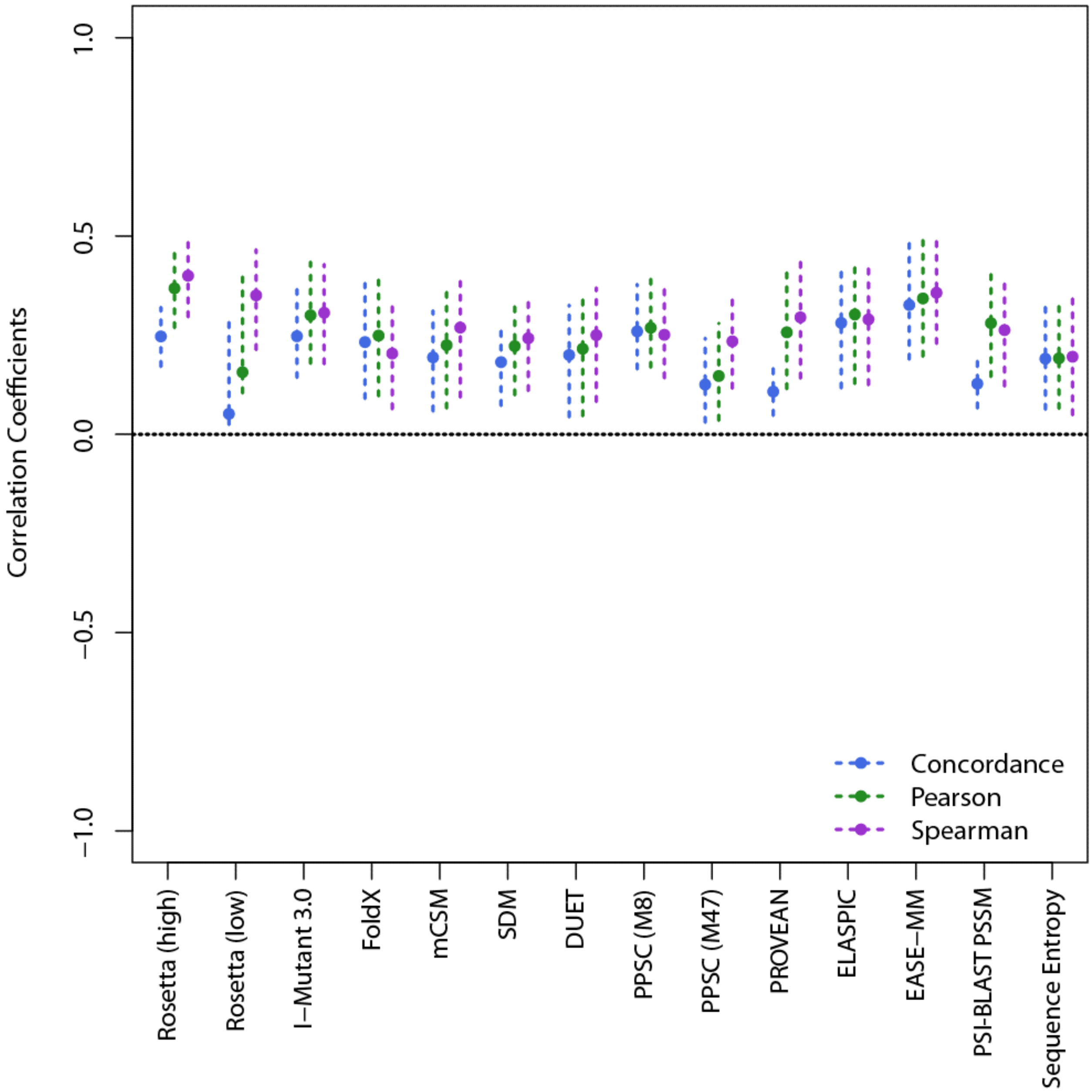


Figure 3.8. Comparison of Concordance, Pearson, and Spearman correlation coefficients of point mutations that do not involve a proline or glycine. Dashes represent a bias-corrected 95% confidence interval.

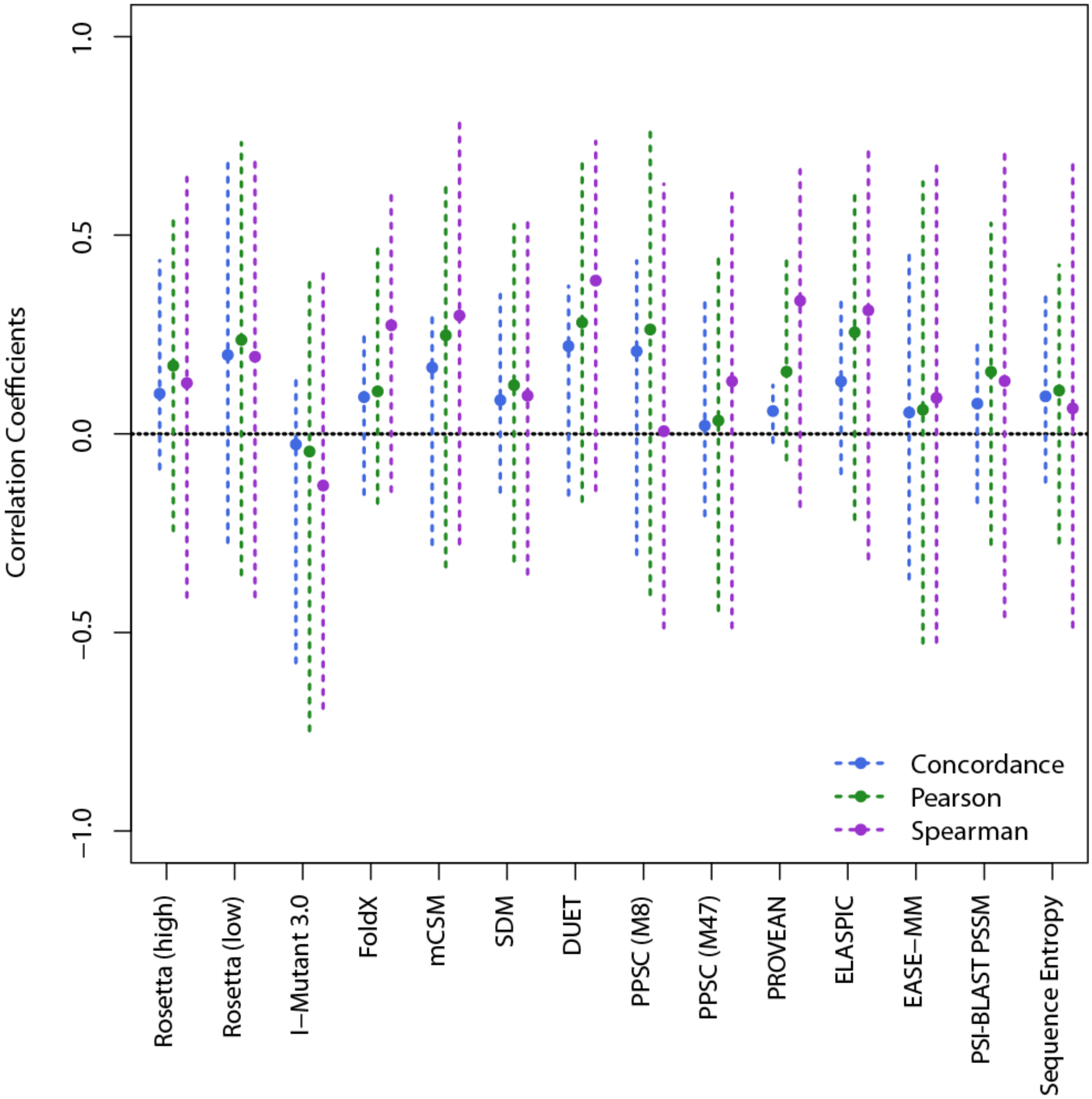


Figure 3.9. Comparison of Concordance, Pearson, and Spearman correlation coefficients for residues in the aqueous phase. Dashes represent a bias-corrected 95% confidence interval.

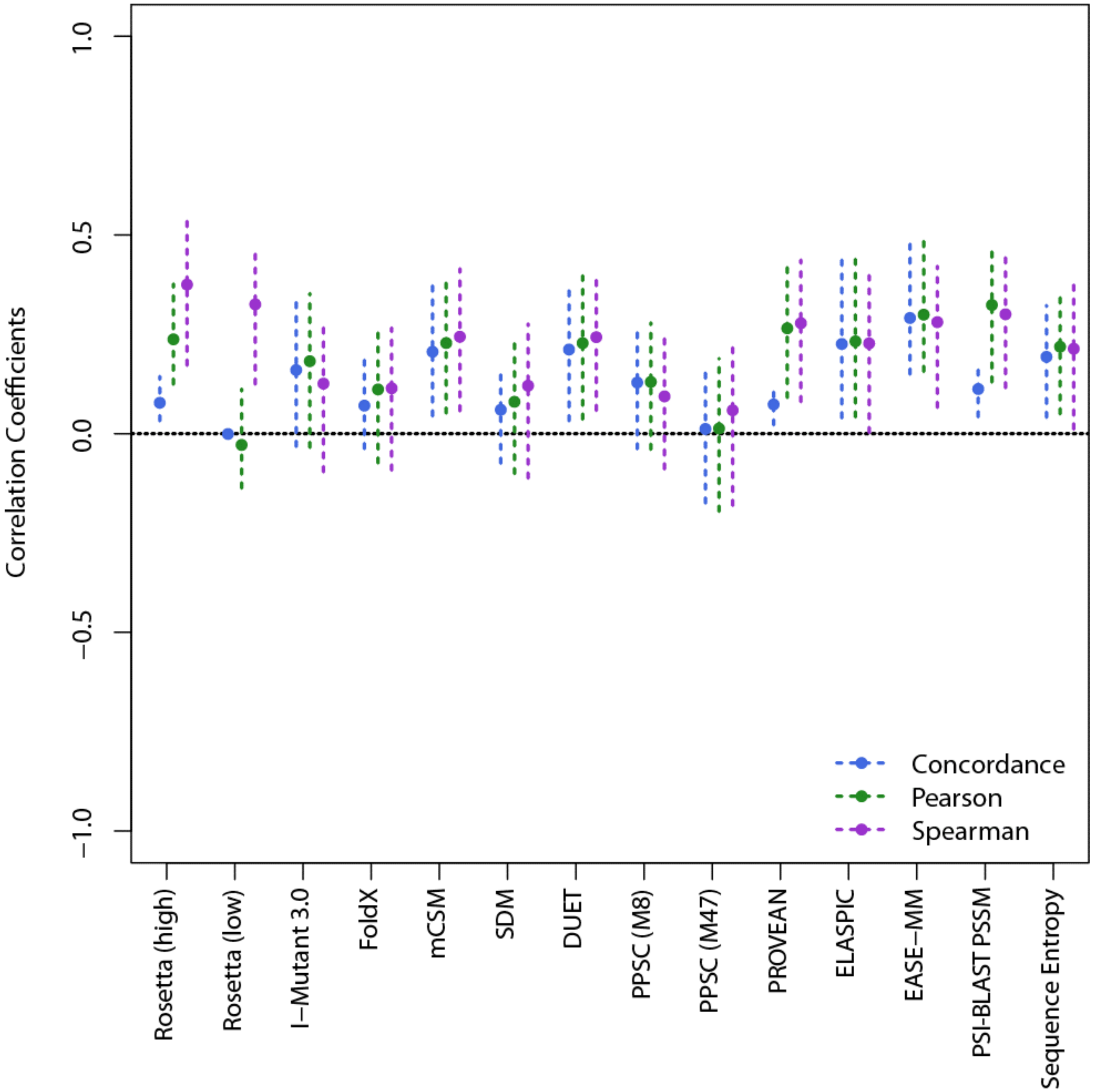


Figure 3.10. Comparison of Concordance, Pearson, and Spearman correlation coefficients for residues at the interface between membrane and aqueous phases. Dashes represent a bias-corrected 95% confidence interval.

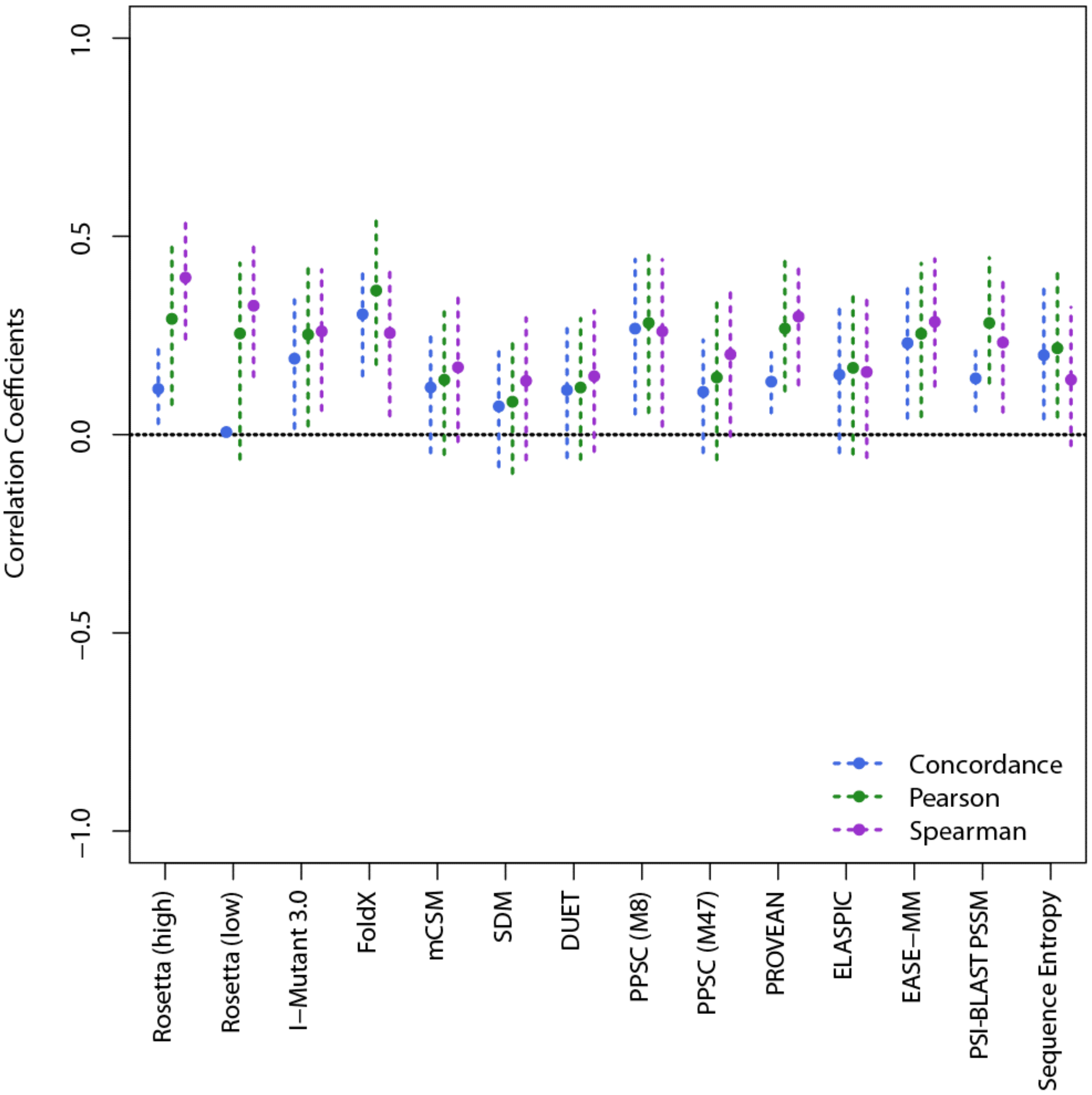


Figure 3.11. Comparison of Concordance, Pearson, and Spearman correlation coefficients residues in the aliphatic phase of the membrane. Dashes represent a bias-corrected 95% confidence interval.

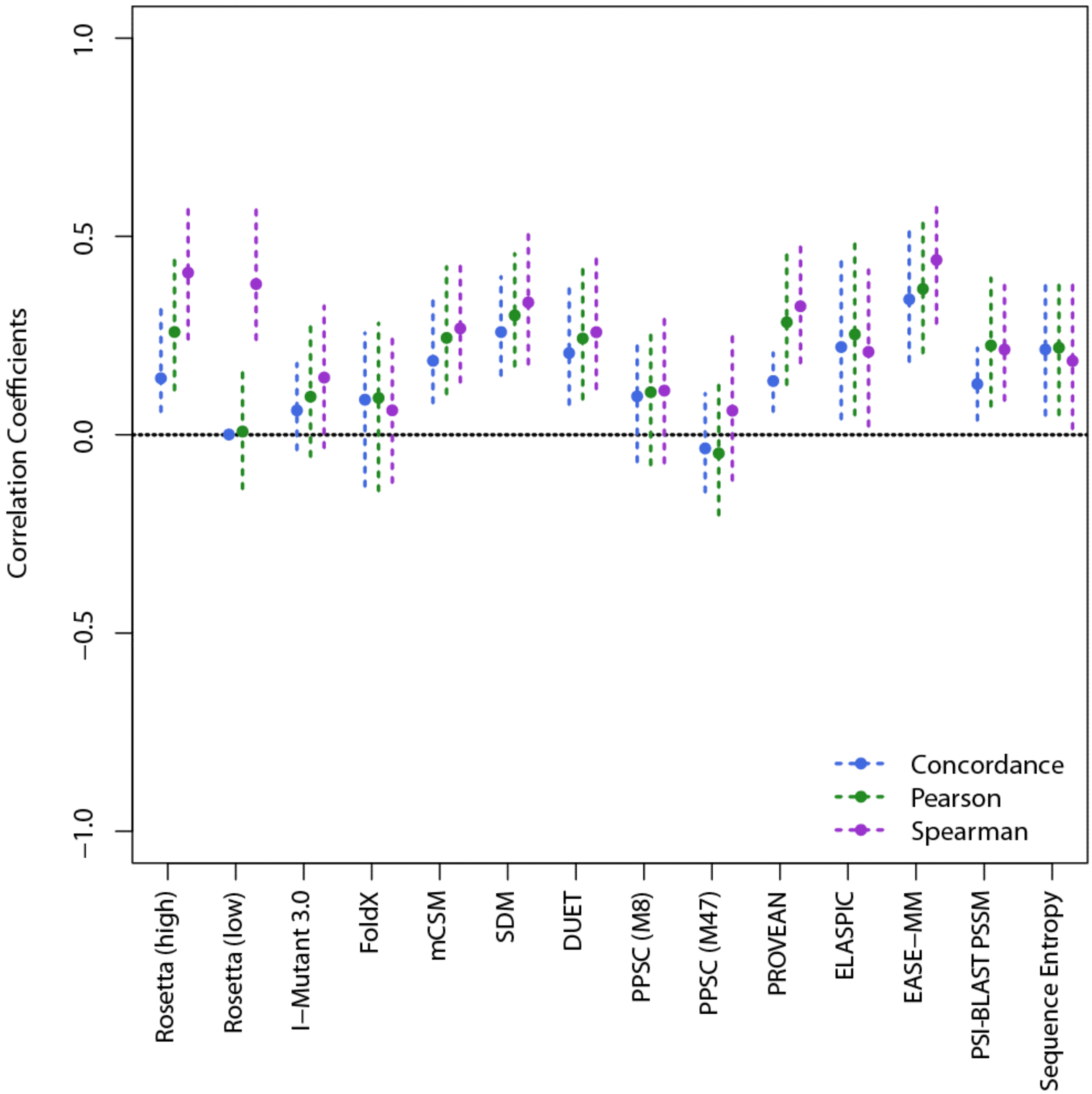


Figure 3.12. Comparison of Concordance, Pearson, and Spearman correlation coefficients for solvent exposed residues. Dashes represent a bias-corrected 95% confidence interval.

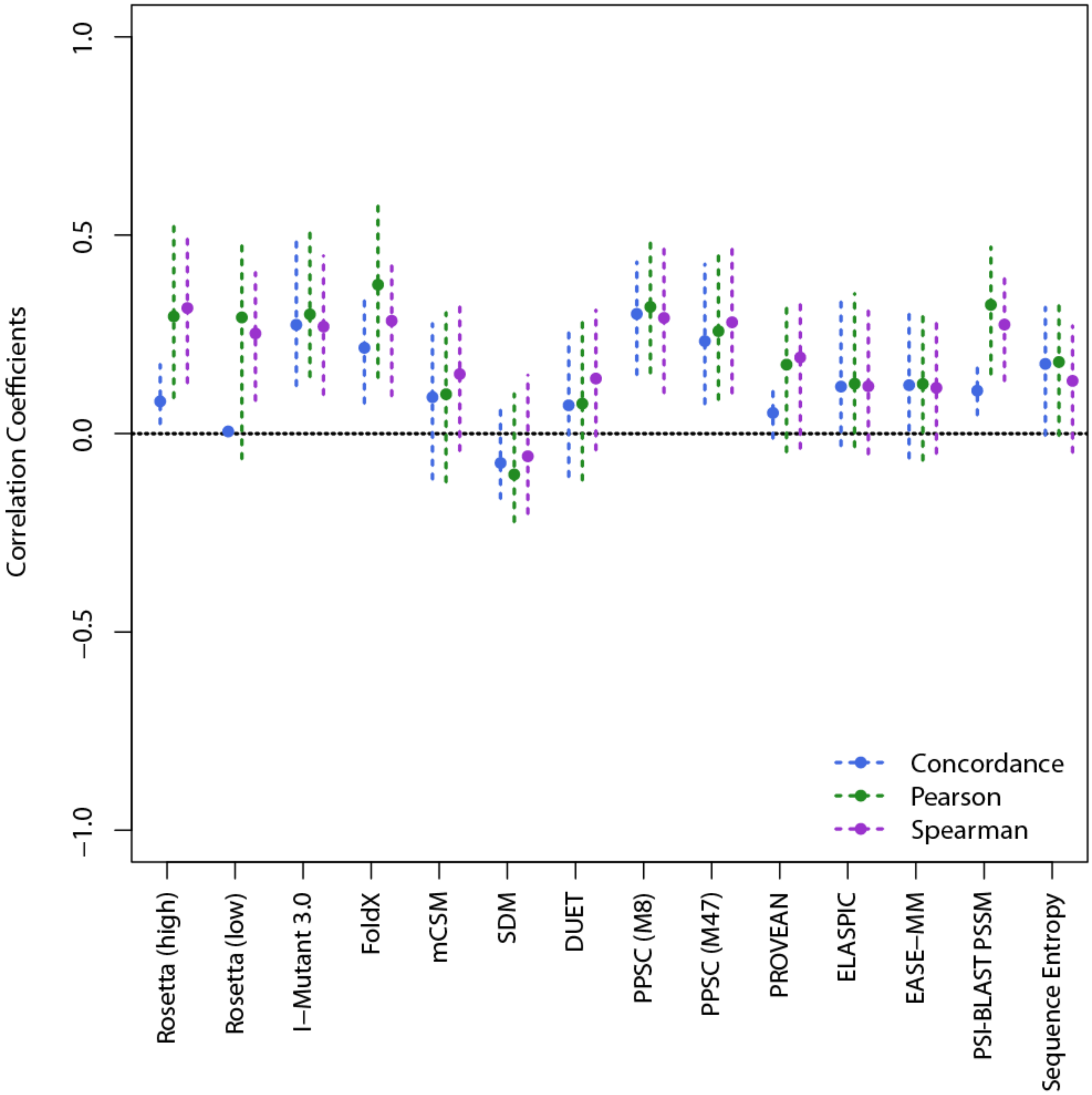


Figure 3.13. Comparison of Concordance, Pearson, and Spearman correlation coefficients for buried residues. Dashes represent a bias-corrected 95% confidence interval.

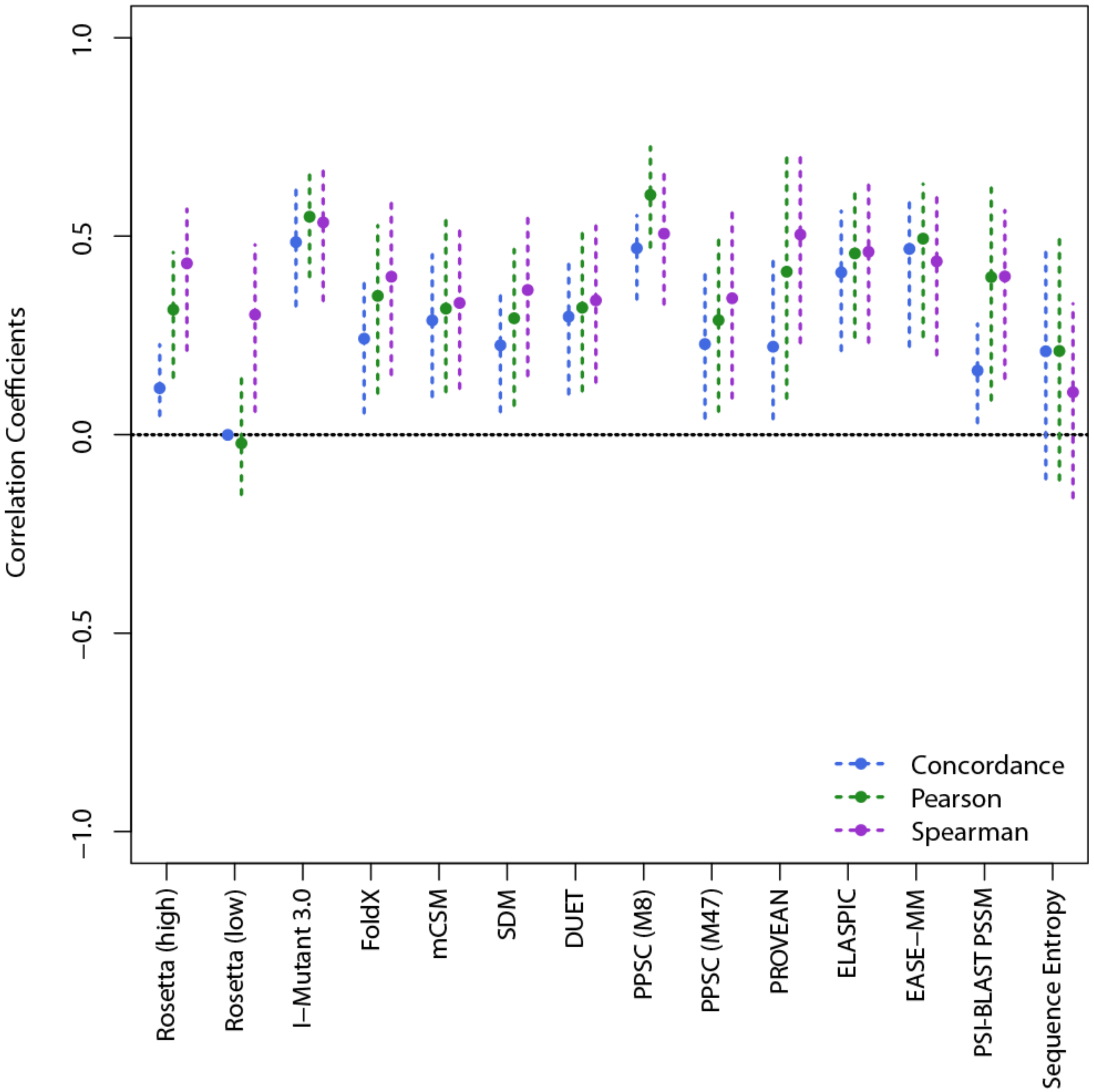


Figure 3.14. Comparison of Concordance, Pearson, and Spearman correlation coefficients for the bacterial proton pump, bacteriorhodopsin. Dashes represent a bias-corrected 95% confidence interval.

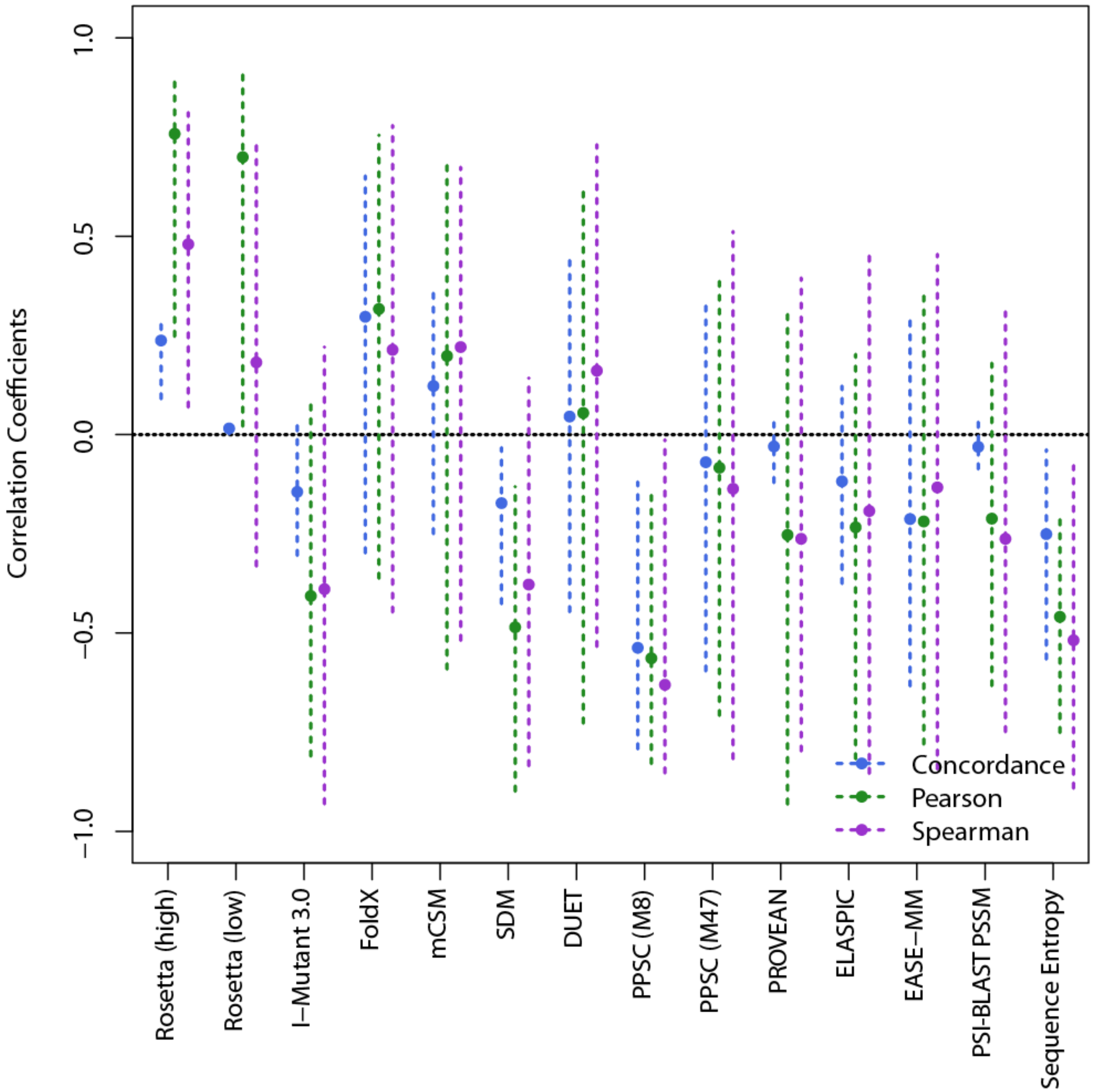


Figure 3.15. Comparison of Concordance, Pearson, and Spearman correlation coefficients for glycophorin A. Dashes represent a bias-corrected 95% confidence interval.

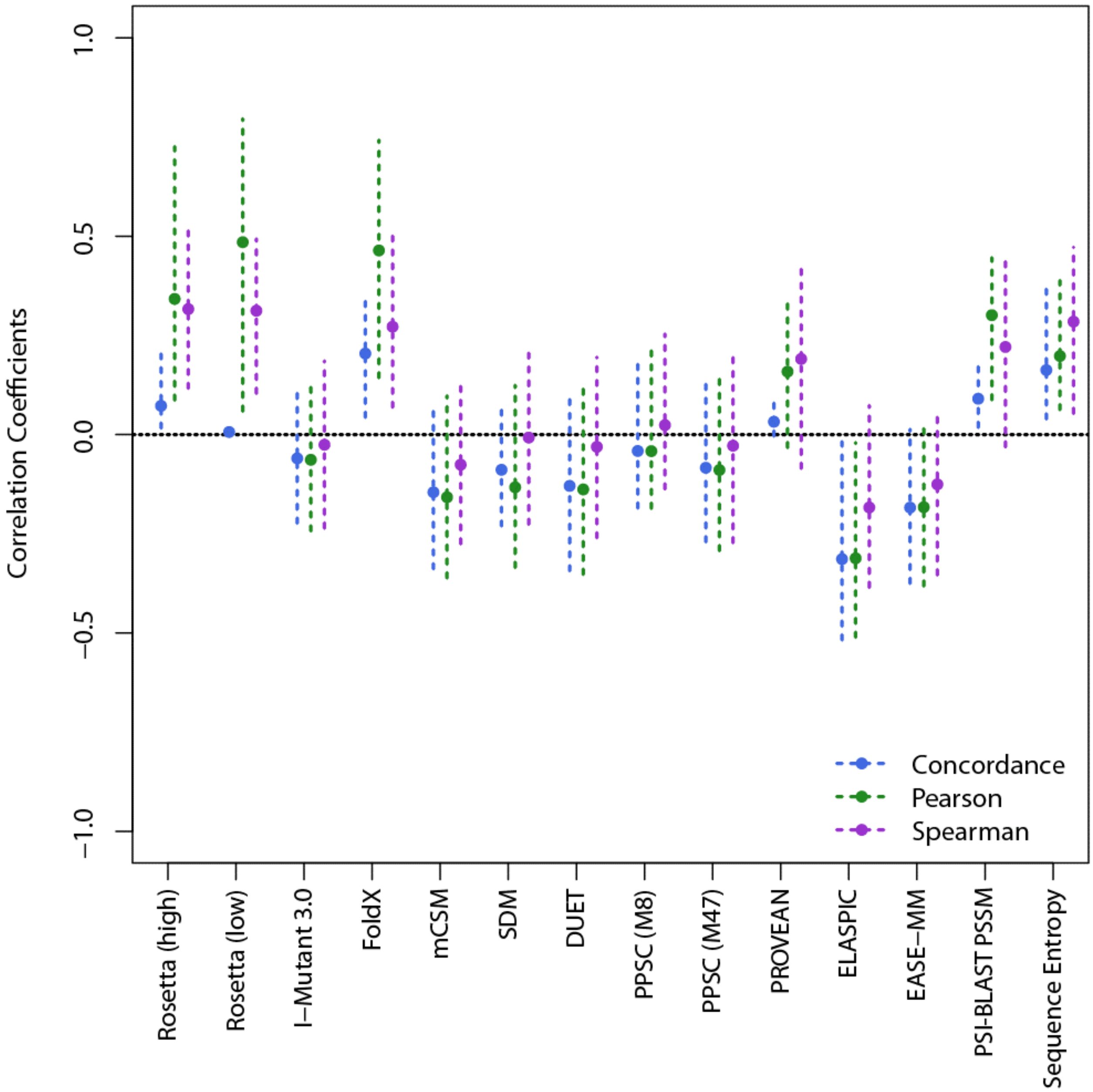


Figure 3.16. Comparison of Concordance, Pearson, and Spearman correlation coefficients for the E. coli rhomboid protease, GlpG. Dashes represent a bias-corrected 95% confidence interval.

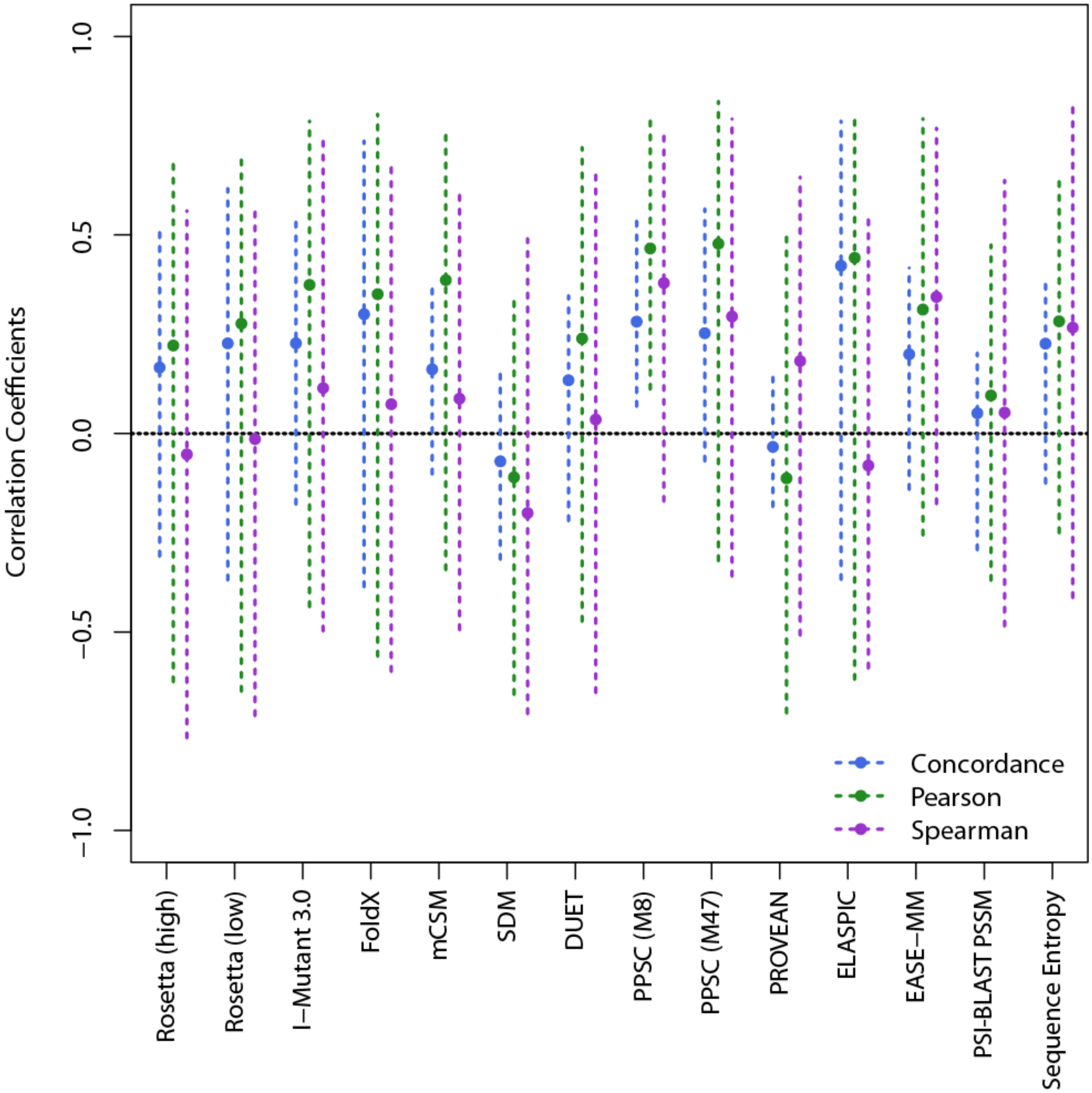


Figure 3.17. Comparison of Concordance, Pearson, and Spearman correlation coefficients for the disulfide formation protein B, DsbB. Dashes represent a bias-corrected 95% confidence interval.

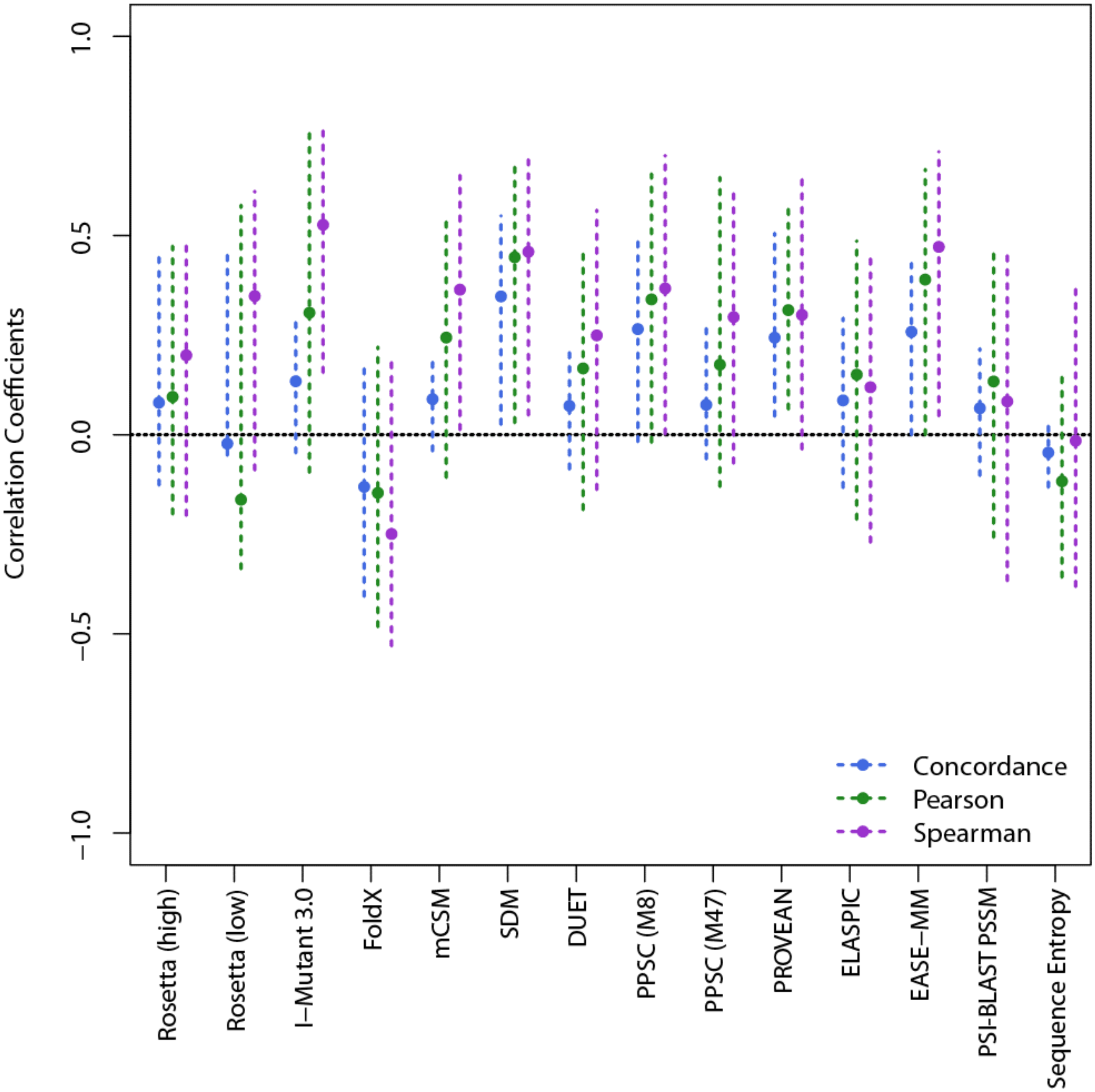


Figure 3.18. Comparison of Concordance, Pearson, and Spearman correlation coefficients for the outer membrane phospholipase A1, OmpLA. Dashes represent a bias-corrected 95% confidence interval.

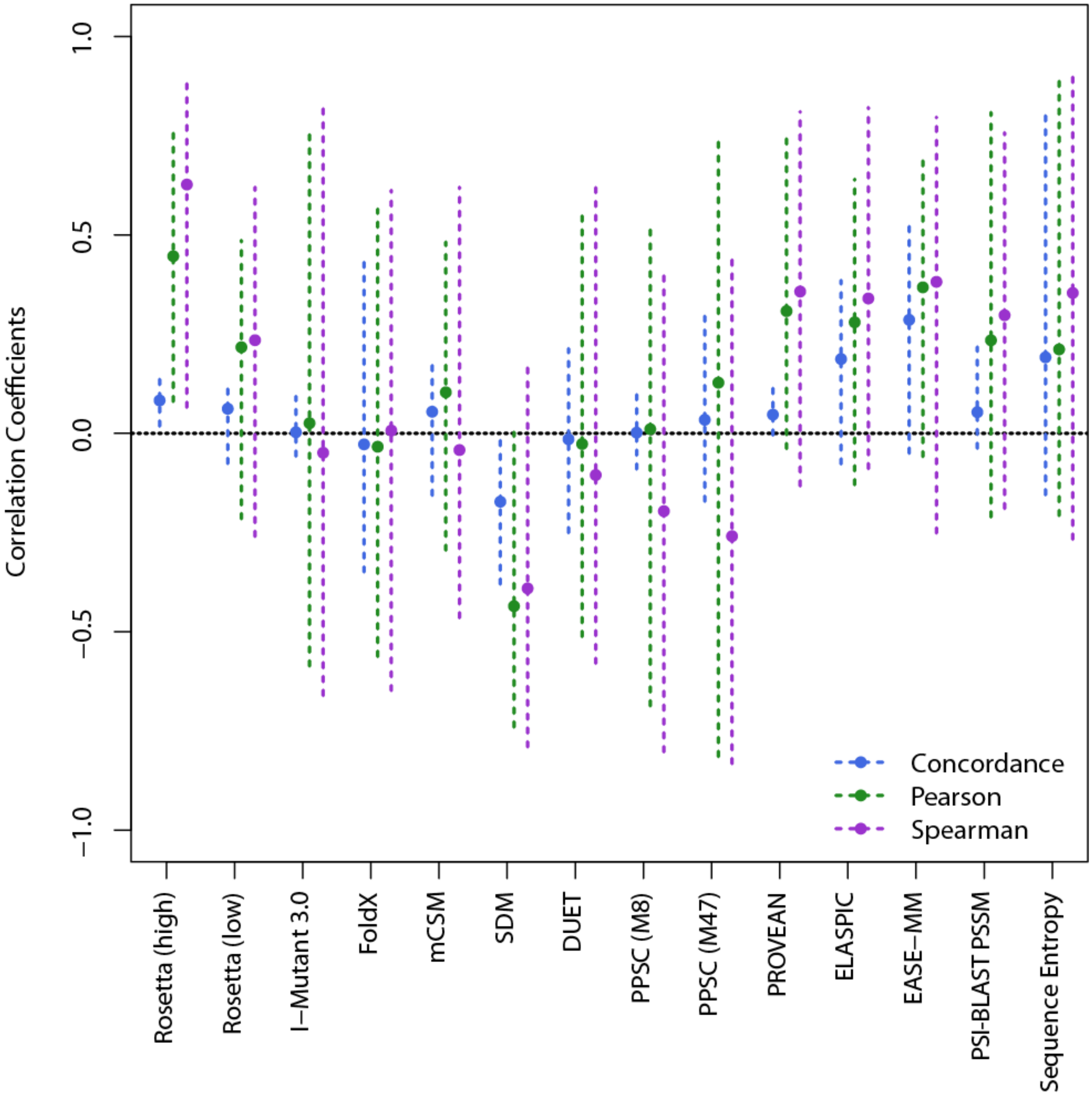


Figure 3.19. Comparison of Concordance, Pearson, and Spearman correlation coefficients for the outer membrane protein A, OmpA. Dashes represent a bias-corrected 95% confidence interval.

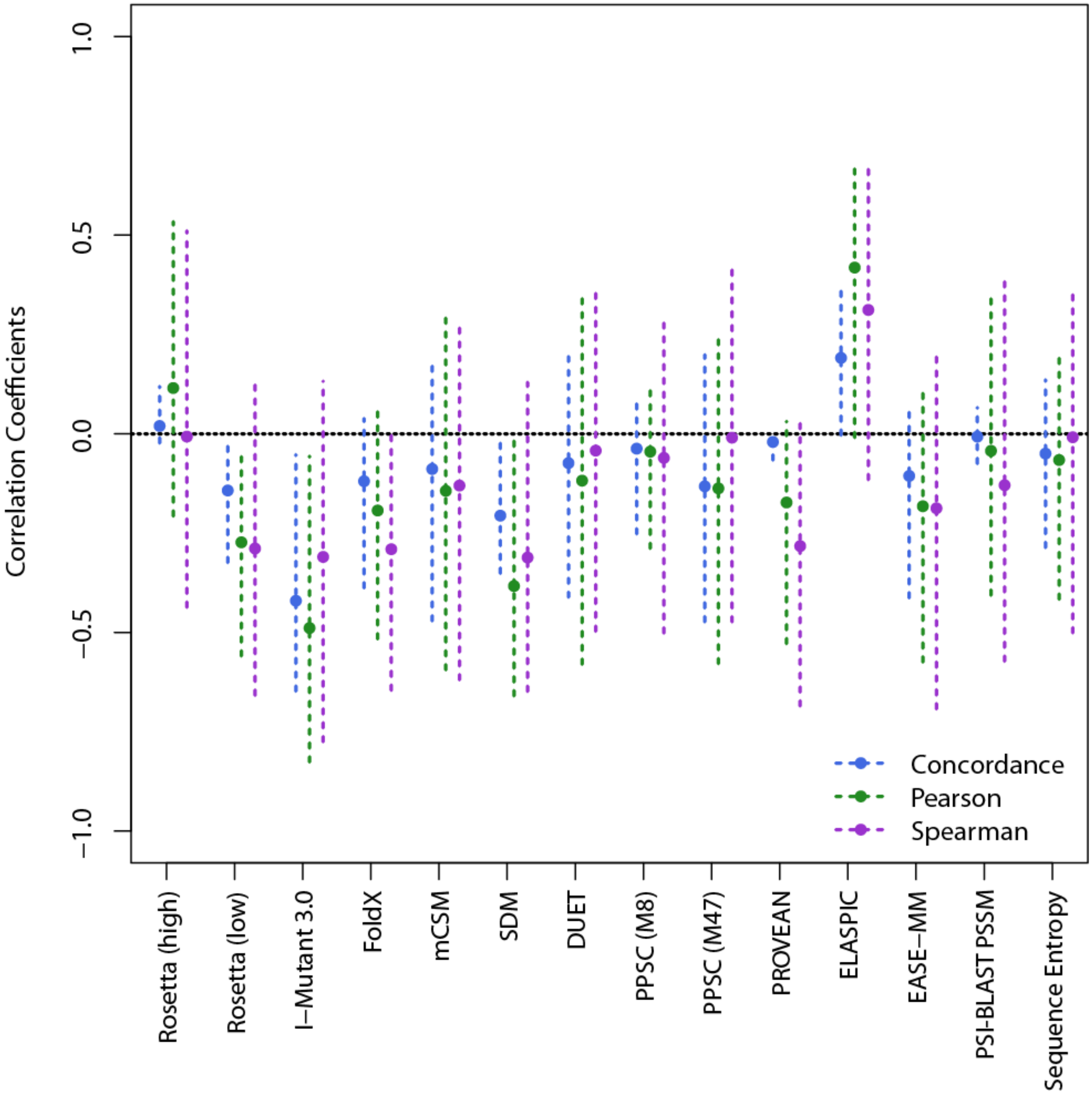


Figure 3.20. Comparison of Concordance, Pearson, and Spearman correlation coefficients for the lipid A palmitoyltransferase, PagP. Dashes represent a bias-corrected 95% confidence interval.

CHAPTER 4

AN IMPROVED ROSETTA ENERGY FUNCTION FOR PREDICTION OF MUTATION-INDUCED STABILITY CHANGES IN MEMBRANE PROTEINS

This chapter contains unpublished work from:

Duran and Meiler 2018

Author contribution: For this chapter, I designed experiments under the direction of Jens Meiler. I generated all data, analyses, and all figures and tables.

4.1 Introduction

Clinical genetic testing has become standard with the increase in information from human genetic analysis. Genome sequencing has generated extensive information for rare and common variants that predispose patients to diseases; however, many of genetic variants still have unknown clinical significance (VUS). Some genetic variants that are linked to diseases such as long-QT syndrome, Alzheimers, cancer, and cystic fibrosis (C R Sanders & Myers, 2004) are membrane proteins. Membrane proteins are a class of proteins that include channels and transporters, and they are often difficult to structurally characterize. Many times, single-nucleotide variants that predispose disease result in loss of function by means of protein destabilization. To experimentally screen the stability of VUSs, especially those that are membrane proteins, would be exhaustive for both time and resources.

Computational methods have the potential to accelerate the process to predict whether VUS have destabilizing effects. Current methods to predict the stabilizing effects of mutations include physical, statistical, and empirical approaches. Physical methods utilize atomic force fields to calculate interactions in the protein. Statistical methods use knowledge from protein structures, such as propensities of amino acids in a particular environment or ideal atom pair distances, to predict the energetic effect of interactions in the protein. Empirical methods make use of known experimental data that characterize interactions to create an energy function from machine learning methods. Additionally, methods have been created using unique combinations of these approaches.

Rosetta is a molecular modeling suite that enables rapid sampling of side-chains and computes energy by calculating score terms such as ones that describe van der Waals interactions, electrostatic interactions, rotamer probability, solvation and hydrogen bonding. Moreover, Rosetta has an energy function that is used to model membrane proteins, which makes it ideal for creating a structure-based approach to predict the stabilizing effect of membrane protein VUSs.

Previously, we reported the performance of multiple programs that predict the mutation-induced perturbation of folding stability (ΔΔG) for a set of membrane proteins (Kroncke et al., 2016). The existing programs had been trained using soluble protein thermostability data. The Pearson's R, Spearman rank R, and Concordance correlations between predicted and experimentally determined ΔΔGs were below 0.4 for all programs. The Rosetta method, ddg_monomer, had been optimized and benchmarked using soluble proteins. We tested the performance of ddg_monomer using the RosettaMembrane energy function and found that the correlations were no different than ddg_monomer using the standard soluble energy function. Herein we evaluate various regression analyses to approximate experimentally determined ΔΔGs from the energy contributions of RosettaMembrane score terms, and propose a new RosettaMembrane weight set for ΔΔG calculations using ddg_monomer.

4.2 Methods

Generating dataset for machine learning

The same set of 224 single point mutations were used to generate a dataset that consisted of score term contributions for the purposes of refitting a new membrane protein specific scoring function for the ddg_monomer application for high resolution proteins. For simplicity, the "soft-rep" energy function for the first stage of minimization was used because a soft repulsive energy function does not exist for membrane proteins. For the second stage, the RosettaMembrane high resolution energy function was used in place of the score12 energy function. The full dataset was run through ddg_monomer five times. The total Rosetta energy score for the mutant was subtracted from the total energy score for the wild-type protein model to obtain the ΔΔG. This calculation is consistent with the direction in the thermodynamic cycle (Figure 4.1), so that a negative score indicates destabilizing. For each run, the top three ΔΔG scores for each mutation were averaged and compiled into a dataset for a total of five scores representing each mutation. The individual score term scores were un-weighted in order to optimize the score term coefficients based on the raw energy contribution. For each mutation, the raw energy contributions of each score term were placed in a table next to their respective experimental ΔΔG values.

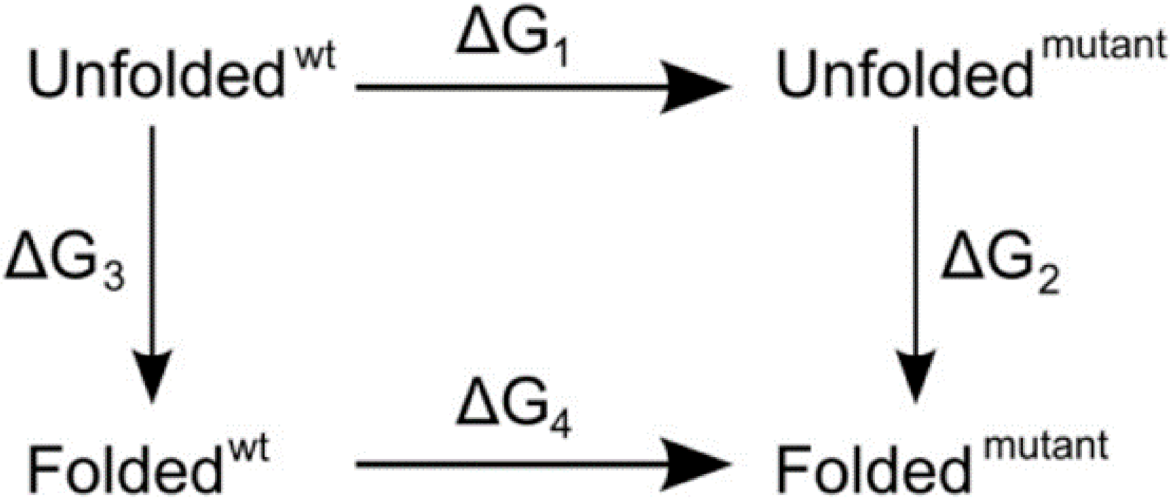


Figure 4.1 Thermodynamic cycle. This illustrates that ΔΔG = ΔG3 – ΔG2 = ΔG1 – ΔG4. Reprinted from, Biophysical Journal, Vol 98, Daniel Seeliger and Bert L. de Groot, Protein Thermostability Calculations Using Alchemical Free Energy Simulations, 2310, 2010, with permission from Elsevier.

Refitting the score term weights

Several machine learning approaches were used to refit the weights of the score terms for an optimized RosettaMembrane high resolution energy function for predicting ΔΔG of single-point mutations. The statistical analysis program, R (3.2.5), was used to perform various machine learning methods in order to approximate the experimental ΔΔG values with a linear combination of the given terms of determined weights. These machine learning methods included: multiple linear regression (MLR), non-negative least squares (NNLS), ridge regression, lasso regression, and elastic net regression. MLR resulted in negative coefficients for some of the score terms, which was problematic as Rosetta does not support having negative weights for score terms. Therefore, I continued the analysis with NNLS. While this approach forced the coefficients to be positive, the residue sum of squares was large indicating that the fit of the line was poor. Additionally, the performance of ddg_monomer while using the resulting weights was inconsistent and poor (Table 4.1). I continued the regression analysis with ridge, lasso, and elastic net regressions.

4.3 Results and discussion

Table 4.1 Summary of performance of weights derived from various regression analysis. These values represent the metrics that resulted from using the respective weights in the ddg_monomer application.

	Regression analysis used to determine weight set			
Performance Metric	RosettaMembrane	Ridge	Lasso	Elastic Net
Pearson R	0.31	0.46	0.5	0.45
Spearman R	0.34	0.49	0.5	0.43
AUC	0.68	0.75	0.74	0.73

An approach using a combination of multiple linear regression and non-negative least squares

MLR was not used as a method alone because it produced negative score term weights. Instead, NNLS was used as an initial study to optimize the RosettaMembrane energy function for predicting ΔΔG values. First, I used the full dataset to perform NNLS using the package nnls and the function nnls. This resulted in a residual sum of squares of 2117. Several score terms were set to zero, and it resulted in a Pearson's R correlation of 0.51. Then, I removed the score terms that were given a coefficient of zero and performed NNLS again on the same dataset to ensure these score terms did not contribute to accurate predictions. The residual sum of squares and the Pearson's R correlation remained the same.

Next, I removed score terms that I suspected would contribute to noise. The dataset contains very few mutations that affect interactions involved in disulfide bonds, therefore, the information available to train this particular interaction is incredibly sparse. As a result, all score terms involving disulfide bonds were removed, despite knowing that this would limit the predictive power of mutations involving disulfide bonds. The residual sum of squares was 2118 while the Pearson's R correlation remained at 0.51. Finally, I tried to find the minimal combination of score terms that resulted in the same performance. I systematically removed the score terms that had very little weight at this point. I determined that score terms omega, fa_dun, and Menv_smooth could be removed and the resulting energy function performed with a Pearson's R correlation of 0.51; however, it resulted in a residual sum of squares of 2179 indicating it was less of a fit than the previous linear combination of terms.

Finally, I used cross-validation to determine the first final weight set. Rather than a traditional, randomized approach such as k-fold cross-validation, I created training and test sets based on the protein. This is to avoid biasing the weight set by testing the performance of the weight set on a protein backbone that the program has already used to train for predicted ΔΔG values. I provided each training set with the combination of score terms that excluded the reference and disulfide energies as well as the score terms that had a coefficient as zero. Because the size of the dataset varied for each training and test set, I computed a normalized residual sum of squares as the residual sum of squares per data point. The Pearson's R correlation coefficient ranged from -0.05 to 0.36 while the Concordance correlation coefficients ranged from -0.03 to 0.31. The area under the curve was also calculated for the prediction of stabilizing effect and this ranged from 0.53 to 0.74. Unfortunately, the variability in the resulting correlation coefficients indicated that the proposed energy function may be overfit for a particular protein of the dataset.

The next approach combining MLR and NNLS involved first a regression analysis that did not constrain the coefficients to only be positive. I prepared datasets that were separated by protein, and all original RosettaMembrane score terms were used for MLR. MLR was performed using the lm function. I then examined the sign of the coefficients for each score term. If the various datasets resulted in both negative and positive coefficients, I assumed this to mean the score term introduced noise to the energy function, and removed it. I then performed MLR again on the datasets this time with the reduced score terms. This resulted in Pearson's R correlations in the range of 0.49 to 0.57 and AUCs in the range of 0.71 to 0.75. This was a much more narrow range compared to the previous approach. Because Rosetta does not handle score terms with negative weights, I then ran an experiment using the same datasets where I removed the score terms with negative coefficients to determine how much they contribute to accurate predictions. This resulted in a Pearson's R of range 0.46 to 0.53, and AUCs in the range of 0.71 to 0.75. While the Pearson's R correlation suffered a minor decrease due to the removal of score terms with negative coefficients, the AUC remained the same. With the remaining score terms from the MLR analysis, I performed NNLS to compare the coefficients found from each analysis. Because score terms with little influence as well as score terms with negative coefficients were removed, it was predicted that these coefficients would be very similar in value to those determined from MLR. From NNLS, I performed cross-validation where training and test sets were divided such that mutations from the same protein remained in the same set. Again, the performance was variable with a Pearson's R of range -0.09 to 0.37 and an AUC range of 0.51 to 0.74.

Finally, I decided to take a different approach to cross-validation. I broke the dataset into four randomly assigned, equally populated pieces to construct a series of training and test sets. Our previous concern was that the training set might be biased if we test the same protein fold; however, the Rosetta ddg_monomer high resolution protocol focuses on changes in conformation at a local level. Additionally, thermostability data for the various proteins in the dataset are not equally distributed as some proteins only have 9 data points and others up to 71, meaning that the size of the training and test sets are highly variable when constructed this way which may have led to the highly variable Pearson R and AUC values. The new cross-validation approach resulted in a Pearson's R of range 0.38 to 0.55 and an AUC range of 0.65 to 0.81. I then averaged the coefficients determined for each score term during cross-validation to construct the weights file for the new scoring function. I rounded the averages where appropriate and consulted the previous cross-validation coefficients when needed to create the MLR/NNLS derived weights file (Table 4.2).

Table 4.2 Selected score terms and weights derived from a combination of MLR and NNLS regression analyses

Score term	Weight
fa_atr	0.17
fa_rep	0.08
fa_pair	0.5
fa_dun	0.07
fa_mbenv	0.14
fa_mbsolv	0.17
hbond_bb_sc	0.37
hbond_sc	0.39

Although these coefficients are predicted to more accurately score mutation-induced stability changes, this does not determine the affect the new weights file would have on Rosetta's sampling. Rosetta ddg_monomer is not a deterministic algorithm, so upon creating the new RosettaMembrane weights file determined from MLR and NNLS, I had to regenerate predicted ΔΔG values to determine where the sampling would be negatively affected. Unfortunately, the MLR/NNLs determined RosettaMembrane weights file resulted in a much worse metrics than the original RosettaMembrane weights file with a Pearson's R of 0.12, Spearman's R of 0.24, and an AUC of 0.6. (Table 4.1)

I used Ridge regression to try to approximate the experimentally derived ΔΔG s from the RosettaMembrane score terms values. While ridge cannot do feature selection on its own, I manually removed coefficients that were nearly zero as well as coefficients that were negative. I used the MASS package in R (version 3.2.5) and the function lm.ridge. First, I used a cross-validation approach where I separated the datasets by protein in order to avoid training on backbones seen in the test set. This resulted in a wide range of values as the proposed coefficients for each of the score terms. Additionally, the issue of having inconsistently sized training and tests set likely contributed to this variance. I then used a more randomized approach where I split the data into five random sets but kept five replicates of each mutation in the same set. I averaged the weights that resulted from each training set and developed the weight set in Table 4.3.

Table 4.3. Selected score terms and weights derived from a combination of Ridge regression analysis

Score term	Weight
fa_atr	0.18
fa_rep	0.08
fa_pair	0.57
fa_dun	0.07
fa_mbenv	0.17
fa_mbsolv	0.23
hbond_bb_sc	0.45
hbond_sc	0.43
omega	0.09

In all of the previous methods, I have done manual selection of features. Lasso is unique in that the approach uses feature selection by setting coefficients to zero. However, it still results in coefficients that are negative, so I manually removed these features and retrained on the reduced feature set. I used the glmnet and cv.glmnet functions from the glmnet package in R (version 3.2.5). It should be noted that the cv.glmnet function has an internal cross-validation with a fold set to ten. For the development of the weight set, I used a cross-validation approach that separated datasets by protein. However, this resulted in high variance of performance metrics between datasets for reasons described above. I used the same five randomized groups as described above. I averaged the weights that resulted from each training set and developed the weight set in Table 4.4.

Table 4.4. Selected score terms and weights derived from LASSO regression analysis

Score term	Weight
fa_atr	0.18
fa_rep	0.06
fa_pair	0.57
fa_dun	0.06
fa_mbenv	0.17
fa_mbsolv	0.23
hbond_bb_sc	0.45
hbond_sc	0.42
omega	0.09

I used Elastic Net as the final regression approach. Elastic Net still results in coefficients that are negative, so I manually removed these features and retrained on the reduced feature set. I used the glmnet and cv.glmnet functions from the glmnet package in R (version 3.2.5). It should be noted that the cv.glmnet function has an internal cross-validation with a fold set to ten. For the development of the weight set (Table 4.5), I used a cross-validation approach that separated datasets by protein. I varied the alpha parameter from 0 to 1 at 0.1 increments to calculate the correlation and AUC resulting from this series of tests. For the Elastic Net weight set, I used averages from the coefficients determined at an alpha value of 0.1.

Table 4.5. Selected score terms and weights derived from a combination of Elastic Net regression analysis.

Score term	Weight
fa_atr	0.12
fa_rep	0.05
fa_pair	0.4
fa_dun	0.06
fa_mbenv	0.07
fa_mbsolv	0.07
hbond_bb_sc	0.3
hbond_sc	0.24
omega	0.1

Although the regression approaches approximated the experimentally derived ΔΔG s from the RosettaMembrane score terms values calculated from the ddg_monomer application, Rosetta is by no means a deterministic algorithm. Rather than developing an energy function that is applied at the end of the ddg_monomer analysis to only impact the final score, I decided to test what affect the proposed energy function has on sampling and ultimately scoring of ddg_monomer. For the first stage, I kept the "soft_rep_design" energy function. For the second stage, the appropriate new energy function was specified. Then the standard ddg_protocol was implemented to create 50 models of each mutant and wildtype proteins. For each mutant, I selected the top three models by score and subtracted this number from the average of the top three models by score of the wildtype to obtain the predicted ΔΔG. I then compared the compared the predicted ΔΔGs to the experimentally determined ΔΔGs using Pearson's R, Spearman's rank R, and AUC (Table 4.1).

The new proposed energy functions derived from Ridge, Lasso, and Elastic Net regressions improved the Pearson and Spearman's R correlation coefficients to nearly 0.5. This is an improvement from the RosettaMembrane correlation coefficients of 0.31 and 0.34 for Pearson's R and Spearman's rank R, respectively. When generating a ROC curve based on binomial classification of either destabilizing or neutral/stabilizing, the AUC improved to nearly

0.75 from 0.68. While Ridge, Lasso, and Elastic Net had similar improvements in these metrics, Ridge and Lasso were the best at improving both Pearson's and Spearman's rank R as well as AUC. Interestingly, when comparing the score contribution of the various score terms to the total score, it is apparent that the ridge and lasso derived energy functions rely heavily on the membrane related score terms fa_memenv and fa_memsol whereas for RosettaMembrane, the score contribution of these terms is minimal (Figure 4.3). This suggests that fa_memenv and fa_memsol score terms contribute to more accurate predictions.

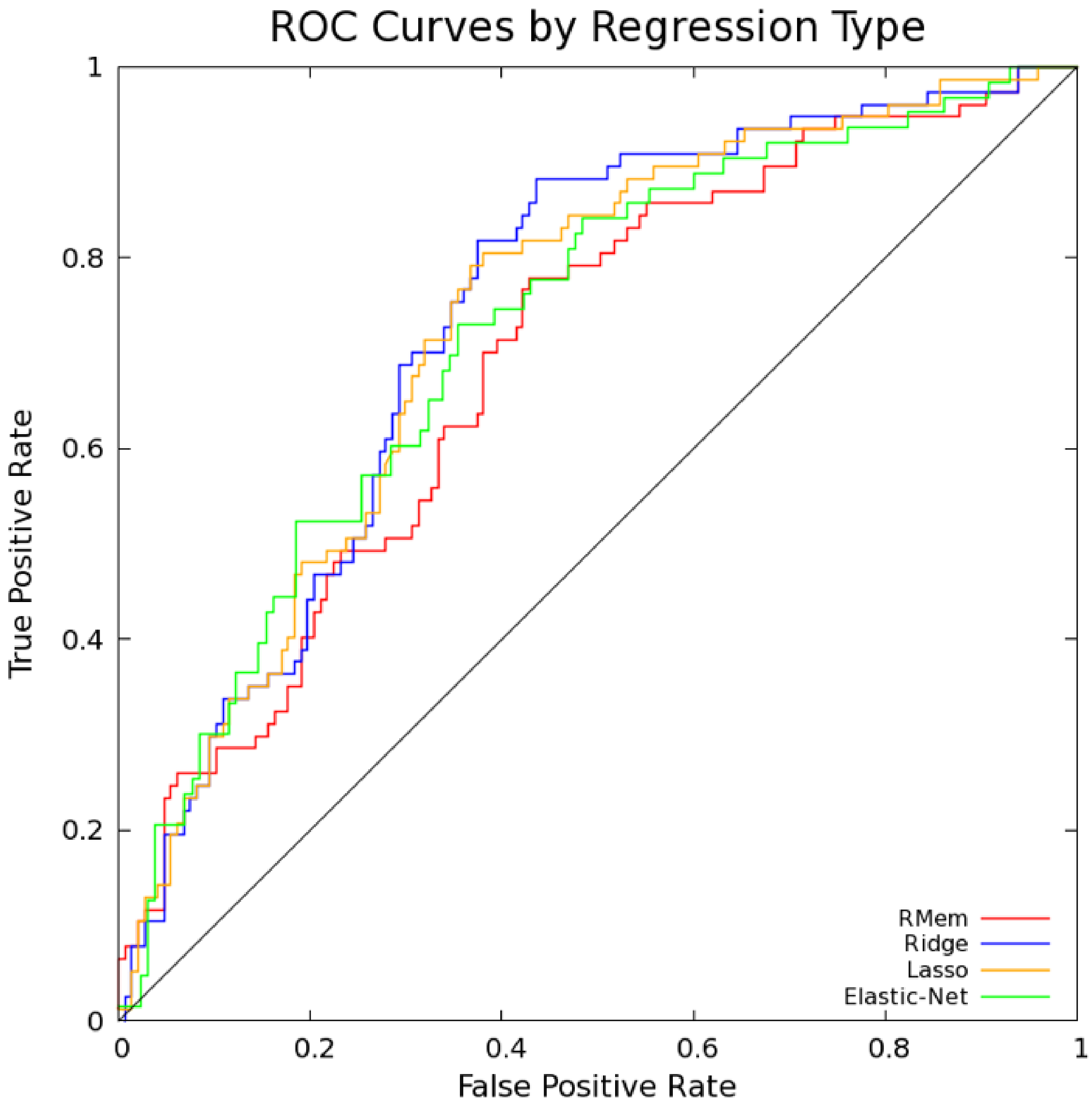


Figure 4.2. ROC curves generated from weights files derived from various regression analyses. RMem is the implementation of ddg_monomer with the original RosettaMembrane energy function. AUCs are provided in Table 4.1.

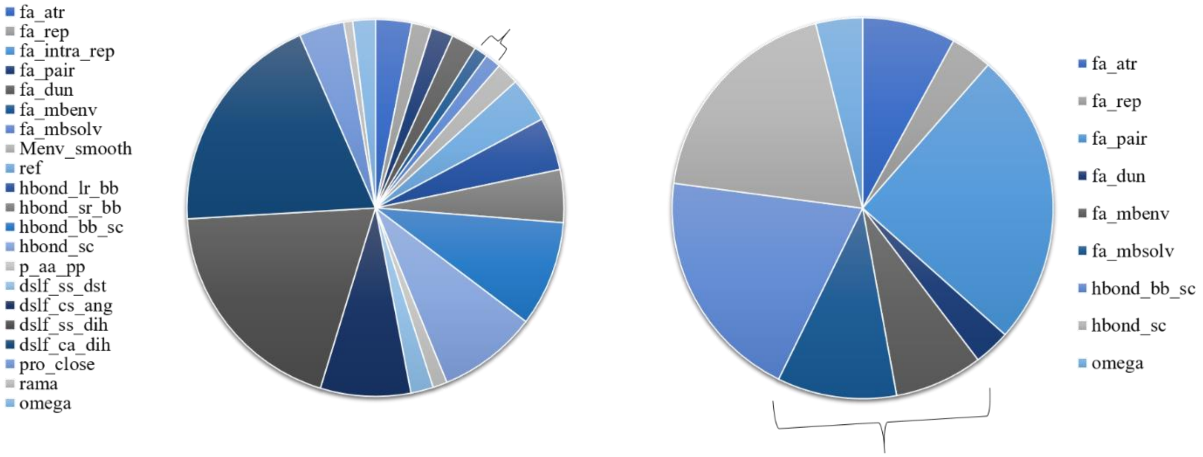


Figure 4.3. The energy contribution of score terms to total energy for RosettaMembrane (left) and Ridge regression derived energy function (right). Membrane specific score terms (bracketed), contribute only 2% to the total energy score for RosettaMembrane and 17% to the total energy score for the Ridge regression derived energy function.

Overall, the regression approaches reduced the number of score terms required for more accurate predictions and reweighted the remaining terms to improve predictions for mutation-induced stability changes in membrane proteins. The dataset consisted of 224 mutations from seven proteins. Four of these proteins are alpha-helical membrane proteins while three of these proteins are beta-barrel outer membrane proteins. Ideally, energy functions for alpha-helical and beta-barrel membrane proteins would be different; however, the available data for experimentally derived ΔΔGs is limited, thus the dataset is small. Additionally, score terms that describe interactions involved in disulfide bonds were excluded in the final weight set as they either had a zero or negative coefficient. This is because there is very little representation of mutations involved or near disulfide bonds in the dataset. Therefore, use of the ridge or lasso derived weights with mutations near sites with disulfide bonds should be cautioned. While this is a case where a specific interaction is not represented in the dataset, it raises the question of whether there are other cases where score terms could contribute to accurate predictions if these interactions were to exist in the training set. Also underrepresented are mutations from small to large amino acids. Large to small mutations are often in abundance due to alanine-scanning mutagenesis. As a result, score terms that describe packing conditions such as fa_atr and fa_rep may be biased such that fa_rep has a lower than expected contribution to the scoring function as the models in the training set have small side chains in the place of larger side chains. Whereas a penalty by fa_rep is seen less frequently because this can be expected to occur in mutations where a large side chain takes the place of small side chain. At present, it is unclear how to weight these particular score terms until more experimentally determined ΔΔGs are available.

Another observation regarding the score terms that remained in the ridge and lasso weight sets is that the hydrogen bond terms that specify interactions between side chains remained while the hydrogen bond terms that describe interactions in the backbone were removed. This could be an artefact from the sampling strategy of ddg_monomer. Recall that the input protein model is minimized with contraints prior to ddg_monomer. The high-resolution protocol first employs an all-residue sidechain repacking (no backbone minimization) while the mutated residue is introduced. The second stage does include backbone and sidechain minimization, however there are uniform constraints on the proteins such that only energetically favorable moves that do not involve movements further than 0.5 Angstroms are allowed. This effectively reduces the conformational search space and limits the backbone movements allowed during minimization. Because the ΔΔG calculation involves a subtraction of the mutant energy from the wildtype energy, these few, small, global backbone movements likely result in very little differences in energy for predicted ΔΔGs. Additionally, the hydrogen bond terms for backbone atoms tend to describe interactions pertinent to secondary structural elements. It is unlikely that these interactions would be disrupted due to the nature of the computational protocol. However, accounting for these interactions in other aspects of membrane protein modeling is important, especially in the cases of comparative and *de novo* modeling. Therefore, the proposed ridge and lasso energy functions should be used specifically for the ddg_monomer protocol in the second stage.

Another challenge to address is the experimental error involved with the methods to obtain experimentally determined ΔΔGs. The dataset contains values that were determined from SDS titrations, Urea titrations, and dimerization analytical ultracentrifugation. However, at times the error involved with these reported values can be upwards of 0.7 kcal/mol. This is particularly a concern because many of these reported values and attributed error overlapped between the destabilized and neutral/stabilized ranges, which in turn can complicate the classification stage. To simplify this, we performed regression analysis on the reported values alone and did not consider the experimental error. In recent years, a lot of progress has been made on methods for calculating thermostability that result in low error. These methods include atomic force microscopy and steric trapping (Edwards & Perkins, 2017; Jefferson, Min, Corin, Wang, & Bowie, 2017).

In the future, regression analysis with an ideal training dataset would be from a set of diverse membrane proteins; include more mutations of small to large amino acids; and derived from experimental methods with low error. Regression analysis with this ideal set would likely improve correlation coefficients to well over 0.5 and improve classification predictions that match the soluble reference set at an AUC beyond 0.8. Ideally, energy functions would also be derived for alpha helical membrane proteins and beta barrel membrane proteins separately. However, at this time, the implicit membrane representation in RosettaMembrane is limited to a fixed dimension which was derived from alpha helical proteins. Additionally, experimentally derived ΔΔGs for beta barrel proteins are sparse.

CHAPTER 5

COMPARATIVE MODELING OF THE RESTING VSD, CLOSED PORE STATE OF KCNQ1 FROM MULTIPLE TEMPLATES

This chapter contains unpublished content.

Author contribution: The manuscript as it is in this state only includes text, tables, figures and data entirely from A Duran. This chapter will be combined with additional studies in the future as a manuscript investigating the modulation of KCNQ1 by KCNE1.

5.1 Introduction

Ion channels are membrane proteins that facilitate the passage of ions across the membrane. For this reason, they are important in regulation and signal transduction in the cell, and, in particular voltage-gated ion channels, are commonly found in neurons, as well as cardiac and smooth muscle cells (Fosmo & Skraastad, 2017). In the human genome, it is estimated that approximately 350 genes encode some type of ion channel (Blin et al., 2016; Schmidt & Peyronnet, 2018). Ion channels were found to be the second largest drug target (Alexander, Mathie, & Peters, 2011; Overington, Al-Lazikani, & Hopkins, 2006), which can likely be attributed to their role in signaling as well as pore gating mechanisms seen in many ion channels. Additionally, genetic variants of ion channels that affect the structure and function of the ion channel complex can lead to serious complications (Heijman & Dobrev, 2017). KCNQ1 is a voltage-gated potassium channel that is associated with Long-QT syndrome, a potentially fatal cardiac arrhythmia that can arise from genetic variants of KVLQT1 (Q. Wang et al., 1996).

The voltage-gated ion channel superfamily is largely responsible for signal transduction in cells and includes voltage-gated sodium channels, voltage-gated calcium channels, and voltage-gated potassium channels (Yu, Yarov-Yarovoy, Gutman, & Catterall, 2005). Members of this superfamily can be characterized as a six transmembrane protein that is comprised of a four transmembrane (S1-S4 helices) voltage-sensing domain (VSD) and a two transmembrane (S5-S6) pore-forming domain. While both voltage-gated sodium and calcium channel tetramers are constructed from four homologous subunits, voltage-gated potassium channels are homotetrameric. It is believed that voltage-gated ion channels all have a common two transmembrane spanning ancestral protein, similar to that of the inwardly rectifying potassium channels, that evolved to accommodate various functions through the addition of domains like that of the voltage-sensing domains that are seen today. As such, other homotetrameric ion channels that resemble the architecture of voltage-gated potassium channels include calcium-activated potassium channels, cyclic nucleotide-gated (CNG) , hyperpolarization-activated cyclic nucleotide-modulated (HCN), and transient receptor potential (TRP) channels (Yu et al., 2005).

Voltage-dependent gating consists of three stages: voltage sensor activation, VSD coupling to the pore gating domain (PGD), and pore opening (Cui, 2016). In voltage-gated ion channels, the S4 helix of the VSD has a number of cationic residues that are believed to be the main voltage sensing component. For KCNQ1, these residues are R228 (R1), R231 (R2), Q234 (R3), R237 (R4), H240 (H5), and R243 (R6). The S2 helix contains two anionic residues, E160 (E1) and E170 (E2), that interact with cationic residues in S4 (D. Wu et al., 2010). These specific residue pairings are key for distinguishing between active and resting states.

KCNQ1 with the charged residue pairing of E1 and R1 has been observed as a VSD in the resting state and a closed pore. The pairing of E1 and R4 is indicative of an active VSD and and presumably an open pore. These end result of the functional states are unaffected by the presence of KCNE1. However, the residue pairing of E1 and R2 is believed to be an intermediate state of the VSD where isolated KCNQ1 has an open pore while this same pairing in the presence of KCNE1 results in a closed pore (Cui, 2016; D. Wu et al., 2010). It is proposed that this is due to the two-phase VSD activation of KCNQ1 in the presence of KCNE1. These phases consist of a fast movement in each of the S4 helices from the resting to activated state in negative voltages which displaces most of the gating charge which is followed by a slow continuation of the displacement of the remaining gating charge through the S4 movement that is coupled to the pore opening during positive voltages (Barro-Soria et al., 2014).

The coupling between the VSD and PGD is also influenced by an interaction between the S4-S5 linker and the S6 helix on the cytosolic side. Interestingly, mutational studies revealed that mutations to either V254 in the S4-S5 linker or the S6 helix at position L353 resulted in a partially constitutively open pore, in particular V254A, V254L, V254E and L353A. However, a double mutant containing V254L/L353A recovered the ability of the pore to close likely due to recovering the contact between these two residues with side-chains of similar size (Cui, 2016; Labro et al., 2011). Additionally, VSD-PGD coupling of an activated VSD and an open PGD was shown to require the presence of phosphatidylinositol 4,5-bisphosphate (PIP2), a lipid that exists in the inner leaflet of the membrane (Zaydman et al., 2013). However, it should be distinguished that studies that involved the depletion of PIP2 revealed that PIP2 was not required for VSD activation directly (Zaydman & Cui, 2014; Zaydman et al., 2013).

Recently, the structure of KCNQ1 from frog (Xenopus laevis) was determined in complex with calmodulin (CaM) by cryoelectron microscopy (cryo-EM) (Sun & MacKinnon, 2017). The structure was determined in in the absence of PIP2, therefore it is representative of an uncoupled state where the voltage sensors are activated and the pore is closed. While studies have been able to determine which residues are interacting in the resting state of the VSD, this has never been structurally resolved for voltage-gated potassium channels. Previous structural models of KCNQ1 with the VSD in the resting state have been created (Smith, Vanoye, George, Meiler, & Sanders, 2007); however, this was before many of the recent studies that identified the pairing of charged residues in active and resting states for the VSD.

The sequence identity of frog KCNQ1 to human KCNQ1 is 78% (Sun & MacKinnon, 2017), which makes the frog KCNQ1 structure an ideal template for homology modeling of human KCNQ1. Herein, Rosetta, a molecular modeling software suite, is used to create a model of KCNQ1 with resting VSD and a closed pore. The obvious challenge is the lack of available templates for the resting VSD of KCNQ1. However, RosettaCM (Song et al., 2013) enables the use of multiple templates in addition to fragment insertion during the modeling process. While there are currently no available structures of voltage-gated potassium ion resting VSDs, I was able to use existing models of the voltage-gated Shaker C3 VSD in the resting state (Henrion et al., 2012), the crystal structure of the resting state of a voltage-sensitive phosphatase VSD from *Ciona intestinalis* (Ci-VSP) (Li et al., 2014), and the main voltage-sensitive domain (VSD2) in the inactive state from the crystal structure of two-pore channel (TPC1) from *Arabidopsis thaliana* (Kintzer & Stroud, 2016).

5.2 Methods

RosettaCM in Rosetta release 3.6 was used in combination with RosettaMembrane (Barth et al., 2007; Yarov-Yarovoy et al., 2006), and Rosetta Symmetry (Dimaio et al., 2011) to approach modeling the resting/closed state of KCNQ1. The first step of RosettaCM required a structural alignment by MUSTANG. The templates (Table 5.1) for the resting state of the voltage sensor, Ci-VSP, TPC1, and Shaker, were structurally aligned to chain A of the frog KCNQ1 structure. The multiple sequence alignment generated by MUSTANG (Konagurthu, Whisstock, Stuckey, & Lesk, 2006) was input to the Clustal OMEGA server (Goujon et al., 2010; Sievers et al., 2011). Several manual adjustments were made such that there were no gaps in the secondary structural elements (see Appendix G, Figure AG.1). Additionally, the alignment of conserved residues involved in the charged interactions between S2 and S4 was enforced. Alignments were converted into the grishin format. Aligned structures and sequence alignments were input into the partial threading application (see Appendix G for command line).

Exhaustive tests were performed to determine the optimal combination of templates. Details of these trials can be found in Appendix G. Because the frog KCNQ1 structure contains an active VSD, the voltage sensor from chain A of the frog KCNQ1 structure was replaced with the Ci-VSP structure. The Ci-VSP structure does not contain structured regions for the S0 helix and intracellular loop between S2 and S3. Fragments from the frog KCNQ1 activated voltage sensor were swapped in for these regions on the initial Ci-VSP VSD- frog KCNQ1 PD hybrid template. The Ci-VSP-frog KCNQ1 hybrid was used as the initial template for RosettaCM hybridization. In addition, within the RosettaCM hybridization mover, detailed protocols was employed to enforce only template sampling for residues in the S0 helix and intracellcular loop between S2 and S3. It was determined that in addition to the Ci-VSP structure, sampling of the VSDs from the TPC1 structure and Shaker models resulted in good-quality models. Approximately 1000 models were built at the hybridization phase which includes an internal minimization. The distance between residues E1 and R1 were calculated, and the 50 models by the shortest E1-R1 distance were selected for further minimization studies. These 50 models will be referred to as parent models.

Several minimization studies were tested to determine the optimal order of protocols. More details can be seen in Appendix G. The final protocol included first a dualspace relax with the constrained to start coordinates option. Each model was minimized 25 times for a total of approximately 1,250 models. The top 100 models were then selected based first on lowest Rosetta energy score (REU), shortest E1-R1 distance, and lowest backbone transmembrane root-mean-square deviation (RMSD) to templates Ci-VSP, frog KCNQ1, and Shaker. These top models were minimized 25 times, a total of 2,500 models, using dualspace relax and a constrain to start coordinates option with the ramp_constraints option set to false. The top ten models were then selected based on lowest REU, and shortest E1-R1 distance (Figure 5.1). These ten belonged to the S20 parent model. In an effort to represent more conformational diversity, a second set of 10 models was selected from a second parent model, S12, whose minimized child models had the next lowest REU and shortest E1-R1 distance after minimized models from S20.

Table 5.1 Annotation of appropriate templates available for constructing a closed pore, resting VSD model of KCNQ1. Not all of the templates in this table were used in the final protocol (see protocol capture in Appendix G for details).

Domain	**Template**	**Sequence Identity**	**Sequence Coverage**	**Type**	**Reasons to use as a template**
Pore	Frog KCNQ1[a]	93%	100%	Cryo-EM	Closed pore of high sequence identity and coverage
Voltage Sensor	Ci-VSP[b]	20%	91%	X-ray crystal	High sequence coverage, resting VSD structure
Voltage Sensor	Shaker (C3)[c]	18%	61%	Model	Shows proposed E1-R1 interaction
Voltage Sensor	KCNQ1[d]	100%	85%	Model	Original closed-state model
Voltage Sensor	TPC1[e]	20%	74%	X-ray crystal	Inactive VSD2 structure
Voltage Sensor	Kv1.2-2.1[f]	23%	66%	Model	Increase conformational diversity
Voltage Sensor	MLotiK[g]	40%	76%	X-ray crystal	High sequence identity
Voltage Sensor	Frog KCNQ1[a]	83%	92%	Cryo-EM	Orientation of domains; secondary structure for S0, S2S3 linker

a (Sun & MacKinnon, 2017); b (Li et al., 2014); c (Henrion et al., 2012); d (Smith et al., 2007); e (Kintzer & Stroud, 2016); f (Jensen et al., 2012); g (Clayton, Altieri, Heginbotham, Unger, & Morais-Cabral, 2008)

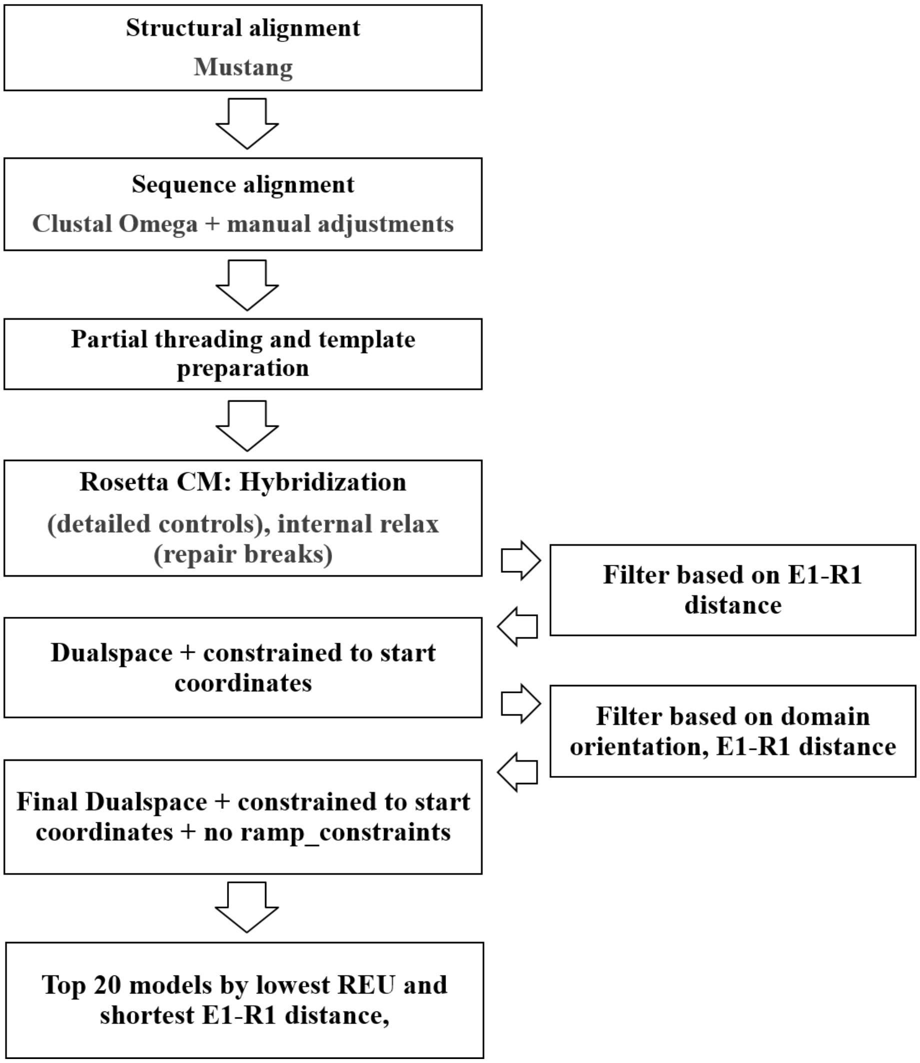

Structural alignment
Mustang
Sequence alignment
Clustal Omega + manual adjustments
Partial threading and template preparation
Rosetta CM: Hybridization
(detailed controls), internal relax (repair breaks)
Filter based on E1-R1 distance
Dualspace + constrained to start coordinates
Filter based on domain orientation, E1-R1 distance
Final Dualspace + constrained to start coordinates + no ramp_constraints
Top 20 models by lowest REU and shortest E1-R1 distance,

Figure 5.1. Flowchart of RosettaCM protocol developed for modelling the resting VSD, closed pore state of KCNQ1. RosettaCM also employed RosettaMembrane and Rosetta Symmetry at the hybridization and dualspace relax stages. Full protocol capture can be found in Appendix G.

5.3 Results and Discussion

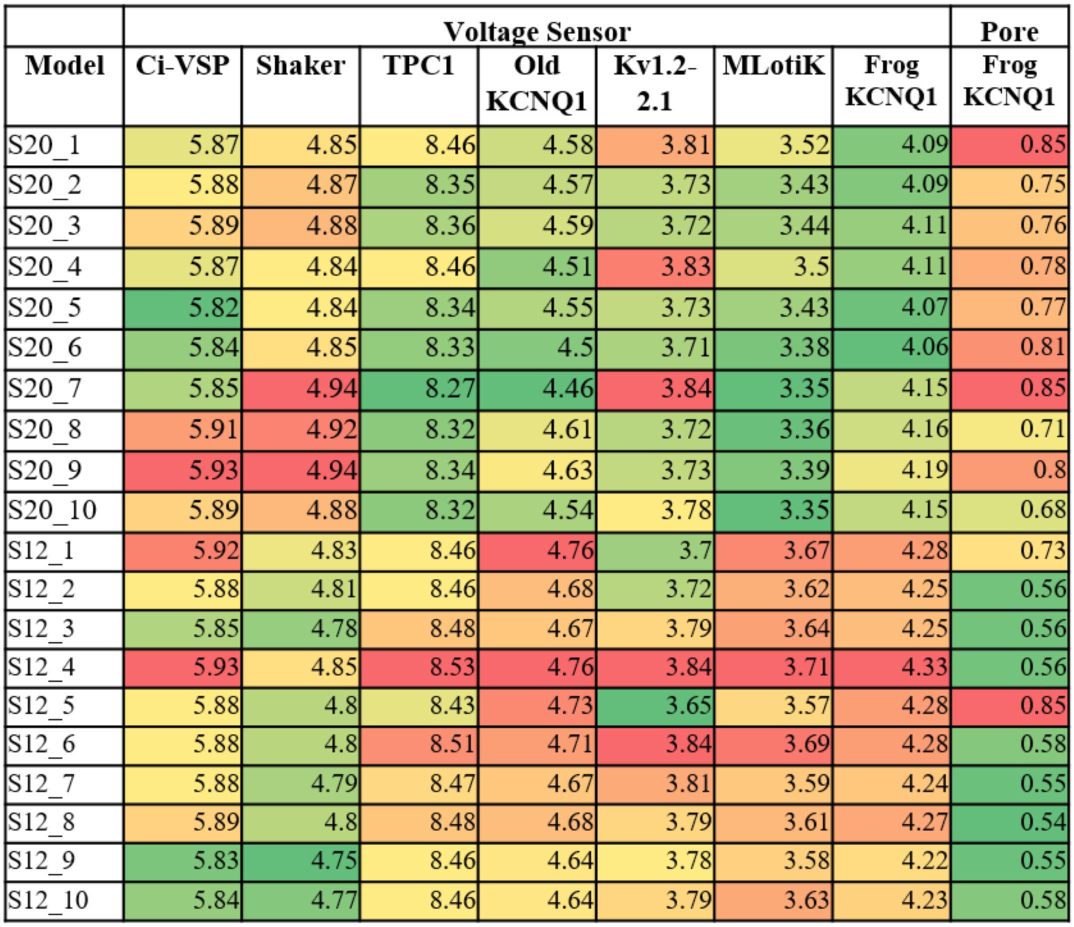
An ensemble of KCNQ1 models with a resting VSD and closed pore were created using RosettaCM. While the frog KCNQ1 structure has high sequence identity and sequence coverage, the VSD is in an active state. The literature shows that there is strong evidence that the residues E1 and R1 of the VSD are interacting in the resting state (Cui, 2016; D. Wu et al., 2010). The distance between E1 and R1 was calculated during the model building process to filter out models that did not contain a possible E1-R1 interaction. The E1-R1 distance was also calculated for the final top 20 models for analysis (Table 5.2). All final models have a distance of 3.7 Angstroms or less. It is believed that E2 interacts with R2, however, this is less clear. Distances between E2-R2 and E2-R3 were also reported.

Table 5.2. Distances, in Angstroms, of charged residues in voltage sensor helices S2 and S4 for the final top 20 models of resting state KCNQ1. Residues E1 and E2 are on the S2 helix, while R1, R2, and R3 are on the S4 helix. Distance between residue pairs E1R1, E2R2, and E2R3 are reported.

Model	E1R1	E2R2	E2R3
S20_1	3.4	3.5	10.3
S20_2	3.4	3.7	10.7
S20_3	3.4	3.7	10.6
S20_4	3.4	3.6	10
S20_5	3.4	3.6	10.2
S20_6	3.4	3.6	10.3
S20_7	3.4	3.8	11.3
S20_8	3.4	4	10.8
S20_9	3.4	3.8	10.5
S20_10	3.4	3.8	10.6
S12_1	3.6	3.4	15.3
S12_2	3.6	3.4	15.3
S12_3	3.6	3.5	15.2
S12_4	3.6	3.5	15.5
S12_5	3.4	3.4	15
S12_6	3.7	3.5	15.3
S12_7	3.6	3.5	15.5
S12_8	3.6	3.4	15
S12_9	3.5	3.5	15.3
S12_10	3.6	3.4	15.2

The transmembrane backbone RMSD was calculated throughout the model building process for model comparisons to frog KCNQ1, Ci-VSP, TPC1, and the Shaker model. Models with lower RMSDs after hybridization were considered to be more favorable. For the final top 20 models, the transmembrane backbone RMSD was calculated for comparisons to the old KCNQ1 model (Smith et al., 2007), Kv1.2-2.1 model (Jensen et al., 2012), MLotiK (Clayton et al., 2008), and the frog KCNQ1 VSD (active state) in addition to templates frog KCNQ1 pore, Ci-VSP, TPC1, and the Shaker model (Figure 5.2).

Interestingly, VSDs for models had the lowest transmembrane backbone RMSD to MLotiK and Kv1.2-2.1. MLotiK was exclude from the model building process because it is not a true voltage sensor but rather an S1-S4 domain (Clayton et al., 2008). Kv1.2-2.1 was excluded from the model building process because all four chains had different conformations with chain B appearing to be decoupled (Jensen et al., 2012). Chain C was used for RMSD calculations. The frog KCNQ1 VSD had the next lowest RMSD for both series. While this represents the active state, it is believed that the S4 helix contributes to the majority of the difference in conformation between active and resting, so a low RMSD here was not surprising. Both the old KCNQ1 model and Shaker model had the next lowest RMSDs when compared to the final models. This was also not surprising considering the homology models in these studies were build using Rosetta, albeit different versions. Interestingly, the Shaker model, Ci-VSP, and TPC1 had the highest RMSDs of all model comparisons to VSDs. Finally, the comparison of the pore domain from final models to the frog KCNQ1 template resulted in a transmembrane backbone RMSD of under 1 Angstrom for all final models.



	Voltage Sensor							Pore
Model	Ci-VSP	Shaker	TPC1	Old KCNQ1	Kv1.2-2.1	MLotiK	Frog KCNQ1	Frog KCNQ1
S20_1	5.87	4.85	8.46	4.58	3.81	3.52	4.09	0.85
S20_2	5.88	4.87	8.35	4.57	3.73	3.43	4.09	0.75
S20_3	5.89	4.88	8.36	4.59	3.72	3.44	4.11	0.76
S20_4	5.87	4.84	8.46	4.51	3.83	3.5	4.11	0.78
S20_5	5.82	4.84	8.34	4.55	3.73	3.43	4.07	0.77
S20_6	5.84	4.85	8.33	4.5	3.71	3.38	4.06	0.81
S20_7	5.85	4.94	8.27	4.46	3.84	3.35	4.15	0.85
S20_8	5.91	4.92	8.32	4.61	3.72	3.36	4.16	0.71
S20_9	5.93	4.94	8.34	4.63	3.73	3.39	4.19	0.8
S20_10	5.89	4.88	8.32	4.54	3.78	3.35	4.15	0.68
S12_1	5.92	4.83	8.46	4.76	3.7	3.67	4.28	0.73
S12_2	5.88	4.81	8.46	4.68	3.72	3.62	4.25	0.56
S12_3	5.85	4.78	8.48	4.67	3.79	3.64	4.25	0.56
S12_4	5.93	4.85	8.53	4.76	3.84	3.71	4.33	0.56
S12_5	5.88	4.8	8.43	4.73	3.65	3.57	4.28	0.85
S12_6	5.88	4.8	8.51	4.71	3.84	3.69	4.28	0.58
S12_7	5.88	4.79	8.47	4.67	3.81	3.59	4.24	0.55
S12_8	5.89	4.8	8.48	4.68	3.79	3.61	4.27	0.54
S12_9	5.83	4.75	8.46	4.64	3.78	3.58	4.22	0.55
S12_10	5.84	4.77	8.46	4.64	3.79	3.63	4.23	0.58

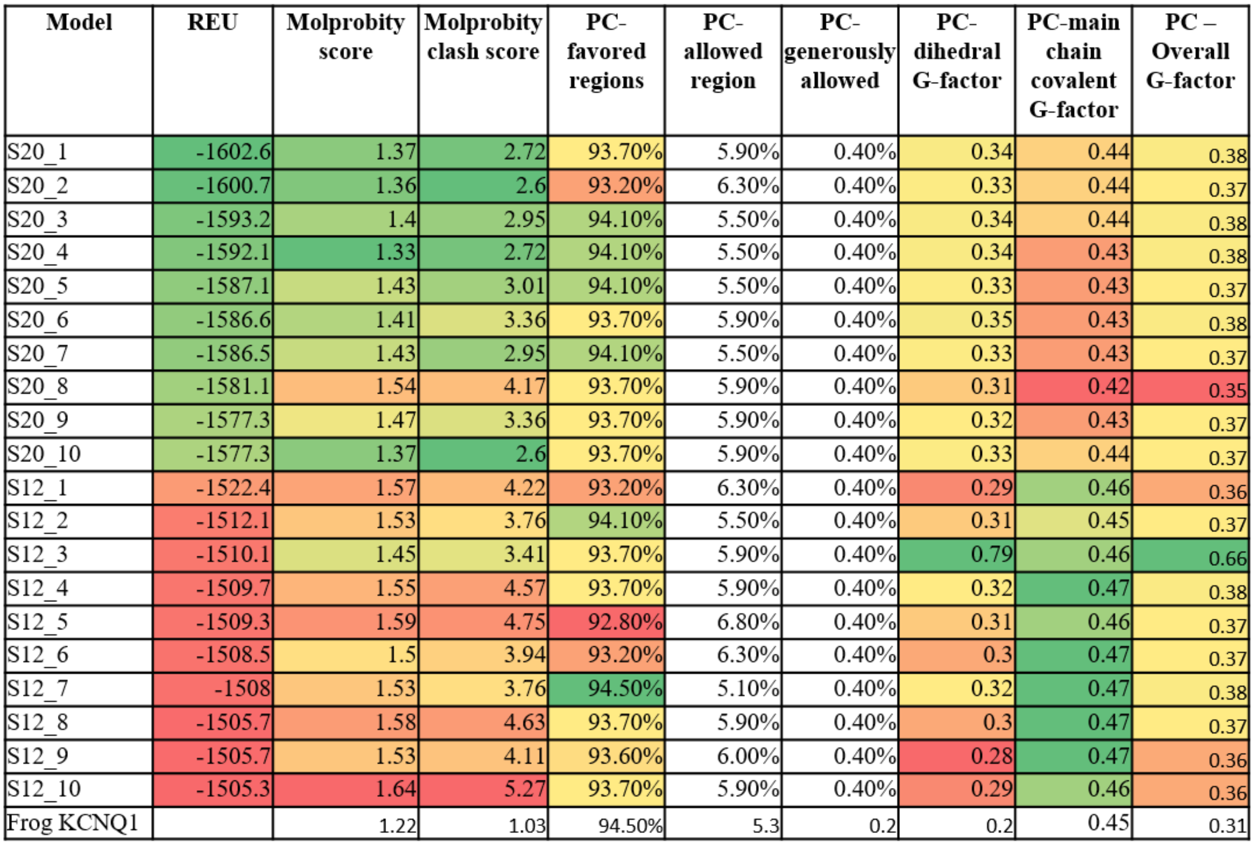
Figure 5.2. Heatmap of transmembrane backbone RMSD for resting VSD and closed pore models of KCNQ1 to VSD and pore templates. Each column has its own color scale with green for the lowest RMSD and red for the highest RMSD. It should be noted that in many cases, the difference in RMSD within the column is negligible, such as in the Kv1.2-2.1 column. Only Ci-VSP, Shaker, and TPC1 were used for modeling building.

While the transmembrane backbone RMSD for the resting VSD models compared to the frog KCNQ1 active VSD showed a moderately low RMSD, it does not provide an appropriate measure for how different the two states are from each other. The movement of S4 from the resting to active state is thought to be the largest conformational change. The S1-S3 helices from each of the top 20 models were structurally aligned to the S1-S3 helices in the VSD of the frog KCNQ1 structure. The RMSD was calculated between fragments covering the same sequence of the S4 helix in each of the top 20 models and frog KCNQ1 (Table 5.3). The S20 series S4 helix was calculated to have moved approximately 8 Angstroms while the S12 series S4 helix was calculated to have moved approximately 11 Angstroms.

Table 5.3. Calculated movement of the S4 helix. Helices S1-S3 from the top 20 models were aligned to the S1-S3 helices in the frog KCNQ1 structure and movement of the S4 helix alone is reported.

Model	**S4 Movement**
S20_1	8.31
S20_2	8.16
S20_3	8.16
S20_4	8.32
S20_5	8.14
S20_6	8.16
S20_7	8.67
S20_8	8.22
S20_9	8.49
S20_10	8.39
S12_1	10.93
S12_2	10.76
S12_3	10.73
S12_4	10.98
S12_5	10.74
S12_6	10.85
S12_7	10.82
S12_8	10.9
S12_9	10.75
S12_10	10.7

The top 20 models were selected in part by the lowest Rosetta energy score. To ensure the selection of models was not biased by the Rosetta energy function, I validated the quality of the models using external servers Molprobity (V. Chen et al., 2010; Davis et al., 2007) and PDBsum (de Beer, Berka, Thornton, & Laskowski, 2014) (ProCheck). The S20 series scored nearly 100 Rosetta energy units (REU) lower than the S12 series indicated that the S20 series is more energetically favorable. The overall Molprobity scores and Molprobity clash scores were reported (Figure 5.3) and lower scores are more favorable as indicated by green coloring. PDBsum (ProCheck) scores were reported for the percentage of rotamers in various regions of acceptance. Nearly all S20 models and a few S12 models had approximately 94% of rotamers in favored regions as colored by green and yellow. Additionally, G-factors for properties relating to torsion angles (PC-dihedral), covalent bond geometry (PC-main chain covalent), and overall G-factors are also reported where positive numbers further from zero indicate a higher probability (green) of favorable properties.



Model	REU	Molprobity score	Molprobity clash score	PC-favored regions	PC-allowed region	PC-generously allowed	PC-dihedral G-factor	PC-main chain covalent G-factor	PC – Overall G-factor
S20_1	-1602.6	1.37	2.72	93.70%	5.90%	0.40%	0.34	0.44	0.38
S20_2	-1600.7	1.36	2.6	93.20%	6.30%	0.40%	0.33	0.44	0.37
S20_3	-1593.2	1.4	2.95	94.10%	5.50%	0.40%	0.34	0.44	0.38
S20_4	-1592.1	1.33	2.72	94.10%	5.50%	0.40%	0.34	0.43	0.38
S20_5	-1587.1	1.43	3.01	94.10%	5.50%	0.40%	0.33	0.43	0.37
S20_6	-1586.6	1.41	3.36	93.70%	5.90%	0.40%	0.35	0.43	0.38
S20_7	-1586.5	1.43	2.95	94.10%	5.50%	0.40%	0.33	0.43	0.37
S20_8	-1581.1	1.54	4.17	93.70%	5.90%	0.40%	0.31	0.42	0.35
S20_9	-1577.3	1.47	3.36	93.70%	5.90%	0.40%	0.32	0.43	0.37
S20_10	-1577.3	1.37	2.6	93.70%	5.90%	0.40%	0.33	0.44	0.37
S12_1	-1522.4	1.57	4.22	93.20%	6.30%	0.40%	0.29	0.46	0.36
S12_2	-1512.1	1.53	3.76	94.10%	5.50%	0.40%	0.31	0.45	0.37
S12_3	-1510.1	1.45	3.41	93.70%	5.90%	0.40%	0.79	0.46	0.66
S12_4	-1509.7	1.55	4.57	93.70%	5.90%	0.40%	0.32	0.47	0.38
S12_5	-1509.3	1.59	4.75	92.80%	6.80%	0.40%	0.31	0.46	0.37
S12_6	-1508.5	1.5	3.94	93.20%	6.30%	0.40%	0.3	0.47	0.37
S12_7	-1508	1.53	3.76	94.50%	5.10%	0.40%	0.32	0.47	0.38
S12_8	-1505.7	1.58	4.63	93.70%	5.90%	0.40%	0.3	0.47	0.37
S12_9	-1505.7	1.53	4.11	93.60%	6.00%	0.40%	0.28	0.47	0.36
S12_10	-1505.3	1.64	5.27	93.70%	5.90%	0.40%	0.29	0.46	0.36
Frog KCNQ1		1.22	1.03	94.50%	5.3	0.2	0.2	0.45	0.31

Figure 5.3. Heatmap of external model validation. Final models in the S20 and S12 series of KCNQ1 with a resting VSD and closed pore are compared by metrics from Rosetta, Molprobity, and ProCheck (PC). Each column has its own color scale where green indicates the better values

within a column and red represents the worst values within a column. Metrics for the Frog KCNQ1 structure are provided as a reference.

The S20 ensemble was superimposed on the frog KCNQ1 structure (Figure 5.4, A-C) and the S12 ensemble was superimposed on the frog KCNQ1 structure (Figure 5.4, D-F). Visual inspection of the models in both series verified that the S5 and S6 helices as well as the central pore, are very similar to the frog KCNQ1 structure, which has a closed pore. Additionally, it appears that the S20 series ensemble is more variable in the region of the voltage sensor than the S12 series ensemble. However, the calculated transmembrane backbone RMSD values of the S20 models to the frog KCNQ1 VSD were in a relatively narrow range (Figure 5.2), indicating that while the models appear to be different from each other, the amount of which they are different from the frog KCNQ1 active VSD is similar.

The variability in the superimposed structures for the S20 series may be attributed to two possible events. The models were build using an initial template that consisted of a pore from one template, frog KCNQ1, and a voltage sensor from Ci-VSP. I trimmed the Ci-VSP template on the c-terminus and the frog KCNQ1 pore template on the n-terminus to create a gap in the structure of the template, rather than introduce clashes immediately into the structure. The hybridization protocol allowed only fragment insertion in this short region so that the connection could be built with each template as an anchor. In the S20 series, this region is unstructured which means that perturbations in this loop could create a leaver-arm effect such that the orientation of the voltage sensor to the S4-S5 linker is different with each outcome. All minimization protocols employed in this pipeline added a constrain to start coordinates, so the global changes were not so drastic as to move the VSD out of the plane of the PD. However, this would also indicate that the perturbations were occurring in the unstructured region frequently perhaps because this was not energetically favorable.

While the S12 series does not show this variability in the positioning of the voltage sensor, one concern is the distortion of the S4-S5 linker. Although in this model, the S4 helix has an additional turn of a helix compared to models in the S20 series, the causes the S4-S5 linker to unravel and bend downwards, away from the additional density imposed by the structured end of the S4 helix. The S4-S5 linker region has not been structurally resolved, so it is difficult to refute this distortion. However, the main concern of the S12 series are the lower Rosetta and Molprobity scores (Figure 5.3). For this reason, I have continued analysis with only the S20 series.

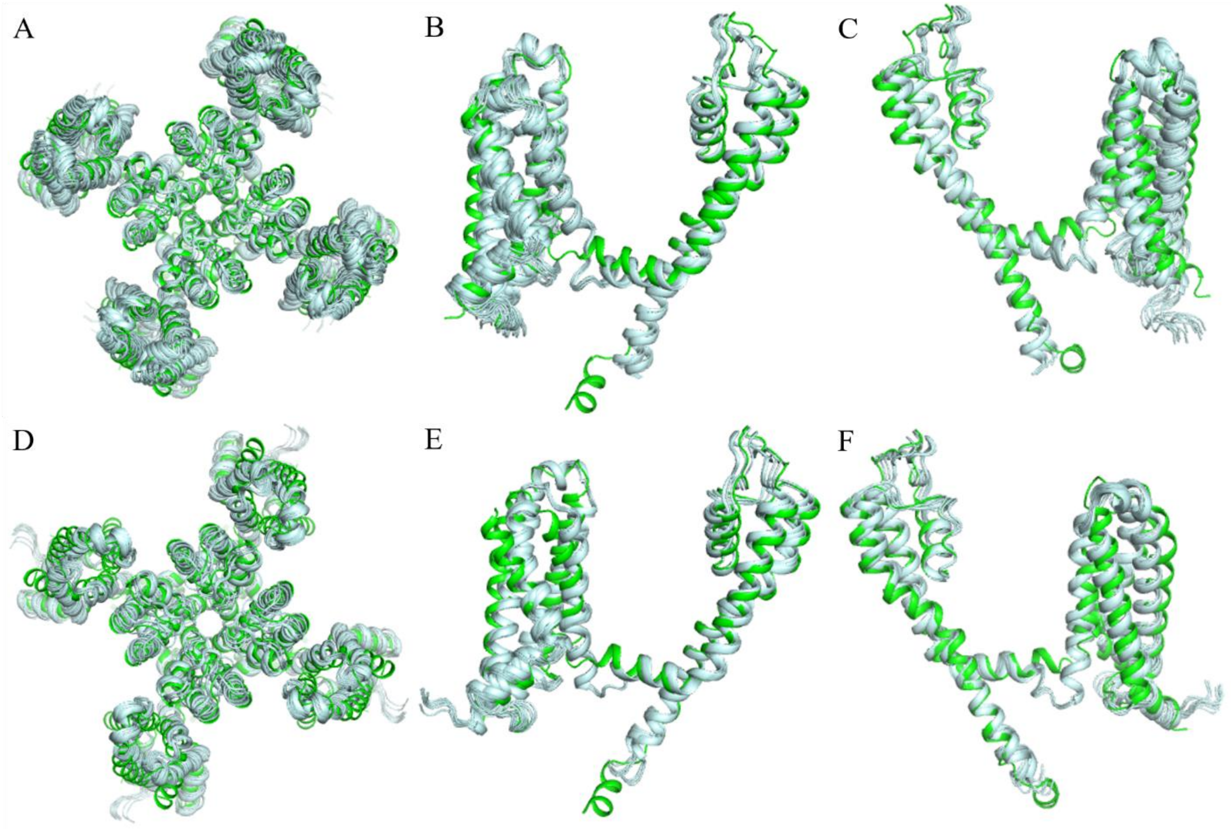


Figure 5.4. Ensemble of S20 and S12 series compared to frog KCNQ1. The top ten models by score of the S20 series (cyan) are superimposed on the frog KCNQ1 structure (green) (A-C). The top ten models by score of the S12 series (cyan) are superimposed on the frog KCNQ1 structure (green) (D-F). A top down view shows a clear view of the central pore (A,D). Views of chain A from the side (B-C,E-F) displays the differences seen in the S4-S5 linker.

The VSD of the S20 series was superimposed on each of the available VSD templates. Because of the aforementioned variability of the VSD in the ensemble, the top three models by score were selected for visualization and are shown in light cyan (Figure 5.5). Interestingly, from a top-down, global perspective, the major differences between the frog KCNQ1 VSD and the top three VSDs are in helices S2 and S3 (Figure 5.5, A). The frog KCNQ1 VSD represents an active state, so it would be predicted that the S4 helix would be the most different; however, it is unclear from this viewpoint if the S4 helix sequence aligns as well as the structure. The Ci-VSP aligns well with the S1-S3 helices of the top models; however, the S4 helix is in a noticeably different conformation (Figure 5.5, B). The TPC1 structure shares a similar conformation in helices S1-S2; however, S3 and S4 helices are quite different (Figure 5.5, C), which is expected because the linker for TPC1 is in a much different orientation because TPC1 is a heterotetramer whereas KCNQ1 is a homotetramer. Finally, the Shaker model shares the feature in the extracellular loop between S1 and S2 as well as a similar conformation of the S1 helix (Figure 5.5, D).

Interestingly, the templates that were not used in model building showed striking resemblance to the top models. The old KCNQ1 model from Smith et al. superimposes well with the S2 and S3 helices of the top models (Figure 5.5, E). Whereas, the Kv1.2-2.1 model aligns well with the top models in all regions except for the extracellular ends of helix S3 and S4 (Figure 5.5, F). Finally, the MlotiK structure aligns well in regions covering S1, S2, S4 and even the S4-S5 linker (Figure 5.5, G). These observations agree with Figure 5.2 where it was shown that templates not used in model building had lower transmembrane backbone RMSDs than templates used in the model building process. I propose two possible reasons for this. As mentioned previously, the Shaker model, old KCNQ1 model, and Kv1.2-2.1 model were all initially homology modeled using Rosetta. While the version of Rosetta used to build the top models is much more recent than others, it could be that Rosetta still scores these conformations favorably resulting in a bias toward the templates that are Rosetta models when comparing to all templates. Lastly, the MLotiK structure has the highest sequence identity of all of the available templates (Table 5.1). It was excluded from the model building process because it does not have a true voltage-sensor; however, it is validating that the final models closely resemble the template with the highest sequence identity when that template was excluded from the model building process.

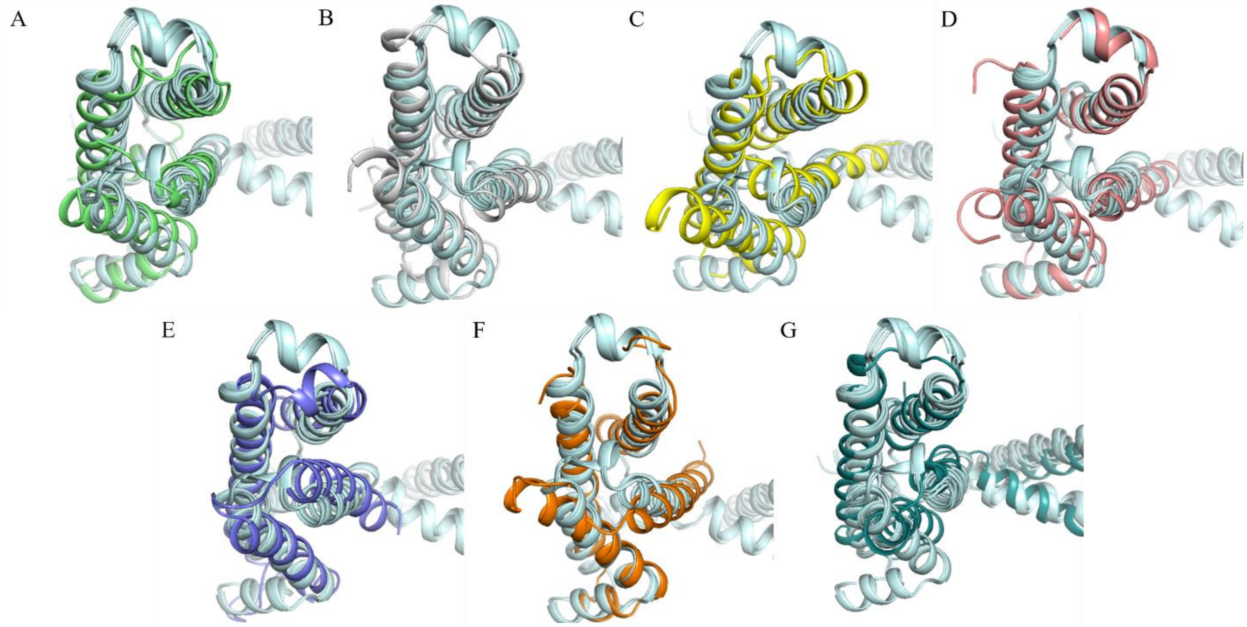


Figure 5.5. The VSD of the top three models by score from the S20 series (light cyan) superimposed with frog KCNQ1 (A), Ci-VSP (B), TPC1 (C), Shaker model (D), old KCNQ1 model (E), Kv1.2-2.1 model (F), MLotiK (G). Templates for model building (A-D) and templates not use in model building (E-G) were compared to the ensemble of the top three models.

To better visualize the differences between the active and resting states, the top three models were superimposed on the frog KCNQ1 structure. Residues involved in gating charges were highlighted as sticks to emphasize the change in register (Figure 5.6). From the active to resting state the R1 residue shifts approximately 1.5 turns of a helix. This shift propagates down the helix and offsets R2 and R3 in similar amounts.

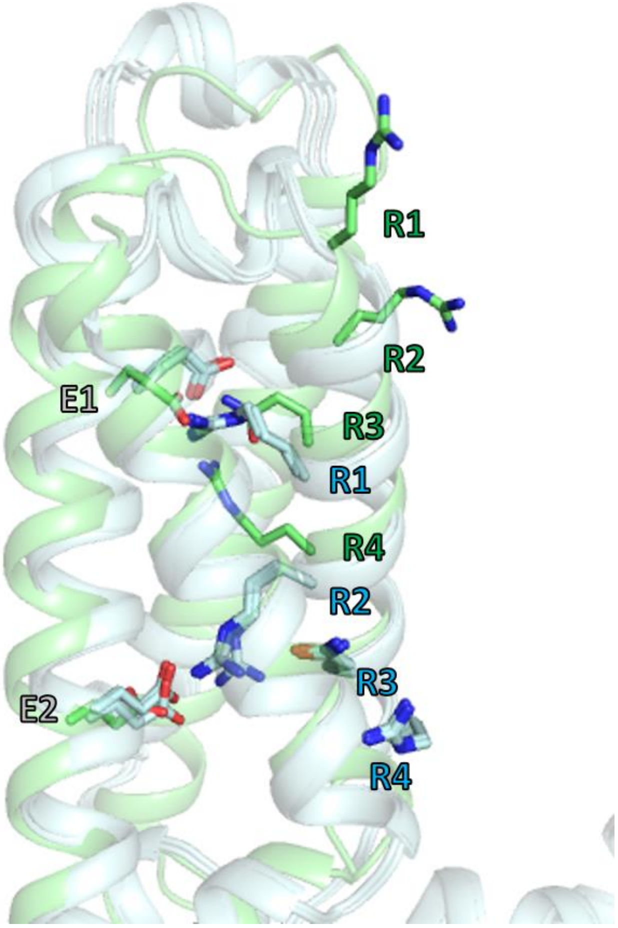


Figure 5.6. Interactions between gating charges in the active and resting state of the KCNQ1 VSD. Active is represented by the frog KCNQ1 structure (green), while resting is represented by the top three models of the S20 series (cyan). Residues involved in gating charges are shown in sticks and are labeled for active (green), resting (cyan), and both (gray).

One of the concerns with modeling membrane proteins with a pore in Rosetta is the collapse of the pore. The attractive and repulsive terms best represent the Leonard Jones potential which is the driving force of the Rosetta energy function. This means that Rosetta favorably scores conformations where residues make the most contact with surrounding residues without creating clashes. Typically in Rosetta, membrane proteins are modeled in the absence of water molecules. Without the water molecules present to take up density, models of pores are more likely to collapse in an effort to fill the empty space in the core of the protein. To overcome this, each minimization step contains a constrain to start coordinates where the protein backbone is not allowed to be perturbed beyond 0.5 Angstroms.

In order to verify that the central pore was not changed during modeling in such a way that distorts it from the shape of the closed pore template, I used the server PoreWalker (Pellegrini-Calace, Maiwald, & Thornton, 2009). PoreWalker calculates the diameter of the pore along 1 Angstroms steps in the x axis and creates a pore profile (Figure 5.7). The frog KCNQ1 closed pore was used as a reference for the representative model of resting VSD, closed pore KCNQ1. It is expected that the profile of the pore between these two would be similar. The frog KCNQ1 structure has a pore diameter of 2 Angstroms at an x coordinate of 40 near the opening of the selectivity filter. Near the internal pore at an x coordinate of 10 the pore diameter is also 2 Angstroms indicating that it is likely representative of a closed pore (Figure 5.7, A-B). The top scoring model of the S20 series was used as a representative model of the S20 series. The S20 closed pore model has a pore diameter of 2 Angstroms at an x coordinate of 20 near the opening of the selectivity filter. The internal pore for this model is near an x coordinate of -10 where there is a pore diameter of 2 Angstroms (Figure 5.7, C-D). This indicates that the pore profile for the frog KCNQ1 structure and the S20 model are similar and that both are likely representative of a closed pore.

KCNQ1 with an active VSD and an open pore was modeled (modeling not included here), and was used to compare the open pore profile to the closed pore profiles. For the open pore model of KCNQ1, the opening of the selectivity filter is near an x coordinate of 15 and has a pore diameter of 2 Angstroms. However, the internal pore is located near an x coordinate of approximately -18 where the pore diameter changes from 3 Angstroms to 4 Angstroms indicating it is an open pore (Figure 5.7, E-F).

Next, I calculated distances between residues near the opening of the selectivity filter (Ts) and near the internal pore (S339). The calculated distance between the S339s in frog KCNQ1 was 4.3 Angstroms (Figure 5.8, B). For the representative S20 model, the distance between the S339s was 3.9 Angstroms (Figure 5.8, D), only 0.4 Angstroms closer than the frog KCNQ1 structure. The open pore model of KCNQ1 had a distance of 12.7 Angstroms between the S339s (Figure 5.8, F), which is substantially higher than the distances between S339s in the closed pore conformations.

In conclusion, I have created a set of models, S20 series, of KCNQ1 with a resting VSD and closed pore using RosettaCM. Model building utilized several templates simultaneously, and transmembrane backbone RMSD calculations verify that the final models are not biased by the conformation of any one of the templates (Figure 5.2). External servers for MolProbity and ProCheck were used to evaluate the quality of the models (Figure 5.3). The overall MolProbity scores of the models were comparable to the score of the frog KCNQ1 structure. PDBSum found that the models all had above 93% of residues in the most favored regions, and above 5.5% of residues in the allowed regions. The server PoreWalker verified that the pore profile of the models reflected that of the pore profile from a structure of the frog KCNQ1 closed pore (Figure 5.7). Models all contained a distance between E1 and R1 of 3.4 Angstroms (Figure 5.3) supporting experimental evidence of an interaction between the two residues involved in gating charges.

In the future, this model will be used to conduct docking studies of KCNE to KCNQ1 in order to gain insight on the modulation of KCNE to KCNQ1. Previous models of KCNQ1 failed to include the interaction between E1 and R1 as only recently the interaction has been elucidated experimentally. Additionally, the recent structure of frog KCNQ1 provided an excellent template for the closed pore as well as the S0 helix and the intracellular loop between S2 and S3. For these reasons, the models presented herein are the best models to move forward with in future studies until additional experimental evidence is reported.

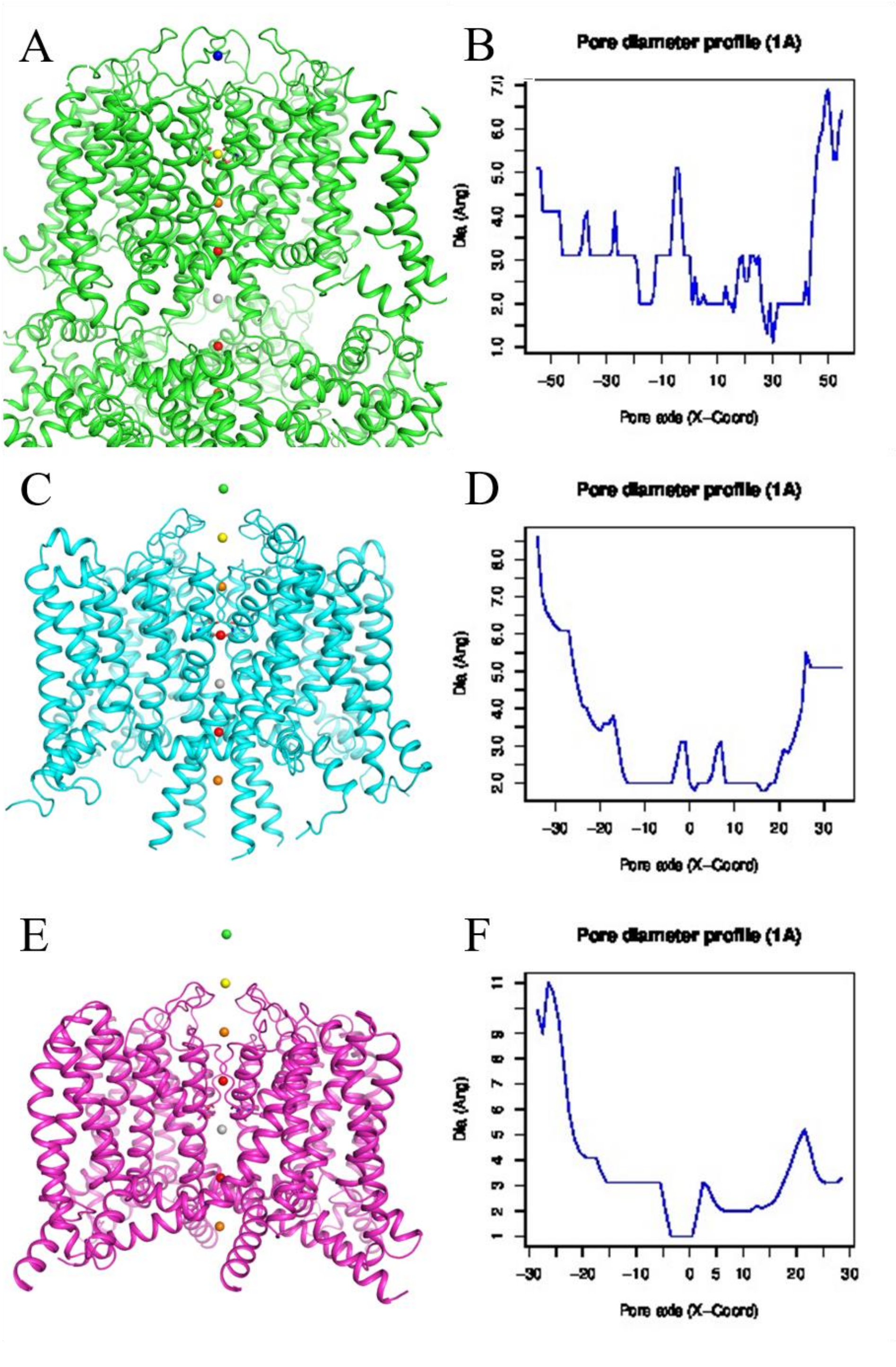


Figure 5.7. Profile of the central pore from PoreWalker. Frog KCNQ1 (A-B), a representative resting/closed state model of KCNQ1 (C-D), and a representative active/open state mode of KCNQ1 (E-F) were analyzed using the server PoreWalker. The gray sphere is at the x coordinate of zero. Each sphere above the gray sphere is in a +10 interval, and each sphere below is in a -10 interval.

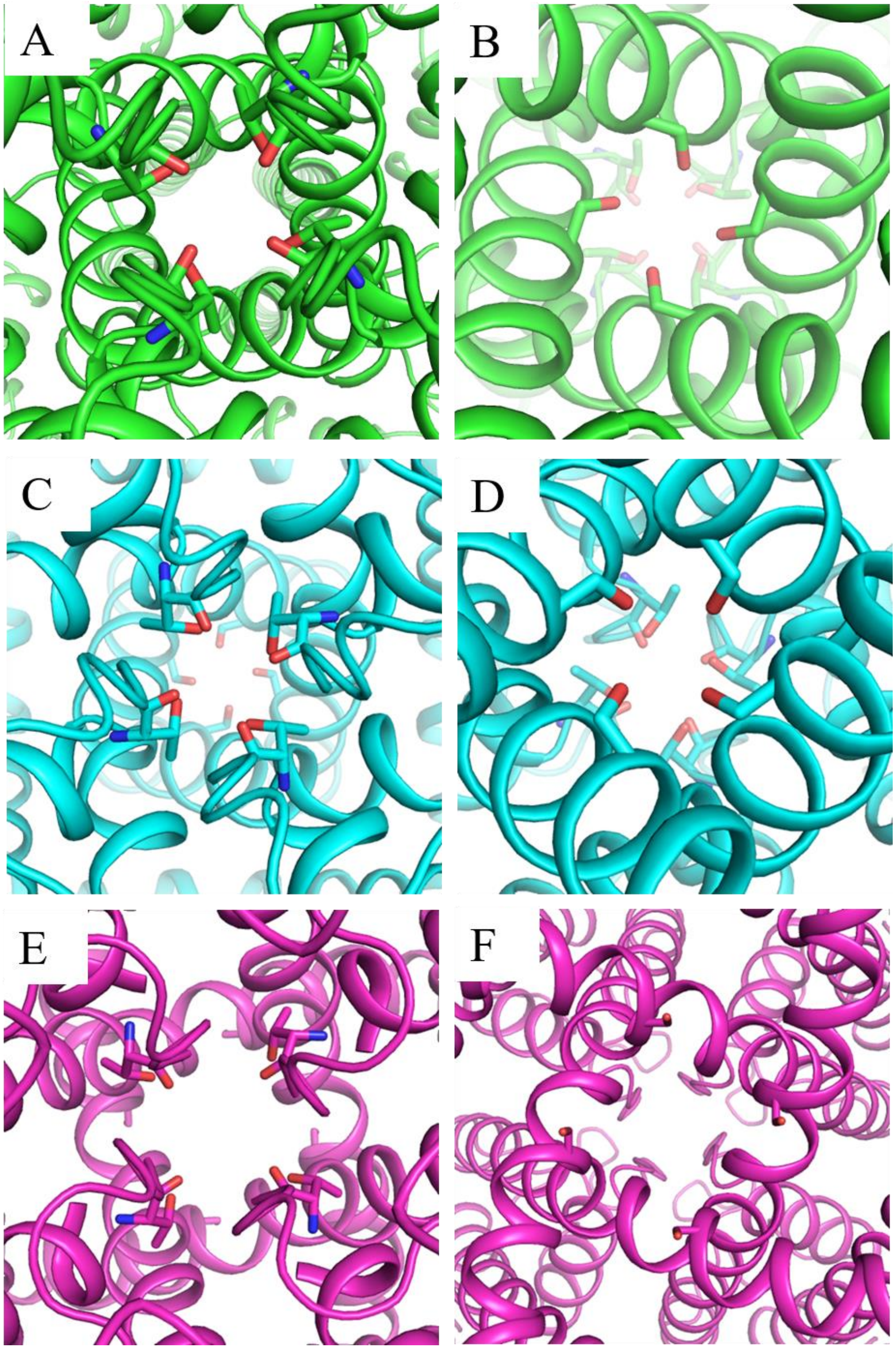


Figure 5.8. Top-down view of TX residues in frog KCNQ1 closed pore (A), representative model of KCNQ1 closed pore (C), and a representative model of KCNQ1 open state (E). Bottom-up view of SX reisdues in frog KCNQ1 (B), representative model of KCNQ1 closed pore (D), and a representative model of KCNQ1 open state (F).

CHAPTER 6

INVERTED TOPOLOGIES IN MEMBRANE PROTEINS

This chapter contains published content from:

Duran and Meiler, 2013.

Author contribution: I created all of the figures and tables for the manuscript entitled “Inverted Topologies in Membrane Proteins”, an open access article in the journal Computational and Structural Biotechnology Journal (Duran & Meiler, 2013). I also wrote the manuscript under the mentorship of Jens Meiler.

Author note: This review has been included as a chapter in order provide the background necessary for Chapter 7.

6.1 Pseudo-symmetry in proteins

Helical membrane proteins such as transporters, receptors, or channels often exhibit structural symmetry. Symmetry is perfect in homo-oligomers consisting of two or more copies of the same protein chain. Intriguingly, in single chain membrane proteins, often internal pseudo-symmetry is observed, in particular in transporters and channels. In several cases single chain proteins with pseudo-symmetry exist, that share the fold with homo-oligomers suggesting evolutionary pathways that involve gene duplication and fusion. It has been hypothesized that such evolutionary pathways allow for the rapid development of large proteins with novel functionality. At the same time symmetry can be leveraged to recognize highly symmetric substrates such as ions. For helical transporter proteins with an inverted two-fold pseudo-symmetry, the symmetry axis lies in the membrane plane. As a result, the putative ancestral monomeric protein would insert in both directions into the membrane and its open-to-the-inside and open-to-the-outside conformations would be structurally identical and iso-energetic, giving a possible evolutionary pathway to create a transporter protein that needs to flip between the two states.

Pseudo-symmetry in soluble proteins

In the realm of soluble proteins, ten folds are over-represented and dominate the structures determined so far experimentally in the Protein Data Bank (PDB)(Berman et al., 2000). Such common ‘superfolds’ in proteins likely exist because nature evolved existent protein folds as opposed to generating new folds (Brych et al., 2004). Six of these ten superfolds display pseudo-symmetry, i.e. can be seen as a repeat of usually two or more copies of nearly identical structural subunits. These folds are: Ferredoxin fold, β-trefoil, up-down bundle, immunoglobulin fold, jelly-roll, and the TIM-barrel fold (Söding & Lupas, 2003).

The TIM-barrel fold is a repeat of eight β- strand-α-helix units where the eight β-strands form an inner barrel surrounded by the eight α-helices. Close inspection of the hydrogen bonding pattern in the barrel reveals that the fold is a 4-fold symmetric arrangement of β-strand-α-helix-β-strand-α-helix units (Söding & Lupas, 2003). Many enzymes share this (βαβα)4 fold some recognizing pseudo-symmetric substrates. Similarly, four-helix bundles with C2 and C4 symmetry are commonly seen as homo-dimers and homo-tetramers (Söding & Lupas, 2003).

It has been postulated that symmetry at the fold level evolved via gene duplication and fusion events from homo-oligomeric proteins (McLachlan, 1972; Rapp, Granseth, Seppälä, & von Heijne, 2006) (Figure 6.1). Fusion of monomer units into a single domain increases thermodynamic stability and kinetic foldability (Wolynes, 1996). Gene duplication is thought to relieve selective pressure which allows for diversification of the subunits on the sequence level before and/or after the fusion event (Figure 6.1) to achieve more complex biological functions

(Söding & Lupas, 2003). As different mutations occur in the two copies of the gene, the evidence of symmetry is masked at the level of the primary sequence. It is assumed that this strategy is one route to evolve large proteins with complex functions rapidly in nature. At the same time symmetry is explored as an avenue for rational or computational design of large protein domains (Blaber, Lee, & Longo, 2012; Gerlt, 2000).

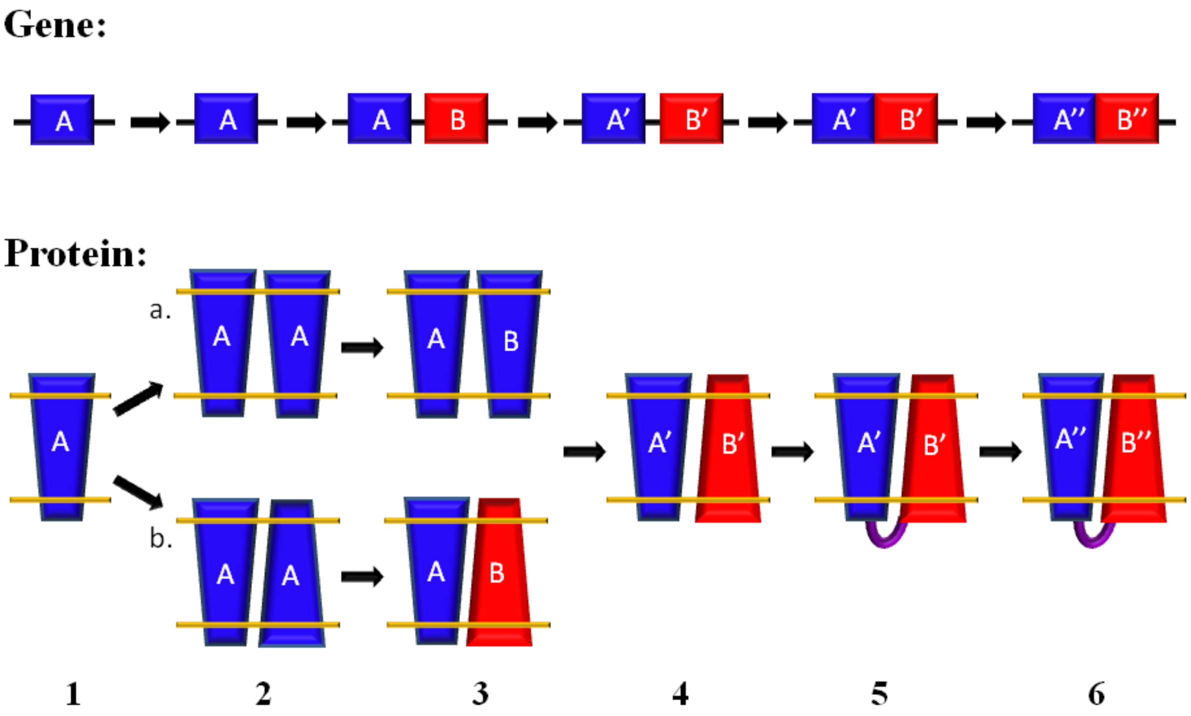


Figure 6.1 Proposed evolutionary pathway for membrane proteins with inverted symmetry involving the gene duplication and fusion hypothesis. Step 1. Prior to a gene duplication event, gene A exists as a singular gene. Step 2. The translation product of the gene, protein A, has an odd number of trans-membrane spans, and has a preferred orientation (no dual topology, 2a) or is attracted to itself and exhibits dual-topology (2b). Step 3. A gene duplication event occurs to produce sequence identical genes A and B, which are composed of the same sequence (3ab). Step 4. Both gene A and B acquire mutations independently of each other resulting in genes A' and B'. For path a, mutations cause a switched in protein's B bias to insert into the membrane resulting in proteins of opposite topology. For path b, this means mutations have stabilized each protein in its respective topology. Step 5. Related genes A' and B' undergo a gene fusion event and are connected by a loop (green). Step 6. Additional mutations cause further sequence divergence resulting in a protein with homologous subunits A'' and B''.

6.2 Self-attraction and self-association of protomers

For gene duplication and fusion as a viable strategy to create large protein domains, interaction of a protein with itself, self-attraction, is a prerequisite. And indeed, homo-oligomers are abundant in the Protein Data Bank (PDB). Homo-oligomers are more stable and therefore more prevalent, as they tend to have a lower energy than their hetero-oligmeric counterparts (André et al., 2008). There are two basic ways in which a protein can be attracted to itself. The first type of self-association is where the same faces of the protein are attracted to each other and form the dimerization interface. The remaining faces are left and can interact with similar remaining faces to form larger oligomers. The second, less common form of self-association occurs when two different faces are attracted to each other. This creates a cyclic oligomeric structure (Figure 6.2) (Levy, Boeri Erba, Robinson, & Teichmann, 2008).

Interestingly, the majority of homo-dimeric complexes in the PDB exhibit a symmetric arrangement of the two protomer units. In this arrangement all interactions between the two protomers are duplicated which halves the total number of unique interactions that are possible. This causes a bias towards very-low-energy symmetric homo-dimeric complexes. With one patch of the protein interacting with the same patch of another copy, such arrangements are evolutionarily stuck in dimeric symmetry as continued evolution into homo-oligomers with higher-order cyclic symmetry requires interaction of two distinct patches (Figure 6.2).

Nevertheless, cyclic symmetry while less frequent is still observed on the homo-oligomeric level. Starting from these cyclic homo-oligomeric proteins internal cyclic symmetry can evolve via gene duplication and fusion. The TIM-barrel and β-propeller superfolds are prominent examples (Söding & Lupas, 2003). However, applying the gene duplication and fusion hypotheses to the study of membrane protein evolution has proven difficult due to sparseness of membrane protein structures.

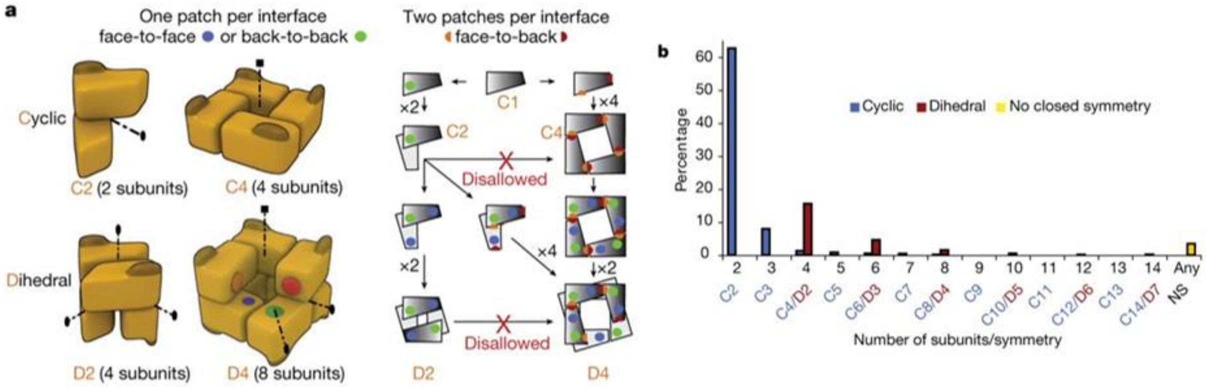


Figure 6.2. Assembly of protomers into oligomers. Assembly can be organized in a cyclic or dihedral manner. Symmetry axes are represented by the dotted lines where two-fold are labeled with ellipses and four-fold are labeled with squares. Cyclic arrangement allows for face-to-back contacts between protomers while dihedral arrangement allows for additional interface contacts between protomers (2a). Cyclic assembly is the overall most common type of arrangement; however, dihedral is common in tetramers (2b). Reprinted by permission from Macmillan Publishers Ltd: Nature, advance online publication, 18 June 2008 (doi: 10.1038/sj.Nature.06942)

6.3 Sparseness of membrane protein structures complicate determination of evolutionary pathways

One of the biggest limiting factors in studying membrane protein topology and symmetry is the small number of membrane protein structures that have been determined (Shimizu, Mitsuke, Noto, & Arai, 2004). Currently, only 289 unique helical membrane protein structures are available (White). These represent only about 120 distinct folds i.e. structurally distinct arrangements of two or more trans-membrane helices. On the other hand, analysis of sequence databases reveals 1,200 families of proteins with more than one predicted trans-membrane helix. These families are distinct in the sense that no inter-family homology can be detected on the sequence level (Hopf et al., 2012). While some of these families might turn out to share a fold on the structural level, this result also implies that many membrane proteins of unknown topology remain to be determined. During the past five years between five and ten novel membrane protein topologies have been determined per year. However, many more structures will need to be determined before the evolutionary pathways are better supported and understood.

6.4 Internal repeat symmetry in monomeric membrane proteins

Symmetry in proteins can improve stability and aids in overcoming energy hurdles in conformational change pathways (Hoang, Trovato, Seno, Banavar, & Maritan, 2004; Wolynes, 1996). In some cases, internal repeat symmetry (IRS) can be detected by sequence analysis. However, because the sequence of membrane proteins evolves quickly, IRS is often only confirmed after the structure of the protein has been determined (S. Choi, Jeon, Yang, & Kim, 2008; Khafizov, Staritzbichler, Stamm, & Forrest, 2010). IRS is hypothesized to originate from gene duplication events or by fusion of similar subunits (S. Choi et al., 2008). In a 2008 study by Choi and coworkers, it was found that almost half of known α-helical membrane proteins have internal repeat symmetry. Types of symmetry include n-fold rotational or cyclic symmetry and inverted symmetry. As the symmetry is only present at the structural level but not at the sequence level it is often referred to as pseudo-symmetry (Forrest et al., 2008).

6.5 The lipid environment restricts the fold space for membrane proteins

For membrane proteins, the lipid environment restricts conformation (J U Bowie, 2001). Along with symmetry, and self-association, these observations have a number of important consequences for membrane protein topology: homo-dimeric proteins with a symmetric arrangement of the two protomer units can align their symmetry axis either parallel or orthogonal to the membrane normal (Granseth, 2010) (Figure 6.3). Higher-order (larger than two) homo-oligomers with cyclic symmetry can only embed into the membrane with the symmetry axis parallel to the membrane normal, i.e. orthogonal to the two-fold symmetry axis of the membrane in the membrane plane. Any other arrangement would break the symmetry in the homo-oligomer. In consequence, we observe two major classes of homo-oligomeric membrane proteins and resulting pseudo-symmetric membrane proteins when considering alignment with respect to the membrane.

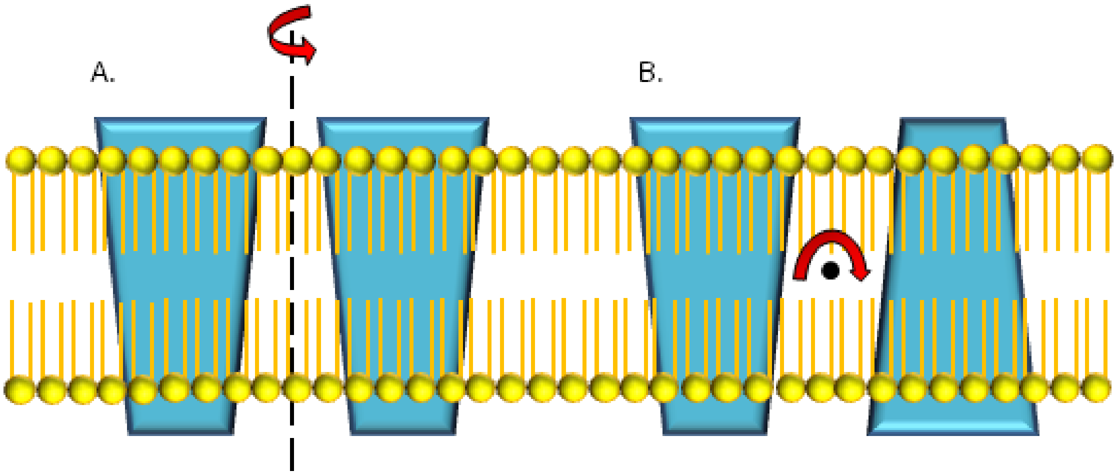


Figure 6.3. Symmetry axes for membrane proteins. The rotational symmetry axis can either be parallel to the membrane normal and orthogonal to the membrane plane (3a). The axis can also be orthogonal to the membrane normal and parallel to the membrane plane. When rotated 180° along this axis, the resulting structure will resemble the starting structure.

Symmetry axis parallel to membrane normal and orthogonal to membrane plane

Proteins embedded in the outer membrane often form β-barrels. These can be seen as cyclic repeats typically consisting of 3-10 ββ-hairpins with the pseudo-symmetry axis parallel to the membrane normal (Remmert, Biegert, Linke, Lupas, & Söding, 2010). β-barrel monomers are also known to assemble into higher order oligomers. For example, cholesterol-dependent cytolysins are capable of forming aqueous pores that consist of up to fifty monomers (Ramachandran, Tweten, & Johnson, 2004). However, approximately 70% of the unique membrane protein structures are α-helical including receptors, transporters, and channels (White). A variety of homo-oligomeric and pseudo-symmetric proteins are observed with the symmetry axis parallel to the membrane normal. For example, the single trans-membrane span glycophorin (M a Lemmon et al., 1992) forms a homo-dimer, diacylglycerol kinase (Vinogradova, Badola, Czerski, Sönnichsen, & Sanders, 1997) forms a homo-trimer, voltage-gated potassium channel (Doyle, 1998) forms a homo-tetramer, and several eukaryotic ABC transporters such as TM287/288 (Hohl, Briand, Grütter, & Seeger, 2012) form a hetero-dimer. Note, that all N- and C-termini of the protomers will always assemble on the same side of the membrane, i.e. the protomers insert into the membrane in the same direction. These homo-oligomers typically follow the positive-inside rule which states that positively charged residues Arginine and Lysine tend to face towards the inner leaflet of the membrane (G von Heijne & Gavel, 1988). Most membrane proteins have such a well-defined orientation based on the distribution of positively charged residues. Resulting homo-oligomers have C2, C3, or C4 symmetry with the rotational axis of symmetry parallel to the membrane normal (Blaber et al., 2012; Granseth, 2010). For example, single chain voltage gated sodium channels exist in humans that resemble the homo-tetrameric structure of the bacterial voltage-gated sodium channel NavAB (Payandeh, Scheuer, Zheng, & Catterall, 2011). For these proteins to evolve into a single chain, monomeric membrane proteins require an even number of trans-membrane spans to satisfy the gene duplication and fusion hypothesis.

Symmetry axis orthogonal to membrane normal and parallel to membrane plane

In the inverted symmetry scenario, protomers insert into the membrane in opposite directions (Granseth, 2010). This arrangement is only feasible for homo-dimers as for higher-order oligomers the symmetry would be broken by the 2-fold symmetry of the membrane when ignoring the differences between inner and outer leaflet in natural membranes. N- and C-termini of the protomers are on opposite sides of the membrane, respectively. Examples of proteins with inverted pseudo-symmetry are the glycerol facilitator channel (James U Bowie, 2006; S. Choi et al., 2008; D. Fu, 2000), the leucine transporter (Yamashita, Singh, Kawate, Jin, & Gouaux, 2005), and the urea transporter (Levin, Quick, & Zhou, 2009). They contain an odd number of helices in the symmetric unit. In some cases, half helices or re-entrance loops are observed which will meet with its symmetric counterpart at the middle of the membrane. An odd number of trans-membrane spans is required for the gene duplication and fusion hypothesis to be a possible evolutionary route to pseudo-symmetric monomeric proteins.

6.6 Effect of inverted symmetry on transporter proteins with open-to-the-inside and -outside conformations

Sequence conserved regions of proteins are referred to as internal repeat cores (IRC) and are typically found at the symmetric interface. It has been proposed that this region is the most conserved because of the self-attractive interactions needed in stabilizing the two symmetric subunits and the role it has in the two-state conformational switch for the inactive and active transport of molecules (S. Choi et al., 2008). Interestingly, inverted pseudo-symmetry is particularly frequent in transporter proteins which can be explained with the necessity of having at least an open-to-the-inside and an open-to-the-outside conformation in an alternate access mechanism of transport. For example, LeuT has an inverted internal repeat of five trans-membrane helices. The inverted structural symmetry inherently creates a channel with a symmetric pathway across the membrane because the structurally symmetric units are placed opposite of each other. The perfectly symmetric structure can be leveraged to create structurally identical and iso-energetic inward and outward facing conformations so that no major energy barriers would need to be overcome to transport substrate across the membrane. As a transporter, the symmetric pathway helps form inward and outward conformations (Forrest & Rudnick, 2009). With core functional residues conserved, chemically similar residues and structures are on either side of the membrane, enabling bidirectional transport of molecules across the membrane (S. Choi et al., 2008).

6.7 Sparseness of membrane protein homo-dimers with inverted symmetry

Despite the abundance of membrane proteins with inverted two-fold pseudo-symmetry, homo-dimers with inverted symmetry seem to be rare. Formation of these requires "dual topology", i.e. the ability for a single subunit to exist in both orientations in the same membrane and environmental conditions (Granseth, 2010; Gunnar von Heijne, 2006). The existence of proteins capable of dual topology is heavily debated, with one of the best studied examples being the homo-dimeric efflux-multidrug transporter from *Escherichia coli* (*E. coli*), EmrE (Granseth, 2010). A recent NMR study suggests that EmrE is able to exist in either orientation as both states are energetically similar (Morrison et al., 2012). EmrE with an even number of trans-membrane spans cannot readily undergo gene duplication and fusion, i.e. it is evolutionarily frustrated.
In 2006, Rapp et al. proposed five proteins that have potential as proteins capable of dual topology. These proteins are small, composed of four trans-membrane spanning helices, and have very little positive charge bias (Rapp et al., 2006). It makes sense that a protein with dual topology would be small to act as a unit of symmetry and have very little positive charge bias to readily be placed in either orientation in the membrane without disobeying the positive-inside rule (G von Heijne & Gavel, 1988).

Additionally, an overall neutral charge causes both topologies to be similar in energy (Crisman, Qu, Kanner, & Forrest, 2009; Forrest et al., 2008). To further understand the significance of a negligible positive charge bias in dual topology, membrane proteins with a positive charge bias of nearly zero were engineered to have a distinct bias. The engineered bias caused a flip in orientation for these proteins (Rapp et al., 2006). In an evolutionary route over time, mutations to a fused dual-topology protein could essentially lock in a particular topology while maintaining functionally important residues.

6.8 Dual topology is not required for evolution of membrane proteins with inverted two-fold pseudo-symmetry

The apparent sparseness of homo-dimers with inverted symmetry seems to be at odds with the abundance of membrane proteins with inverted two-fold pseudo-symmetry. However, it is important to note that a homo-dimer with inverted symmetry is not a prerequisite for the evolution of a membrane protein with inverted two-fold pseudo-symmetry. Consider the following putative evolutionary pathway (Figure 6.1): a membrane protein gene with preferred orientation in the membrane gets duplicated. In one copy mutations occur that change the preferred orientation within the membrane. An interaction between the two proteins evolves that because of their similarity is still likely to be pseudo-symmetric. At this time the protein develops its transport functionality. A gene fusion event creates the inverted two-fold pseudo-symmetric protein.

In this context a 2006 study by Rapp et al. used E. coli membrane proteins and anti-parallel hetero-dimer pair YdgE and YdgF as examples of homologous proteins with different positive charge biases and opposite orientations. E. coli proteins YdgE and YdgF are overlapping genes on the chromosome, but are expressed separately (Rapp et al., 2006). YdgE is known to consist of four trans-membrane spans whereas YdgF is predicted to consist of four (Drew et al., 2002). YdgQ and YdgL are another example of a homologous gene pair in E. coli that results in proteins with opposite orientations. For both of these pairs of proteins, each protein has a positive charge bias favoring its respective orientation (Rapp et al., 2006). Because of the opposing orientations, each homologous pair is able to form an anti-parallel hetero-dimer. These anti-parallel hetero-dimers are likely the result of gene duplication and topology evolution events (Lolkema, Dobrowolski, & Slotboom, 2008).

Rapp et al. suggested five dual topology possibilities (Table 6.1). Two pairs of homologous hetero-dimers which form anti-parallel topologies are also included in this table. Positive charge bias was calculated similarly to Rapp et al. where counts of K and R in the even loops are subtracted from the odd loops, where the N-terminal loop is loop 1.

Table 6.1 Dual topology and anti-parallel hetero-dimer candidates.

Protein	**Predicted Topology**	**TM Spans**	**Positive Charge Bias**
EmrE	Dual	4	−2
SugE	Dual	4	−1
CrcB	Dual	4	0
YdgC	Dual	3	1
YnfA	Dual	4	0
YdgE	Anti-parallel hetero-dimer	4	7
YdgF	Anti-parallel hetero-dimer	4	6
YdgQ	Anti-parallel hetero-dimer	05–06	−6
YdgL	Anti-parallel hetero-dimer	4	−7

6.9 Gene duplication and fusion as it applies to monomeric membrane proteins

In a 2008 study, Lolkema et al. studied the DUF606 family to get clues for the possible order of evolutionary events. In the DUF606 family, there exist homo-dimeric proteins, hetero-dimeric proteins, and two-domain fusion proteins which are proposed to be indicative of single gene dual topology proteins, homologues of opposite orientations, and fused genes creating an anti-parallel topology, respectively (Lolkema et al., 2008). They found no existing fused homo-dimeric protein in the DUF606 family as evidence for direct fusion of duplicated genes. The evolutionary pathway that was proposed as a result of this study involved a gene duplication event followed by sequence divergence and finally a gene fusion event between the homologues. Previously, it has been suggested that one of the likely evolutionary routes begins with gene duplication shortly followed by gene fusion. Following fusion, divergence further stabilizes the energetics, anti-parallel topology, and function (Granseth, 2010). However, evidence of any fused homo-dimer have not yet been found (Lolkema et al., 2008). Yet, extensive divergence prior to a fusion event would seem to affect the self-attraction between the two domains. Therefore, in Figure 6.1, we propose a slightly modified version of the alternative evolutionary route for inverted membrane protein topologies. First, the gene capable of dual topology is duplicated by the appropriate evolutionary mechanism, a gene duplication event. Next, the domains of the homo-dimers are stabilized into opposite orientations by mutations which stabilize the overall anti-parallel topology. Then, the similar domains undergo fusion followed by even further sequence divergence to stabilize structure and improve function. However, there is currently insufficient evidence to support one route over the other.

Major Facilitator Superfamily

The major facilitator superfamily (MFS) transporters have been extensively studied for their symmetry. Proteins in the MFS transporter family are composed of 12 trans-membrane spans. The sequence homology between other members of this family is weak; however, proteins in this family are structurally similar (Reddy, Shlykov, Castillo, Sun, & Saier, 2012). There have been differences in opinion for the breakdown in symmetry. In 2012, a review proposed that the smallest symmetric unit is a two trans-membrane spanning domain (Reddy et al., 2012). This would mean that there is three-fold symmetry within the six helix bundles and then an additional two-fold inverted symmetry for the six helix bundles. However, previous studies have supported the idea of a three trans-membrane spanning structural motif resulting in two-fold symmetry in the six helix bundle (Hirai, Heymann, Maloney, & Subramaniam, 2003). Recently in 2013, Madej et al. conducted an experiment where the symmetry motifs in MFS protein L-fucose H+ symport protein FucP were rearranged (Madej, Dang, Yan, & Kaback, 2013). The result was a structure strikingly similar to LacY, another member of the MFS. The conclusion was that FucP and LacY likely evolved from the same primordial helix triplets, but the order of assembly of these structural motifs into larger proteins differed which created an avenue for diversity in function (Madej et al., 2013).

Neurotransmitter Sodium Symporters

Neurotransmitter sodium symporters are also a type of transporter proteins which display internal symmetry. The most well-known examples display two-fold pseudo-symmetry and include the glutamate transporter (GltPh), the sodium and proton antiporter (NhaA), and the leucine transporter (LeuT) (Krishnamurthy, Piscitelli, & Gouaux, 2009). A rocker switch mechanism of transport favors the internal two-fold symmetry because the conformation is easily exchanged (Forrest & Rudnick, 2009; Krishnamurthy et al., 2009). In LeuT, the trans-membrane spans 1-5 are symmetric to 6-10. LeuT can be considered an occluded state that can convert into outward and inward facing conformations due to the internal symmetry. Furthermore, the addition of non-symmetric helices act as hinges to promote conformational change during transport (Forrest & Rudnick, 2009; Forrest et al., 2008).

Aquaporins

The aquaporins are another type of membrane protein that exhibit pseudo-symmetry. In fact, the very first high resolution example of inverted topology was from E. coli's aquaglyceroporin the glycerol facilitator protein (GlpF) (James U Bowie, 2013). Aquaporins are a great example of symmetry observed on a single polypeptide chain. They are made up of six trans-membrane spanning helices and two half-spanning helices with the symmetric unit being three and a half helices. The α carbon root mean square deviation between the two halves of GlpF is 1.8 Angstroms (S. Choi et al., 2008). Channel proteins' primary function is the transport of water and small molecules across the membrane. Inverted symmetry is advantageous for the formation of a symmetric pathway across the channel (Forrest & Rudnick, 2009). However, because transport through the channel is permeation instead of a two-switch conformational change, the advantage of inverted symmetry is largely for stability of the protein. For this reason, channels are sometimes referred to as broken transporters (Forrest & Rudnick, 2009). GlpF and other aquaporins have an aspartic acid-proline-alanine motif seen in both halves at the symmetric interface (S. Choi et al., 2008; D. Fu, 2000; Stroud et al., 2003). In this example, stability is improved because of the interaction between the proline rings on either half (Daxiong Fu et al., 2000).

Chloride Channel

Another type of channel protein that exhibits symmetry is the ClC chloride channel (James U Bowie, 2013; Dutzler, Campbell, Cadene, Chait, & MacKinnon, 2002). A single subunit in the homo-dimeric complex is made up of 18 helices. Eight helices on the N-terminal half display striking inverted two-fold pseudo-symmetry with the C-terminal half. Like the aquaporins, the anti-parallel structure is useful for this channel protein because the symmetric polar ends of helices are able to face the outside of the membrane. This is energetically favorable in that the polar ends are not buried in the membrane (Dutzler et al., 2002). Interestingly, another ion channel, the potassium channel, does not take advantage of anti-parallel topology. The potassium channel works very differently in that the cavity widens near the center of the membrane. The helix dipoles are also positioned very differently, in a parallel fashion, to help overcome the dielectric barrier which is the nature of the membrane. However, in this anion channel, the anti-parallel topology creates a selectivity filter for chloride ions. It is predicted that the reason for this vast difference in topology is because hydrophobic anions partition into membranes much more readily than hydrophobic cations, so channels transporting cations would need a much larger cavity to stabilize the cation (Dutzler et al., 2002).

6.10 Effect of lipid composition on membrane protein topology

A factor largely ignored in this review is lipid composition and differences between inner and outer leaflet of the membrane (Vitrac, Bogdanov, & Dowhan, 2013). In 2013, Vitrac and colleagues found that when the composition of phosphatidylethanolamine (PE) was varied in a lipid environment, proteins were capable of complete inverted topology. Native and inverted conformations of lactose permease (LacY) from E. coli were found to exist in the membrane at the same time. Thermodynamically, dual topology is partially determined based off of the inherent properties of the protein interaction with the lipids in the membrane. Studies both in vitro (Bogdanov, Xie, Heacock, & Dowhan, 2008; Vitrac et al., 2013) and in vivo (Bogdanov & Dowhan, 2012) show that through the manipulation of protein domain charge or lipid composition, dual topological arrangements of a protein can co-exist in the same membrane. It is important to keep in mind that membrane proteins not only evolve with time but in concert with lipid environments that can also affect topology between homologous proteins.

6.11 Overcoming insufficient structural information

Many cases of internal repeat symmetry in membrane protein have been difficult to recognize until after structure determination (S. Choi et al., 2008; Khafizov et al., 2010). Often times, as shown in many of the aforementioned cases, the sequence identity is low because of such extensive sequence divergence despite maintaining structural symmetry. Because it is not feasible to determine the structure for all proteins of interest in order to detect symmetry, other physical properties have been employed to provide additional information towards the prediction of internal symmetry. In particular, hydropathy profiles have recently been used to detect internal symmetry of transporters (Khafizov et al., 2010). Instead of looking at raw sequence similarity, AlignMe (Khafizov et al., 2010; Stamm, Staritzbichler, Khafizov, & Forrest, 2013) takes into consideration the hydrophobicities of amino acids as a tool for alignment. The advantage is that physical properties like hydrophobicites will be more conserved over time and will match proteins that resemble each other chemically. This can improve the ability to detect internal symmetries where structural information is unavailable.

The most obvious limiting factor in understanding more about membrane protein evolution and pseudo-symmetry is the limited number of known membrane protein structures. Table 6.2 displays proteins with detected internal symmetry in Choi et al (S. Choi et al., 2008).

Additional, selected membrane protein structures determined since 2007 were added, when symmetry was obvious. For these, we calculated Cα RMSD for the trans-membrane spanning helices using PyMol (*The PyMol Molecular Graphics System*) software. The OCTOPUS (Viklund & Elofsson, 2008) server was used to determine the location of the loops with respect to the membrane. Here, positive charge bias was calculated by the number of "inside" K and R residues minus the number of "outside" K and R residues. In Figure 6.4, six of these structures were chosen to visualize the symmetry from both side and top views with corresponding trans-membrane helices colored accordingly.

Table 6.2. Internal Symmetry in select membrane protein structures.

Protein Name	**PDB ID**	**#TM spans per unit**	**% Identity**	**Cα RMSD**	**Symmetry Type**	**Membrane Symmetry Axis**	**Positive Charge Bias**
Cytochrome C Oxidase	1OCC[a]	4	15.6	3	C3	Normal	6
Formate dehydrogenase-N	1KQF[a]	2	18.1	3.8	C2	Normal	21
Mitochondrial ADP/ATP carrier	10KC[a]	2	23.5	1.8	C3	Normal	16
Rotor of V-type Na ATPase	2BL2[a]	2	28.9	1.2	C2	Normal	5
Spinach photosystem II	1RWT[a]	1	23.1	3.3	C2	Normal	6
BtuCD vit B13 transporter	1L7V[a]	2.5	21.5	3.4	Inverted C2	Plane	7
Bovine rhodopsin	1U19[a]	3	17.1	4.7	C2	Normal	11
Archaerhodopsin-2	1VGO[a]	3	9.2	4.4	C2	Normal	7
AQP1 water channel	1J4N[a]	3.5	17.6	2.5	Inverted C2	Plane	10

Protein Name	PDB ID	#TM spans per unit	% Identity	Cα RMSD	Symmetry Type	Membrane Symmetry Axis	Positive Charge Bias
Glycerol Facilitator channel	1FX8[a]	3.5	18.5	1.8	Inverted C2	Plane	5
H/Cl exchange transporter	1KPK[a]	5	17.9	2.7	Inverted C2	Plane	23
Amt-I ammonia transporter	2B2F[a]	4.5	11.8	2.3	Inverted C2	Plane	12
LeuTAa leucine transporter	2A65[a]	2.5	17.8	4.5	Inverted C2	Plane	9
AcrB bacterial multi-drug efflux transporter	1IWG[a]	5	16.4	2.1	C2	Normal	16
Nha Na/H antiporter	1ZCD[a]	3	19.5	3.3	Inverted C2	Normal	11
CusA transporter	3K0I	5	21.9 (360 Residues)	3.46	C2	Normal	12
AcrB bacterial multi-drug efflux transporter	2HQF	3	17.9 (218 Residues)	3.679	C2	Normal	15
Phosphate Transporter	4J05	6	21.4 (131 Residues)	3.794	C2	Normal	2
Formate Channel	3KCU	3.5	34.6 (122 Residues)	5.611	Inverted C2	Plane	5
Urea Transporter	4EZC	5.5	27.9 (147 Residues)	1.926	Inverted C2	Plane	8

a Membrane proteins from Choi et al 2007

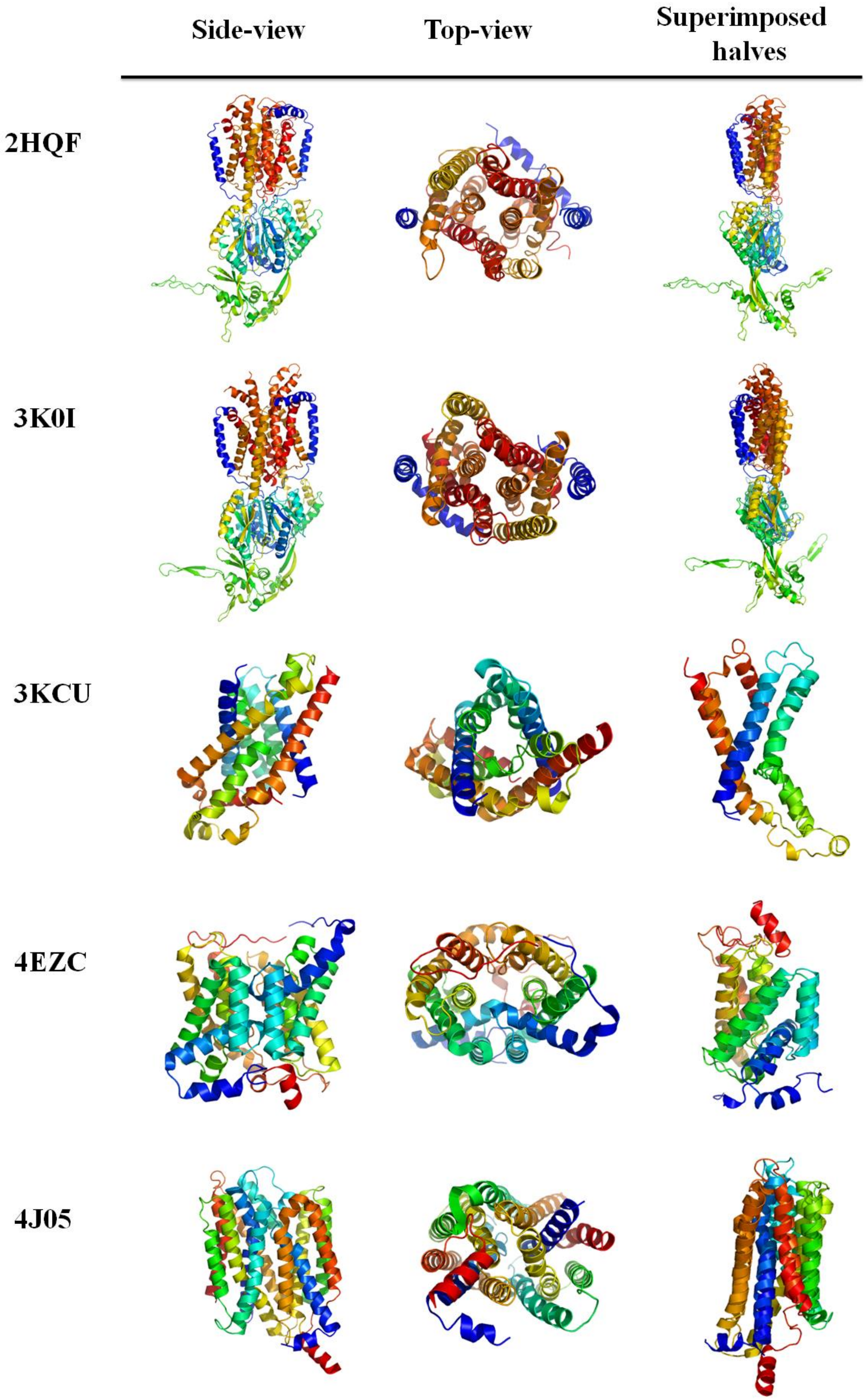

Side-view
Top-view
Superimposed halves
2HQF
3K0I
3KCU
4EZC
4J05

Figure 6.4. Superimposition of pseudo-symmetric halves. Five symmetric membrane proteins since 2007 are shown on the left as a monomer. The middle shows a view of the symmetry from the top. On the right, the pseudo-symmetric halves are superimposed to show the striking structural similarity. Cα RMSD for these proteins can be found in Table 6.2.

In summary, inverted topology in membrane proteins could have evolved via multiple evolutionary routes. While symmetric self-association is known as a stabilizing factor for protein structure, inverted topology within membrane proteins adds an interesting twist to the puzzle as it implies dual topology membrane proteins, i.e. proteins that can insert into the membrane in both directions. However, it is also possible that attraction between the two protomers only evolved after gene duplication and after one copy of the gene underwent mutations that inverted its topology. Such symmetric interactions between almost identical proteins would still be energetically favorable as many residues in the interface would adhere to the symmetry condition. With insufficient evidence to prefer one route over the other, efforts continue to understand how inverted symmetry in membrane proteins evolved.

CHAPTER 7

TOWARDS THE COMPUTATIONAL ENGINEERING AND DESIGN OF A STABLE, SYMMETRIC MEMBRANE PROTEIN

This chapter contains unpublished content from A.M. Duran and J. Meiler.

Author contribution: I designed experiments and analyses under the direction of Jens Meiler. I created scripts to generate engineered proteins by *in silico* circular permutation. I generated all of the computational models, data, and figures in this chapter. Xuan Zhang was trained by me for medium throughput expression screens for membrane protein. The expression tests in this chapter, Tables 7.5, 7.6 and 7.7 were performed by Xuan under my direction.

7.1 Introduction

Most proteins have fundamental folds that classify as one of the ten major superfolds (Thornton, Orengo, Todd, & Pearl, 1999). The existence of so few superfolds is likely due to the thermodynamic stability exhibited in these folds (Brych et al., 2004) despite the large amount of degrees of freedom involved in folding a protein. Common superfolds in proteins likely exist because nature chose to evolve the existing folded proteins as opposed to generating completely new folds (Andrade, Perez-iratxeta, & Ponting, 2001; Brych et al., 2004). Interestingly, six of the ten superfolds exhibit structural symmetry (Blaber et al., 2012; Brych et al., 2004).

Symmetry in proteins aids in overcoming energy hurdles in conformational change pathways and can improve stability (Hoang et al., 2004; Wolynes, 1996). Structurally symmetric proteins are thought to be the result of gene duplication (Brych et al., 2004). In microorganisms alone, nearly 50% of genes are hypothesized to be the result of gene duplication events (Eisenbeis & Höcker, 2010). Although often times silenced (Lynch, 2000), gene duplicates are thought to relieve selective pressure for its intended function and enables diversification for novel function (Söding & Lupas, 2003). The fusion event following duplication restricts translational and side chain entropy, which ultimately stabilizes the resulting protein (Eisenbeis & Höcker, 2010). Over time, sequence divergence diversifies the sequence for additional protein functions while maintaining the overall architecture of the protein through key residue contacts. Because of sequence divergence, it is often difficult to immediately detect proteins that have undergone gene duplication and fusion events. Motifs called internal repeats cores (IRC) help to identify possible duplicated genes. IRCs often are found at the interface of the symmetric units (S. Choi et al., 2008). Proteins that exhibit structural symmetry despite divergent amino acid sequences are proposed to be the result of gene duplication, fusion, and diversification events.

Membrane proteins often have IRCs at the interface between two halves. A study in 2008 found that nearly half of known α-helical membrane proteins were detected to have internal repeat symmetry (S. Choi et al., 2008). Types of structural symmetry include n-fold rotational or cyclic symmetry and inverted symmetry. Despite the abundance of membrane proteins with inverted two-fold pseudo-symmetry, homo-dimers with inverted symmetry seem to be rare. For this to occur, it would require dual topology which is a property where the membrane protein can exist in both orientations in the same membrane and environmental conditions (Crisman et al., 2009; Granseth, 2010).

The dual topology phenomenon is highly controversial. The homo-dimeric efflux-multidrug transporter from *Escherichia coli* (*E. coli*), EmrE has been studied extensively for dual topology (Granseth, 2010). One nuclear magnetic resonance study suggested that EmrE is capable of existing in either orientation because they are energetically similar (Morrison et al., 2012). Because EmrE has four trans-membrane spans, it would not readily create an inverted, symmetric topology. Small membrane proteins with an odd number of helices and little positive charge bias make ideal candidates for dual topology (Rapp et al., 2006) as they can be oriented in the membrane without disobeying the positive-inside rule, and because the topologies have similar energies (Crisman et al., 2009; Forrest et al., 2008). The positive-inside rule is described where positively charged amino acids, Lysine and Arginine, tend to be positioned towards the inside of the cell versus in the membrane and outside of the cell (G von Heijne & Gavel, 1988). To examine this, one study engineered a protein with a charge bias of nearly zero to be charged which resulted in a flip in the orientation of the protein in the membrane (Rapp, Seppälä, Granseth, & von Heijne, 2007). In evolution, it is possible that a small amount of mutations could create a preferred topology for these dual topology proteins (Duran & Meiler, 2013) giving rise to the symmetric, inverted topologies we see today (Schuldiner, 2009).

Pseudo-symmetry has been described as a protein architecture that is mostly symmetric in structure, but asymmetric in sequence (Forrest et al., 2008). Recent advances in structural techniques have revealed that many membrane protein structures exhibit inverted pseudo-symmetry. If gene duplication and fusion was an evolutionary route for inverted topologies, then monomers are presumed to have inserted into the membrane in both orientations and assembled in an anti-parallel manner, implicating a possible dual-topology.

Aquaporins are a type of membrane proteins that have been shown to exhibit pseudo-symmetry (Duran & Meiler, 2013). Aquaporins are channel proteins whose primary function is the transport of water and small molecules, such as glycerol in the case of glyceroaquaporins, across the membrane. Inverted symmetry is advantageous for the formation of a symmetric pathway across the channel (Forrest & Rudnick, 2009). However, because transport through the channel is permeation instead of a two-switch conformational change, the advantage of inverted symmetry is largely for stability of the protein. Additionally, aquaporins are a great example of symmetry observed on a single polypeptide chain. They are made up of six trans-membrane spanning helices and two half-spanning helices with the symmetric unit being three and a half helices. *E. coli*'s aquaglyceroporin the glycerol facilitator protein (GlpF) (Daxiong Fu et al., 2000) was the first high resolution example of inverted topology (James U Bowie, 2013). For GlpF, the α carbon root mean square deviation between the two halves of the single chain is 1.8 Angstroms and the amino acid sequence identity of the symmetric halves was calculated to be 18.2% (S. Choi et al., 2008).

Protein design that utilizes symmetry and symmetric assembly of proteins is appealing for the design of larger, symmetric complexes. Previously, our lab successfully designed a sequence symmetric variant of a TIM-barrel protein (Fortenberry et al., 2011). The resulting symmetric protein was name FLR. The X-ray crystallographic structure of FLR was within 0.87 Å of the computationally predicted model. The computational approach involved modeling sequence symmetric variants of the imidazole glycerol phosphate synthase (HisF) protein in the molecular modeling software Rosetta and selecting models with the lowest Rosetta energy score. This approach can easily be applied to membrane protein systems because Rosetta has an energy function specifically for modeling membrane proteins (Barth et al., 2007; Yarov-Yarovoy et al., 2006). Because of their high degree of structural pseudosymmetry, aquaporins are an ideal protein with which to engineer sequence and structure symmetric membrane protein variants.

I took several approaches at computationally engineering and designing GlpF using feedback from experimental studies involving the expression and purification of symmetric variants. Details of these approaches can be found in Appendix H. Herein, I used the backbone and sequences from 13 unique aquaporin structures, including a sub-angstrom structure

(Eriksson et al., 2013), to engineer and model thousands of symmetric variants from multiple backbones.

The protein engineered in this study will give insight for whether a membrane protein is capable of dual topology. Additionally, the asymmetric unit of the engineered protein will also be expressed which can test self-attraction and assembly of multiple symmetric membrane protein units (in this case, two units). If the asymmetric unit is seen in both orientations, it would argue against the current proposed mechanisms for insertion and folding of membrane proteins into the membrane.

7.2 Methods

For preparation of structural alignment between the inverted halves, the asymmetric units were created from 12 aquaporin backbones (Table 7.1). To determine where in the sequence to halve each protein, the start and end of each transmembrane (TM) helix were determined (Table 7.2). Two asymmetric units were created for each protein by cutting at a position between TM4 and TM5. For each protein, the two symmetric units were input into the MAMMOTH structural alignment program (Ortiz & Strauss, 2002) to superimpose the asymmetric units in order to acquire the coordinates necessary to create an inverted version of the original protein. The c-alpha RMSD calculated between the two asymmetric units was reported (Table 7.3).

Table 7.1. Aquaporin sequences and backbones used in the construction of engineered proteins.

PDB	Name	Species	Resolution
1FQY	AQP1 red blood cell aquaporin water channel	Homo sapiens	3.8
1FX8	Glycerol facilitator channel	Escherichia coli	2.2
1J4N	AQP1 aquaporin red blood cell water channel	Bos taurus	2.2
1RC2	AqpZ aquaporin water channel	Escherichia coli	2.5
1YMG	AQP0 aquaporin water channel	Bos taurus	2.24
2B6P	AQP0 aquaporin sheep lens junction	Ovis aries	1.9
2D57	AQP4 aquaporin rat glial cell water channel	Rattus norvegicus	3.2
2F2B	AqpM aquaporin water channel	Methanothermobacter marburgensis	1.68
3C02	PfAQP aquaglyceroporin	Plasmodium falciparum	2.05
3D9S	AQP5 aquaporin water channel	Homo sapiens	2.0
3GD8	AQP4 aquaporin water channel	Homo sapiens	1.8
3ZOJ	Aqy1 yeast aquaporin	Pischia pastoris	0.88
4NEF	AQP2 Aquaporin from kidney	Homo sapiens	2.75

Table 7.2. Transmembrane helix starting and ending residues for each backbone. TM3 and TM7 span only half of the membrane.

PDB	TM1	TM2	TM3	TM4	TM5	TM6	TM7	TM8
1FQY	5-26	44-59	70-77	87-108	130-151	160-175	186-193	202-223
1FX8	3-27	37-52	63-72	82-103	139-163	175-190	199-208	228-249
1J4N	12-32	54-72	79-88	96-118	139-159	170-188	195-204	211-233
1RC2	5-25	38-57	64-71	81-104	131-151	161-180	189-196	202-225
1YMG	5-26	38-56	63-72	80-99	122-143	154-172	179-188	195-214
2B6P	9-30	42-61	67-76	84-103	126-147	158-177	183-192	199-218
2D57	5-26	43-58	68-77	86-104	126-147	159-174	184-193	201-219
2F2B	4-27	57-74	83-92	97-122	143-166	174-191	200-209	219-244
3C02	2-24	36-53	64-73	81-103	127-149	159-176	187-196	213-235
3D9S	10-32	45-64	70-79	87-108	128-150	161-180	187-196	203-224
3GD8	3-25	41-60	67-76	84-105	125-147	158-177	183-192	199-220
3ZOJ	11-33	52-73	79-88	95-116	134-156	163-184	190-199	207-228
4NEF	8-30	42-62	68-77	84-106	126-148	158-178	184-193	200-222

Table 7.3. Results of MAMMOTH structural alignment of asymmetric units for each backbone. The residues around the cutpoint for creating asymmetric units is reported along with the resulting c-alpha RMSD, in angstroms, and the number of residues aligned.

PDB	Asymmetric cutpoint	Mammoth align (Å)	Residues aligned
1FQY	120-121	3.95	174
1FX8	131-132	3.98	158
1J4N	127-128	3.74	162
1RC2	117-118	3.91	180
1YMG	111-112	3.09	179
2B6P	118-119	3.4	178
2D57	117-118	3.62	200
2F2B	134-135	3.25	153
3C02	118-119	3.75	199
3D9S	119-120	3.58	193
3GD8	116-117	2.82	190
3ZOJ	127-128	3.72	139
4NEF	118-119	3.8	191

The coordinates from the native protein and the inverted protein were then used to construct symmetric variants. Symmetric variants were engineered using an approach similar to circular permutation. Each helix on the N-terminal side of the protein had a helix on the C-terminal side with which they were superimposed when aligning native to inverted protein. These helix partners will be referred to as symmetric counterparts. The alignments of the symmetric counterparts were trimmed so that the TMs in each pair were of the same length. These residue pairs were used to create a list of all possible cutpoints for creating a symmetric backbone from the native and inverted backbones for each protein. To ensure exhaustive sampling of backbone and allow additional flexibility with the residue pairs, two additional sets of residue pairs were created for plus one and minus one sequence positions to overcome any poor structural alignments.

Cutpoints for assembling the engineered protein from native and inverted structures are best selected when residue pairs are close in space to avoid offsetting the backbone. The c-alpha distance between all determined residue pairs were calculated. Several c-alpha distance values were used as a threshold to determine how many symmetric variants would result from each cutoff. The final threshold for allowing the construction of symmetric variants from residue pairs was 3.5 angstroms (Table 7.4).

Table 7.4. Threshold values for c-alpha distance between residue pairs considered as cutpoints. The resulting number of symmetric variants that would result from all 13 backbones was reported.

Threshold	Number of structures
1	172
1.5	370
2	570
2.5	763
3	1006
3.5	1302
4	1673

The 1302 symmetric backbones were constructed by concatenating the appropriate fragments from native and inverted coordinates (Figure 7.1) using Python. Dualspace relax (Conway et al., 2014) along with RosettaMembrane was used to idealize bond lengths in the gap regions between fragments 20 times for each symmetric backbone. RosettaMembrane and the score_jd2 application were then used to determine the energies of each symmetric variant model. The total Rosetta energy was normalized for comparison across symmetric variants of varying lengths. These energies were compared to the normalized Rosetta energy of their respective native protein. Additionally, the symmetric energy of the native protein was calculated by the sole inclusion of the energy of relevant fragments from the native protein structure that were used to construct the protein. Then the normalized Rosetta symmetric energy of the native protein was subtracted from the normalized total Rosetta energy from the symmetric variant.

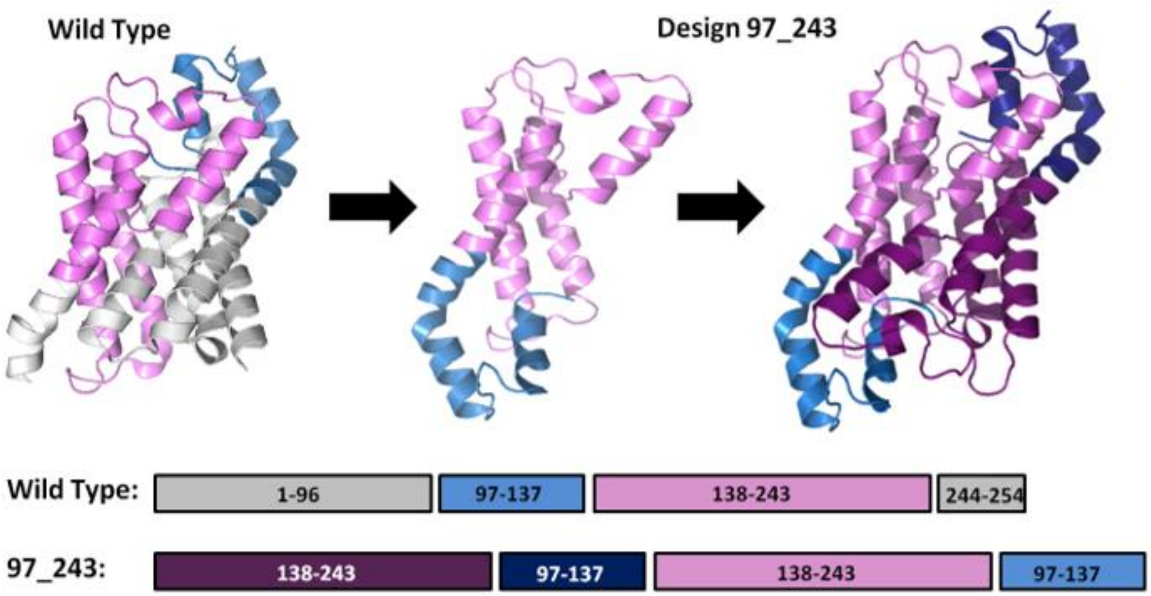


Figure 7.1 Construction of symmetric backbones from native and inverted native structure coordinates. For an engineered GlpF protein with cutpoints at 97 and 243, the start and end of the asymmetric unit are at residue positions 97 and 243, respectively. In order to have a complete helix from a continuous fragment, the new asymmetric unit becomes residues 138-243, 97-137. This asymmetric unit is duplicated to create a symmetric variant backbone.

Normalized total and symmetric energy differences from native were plotted for each symmetric variant. Variants with negative values for both calculations were considered further as candidates for experimental validation. The normalized score for all score terms were plotted to evaluate the driving force behind the negative scores. For backbones based on 1FX8, the Menv_smooth term, a term added to smooth the potential rather than represent physical features, was the driving force. It was determined that the best way to identify ideal candidate symmetric variants was to use a weighted sum of the normalized total and normalized symmetric energy differences. The normalized symmetric energy differences were given a weight of 0.75 while the normalized total energy differences were given a weight 0.25. The symmetric variants with the lowest weighted sum of normalized energy differences were consider as candidates for further analysis.

Favor Native Residue and Favor Symmetric Sequence bonuses were used to explore the sequence space of the mutations. Favor Native Residue bonuses of 0.5, 0.75, 1, 1.25, and 1.5 were used along with a constant Favor Symmetric Sequence bonus of 1 were used to redesign all symmetric variant candidates. The sequence recovery rate was calculated for the best and worse scoring designs, and the sequence recovery rate of the worst was subtracted from the sequence recovery rate of the worst best design for each backbone. Backbones with a difference in sequence recovery rates that were positive across the various bonuses were selected as candidates for experimental studies.

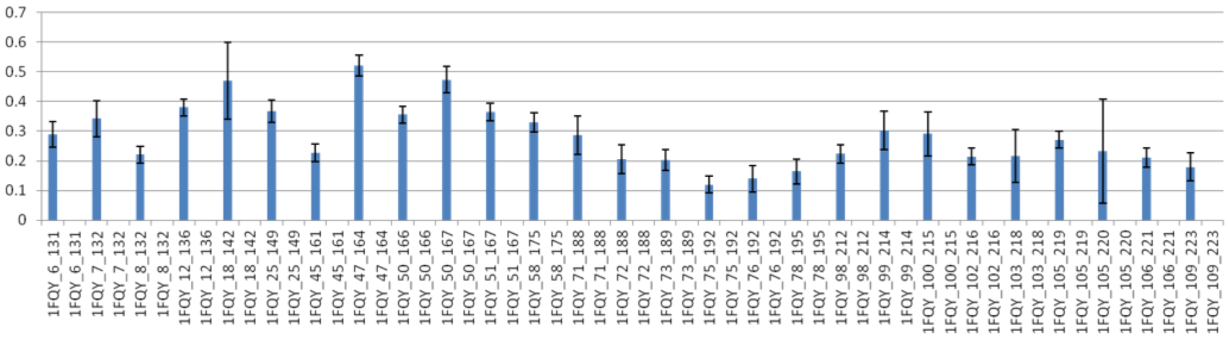
The individual backbones were analyzed by the number of mutations and the resulting change in energy between the symmetric backbone and the symmetrically mutated symmetric backbone. The single symmetric mutations were analyzed to determine the degree of their energy contribution. Those with little to no improvement in energy compared to the original symmetric backbone were selected as candidate for symmetric variants.

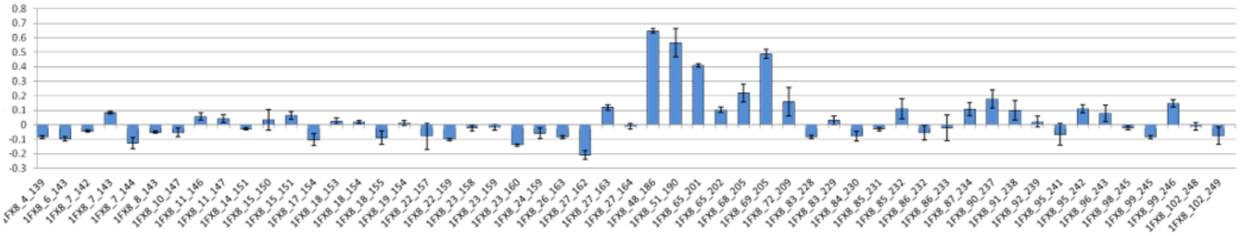
With the top 20 symmetric variants identified from the studies above, the amino acid sequences of the asymmetric unit for each backbone was extracted. The DNA sequence for the asymmetric unit was optimized for *E.coli* expression on GenScript’s website. The asymmetric genes were ordered from GenScript in the pUC19 vector. The asymmetric gene was assembled as a dimer into the pBG100 vector using SLIC (CITE). SLIC primers were designed manually and the analyzed using clonemanager. The assembled genes in pBG100 were sequence verified.

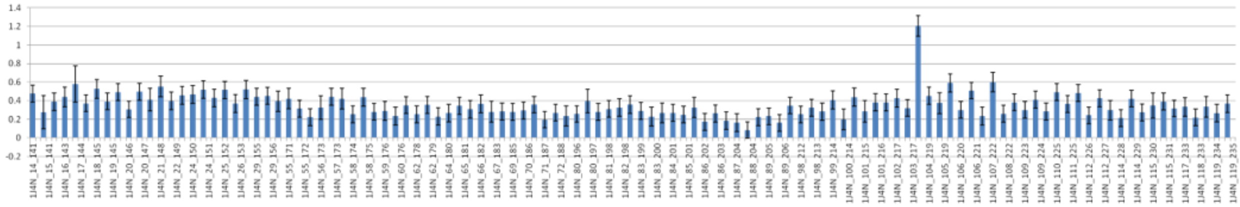
The pBG100 with assembled symmetric genes were transformed into BL21 (DE3), BL21 (DE3) STAR, BL21 (DE3) pLysS, Rosetta (DE3), Rosetta2 (DE3) pLysS, C-41 (DE3), and C-41 (DE3) pLysS host strains. Expression tests were performed using a medium-throughput approach that involves 10 mL glass test tubes containing 5 mL of LB broth and 10 µL of an overnight starter culture in LB grown at 37C. Cultures were grown in 37C for several hours until it reached an OD600 of between 0.5 and 0.7 at which point cultures were induced with 1 mM IPTG. Post-induced, cultures were grown either in 37C for 4 hours, or 25C for 12 hours. At the point of harvesting, the OD600 was recorded. The samples were normalized based on recorded OD600 and run on a 12% gel. Western blot was conducted using an anti-his antibody for detection of his-tagged proteins in pre-induction and post-induction whole cell lysates. Protein identification by mass spectrometry of trypsin-cleaved samples was performed by the Vanderbilt Mass Spectrometry Core.

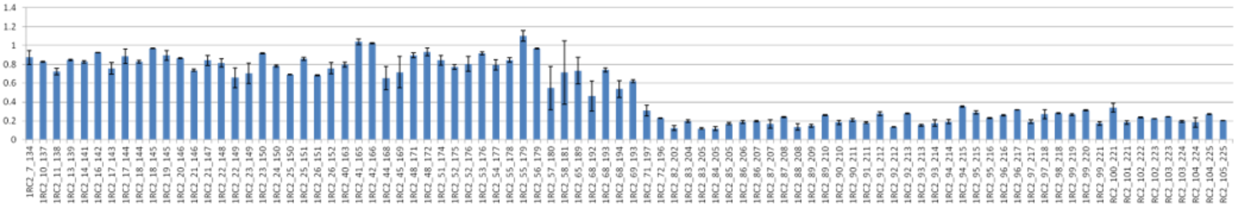
7.3 Results and discussion

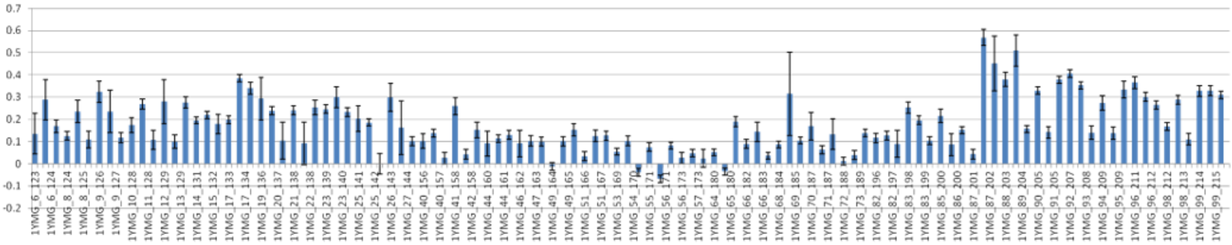
From 13 aquaporin sequences and backbones, 1302 symmetric backbones were created with cutpoints where symmetric counterparts were no further than a distance of 3.5 Angstroms. The Rosetta score in Rosetta energy units (REU) was normalized by protein length to determine the change in REU per amino acid by subtracting the REU per amino acid of the native protein minus the REU per amino acid of the symmetric backbone. These values were plotted for each original 13 protein backbones with the symmetric backbones ordered such that cutpoints were in sequence order (Figure 7.2).

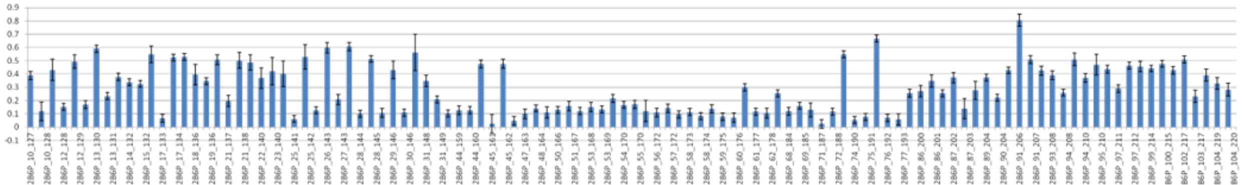

0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1FQY_6_131
1FQY_6_131
1FQY_7_132
1FQY_7_132
1FQY_8_132
1FQY_8_132
1FQY_12_136
1FQY_12_136
1FQY_18_142
1FQY_18_142
1FQY_25_149
1FQY_25_149
1FQY_45_161
1FQY_45_161
1FQY_47_164
1FQY_47_164
1FQY_50_166
1FQY_50_166
1FQY_50_167
1FQY_50_167
1FQY_51_167
1FQY_51_167
1FQY_58_175
1FQY_58_175
1FQY_71_188
1FQY_71_188
1FQY_72_188
1FQY_72_188
1FQY_73_189
1FQY_73_189
1FQY_75_192
1FQY_75_192
1FQY_76_192
1FQY_76_192
1FQY_78_195
1FQY_78_195
1FQY_98_212
1FQY_98_212
1FQY_99_214
1FQY_99_214
1FQY_100_215
1FQY_100_215
1FQY_102_216
1FQY_102_216
1FQY_103_218
1FQY_103_218
1FQY_105_219
1FQY_105_219
1FQY_105_220
1FQY_105_220
1FQY_106_221
1FQY_106_221
1FQY_109_223
1FQY_109_223

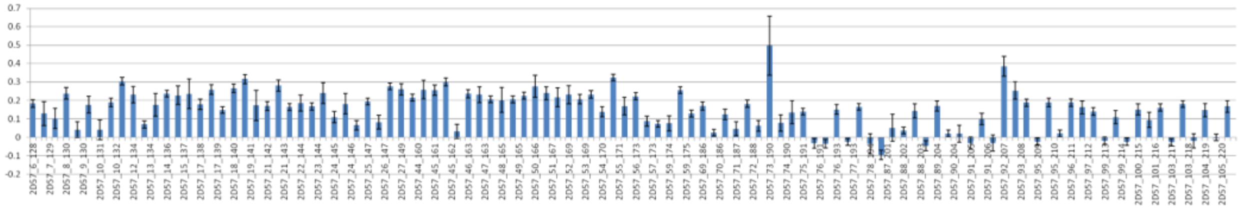

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
1FX8_4_139
1FX8_6_143
1FX8_7_142
1FX8_7_143
1FX8_7_144
1FX8_8_143
1FX8_10_147
1FX8_11_146
1FX8_11_147
1FX8_14_151
1FX8_15_150
1FX8_15_151
1FX8_17_154
1FX8_18_153
1FX8_18_154
1FX8_18_155
1FX8_19_154
1FX8_22_157
1FX8_22_159
1FX8_23_158
1FX8_23_159
1FX8_23_160
1FX8_24_159
1FX8_26_163
1FX8_27_162
1FX8_27_163
1FX8_27_164
1FX8_48_186
1FX8_51_190
1FX8_65_201
1FX8_65_202
1FX8_68_205
1FX8_69_205
1FX8_72_209
1FX8_83_228
1FX8_83_229
1FX8_84_230
1FX8_85_231
1FX8_85_232
1FX8_86_232
1FX8_86_233
1FX8_87_234
1FX8_90_237
1FX8_91_238
1FX8_92_239
1FX8_95_241
1FX8_95_242
1FX8_96_243
1FX8_98_245
1FX8_99_245
1FX8_99_246
1FX8_102_248
1FX8_102_249

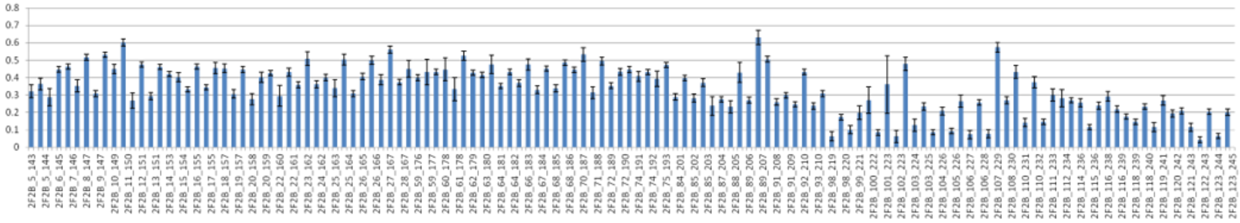

1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
1J4N_14_141
1J4N_15_141
1J4N_16_143
1J4N_17_144
1J4N_18_145
1J4N_19_145
1J4N_20_146
1J4N_20_147
1J4N_21_148
1J4N_22_149
1J4N_24_150
1J4N_24_151
1J4N_25_152
1J4N_26_153
1J4N_29_155
1J4N_29_156
1J4N_55_171
1J4N_55_172
1J4N_56_173
1J4N_57_173
1J4N_58_174
1J4N_58_175
1J4N_59_176
1J4N_60_176
1J4N_62_178
1J4N_62_179
1J4N_64_180
1J4N_65_181
1J4N_66_182
1J4N_67_183
1J4N_69_185
1J4N_70_186
1J4N_71_187
1J4N_72_188
1J4N_80_196
1J4N_80_197
1J4N_81_198
1J4N_82_198
1J4N_83_199
1J4N_83_200
1J4N_84_201
1J4N_85_201
1J4N_86_202
1J4N_86_203
1J4N_87_204
1J4N_88_204
1J4N_89_205
1J4N_89_206
1J4N_98_212
1J4N_98_213
1J4N_99_214
1J4N_100_214
1J4N_101_215
1J4N_101_216
1J4N_102_217
1J4N_103_217
1J4N_104_219
1J4N_105_219
1J4N_106_220
1J4N_106_221
1J4N_107_222
1J4N_108_222
1J4N_109_223
1J4N_109_224
1J4N_110_225
1J4N_111_225
1J4N_112_226
1J4N_112_227
1J4N_114_228
1J4N_114_229
1J4N_115_230
1J4N_115_231
1J4N_117_233
1J4N_118_233
1J4N_119_234
1J4N_119_235

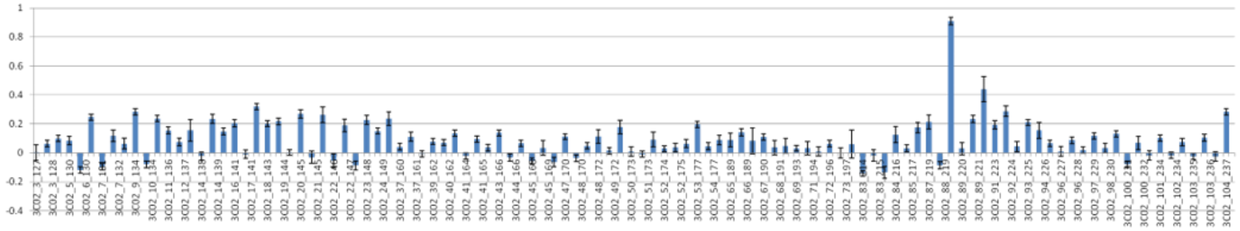

1.4
1.2
1
0.8
0.6
0.4
0.2
0
1RC2_7_134
1RC2_10_137
1RC2_11_138
1RC2_13_139
1RC2_14_141
1RC2_16_142
1RC2_17_143
1RC2_17_144
1RC2_18_144
1RC2_18_145
1RC2_19_145
1RC2_20_146
1RC2_21_146
1RC2_21_147
1RC2_22_148
1RC2_22_149
1RC2_23_149
1RC2_23_150
1RC2_24_150
1RC2_25_150
1RC2_25_151
1RC2_26_151
1RC2_26_152
1RC2_40_163
1RC2_41_165
1RC2_42_166
1RC2_44_168
1RC2_45_169
1RC2_48_171
1RC2_48_172
1RC2_51_174
1RC2_52_175
1RC2_52_176
1RC2_53_176
1RC2_54_177
1RC2_55_178
1RC2_55_179
1RC2_56_179
1RC2_57_180
1RC2_58_181
1RC2_65_189
1RC2_68_192
1RC2_68_193
1RC2_68_194
1RC2_69_193
1RC2_71_197
1RC2_72_196
1RC2_82_202
1RC2_83_204
1RC2_83_205
1RC2_84_205
1RC2_85_205
1RC2_86_206
1RC2_86_207
1RC2_87_207
1RC2_87_208
1RC2_88_208
1RC2_89_209
1RC2_89_210
1RC2_90_210
1RC2_90_211
1RC2_91_211
1RC2_91_212
1RC2_92_212
1RC2_92_213
1RC2_93_213
1RC2_93_214
1RC2_94_214
1RC2_94_215
1RC2_95_215
1RC2_95_216
1RC2_96_216
1RC2_96_217
1RC2_97_217
1RC2_97_218
1RC2_98_218
1RC2_99_219
1RC2_99_220
1RC2_99_221
1RC2_100_221
1RC2_101_221
1RC2_102_222
1RC2_102_223
1RC2_103_223
1RC2_103_224
1RC2_104_224
1RC2_104_225
1RC2_105_225


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
1YMG_6_123
1YMG_6_124
1YMG_8_124
1YMG_8_125
1YMG_9_126
1YMG_9_127
1YMG_10_128
1YMG_11_128
1YMG_12_129
1YMG_13_129
1YMG_14_131
1YMG_15_132
1YMG_17_133
1YMG_17_134
1YMG_19_136
1YMG_20_137
1YMG_21_138
1YMG_22_138
1YMG_23_139
1YMG_23_140
1YMG_25_141
1YMG_25_142
1YMG_26_143
1YMG_27_144
1YMG_40_156
1YMG_40_157
1YMG_41_158
1YMG_42_158
1YMG_44_160
1YMG_44_161
1YMG_46_162
1YMG_47_163
1YMG_49_164
1YMG_49_165
1YMG_51_166
1YMG_51_167
1YMG_53_169
1YMG_54_170
1YMG_55_171
1YMG_56_171
1YMG_56_173
1YMG_57_173
1YMG_64_180
1YMG_65_180
1YMG_66_182
1YMG_66_183
1YMG_68_184
1YMG_69_185
1YMG_70_187
1YMG_71_187
1YMG_72_188
1YMG_73_189
1YMG_82_196
1YMG_82_197
1YMG_83_198
1YMG_83_199
1YMG_85_200
1YMG_86_200
1YMG_87_201
1YMG_87_202
1YMG_88_203
1YMG_89_204
1YMG_90_205
1YMG_91_205
1YMG_92_207
1YMG_93_208
1YMG_94_209
1YMG_95_209
1YMG_96_211
1YMG_96_212
1YMG_98_212
1YMG_98_213
1YMG_99_214
1YMG_99_215


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4

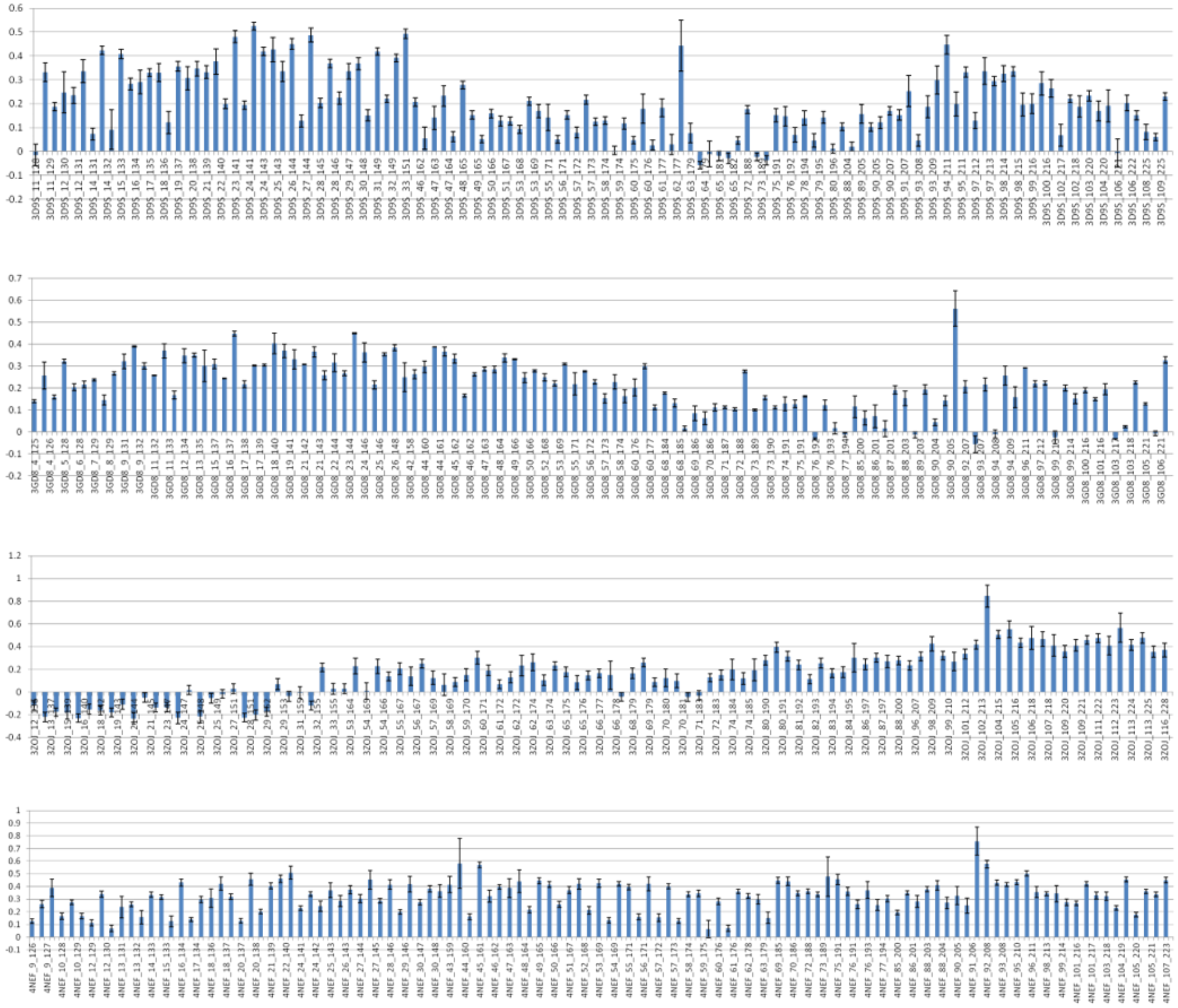


Figure 7.2. Energetic analysis of all symmetric backbones. Symmetric variants are grouped by native protein and ordered by cut point in the sequence. The units showed are normalized Rosetta energy (REU/amino acid). For each protein, oscillations are visible for regions in the protein that are ideal cut points (lower energy) versus regions that are not ideal.

Of the 1302 symmetric backbones, only 77 resulted in relaxed models that score better than the native protein from which they were based (Figure 7.3). Only seven of the original 13 backbones, 1FX8, 1YMG, 2D57, 3C02, 3D9S, 3GB8, and 3ZOJ have symmetric variants that resulted in an improved Rosetta score. In many cases, the standard deviation of the change in REU per amino acid would be in a reach close to zero. When a threshold was set for the average of the change of REU per amino acids to below -0.1, only 21 symmetric variants remained from three of the original backbones, 1FX8, 3C02, and 3ZOJ. (Figure 7.4).

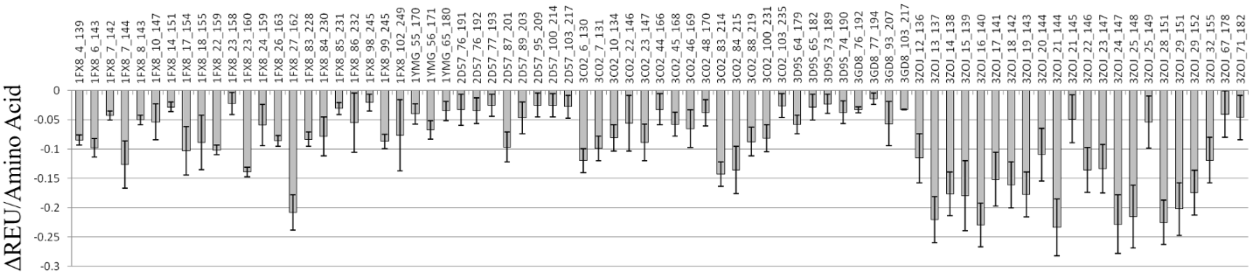


Figure 7.3. The normalized change in Rosetta energy (ΔREU/Amino Acid) was calculated for all symmetric backbones and native proteins. Only symmetric backbones that showed an improved score compared to the native protein are plotted here. Bars represent the standard deviation between the calculated values for the top 10 percent of relaxed models for each symmetric variant.

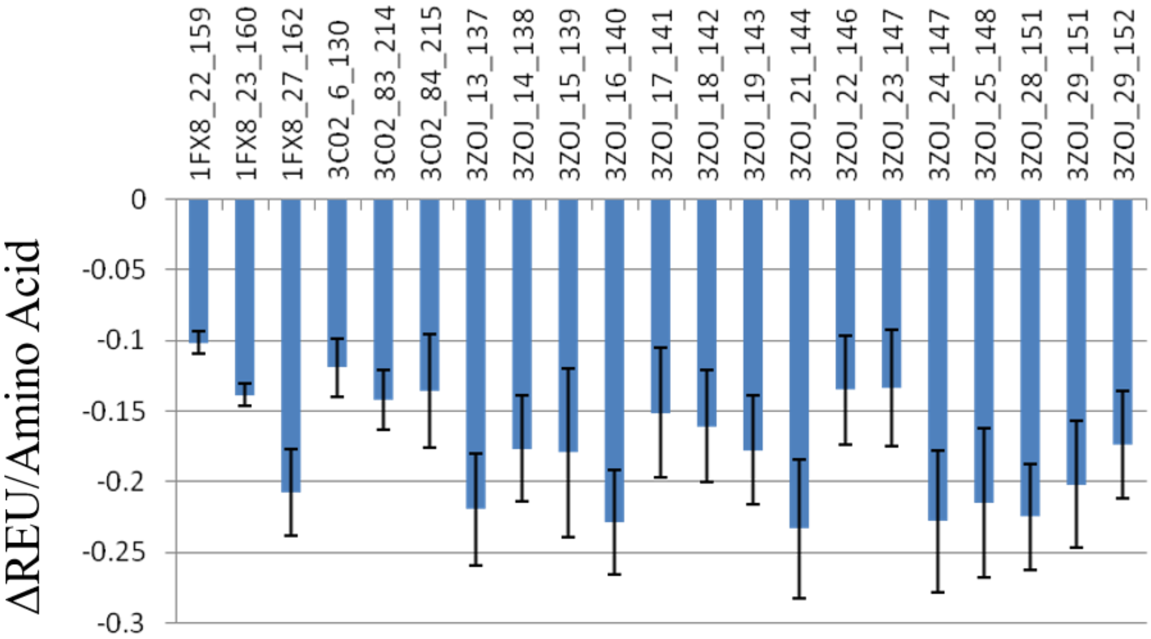


Figure 7.4. The normalized change in Rosetta energy (ΔREU/Amino Acid) for symmetric backbones that showed an improved score of 0.1 or more when compared to the native protein are plotted here. Bars represent the standard deviation between the calculated values for the top 10 percent of relaxed models for each symmetric variant.

However, the native protein energy may not be sufficient to determine whether the symmetric variant is an improvement. As shown in Figure 7.1, not all fragments of the native protein are used for the construction of each symmetric variant. Therefore, the native proteins were rescored such that only the fragments used in the respective symmetric construction contributed to the normalized score. The normalized difference in symmetric score was plotted with the standard normalized difference in score to determine if there was a correlation (Figure 7.5). Only symmetric backbones with scores that were improved from the native using the standard normalized difference in score were used for this analysis. Models that fell into quadrant III were considered to be ideal candidates as the values for both were negative indicating that the symmetric variant is more stable than the native protein as it is as well as the fragments used from the native protein for symmetric construction.

Interestingly, 2D57 symmetric variants were in quadrant III and were near the diagonal, indicating that 2D57 symmetrically calculated normalized difference in score correlated well with the standard normalized difference in score values. Symmetric variants for 3ZOJ were clearly scored better using the standard normalized difference in score as all variants for this protein fell above the diagonal. Symmetric variants for 1FX8 were split into two groups where one group was above the diagonal in quadrant II and the other group was below the diagonal in quadrant III. However, both of these groups had their own positive correlation between the calculated differences in energy. Symmetric variants for 3C02 also split into two groups, however, these groups did not show a strong correlation in this manner.

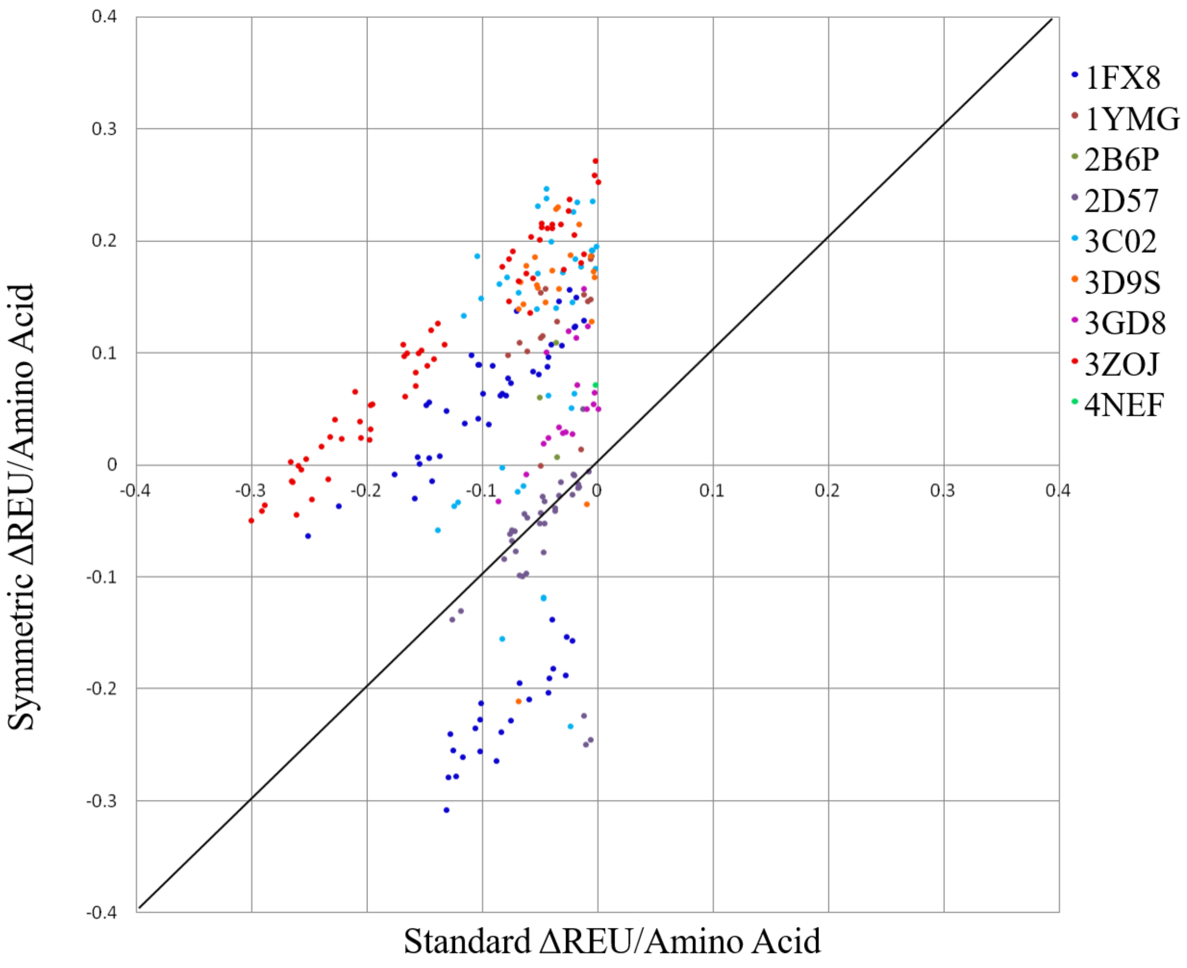


Figure 7.5. Correlation plot of the normalized difference in the calculated symmetric score and standard Rosetta score. Potential candidates can be identified in quadrant III.

The symmetric variants from quadrant III in Figure 7.5 were extracted and the score differences for normalized standard Rosetta score and normalized symmetric score were plotted. A few 1FX8 symmetric variants constructed an N-terminal asymmetric unit had negative scores for both normalized standard Rosetta and normalized symmetric scores; however, the symmetric scores were much closer to zero than the normalized standard Rosetta scores. For 2D57, the symmetric variants constructed from an N-terminal asymmetric unit had much better normalized symmetric scores than normalized standard Rosetta scores, while the symmetric variants that were constructed from a C-terminal asymmetric unit had similar normalized Rosetta scores and normalized symmetric scores.

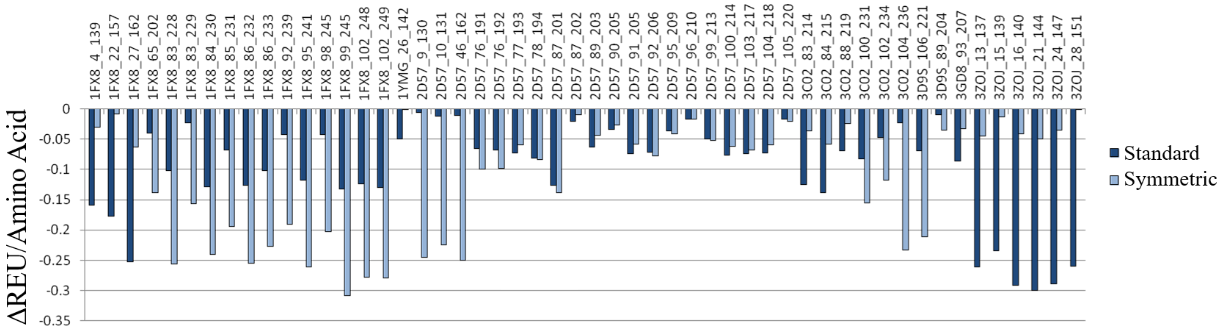


Figure 7.6. Difference in score between symmetric variants in quadrant III and the respective native proteins using the normalized standard Rosetta score and normalized symmetric score.

The normalized standard Rosetta score of the native protein was subtracted from the normalized symmetric score calculated for the native protein based on fragments used to construct the symmetric variant. Therefore, values above the x-axis are when the normalized standard Rosetta score for the native protein is better than the normalized symmetric score, whereas values below the x-axis indicate that the normalized symmetric score is better than the normalized standard Rosetta score (Figure 7.7). Interestingly, for symmetric variants of 1FX8 constructed from an asymmetric unit on the N-terminal side, the normalized symmetric score is better than the normalized standard Rosetta score; however, for variants of 1FX8 constructed from an asymmetric unit on the C-terminal side, the normalized standard Rosetta score was better than the normalized symmetric score. The N-terminal side of 1FX8 has an additional structured region in the loop which could account for this change in direction.

All of the 3ZOJ native scores are below the x-axis, so this could explain why the symmetric score of the symmetric variants was not able to overcome a much better native score whereas 1FX8 symmetric variants constructed from a C-terminal region had more ideal candidates from the normalized symmetric score because the symmetric versions of the native

1FX8 scored worse than the standard Rosetta native score (Figure 7.5). Additionally, the difference between the native scores for 2D57 are close to zero in all cases which supports the strong correlation on the diagonal in Figure 7.5 for the comparison of normalized symmetric and normalized standard Rosetta score for symmetric variants.

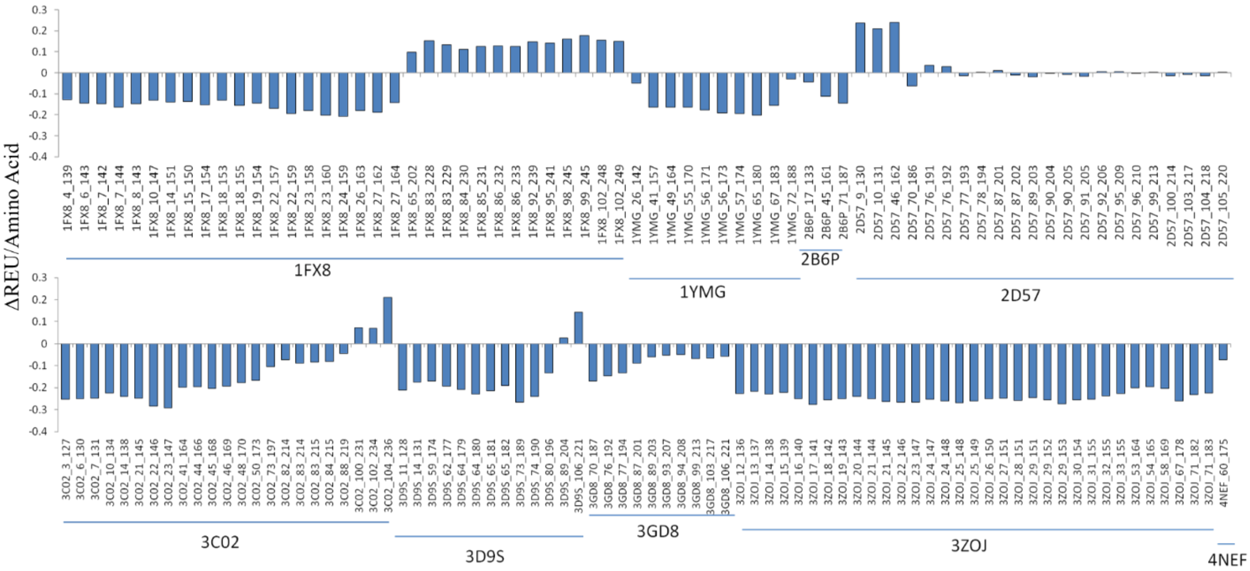


Figure 7.7. Calculated difference in REU per amino acid for native proteins. Symmetric scores were calculated from the native proteins based on fragments used in construction of the symmetric backbones and normalized by size of the protein. The normalized standard Rosetta score is subtracted from the normalized symmetric score to obtain the values plotted here. Values above the x-axis indicate that the normalized standard Rosetta score is better than the normalized symmetric score, whereas values below the x-axis indicate that the normalized symmetric score is better than the normalized standard Rosetta score.

Although the plot above showed a nice correlation between normalized symmetric and normalized Rosetta scores (Figure 7.5), it may have been an artifact of insufficient relaxation because of the observed clustering of similar models in the same energy space. Models were then relaxed an additional 25 times. This resulted in a correlation plot that still had clusters of the same native proteins, but the correlation was unsuspiciously absent (Figure 7.8). Quadrant III contained symmetric variants from 3ZOJ, 1FX8, 2D57, and 3GD8. Although more spread out, there were still clear correlations on the diagonal including scores from symmetric variants of 1FX8 and 2D57. Symmetric variants for 3ZOJ appeared to be positively correlated but above the diagonal meaning that the normalized standard Rosetta score was better than the normalized symmetric score.

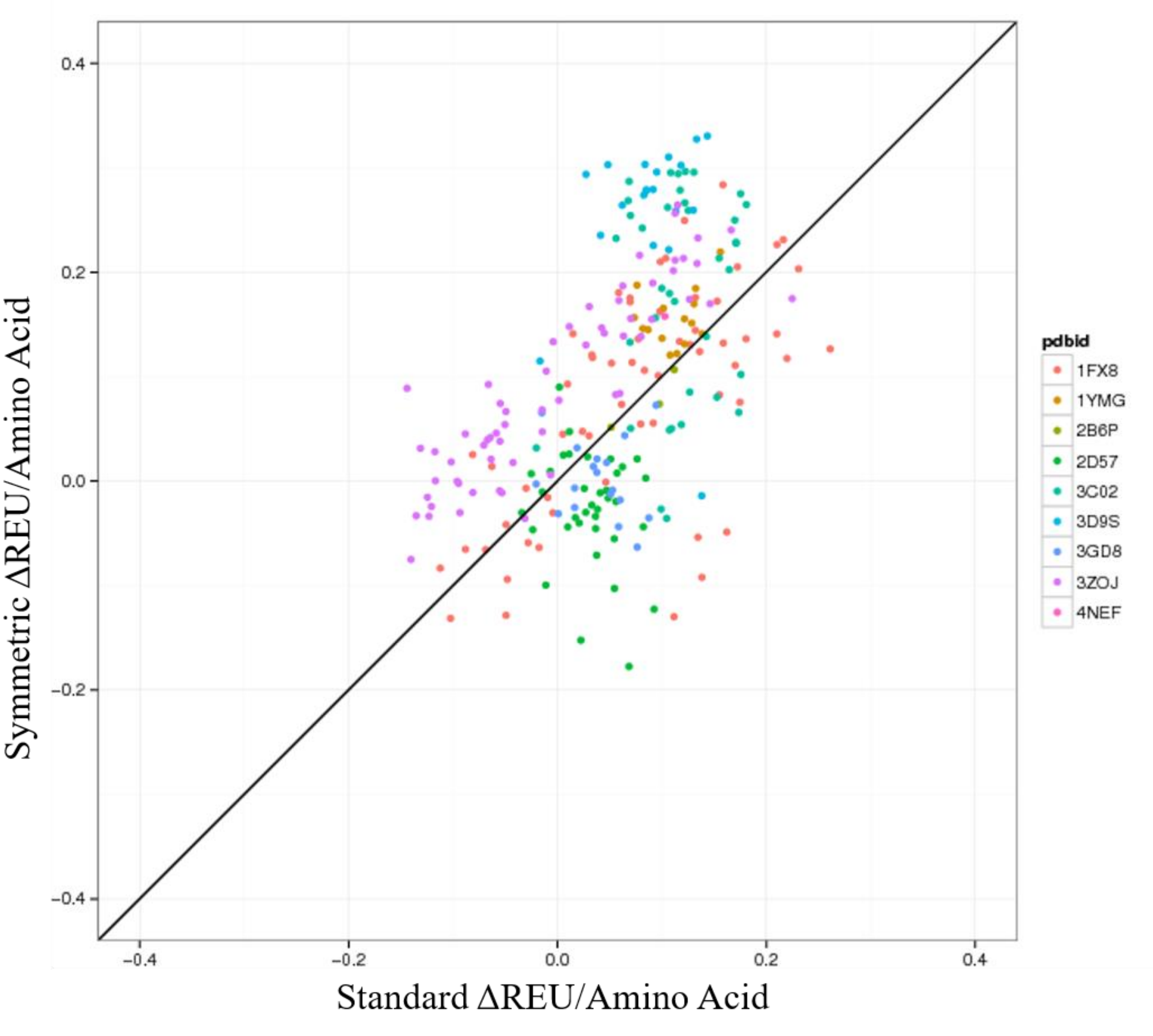


Figure 7.8. Correlation plot of the normalized difference in the calculated normalized symmetric score and normalized standard Rosetta score. Positive correlations can be seen in 1FX8 and 3ZOJ. Potential candidates can be identified in quadrant III.

With the additional relaxation trials, native proteins were also relaxed to have a fair energy comparison. While this created more diversity in the models which is apparent through the scattered scores in Figure 7.8, it shifted many of the symmetric backbones over the y-axis leaving few candidates for experimental testing. However, the aggressive relaxation approach may have also created additional noise in the dataset. The normalized standard Rosetta scores resulting in comparison to residues in an even more optimal conformation, many of which were not representative of the symmetric variants. It is likely that the normalized symmetric scores are more reliable than the normalized standard Rosetta scores. To account for this, the normalized symmetric score was weighted by 0.75 while the normalized standard Rosetta score was weighted by 0.25 (Figure 7.9).

Down weighting the normalized standard Rosetta score collapsed the range of scores and causes nearly every symmetric backbone to have a normalized standard Rosetta score of close to zero. However, this graph does show that despite the significant down weighting, at normalized symmetric scores above 0.1, the normalized standard Rosetta score is also positive, indicating a poor choice as a candidate for experimental studies. Good candidates are still likely to be in quadrant III; however, because of the down weighting of the normalized standard Rosetta score, candidates just over the y-axis can be considered with little penalty. Additionally, any very low weighted, normalized standard Rosetta scores with a normalized symmetric score of near zero could be considered as a candidate. The weighted scores were combined to create a weighted sum of scores where 75% of the score comes from the normalized symmetric score while 25% of the score comes from the normalized standard Rosetta score. From the weighted sum analysis, 18 candidates were identified (Figure 7.10).

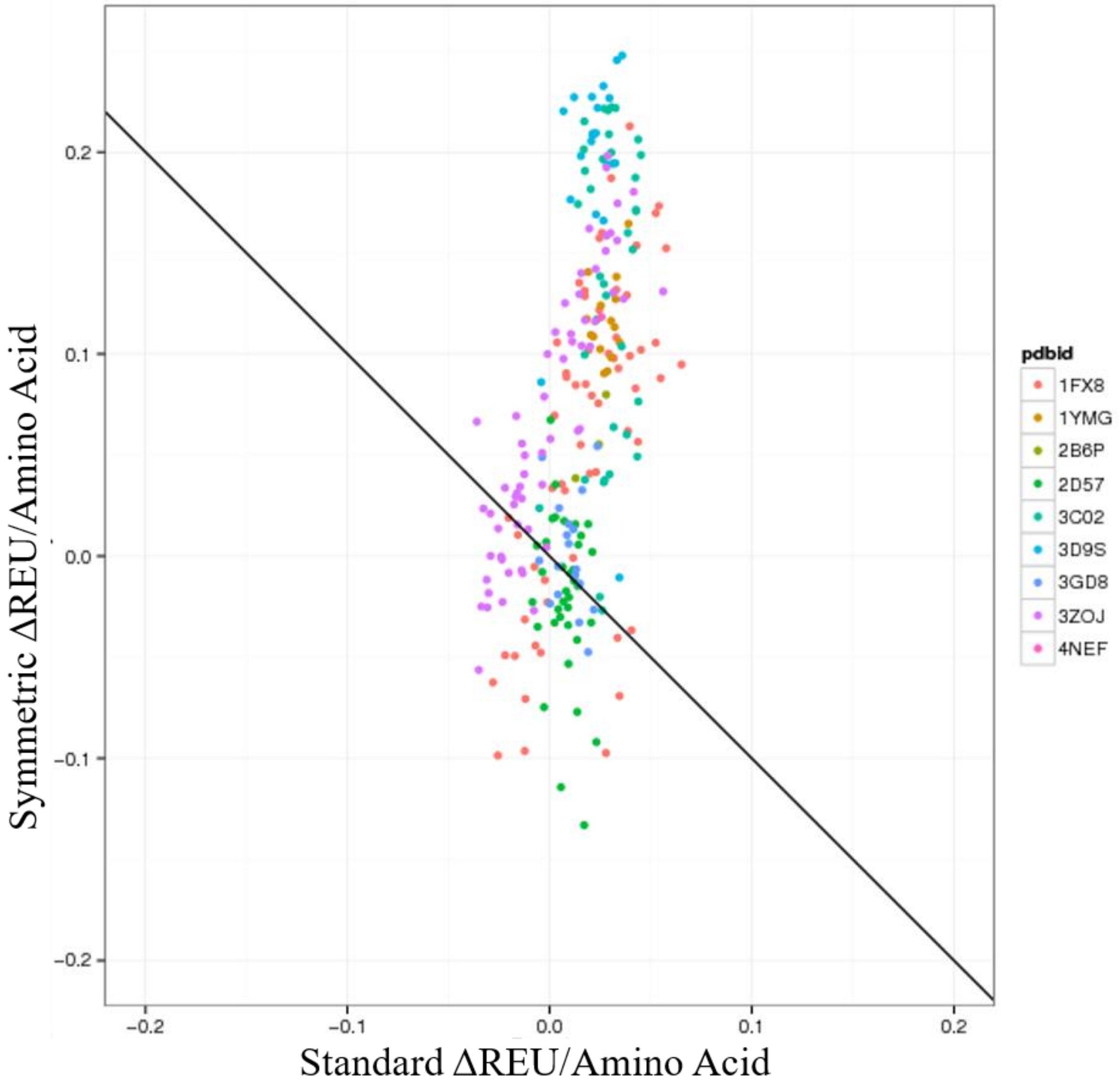


Figure 7.9. Weighted normalized symmetric and standard Rosetta scores. Symmetric scores had a weight of 0.75 while the standard Rosetta score had a weight of 0.25. Below the diagonal are the symmetric backbones of interest with either a negative symmetric score and near zero standard Rosetta score or a near zero symmetric score and a negative standard Rosetta score.

Symmetric backbones that are in the ranges to qualify as ideal candidates include symmetric variants of 1FX8, 2D57, 3C02, 3GD8, and 3ZOJ. A total of 45 symmetric backbones had a weighted sum of normalized scores of nearly zero or less than zero (Figure 7.10). A mixture of symmetric variants of 1FX8 that were constructed from both N-terminal and C-terminal asymmetric units had a weighted sum of normalized change in scores of less than -0.06. Of only two symmetric variants of 2D57 constructed from an N-terminal asymmetric unit, one, 9_130, had a weighted sum of nearly zero whereas 46_162 had a weighted sum of less than -0.06. Interestingly, there were three variants with starting cutpoints in the range of 76-78. The 2D57 76_191 variant had a weighted sum of less than -0.02; the 2D57 77_193 variant had a weighted sum of nearly -0.12; and the 2D57 78_194 variant had a weighted sum of nearly zero. This is a good example of how a shift in the frame for an asymmetric unit by only one residue can result in a vastly different engineered protein energetically. The only symmetric variant for 3C02 and the variants of 3GD8 had a weighted sum of nearly zero. Only variants constructed from the N-terminal half of 3ZOJ had a weighted sum less than zero and the range of weighted sums for these ranged from nearly zero to -0.08.

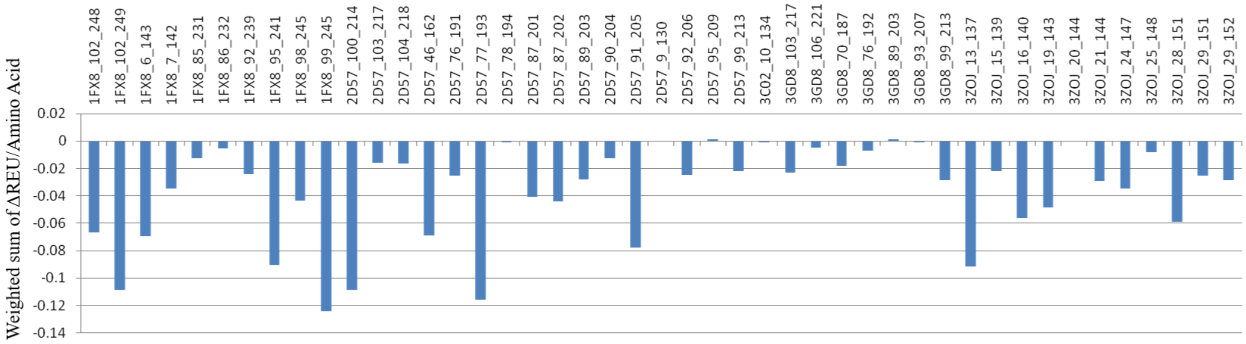


Figure 7.10. Weight sum of normalized symmetric and standard Rosetta scores. The weighted sum was calculated for symmetric variants where symmetric scores had a weight of 0.75 while the standard Rosetta score had a weight of 0.25. Symmetric variants with a weighted sum of nearly zero or less are presented here.

The top 20 symmetric variants were selected by the lowest weighted sum of normalized symmetric and standard Rosetta scores. The sequence space of the symmetric variants was explored using Rosetta fixed backbone design with a scoring bonus that favors the native, in this case original residue from symmetric variant, residue (Favor Native Residue, FNR) along with a scoring penalty that disfavors internal asymmetric sequences (Favor Symmetric Sequence, FSS).

In separate design experiments, FNR was varied at bonuses of 0.5, 0.75, 1, 1.25, and 1.5 while the FSS penalty was set at 1. For each symmetric backbone and varied FNR bonus, the top and bottom designs by Rosetta score were selected and analyzed using sequence recovery. The difference in sequence recovery between favorably scoring and poorly scoring designs was calculated for each of the top 20 symmetric backbones (Figure 7.11).

Positive scores for the difference in sequence recovery indicate that the design models that scored best in Rosetta resulted in a sequence closer to the original symmetric backbone. This suggests that the sequence from the original symmetric backbone is relatively optimal, and that the Rosetta score discriminates well between good and poor quality designs. For 1FX8, symmetric variants constructed from C-terminal asymmetric units had positive scores for all FNR bonuses indicating that the designed sequences from the best scoring models were closer to the original sequence than design models that scored poorly. Interestingly, the symmetric variants constructed from N-terminal asymmetric units had negative or nearly zero difference in sequence recovery. For symmetric variants of 2D57, the three with the closest cutpoints, 87_201, 87_202, and 91_205, had positive values calculated for difference in sequence recovery. This could indicate that this region of 2D57 is optimal for creating asymmetric units. Finally, 3ZOJ had a mix of positive, near zero, and negative values for calculated differences in sequence recovery. While all variants were constructed from an N-terminal asymmetric unit, the two closest to the N-terminus had mostly positively values.

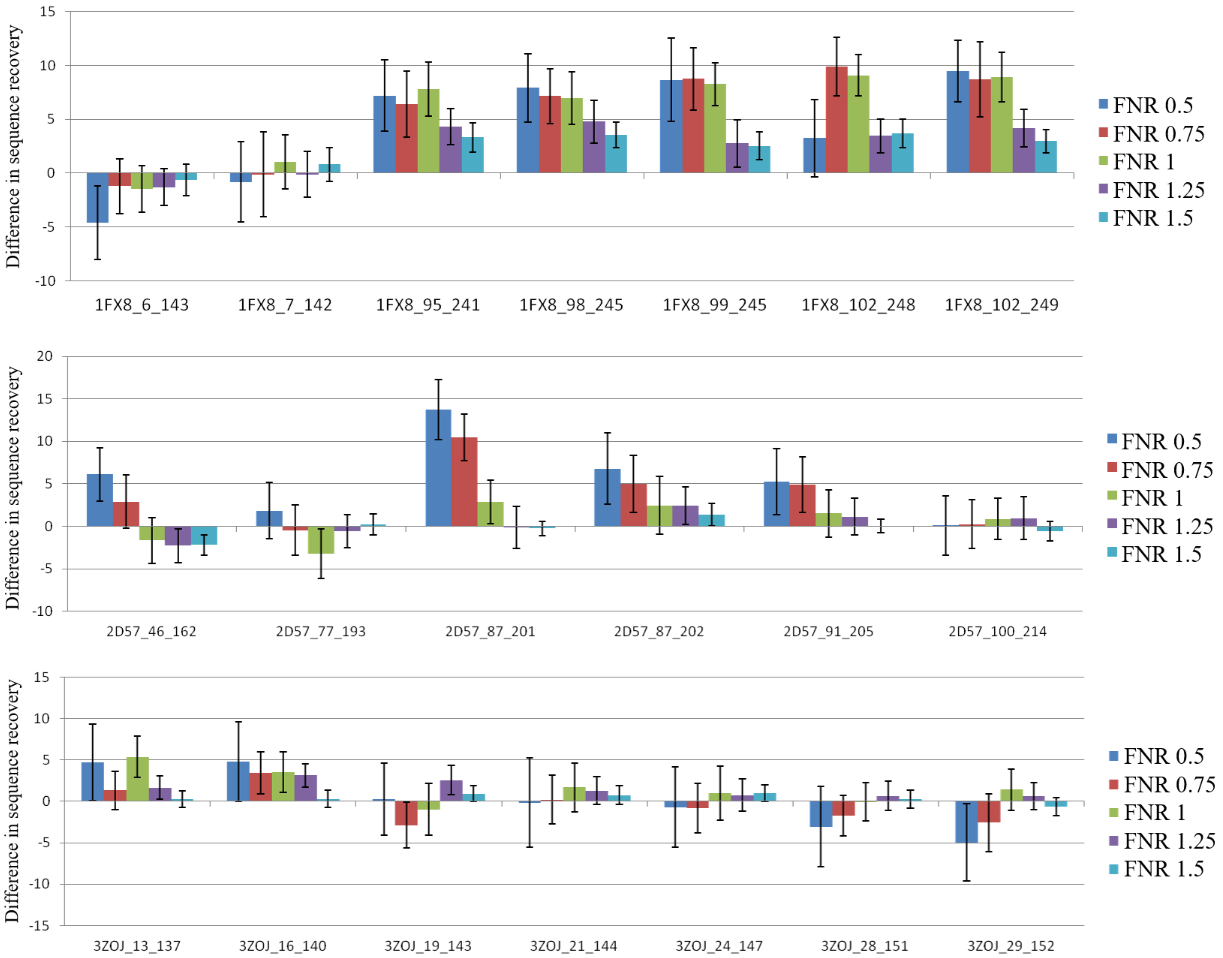


Figure 7.11. Exploring the sequence space of the top 20 symmetric variants by lowest weighted sum of normalized scores. Each symmetric backbone underwent fixed backbone design with a penalty for internal asymmetric mutations and varying bonuses for favoring the original sequence (FNR). An FNR of 0.5 indicates it is favoring the original sequence the least whereas a bonus of 1.5 favors the original sequence the most. The resulting best and worst models by Rosetta score were selected for each trial, and the sequence recovery was calculated. The sequence recovery of the worst scoring designed model was subtracted from the sequence recovery of the best scoring designed model.

Variants tested were identified on the original correlation plot for weighted normalized symmetric and standard Rosetta scores. The variants that had a positive difference in sequence recovery, therefore optimal sequences, are shown mostly in quadrants III and IV.

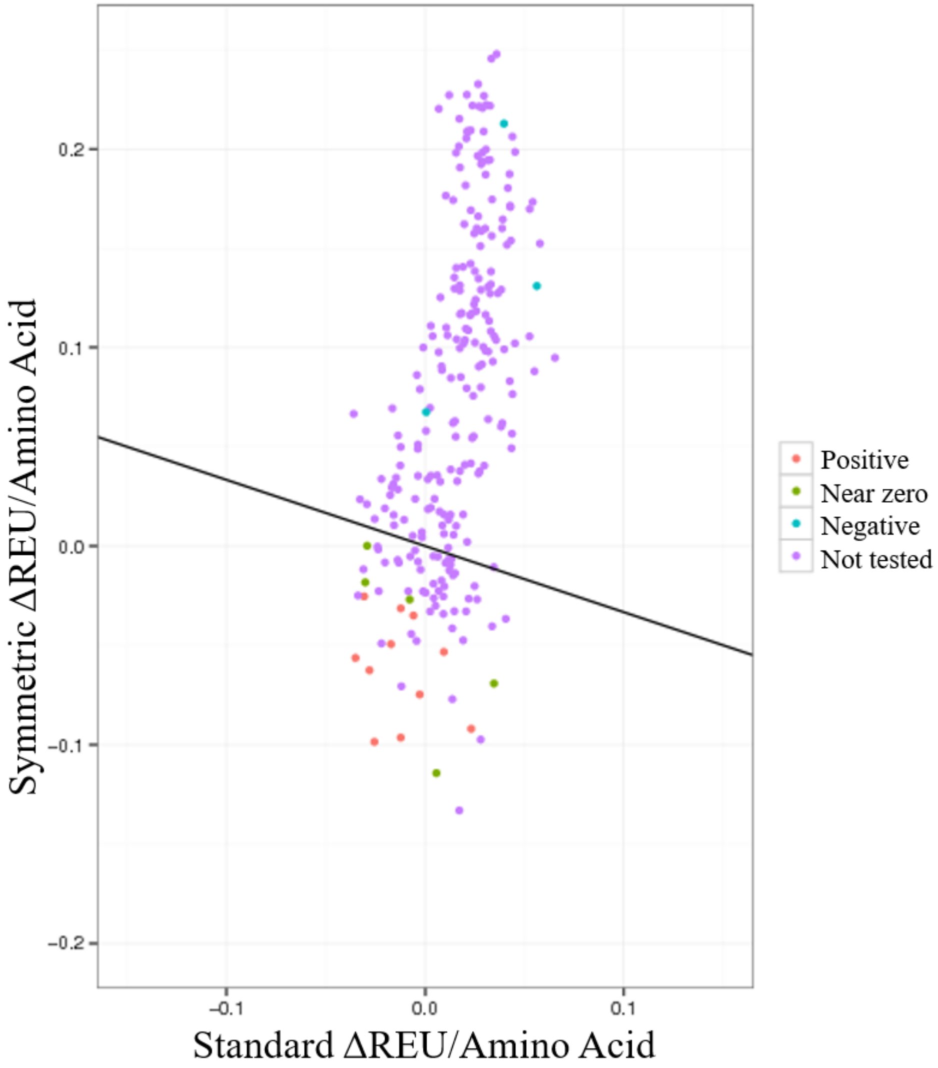
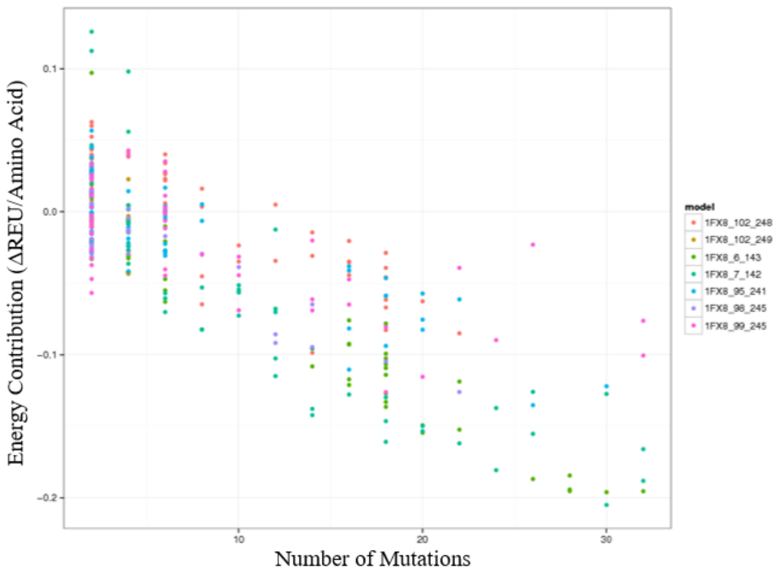
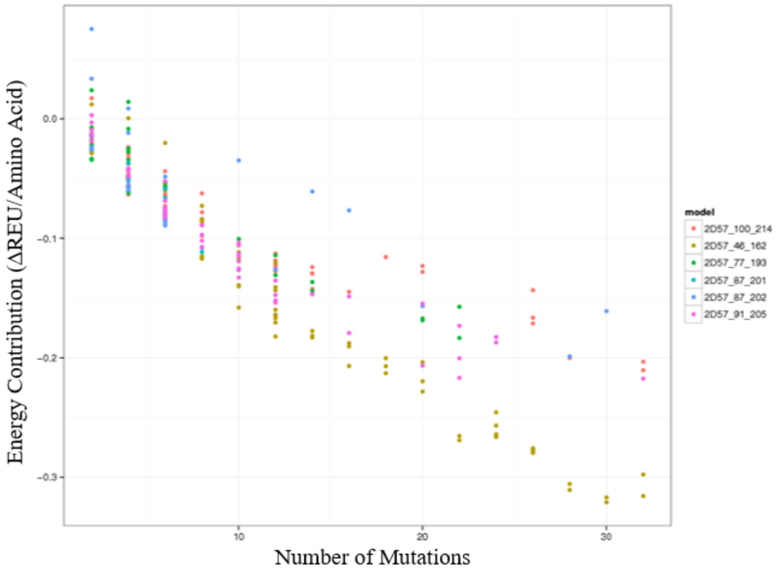


Figure 7.12. Location of symmetric variants in weighted normalized symmetric and standard Rosetta score correlation plot. Symmetric variants that were identified to have relatively optimal sequences from sequence recovery experiments are shown as positive. Those with nearly zero difference in sequence recovery between good and poorly scoring models are labeled as Near zero, while poorly scoring models with better sequence recovery than good scoring models are labeled as negative. Symmetric variants that did not undergo design experiments are labeled as not tested.

The energy contribution of each pair of symmetric mutations was calculated. The energy contribution was determined by the normalized change in REU between the symmetric backbone and designed symmetric backbone. As a pair of symmetric mutations is added, the energetic contribution of the mutations increased by lowering the energy of the design protein, as expected. However, this analysis was to identify whether a single pair of symmetric mutations is worth the energetic cost alone, as well as to determine how many symmetric pairs of mutations are required to reduce the energy consistently (Figure 7.13). This was also plotted for all of the top 20 symmetric backbones individually (Appendix H). For nearly all symmetric backbones, 12 symmetric mutations, a total of 24 mutations, were sufficient to consistently lower the energy of the designs beyond that of the symmetric backbone. As seen in the first stages of design, the initial 10 symmetric mutations came at a cost on their own, but with each addition of a symmetric mutation, the mutations overcame the cost of the barrier to effectively lower the energy of the designed protein.


Energy Contribution (ΔREU/Amino Acid)
0.1
0.0
-0.1
-0.2
10
20
30
Number of Mutations
model
1FX8_102_248
1FX8_102_249
1FX8_6_143
1FX8_7_142
1FX8_95_241
1FX8_98_245
1FX8_99_245


Energy Contribution (ΔREU/Amino Acid)
0.0
-0.1
-0.2
-0.3
10
20
30
Number of Mutations
model
2D57_100_214
2D57_46_162
2D57_77_193
2D57_87_201
2D57_87_202
2D57_91_205

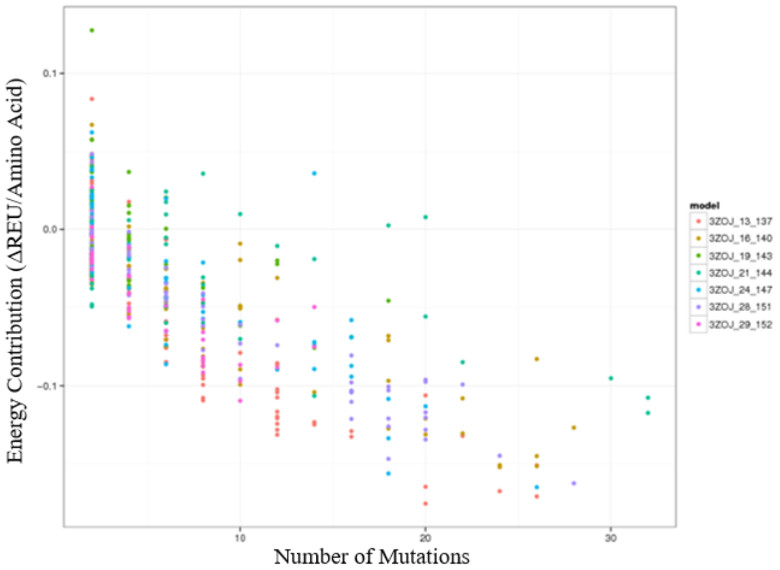

Energy Contribution (ΔREU/Amino Acid)
0.1
0.0
-0.1
10
20
30
Number of Mutations
model
3ZOJ_13_137
3ZOJ_16_140
3ZOJ_19_143
3ZOJ_21_144
3ZOJ_24_147
3ZOJ_28_151
3ZOJ_29_152

Figure 7.13. The normalized energy contribution for the number of mutations exhibited in designs of symmetric backbones in 1FX8 (top), 2D57 (middle), 3ZOJ (bottom). The number of mutations are stepwise by two because each mutation happened twice, once in each half.

From these studies, the top 20 symmetric variants were selected using a weighted sum of normalized symmetric energy and standard Rosetta score. The top 20 symmetric variants were designed while maintaining sequence symmetry. The best scoring versus worst scoring sequence recoveries was calculated and the difference was attributed to how discriminating the Rosetta score was for each backbone as well as how favorably Rosetta scored the original backbone sequence. Of the top 20, eleven had both relatively optimal sequences and discriminatory power between best and worst scoring designs. An additional six had nearly zero differences between best and worst scoring designs. The top 20 were then analyzed to determine the energetic cost of designs with respect to the number of symmetric designs. It was determined that the energetic cost of up to 10 symmetric mutations, 20 total mutations, was not sufficient to lower the normalized Rosetta score of symmetric backbones. However, 12 and up to 30 symmetric mutations lower the energy compared to the original symmetric backbone. This is the case for all of the top 20 symmetric backbones, as such, all were tested experimentally.

Expression tests were done on all 20 symmetric assembled genes under a number of different expression conditions that varied the host strain, induction temperature, and in a few cases the expression vector (Table 7.5). Ultimately, Western blotting for a his-tag verified an induction band for six constructs. Expression conditions were further optimized for these six constructs for variables including induction temperature, amount of IPTG at induction, OD600 at induction, and induction time (Table 7.6). From this expression screen, Western blotting revealed the optimal conditions for each of the six constructs (Table 7.7). Unfortunately, mass spectrometry did not identify our proteins of interest in any of the bands.

Table 7.5. Conditions tested for the 20 symmetric backbone constructs. Constructs with lower energy models were prioritized for extensive testing. Variables include host strain, induction temperature (Ind. Temp), and expression vectors.

Construct	**Host strain**	**Ind. Temp. C**	**Expression Vector**	**Comments on the best expression condition**
1FX8_99_245	BL21,BL21_Plyss, Rosetta_Plyss, C41_PLyss	25, 37	PBG100	PBG100_BL21_Plyss, Rosetta_Plyss@25°C induction show induced bands
2D57_100_214	BL21,BL21_Plyss, Rosetta_Plyss, C41_PLyss	25, 37	PBG100, pET21b	PBG100_BL21_Plyss@ 25°C induction show induced bands
2D57_77_193	BL21,BL21_Plyss, Rosetta_Plyss, C41_PLyss	25, 37	PBG100, pET21b	PBG100_BL21_Plyss, Rosetta_Plyss@25°C induction show induced bands
3ZOJ_13_137	BL21,BL21_Plyss, Rosetta_Plyss, C41_PLyss	25, 37	PBG100, pET21b	PBG100_BL21_Plyss, Rosetta_Plyss@25°C induction show induced bands
1FX8_102_248	BL21_Plyss, C41_Plyss, Rosetta_Plyss, C43	25	PBG100	PBG100_C41_Plyss@ 25°C induction show induced bands
1FX8_102_249	BL21_Plyss, C41_Plyss, Rosetta_Plyss, C43	25	PBG100	PBG100_C41_Plyss, C43@ 25°C induction show induced bands
1FX8_6_143	BL21_Plyss, C41_Plyss, Rosetta_Plyss, C43	25	PBG100	PBG100_C43 @25°C induction show induced bands
1FX8_7_142	BL21_Plyss, C41_Plyss, Rosetta_Plyss, C43	25	PBG100	None
1FX8_95_241	BL21_Plyss, C41_Plyss, Rosetta_Plyss	25	PBG100	None
1FX8_98_245	BL21_Plyss, C41_Plyss, Rosetta_Plyss	25	PBG100	None
2D57_46_162	BL21_Plyss, C41_Plyss, Rosetta_Plyss	25	PBG100	None
2D57_87_201	BL21_Plyss, C41_Plyss, Rosetta_Plyss	25	PBG100	None
2D57_87_202	BL21_Plyss, C41_Plyss	25	PBG100	None

2D57_91_205	BL21_Plyss, C41_Plyss	25	PBG100	None
3ZOJ_16_140	BL21_Plyss, C41_Plyss	25	PBG100	None
3ZOJ_19_143	BL21_Plyss, C41_Plyss	25	PBG100	None
3ZOJ_21_144	BL21_Plyss	25	PBG100	None
3ZOJ_24_147	BL21_Plyss	25	PBG100	None
3ZOJ_28_151	BL21_Plyss	25	PBG100	PBG100_BL21_Plyss@ 25°C induction show induced bands
3ZOJ_29_152	BL21_Plyss	25	PBG100	None

Table 7.6. The six best constructs identified by Western blotting were further optimized for expression. The expression conditions screened included induction temperature, the OD600 at induction, concentration of IPTG, and the length of the induction period.

Construct	**Host strain**	**Expression Vector**	**Induction Temp. C**	**Induction OD600**	**IPTG Conc (mM))**	**Post-induction time (h)**
1FX8_102_248	C41_Plyss	PBG100	25, 30, 37,16	0.2, 0.4, 0.6, 0.8	0.2, 0.4, 0.6, 0.8, 1.0	2, 4, 6, O/N
1FX8_102_249	C43	PBG100	25, 30, 37,16	0.2, 0.4, 0.6, 0.8	0.2, 0.4, 0.6, 0.8, 1.0	2, 4, 6, O/N
1FX8_6_143	C43	PBG100	25, 30, 37,16	0.7	0.6, 1.0	4, O/N
1FX8_99_245	Rosetta_Plyss	PBG100	25, 37	0.2, 0.4, 0.6, 0.8	0.8	2, 4, 6, O/N
2D57_77_193	Rosetta_Plyss	PBG100	25, 30, 37	0.8	0.6, 1.0	4, O/N
3ZOJ_28_151	BL21_Plyss	PBG100	25, 30, 37,16	0.2, 0.4, 0.6, 0.8	0.2, 0.4, 0.6, 0.8, 1.0	2, 4, 6, O/N

Table 7.7. Optimized expression conditions for the six best constructs. The final optimized conditions from screening in Tables 7.5 and 7.6 resulted in identifying the following conditions as optimal.

	1FX8 102_248	1FX8 102_249	1FX8 6_143	1FX8 99_245	2D57 77_193	3ZOJ 28_151
Induction Temp. C	25	30	30	37	30	37
Induction Timing (OD600)	0.6	0.6	0.7	0.8	0.8	0.6
IPTG Conc (mM)	0.8	1	1	0.8	1	0.2
Post-induction time (hr)	O/N	O/N	O/N	O/N	O/N	O/N

Moving forward, a more thorough screen of vectors is needed. Previous studies (Appendix J) show that one specific expression condition worked for the four symmetric designs evaluated. Many of the amino acid sequences that were used to construct symmetric backbones were from species other than *E.coli* (Table 7.1). When ordering genes, the company, GenScript, provides a gene optimization software that prepares synthesized genes with DNA sequences that are optimal for expression in *E.coli*. However, 1FX8, the glycerol facilitator protein, is native to *E.coli*. It would be an interesting experiment to use the exact nucleotide sequences from the fragments used by the respective backbone, as long as the construction of the symmetric gene from fragments does not introduce restriction enzyme cleavage sites. Studies have shown that for membrane proteins especially, silent mutations can have a drastic effect on protein expression levels (Norholm et al., 2012).

CHAPTER 8

DISCUSSION AND FUTURE DIRECTIONS

8.1 Summary

Progress in structural biology of membrane proteins lags far behind that of soluble proteins because the study of membrane proteins is a formidable challenge. Difficulties in structural characterization of membrane proteins is due their complex environment and inherent flexibility. However, membrane proteins are a key class of proteins that are relevant in many diseases and are often times targets for drugs. This has really been the driving force in the community for elucidating structural information and understanding the mechanisms of membrane proteins.

Although soluble proteins have seen much success in protein design, it continues to be a difficult task for membrane proteins. This is especially true for computational protein design of membrane proteins and is due to the limited amount of information known about the structure of membrane proteins stemming from the aforementioned challenges that arise from studying membrane proteins experimentally. This work aims to leverage the experimental information known regarding structure and stability of membrane proteins to improve existing modeling approaches.

Previous studies established energy functions for membrane proteins in Rosetta; however, a thorough evaluation of the performance of the energy function during design had been lacking. One study did evaluate design but it was unclear which proteins were tested and whether this included homo-oligomeric proteins. In Chapter 2, Rosetta design was benchmarked on a diverse set of high-resolution monomeric and homo-oligomeric proteins. First, a set of minimization experiments were performed to evaluate how different types of minimization in

Rosetta play a role in the design outcome. Sequence recovery of the top scoring designs was used as a metric to asses which minimization strategy relieved defects in the input backbone without moving the backbone far from the input, and without the risk of over-constraining the protein to its native sequence. It was found that for both monomeric and homo-oligomeric membrane proteins, the constraint to start coordinates option along with FastRelax is the best strategy for preparing membrane protein structures.

Next, the design algorithm was evaluated using the default soluble energy function and the membrane protein specific energy function RosettaMembrane. While sequence recovery in the core of the protein was found to be higher using the soluble energy function, the surface sequence recovery for monomeric membrane proteins was higher in RosettaMembrane. For homo-oligomeric membrane proteins the energy functions performed similarly for surface recovery; however, design of the symmetric complex did result in a significantly higher sequence recovery than homo-oligomers designed as monomers. This was the first study that we know of that demonstrated that Rosetta Symmetry is advantageous to use in concert with RosettaMembrane.

Finally, this study concluded by showing that Leucine was selected by Rosetta design as a more energetically favorable residue much more frequently than what is seen in native amino acid compositions. Further analysis showed that this most frequently occurred in the inner and outer hydrophobic regions of the protein. This study ultimately determined that RosettaMembrane has a bias towards Leucine at the cost of other hydrophobic amino acids

The RosettaMembrane energy function was originally created based on 28 high-resolution membrane proteins (Yarov-Yarovoy et al., 2006). Since then, there has been tremendous progress with various types of membrane protein structures including the burgeoning fields of G-protein coupled receptors and Beta barrels. Chapter 2 revealed that there is still progress to be made in creating a Rosetta energy function that recapitulates a native-like sequence composition.

There are many computational programs that have tackled the difficult task of predicting mutation-induced stability changes from either sequence or structure-based approaches. Many of these programs use machine learning algorithms to calibrate values to match that of experimentally determined ΔΔG of unfolding. Rosetta contains an application, ddg_monomer, that was previously optimized from a sampling standpoint to accurately predict the energetic effect of a mutation with a correlation of 0.69. However, none of the current available methods have been trained specifically for membrane proteins.

In Chapter 3, ten existing programs were evaluated for their ability to predict mutation-induced stability changes for a membrane protein dataset of over 200 mutations. Concordance, Pearson, and Spearman correlations between the predicted energy and experimentally determined ΔΔG of unfolding did not reach 0.4 in all cases, indicating that these methods are poor predictors for membrane proteins. In addition to testing the correlation of the predicted energies with the experimentally derived energies, a program should be able to provide results for predicting the stabilizing effect the mutation would have on the protein i.e. whether it is destabilizing or stabilizing. Using ROC curve analysis, the AUC was determined to be 0.7 or lower with the original implementation of Rosetta ddg_monomer having the best AUC.

Rosetta was an ideal program for predicting mutation-induced stability changes in membrane proteins because of the existence of the membrane protein specific energy function. The RosettaMembrane energy function was used in place of the high-resolution energy function used during calculations in ddg_monomer. The same dataset of membrane protein mutations was used and ultimately the ddg_monomer implementation using the RosettaMembrane energy function performed similarly to the original implementation of ddg_monomer using the soluble energy function.

Chapters 2 and 3 provide an evaluation of the RosettaMembrane energy function from a sequence and structure perspective, respectively. However, both approaches stem from the same set of score terms that approximate physical forces and internal energetic costs based on Bayesian probabilities. In Chapter 2, it was revealed that the cost of designing in Leucine is a much lower energetic cost than other residues, in particular other hydrophobic residues. In Chapter 3, the RosettaMembrane energy function performed similarly to the soluble energy function when using the application ddg_monomer to predict mutation-induced stability changes in membrane proteins.

While experimental validation is an ideal form of validation for computational designs, it is highly unlikely that membrane proteins fully redesigned by RosettaMembrane could be experimentally structurally characterized in order to derive some sort of value to improve the energy function. It would involve working with mutants of a class of proteins that is already difficult to characterize, and the effects of multiple mutations simultaneously would be difficult, if not impossible, to disentangle. Therefore, the logical step forward to improving the energy function is to leverage existing membrane protein thermostability data from single-point mutations.

In Chapter 4, machine learning methods were employed to re-fit the weights of the score terms in the RosettaMembrane energy function to approximate the experimentally determined ΔΔG of unfolding for the dataset of single-point mutant membrane proteins from Chapter 3. Ridge, Lasso, and Elastic Net regressions resulted in near zero, in the case of Lasso, zero coefficients for terms that did not contribute to accurate predictions of ΔΔG of unfolding. These approaches were cross-validated and determined new weights for score terms in RosettaMembrane as well as removed score terms that introduced noise into the calculation. The newly weighted energy functions were used to sample and calculate the membrane protein single-point mutation dataset and resulted in a correlation of nearly 0.5 and an AUC of up to 0.75. Perhaps the most interesting finding from this study was the drastic increase in the contribution of the membrane protein relevant score terms to the energy function. While this exercise was an incremental improvement in performance metrics, it was informative and a powerful proof of principle that empirical data can be used to improve the RosettaMembrane energy function and effectively improve membrane protein modeling and design.

An improved RosettaMembrane energy function like that described in Chapter 4 would not only have applications for thermostability and design calculations on proteins of known structure, but also for proteins of unknown structure. Many variants of unknown significance implicated in disease are from proteins of unknown structure. In the case of Long-QT Syndrome, some variants of KCNQ1 have been linked to loss of function; however, many variants of unknown significance remain. It is believed that variants of disease-linked proteins act by destabilizing the protein. Prediction of the destabilizing effect of mutations, like that seen in Chapter 4, could provide valuable insight for treatment of patients with these variants; however, the structure of KCNQ1 is unknown. In Chapter 5, a model of the resting, closed state of KCNQ1 is developed to ultimately test predictions of the stabilizing effects of variants of unknown significance.

Chapter 5 details the development of a protocol to create a model of KCNQ1 from multiple templates of low sequence identity using RosettaCM. While RosettaCM enables an increase in conformational space due to the inclusion of multiple templates, the templates have very different structural features in the areas of interest, in particular, the gating charges. I was able to leverage existing mutational data to create filters that resulted in the selection of models that fit experimental criteria and were characteristic of the field's understanding of the resting state of the VSD. These filters were used throughout the minimization process and ultimately for the final selection of models. Final models also low energy Rosetta scores and scored well with external servers MolProbity and PDBSum. Additionally, an external software, PoreWalker, was used to create a profile of the pore's diameter to confirm the pore was closed. Supplementing multiple template comparative modeling with experimentally derived information as well as external model validation are key components to creating a high-quality model. The mutation-induced stability change prediction method described in Chapter 4 relies on high-quality models or structures in addition to an accurate energy function.

In Chapter 6, the relevant evolutionary pathways and biological implications for pseudo-symmetric membrane proteins are reviewed. This sets up the relevant background for my project that involved the engineering of a symmetric membrane protein in Chapter 7. This was truly the inspiration for Chapters 2, 3 and 4 where the RosettaMembrane energy function was evaluated for shortcomings. In Chapter 7, the backbones of 13 aquaporin proteins were used to construct sequence and structure symmetric variants of the native proteins. The engineering strategy, circular permutation, is an exhaustive search of all possibilities. Prior to this study, I used one glyceroaquaporin, GlpF, to construct symmetric variants. After several rounds of design (see Appendix J), I was unable to stabilize the symmetric variants in experimental studies. By increasing the amount of backbones used in the engineering strategy in Chapter 7, we hoped to identify symmetric variant candidates that were even more stable than those constructed in previous trials. Symmetric variants were energetically minimized and evaluated using RosettaMembrane. Ultimately, after extensive expression screens, we were unable to confirm the expression of any of the top 20 scoring symmetric variants.

The experiments conducted in Chapter 7 and Appendix J were performed prior to the development of the improved RosettaMembrane energy function in Chapter 5. While it is unclear at this point in time why expression screens failed for all of the top 20 of symmetric variants by Rosetta score, the inaccuracy of the current RosettaMembrane energy function is likely a contributing factor. Studies in Chapters 2 and 3 revealed shortcomings of the RosettaMembrane energy function and display the improvements that are required to continue pushing the field of membrane proteins further.

Membrane proteins require a translocon to assist their insertion into the membrane bilayer. The composition of the translocon varies for each type of organism, but their role as protein-conducting channels remains the same. In order to be recognized by the translocon, membrane proteins contain signal sequences, also known as signal peptides, on the N-terminal side of the protein sequence. Most membrane proteins are inserted into the membrane during translation as opposed to post-translation (Cymer et al, 2015; Rapoport, 2007).

Membrane proteins are able to overcome the energetic cost of partitioning into the membrane interface due to the Hydrogen bonds in the backbone and hydrophobic side chains in trans-membrane helices. The energetic cost for dehydrating the backbone is much higher than the cost of dehydrating non-polar side chains. Thus, formation of secondary structural elements is more favorable than an unfolded protein in the trans-membrane region. The significance of Hydrogen bonds in membrane protein structures can also be shown through denaturation studies where the resulting unfolded states maintain some alpha-helical structure (Cymer et al., 2015;

Wimley, Hristova, Ladokhin, Silvestro, Axelsen & White, 1998; Ladokhin and White, 1999). For multi-spanning membrane proteins and homo-oligomeric complexes, the interactions between trans-membrane spanning helices play a large role in protein folding and stability.

Membrane proteins are proposed to be equilibrium structures and have been found to fold into the correct protein regardless of the assembly pathway. If this is true, then membrane proteins should be able to fold into their native structures independently of the translocon, assuming they are in the lipid bilayer. Several experiments have taken both alpha helical and beta barrel proteins fragments and found that the protein was able to assemble and was indistinguishable from the continuous wild-type protein. This included the use of freeze thaw cycles to disrupt and reform the membrane mimetic. These experiments showed the importance of helix-helix interactions in structures with multiple trans-membrane spanning helices.

Recently, a cell-free system has become popularized for the expression of toxic proteins. The cell-free system consists of only the components that are necessary for protein expression. For membrane proteins, this would mean including membrane mimetics in the cell-free system. However, due to the abundance of membrane mimetics in this particular setup, membrane proteins folding should still be successful without the translocon.

The symmetric variants of aquaporins that were created in Chapter 7 would only contain a signal sequence if the N-terminus was included in the construction of the asymmetric unit. This would indicate that without the inclusion of the signal sequence, symmetric variants would likely not be recognized by the translocon for insertion into the membrane. While it is possible that intact symmetric variant proteins could have been refolded into detergents or lipids, protein degradation, indicated by laddering seen in Western blots, remained as an issue. Another advantage of the cell-free expression system is the exclusion of components unnecessary for protein synthesis. If proteases are indeed the culprits, then expression of the symmetric variants in a cell-free system would be ideal as the proteases would most likely be removed. Finally, one of the native proteins that symmetric variants are based on is a protein native to E. coli. While we are curious about the symmetric architecture of these variants, we do not expect the variants to be functional. Thus, we did not knock-out the endogenous glycerol facilitator protein during the expression of the symmetric variant as it would more than likely result in cell-death. However, the cell-free system does not require the transport of glycerol to continue functioning, therefore it is again ideal for the synthesis of the symmetric variants. Additionally, membrane protein overexpression can be toxic to cells, leading to cell death (Wagner et al., 2007), so the use of a cell-free system is again appealing for the expression of engineered membrane proteins.

8.2 Implications

Ultimately, the work described herein was aimed towards one major goal: improvement in membrane protein structural modeling and design. The implications range from a deeper understanding of the current state of methods to structural insights into mutations linked to diseases. First, the evaluations of RosettaMembrane for design and prediction of mutation-induced stability changes provide a baseline for the current performance of the membrane specific energy function in Rosetta on these specific approaches.

These studies uncovered the strengths and limitations of RosettaMembrane which is important for the development of more accurate methods as well as experimental applications involving mutations and design in addition. For example, in Appendix K, I predicted a number of stabilizing mutations for a membrane protein. In trafficking assays, it was found that the mutation to Asparagine did not traffick to the membrane. Reflecting on findings from Chapter 2, Rosetta designs of membrane proteins had a much higher composition of Arginine as compared to native, indicating that the proposition of Arginine as a mutation may have been the result of a bias towards Arginine in the existing membrane protein energy function. Experimental validation such as this is essential for the continued development and improvement of computational structural biology methods.

Membrane proteins are typically structurally characterized at a lower resolution than that of soluble proteins. Therefore, Chapter 2 evaluated the performance of Rosetta Design on membrane proteins of various resolutions and determined an ideal way to prepare membrane protein structures for design calculations regardless of the resolution of the input structure. Additionally, Chapter 2 sought to establish an ideal protocol for preparing homo-oligomeric membrane proteins for design. These findings were used in application projects that involved the modeling of various membrane proteins (Appendix K), and may continue to be used until higher resolution protein structures are produced, and beyond.

Iterative feedback between computational and experimental methods is key for improving our understanding of membrane proteins. It is difficult to attain high accuracy prediction methods with very little data. In Chapter 1 of my work, I tied a membrane protein energy function based on statistics from known structures along with a design algorithm to predict sequences that have been optimized for its structure. The resulting sequences were compared to sequence compositions of known proteins and pointed to shortcomings in the energy function. For the second part of my work, I used experimentally determined thermostability measurements to test the existing sampling strategy and membrane protein energy function for mutation-induced stability predictions and found a very clear disconnect between biophysical measurements and computational predictions of thermostability. In Chapter 3, I leveraged the existing thermostability calculations to detect which score terms in the membrane protein energy function were contributing to accurate predictions, as well as which terms created noise. This resulted in a preliminary improved energy function, but the energy function only performs as well as the extend that we know of membrane proteins, so much more additional data is required in order to further improve the energy function beyond what is shown in Chapter 4.

The implications for improved predictions of mutation-induced stability changes are far and wide. The prediction of stabilizing mutations in membrane proteins can be a powerful tool for experimental structural characterization studies. Membrane proteins have inherent flexibility; therefore, stabilization of these flexible regions could enable structural studies such as X-ray crystallization. In turn, an increase in membrane protein structures can aid in future computational predictions. Many membrane proteins have had the addition of proteins such as T4 lysozyme to aid in structural characterization. Stabilizing mutations are arguably less detrimental to the overall structure of the protein when compared to the addition of a water-soluble domain.

Moreover, many programs that act to classify variants of unknown significance use sequence-based approaches. Structure-based predictions of changes in energy can aid these purely sequence-based predictions by bridging the gap between sequence and functional outcomes. Structure-based predictions can also help gain some insight regarding the process by which the changes in sequence affect the overall structural integrity of the protein due to the gain or loss of interactions. With the increase of interest in human genome sequence and personalized medicine, rapid and accurate structure-based modeling and predictions are likely to be a key tool in understanding the effects of mutations, as well as a means of screening possible candidates for drugs to overcome the structural defects and treat patients with these variants.

To this matter, the model of the resting VSD, closed pore state of KCNQ1 in Chapter 5 has related insight and implications. KCNQ1 is associated with diseases such as Long-QT syndrome, and is a protein of interest due to the many variants associated with characterized loss or gain-of function. By creating an accurate model of KCNQ1, we can begin to understand what structural effects these mutations have on the protein, thereby altering its function. This information can supplement sequence-based models like that seen in Li et al. and further improve the understanding and characterization of variants of unknown significance.

Finally, the potential implications of the work described in Chapter 7 encompass proposed evolutionary routes as well as protein engineering strategies. First, because the work aims to study specifically inverted topologies in membrane proteins, it could provide evidence of dual-topology and further support the hypothesis that membrane proteins with internal repeats were created through the evolutionary mechanisms of gene duplication, fusion, and diversification. Second, had the study produced a stable unit capable of self-assembling into symmetric complexes, this could provide an approach for creating large stable proteins from smaller, symmetric units, whether as a unit for material science interests or as a stable scaffold for the design of novel therapeutics.

8.3 Future directions

With the rapid increase in the amount and diversity of membrane protein structures in recent years, it follows naturally that the best way to move forward is to leverage this new information to improve existing methods even further. I demonstrated in Chapter 4 that computational predictions can be improved through the use of empirical data and machine learning methods. However, one of the biggest limitations with this study was the deficiency of membrane protein thermostability data. Of the thermostability data available, we selected data points that were representative of the ΔΔG of unfolding, which reduced the dataset to 224 points.

Of the mutations represented, very few involved or were near disulfide bonds. As a result, the four score terms that approximate the energy involved in disulfide bonds were given coefficients of zero or 100 fold larger than other coefficients as most mutations had zero energy attributed with these score terms. Because this interaction was not represented well in the dataset, the machine learning algorithms had very little to learn regarding these score terms. The score terms were removed from the improved energy functions which poses a concern about how these interactions will be accounted for in the event that mutations are near or at the mutation site. Of course, additional thermostability studies involving disulfide bonds in membrane proteins is needed to properly weigh this in the membrane protein energy function; however, because only three such measurements exist currently, it is unlikely that enough measurements will be made in the coming years. To overcome this for the present time, information may be gleaned from thermostability measurements of soluble proteins involving disulfide bonds to provide a means of calculating such an interaction until additional measurements can be made in membrane proteins.

Moreover, since the creation of the RosettaMembrane energy function, a coulombic score term, fa_elec, has replaced the pair potential, which was a crude estimate for electrostatic interactions in proteins. It was proposed in Chapter 2 that the added accuracy from the coulombic score term helped to put the newer default soluble energy function at an advantage for core sequence recoveries when compared to RosettaMembrane. In future studies, the same approach as what was seen in Chapter 4 should be used to generate the dataset with the updated term representing electrostatics, fa_elec. In line with this, a new disulfide score term has been created to encompass the previous four disulfide relevant score terms. In addition to generating weights for representing disulfides using soluble proteins, the updated implementation of a single term should be used.

In addition to the dataset for membrane protein thermostability measurements being sparse, it is further complicated by the fact that it consists of both alpha-helical and beta-barrel membrane proteins. It is proposed that there are differences between the unfolding of these two classes of proteins due to the differences in secondary structure. Although in Chapter 3, parsing of these two types of membrane proteins did not improve correlations, it would be interesting to do machine learning approaches, like those described in Chapter 4, on datasets for alpha-helical and beta-barrel proteins. One could compare the different coefficients resulting from such an experiment as well as which score terms resulted in coefficients of zero. This would give a sense of whether it is necessary at this time to have separate energy functions, or if one energy function for membrane proteins describes interactions broadly enough to produce accurate predictions.

Furthermore, the dataset consisted of ΔΔGs derived from SDS or urea titrations and analytical ultracentrifugation in the case of glycophorin A. Analytical ultracentrifugation is a technique that measures the domain oligomer stability (Hong, Joh, Bowie, & Tamm, 2009). While it is a reversible process, it more so measures the stability of a protein-protein interaction (Fleming, 2016), so perhaps it should be excluded from this study. SDS and urea titration are reversible and as such are the best sources of experimental values of mutation-induced stability changes at this point of time. However, extrapolation from unfolding curves is debated as error prone. The error associated with measurements made from these techniques range from 0.1 to 0.7 kcal/mol. While many measurements of Gibbs free energy ΔG exist for membrane proteins from thermal denaturation experiments, these measurements are performed irreversibly (Hong et al., 2009). For this reason, these data were excluded from the training and test sets in Chapters 3 and 4. However, perhaps in the future they may be utilized as either a Spearman rank correlation or binary classification. In recent years, atomic force microscopy and steric trapping have shown promise in producing accurate and precise measurements within this same range (Edwards & Perkins, 2017; Jefferson et al., 2017). These additional techniques could provide rapid generation of data that is consistent and reliable and therefore better for training by machine learning algorithms.

A membrane protein energy function that is improved through the use of plentiful and diverse empirical data could drastically aid in the accuracy of computational modeling of membrane proteins. From improvements like those described above, the energy function could then be applied to the computational design of membrane proteins. The experimentally derived stability data could be harnessed to predict realistically stabilizing mutations for membrane proteins. This could be done either through the machine learning methods described above or through the development of an additional membrane protein score term that characterizes the findings from experimental stability data in the form bonuses or penalties based on a look up matrix of interactions that have been seen. However, one must be cautioned that the available experimental stability data has a much larger amount of large to small side chains of amino acid mutations compared to small to large mutations. This is in large part due to the success of Alanine scanning studies. Additionally, while sampling possible mutations, Rosetta design utilizes reference energies which are energies that describe the cost of designing in a residue. If the data for a matrix of interactions is too sparse, reference energies could be re-fit to better match what is shown experimentally.

Finally, in addition to harnessing and leveraging available empirical data regarding the structure and stability of membrane proteins, information from sequence can provide insights into design. Evolutionary information, in the form of direct coupling analysis (DCA), identifies sequence positions in evolution from deep sequence alignments that have co-evolved. These positions often times of amino acid identities that are strongly coupled due to compensatory mutations. In other words, a favorable mutation seen in nature is often aided by the mutation of another residue that compensates for the loss or gain of an interaction by the gain or loss of a second interaction in an effort to maintain the overall structural integrity of the protein. This type of evolutionary information could reduce the sequence space of design calculations into a range that is already used by nature, thereby providing predictions that are more native-like and have a higher probability of being successful in experimental settings.

In addition to aiding computational design of membrane proteins in a general sense, DCA would be an ideal approach to finding mutations that could stabilize symmetric backbones of membrane proteins from Chapter 7. While in previous rounds of design, sequences were used to in an attempt to stabilized the engineered proteins (Appendix J), it was through the use of a residue mutation file in Rosetta which allows Rosetta to sample only what is allowed based on sequences rather than favoring mutations seen in sequences. Additionally, these mutations were selected by determining the symmetric counterpart residues in sequences and only allows these during design. An approach using DCA would not only be more elegant, but it would provide a much deeper sequence alignment than what can be done by one person manually. Moreover, the depth of the sequence alignment is key to identifying residues that have been shown to be important throughout evolution in order for the protein to maintain key residue contacts to maintain structural integrity. Thus, this is the most logical step to improving the prediction of stabilizing mutations in membrane proteins.

Future studies should include the expression of symmetric variants from Chapter 7 in the cell-free expression system. Expression of the intact duplicated protein will allow for additional structural studies to understand the relationship between the symmetric sequence and resulting structure. Additionally, these proteins would be placed into liposomes to observe the osmotic function and compare with studies from the wild-type protein. Finally, asymmetric mutations would be done to rescue the function. Whereas expression of the asymmetric unit of the symmetric variants could answer questions about how the protein may have assembled prior to the fusion event. The use of freeze-thaw cycles to disrupt and reassemble lipids could provide clues as to whether the halves are attracted to each other in the inverted topology seen today.

All of the protocols described herein involve the use of RosettaMembrane. In recent years, a new framework for membrane proteins in Rosetta has been developed called RosettaMP. RosettaMP aims to tie the implicit membrane object with the membrane protein model during conformational sampling rather than setting an implicit membrane in cartesian space only to have the protein move out of the membrane during conformational sampling. Currently, the use of the constrain to backbone coordinates option is also used to overcome this issue of having the protein move outside of the membrane when perturbed in a loop region through a lever-arm effect. In the future, creating protocols for RosettaMP from the existing RosettaMembrane protocols will be important for evolving membrane protein modeling in Rosetta. However, RosettaMP is not yet able to work in concert with Rosetta Symmetry, therefore the implementation of both of these modes needs to be reconciled before moving forward because membrane proteins often have homo-oligomeric states that are essential for modeling and predictions.

Recent developments in Rosetta are primarily done using the XML interface (Bender et al., 2016; Fleishman et al., 2011). Rosetta XML scripts provide the user with much more control over the development of the protocol, as was shown in Chapter 6. The community is moving towards the use of Rosetta XML scripts almost exclusively, so protocols are often presented in this format so as to keep up with development. Rosetta ddg_monomer is an application that was exhaustively evaluated and set with optimal sampling conditions, and therefore is difficult to evolve to accommodate new approaches or information. Therefore, a protocol that mirrors the ddg_monomer sampling strategy should be developed in the RosettaScripts format to easily accommodate new developments such as RosettaMP. The performance of such a protocol can then be benchmarked against the performance of the ddg_monomer application to determine if it matches or exceeds the ddg_monomer application so as to establish it as the new standard protocol for mutation-induced stability changes.

Lastly, one of the major limitations of RosettaMembrane is the discrete range of distances that define hydrophobic layers. While this implementation was clever for the time of the initial study, it has been clear that there is a now a need for a flexible membrane bilayer representation. This is for a number of reasons: First, alpha-helical membrane proteins and outer membrane proteins have very different environments, as the membrane spanning region of outer membrane proteins is much smaller than that of alpha-helical membrane proteins. Second, membrane protein structures are characterized in various membrane mimetics of varying thicknesses. The danger in ignoring this is that all structures may be computationally evaluated with the same membrane thickness, but the layers may be very different depending on the protein and the type of membrane mimetics used.

In conclusion, the studies described herein reveal the current state of computational membrane protein modeling for applications of protein design, thermostability calculations, and homology modeling. The relevance of these projects spans from establishing a baseline for further development of computational design methods for membrane proteins and the development and optimization of protocols to classifying stabilizing effect of mutations for diseases. I have established an ideal protocol for preparing membrane protein structures for computational design; evaluated the current membrane protein energy function; improved the membrane protein energy function for mutation-induced stability changes using empirical data; and proposed a computational approach at engineering pseudo-symmetric membrane proteins to be symmetric.

The conclusions drawn from these experiments have demonstrated how to approach modeling such complicated systems of low resolution structures, sparse empirical datasets, low sequence homology, and homo-oligomeric as well as internal pseudo-symmetric structures. The findings have been made possible through the collaboration and feedback of computational and experimental studies. These studies have defined limitations of computational membrane protein structural modeling and predictions and will aid in the careful development of computational membrane protein protocols in the future.

REFERENCES

Alexander, S., Mathie, A., & Peters, J. (2011). ION CHANNELS. *British Journal of Pharmacology, 164*, S137-S174.

Alford, R. F., Koehler Leman, J., Weitzner, B. D., Duran, A. M., Tilley, D. C., Elazar, A., & Gray, J. J. (2015). An Integrated Framework Advancing Membrane Protein Modeling and Design. *PLoS Comput Biol, 11*(9), e1004398. doi:10.1371/journal.pcbi.1004398

Allison, B., Combs, S., DeLuca, S., Lemmon, G., Mizoue, L., & Meiler, J. (2014). Computational design of protein-small molecule interfaces. *Journal of Structural Biology, 185*, 193-202. doi:10.1016/j.jsb.2013.08.003

Andrade, M. A., Perez-iratxeta, C., & Ponting, C. P. (2001). Protein Repeats : Structures , Functions , and Evolution. *Journal of Structural Biology, 131*, 117-131. doi:10.1006/jsbi.2001.4392

André, I., Strauss, C. E. M., Kaplan, D. B., Bradley, P., & Baker, D. (2008). Emergence of symmetry in homooligomeric biological assemblies. *Proceedings of the National Academy of Sciences of the United States of America, 105*, 16148-16152. doi:10.1073/pnas.0807576105

Arinaminpathy, Y., Khurana, E., Engelman, D. M., & Gerstein, M. B. (2009). Computational analysis of membrane proteins: the largest class of drug targets. *Drug discovery today, 14*, 1130-1135. doi:10.1016/j.drudis.2009.08.006

Baker, R. P., & Urban, S. (2012). Architectural and thermodynamic principles underlying intramembrane protease function. *Nat. Chem. Biol., 8*, 759-768.

Barro-Soria, R., Rebolledo, S., Liin, S. I., Perez, M. E., Sampson, K. J., Kass, R. S., & Larsson, H. P. (2014). KCNE1 divides the voltage sensor movement in KCNQ1/KCNE1 channels into two steps. *Nat Commun, 5*, 3750. doi:10.1038/ncomms4750

Barth, P., Schonbrun, J., & Baker, D. (2007). Toward high-resolution prediction and design of transmembrane helical protein structures. *Proceedings of the National Academy of Sciences of the United States of America, 104*, 15682-15687. doi:10.1073/pnas.0702515104

Bender, B. J., Cisneros, A., 3rd, Duran, A. M., Finn, J. A., Fu, D., Lokits, A. D., Mueller, B.K., Sangha, A.K., Sauer, M.F., Sevy, A.M., Sliwoski, G., Sheehan, J.H., DiMaio, F., Meiler, J., Moretti, R. (2016). Protocols for Molecular Modeling with Rosetta3 and RosettaScripts. *Biochemistry, 55*(34), 4748-4763. doi:10.1021/acs.biochem.6b00444

Berliner, N., Teyra, J., Çolak, R., Lopez, S. G., & Kim, P. M. (2014). Combining structural modeling with ensemble machine learning to accurately predict protein fold stability and binding affinity effects upon mutation. *PLoS One, 9*. doi:10.1371/journal.pone.0107353

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I.N., Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research, 28*, 235-242.

Binz, H. K., Amstutz, P., & Plückthun, A. (2005). Engineering novel binding proteins from nonimmunoglobulin domains. *Nature biotechnology, 23*, 1257-1268. doi:10.1038/nbt1127

Blaber, M., Lee, J., & Longo, L. (2012). Emergence of symmetric protein architecture from a simple peptide motif: evolutionary models. *Cellular and molecular life sciences : CMLS*. doi:10.1007/s00018-012-1077-3

Blin, S., Soussia, I., Kim, E.-J., Brau, F., Kang, D., Lesage, F., & Bichet, D. (2016). Mixing and matching TREK/TRAAK subunits generate heterodimeric K2P channels with unique properties. *Proc Natl Acad Sci U S A, 113*, 4200-4205.

Bogdanov, M., & Dowhan, W. (2012). Lipid-dependent generation of dual topology for a membrane protein. *J Biol Chem, 287*(45), 37939-37948. doi:10.1074/jbc.M112.404103

Bogdanov, M., Xie, J., Heacock, P., & Dowhan, W. (2008). To flip or not to flip: lipid-protein charge interactions are a determinant of final membrane protein topology. *J Cell Biol, 182*(5), 925-935. doi:10.1083/jcb.200803097

Bokil, N., Baisden, J., Radford, D., & Summers, K. (2010). Molecular genetics of long QT syndrome. *Mol Genet Metab, 101*, 1-8.

Borgo, B., & Havranek, J. J. (2012). Automated selection of stabilizing mutations in designed and natural proteins. *Proc Natl Acad Sci U S A, 109*(5), 1494-1499. doi:10.1073/pnas.1115172109

Bowie, J. U. (2001). Stabilizing membrane proteins. *Current opinion in structural biology, 11*, 397-402.

Bowie, J. U. (2006). Flip-flopping membrane proteins. *Nature structural & molecular biology, 13*, 94-96. doi:10.1038/nsmb0206-94

Bowie, J. U. (2013). Structural biology. Membrane protein twists and turns. *Science (New York, N.Y.), 339*, 398-399. doi:10.1126/science.1228655

Brych, S. R., Dubey, V. K., Bienkiewicz, E., Lee, J., Logan, T. M., & Blaber, M. (2004). Symmetric Primary and Tertiary Structure Mutations within a Symmetric Superfold : A Solution , not a Constraint , to Achieve a Foldable Polypeptide. *J. Mol. Biol., 344*, 769-780. doi:10.1016/j.jmb.2004.09.060

Calamini, B., & Morimoto, R. (2012). Protein Homeostasis as a Therapeutic Target for Diseases of Protein Conformation. *Curr Top Med Chem, 12*(22), 2623-2640.

Cao, Z., Schlebach, J. P., Park, C., & Bowie, J. U. (2012). Thermodynamic stability of bacteriorhodopsin mutants measured relative to the bacterioopsin unfolded state. *Biochim Biophys Acta, 1818*(4), 1049-1054. doi:10.1016/j.bbamem.2011.08.019

Capriotti, E., Fariselli, P., & Casadio, R. (2005). I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Res, 33*(Web Server issue), W306-310. doi:10.1093/nar/gki375

Carugo, O., & Pongor, S. (2008). A normalized root-mean-spuare distance for comparing protein three-dimensional structures. *Protein Science, 10,* 1470-1473. doi:10.1110/ps.690101

Casadio, R., Vassura, M., Tiwari, S., Fariselli, P., & Luigi Martelli, P. (2011). Correlating disease-related mutations to their effect on protein stability: a large-scale analysis of the human proteome. *Hum Mutat, 32*(10), 1161-1170. doi:10.1002/humu.21555

Chen, K.-Y. M., Zhou, F., Fryszczyn, B. G., & Barth, P. (2012). Naturally evolved G protein-coupled receptors adopt metastable conformations. *Proc Natl Acad Sci U S A, 109,* 13284-13289. doi:10.1073/pnas.1205512109

Chen, V., Arendall, W. r., Headd, J., Keedy, D., Immormino, R., Kapral, G., . . . Richardson, D. (2010). MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Crystallographica, D66,* 12-21.

Choi, S., Jeon, J., Yang, J.-S., & Kim, S. (2008). Common occurrence of internal repeat symmetry in membrane proteins. *Proteins, 71,* 68-80. doi:10.1002/prot.21656

Choi, Y., Sims, G. E., Murphy, S., Miller, J. R., & Chan, A. P. (2012). Predicting the Functional Effect of Amino Acid Substitutions and Indels. *PLoS One, 7.* doi:10.1371/journal.pone.0046688

Clayton, G., Altieri, S., Heginbotham, L., Unger, V., & Morais-Cabral, J. (2008). Structure of the transmembrane regions of a bacterial cyclic nucleotide-regulated channel. *Proc Natl Acad Sci U S A, 105*(5), 1511-1515.

Combs, S. a., Deluca, S. L., Deluca, S. H., Lemmon, G. H., Nannemann, D. P., Nguyen, E. D., Willis, J.R., Sheehan, J.H., Meiler, J. (2013). Small-molecule ligand docking into comparative models with Rosetta. *Nature protocols, 8,* 1277-1298. doi:10.1038/nprot.2013.074

Conn, P. J., Lindsley, C. W., & Jones, C. K. (2008). Activation of metabotropic glutamate receptors as a novel approach for the treatment of schizophrenia. *Trends in Pharmacological Sciences, 30,* 25-31. doi:10.1016/j.tips.2008.10.006

Conway, P., Tyka, M. D., DiMaio, F., Konerding, D. E., & Baker, D. (2014). Relaxation of backbone bond geometry improves protein energy landscape modeling. *Protein Science, 23*, 47-55. doi:10.1002/pro.2389

Crisman, T. J., Qu, S., Kanner, B. I., & Forrest, L. R. (2009). Inward-facing conformation of glutamate transporters as revealed by their inverted-topology structural repeats. *Proceedings of the National Academy of Sciences of the United States of America, 106*, 20752-20757. doi:10.1073/pnas.0908570106

Cuesta-Seijo, J. A., Neale, C., Moktar, J., Tran, C. D., Bishop, R. E., Pomes, R., & Prive, G. G. (2010). PagP crystallized from SDS/cosolvent reveals the route for phospholipid access to the hydrocarbon ruler. *Structure, 18*, 1210-1219.

Cui, J. (2016). Voltage-Dependent Gating: Novel Insights from KCNQ1 Channels. *Biophys J, 110*(1), 14-25. doi:10.1016/j.bpj.2015.11.023

Cymer, F., von Heijne, G., & White, S. H. (2015). Mechanisms of integral membrane protein insertion and folding. *J Mol Biol, 427*(5), 999-1022. doi:10.1016/j.jmb.2014.09.014

Davis, I. W., Leaver-Fay, A., Chen, V., Block, J., Kapral, G., Wang, X., Murray, L.W., Arendall, W.B., Snoeyink, J., Richardson, J.S., Richardson, D. (2007). MolProbity: all-atom contacts and structure validation for proteins and nucleic acids. *Nucleic Acids Res, 35*, W375-W383.

de Beer, T., Berka, K., Thornton, J. M., & Laskowski, R. (2014). PDBsum additions. *Nucleic Acids Res, 42*, D292-D296.

Deluca, S., Dorr, B., & Meiler, J. (2011). Design of native-like proteins through an exposure-dependent environment potential. *Biochemistry, 50*, 8521-8528. doi:10.1021/bi200664b

Dimaio, F., Leaver-fay, A., Bradley, P., Baker, D., & Andre, I. (2011). Modeling Symmetric Macromolecular Structures in. *PLoS One, 6*. doi:10.1371/journal.pone.0020450

Doyle, D. a. (1998). The Structure of the Potassium Channel: Molecular Basis of K+ Conduction and Selectivity. *Science, 280*, 69-77. doi:10.1126/science.280.5360.69

Drew, D., Sjöstrand, D., Nilsson, J., Urbig, T., Chin, C.-n., de Gier, J.-W., & von Heijne, G. (2002). Rapid topology mapping of Escherichia coli inner-membrane proteins by prediction and PhoA/GFP fusion analysis. *Proceedings of the National Academy of Sciences of the United States of America, 99*, 2690-2695. doi:10.1073/pnas.052018199

Duran, A. M., & Meiler, J. (2013). Inverted Topologies in Membrane Proteins : A Mini-Review. *Computational and Structural Biotechnology Journal, 8*. doi:10.1038/sj.

Duran, A.M., & Meiler, J. (2017). Computational design of membrane proteins using RosettaMembrane. Protein Science, 27:341-355. doi:10.1002/pro.3335

Dutzler, R., Campbell, E. B., Cadene, M., Chait, B. T., & MacKinnon, R. (2002). X-ray structure of a ClC chloride channel at 3.0 A reveals the molecular basis of anion selectivity. *Nature, 415*, 287-294. doi:10.1038/415287a

Eberly, L. (2007). Multiple Linear Regression. *Topics in Biostatistics, 404*, 165-187.

Edwards, D. T., & Perkins, T. T. (2017). Optimizing force spectroscopy by modifying commercial cantilevers: Improved stability, precision, and temporal resolution. *J Struct Biol*, *197*(1), 13-25. doi:10.1016/j.jsb.2016.01.009

Eisenbeis, S., & Höcker, B. (2010). Evolutionary mechanism as a template for protein engineering. *Journal of peptide science : an official publication of the European Peptide Society, 16*, 538-544. doi:10.1002/psc.1233

Eisenbeis, S., Proffitt, W., Coles, M., Truffault, V., Shanmugaratnam, S., Meiler, J., & Höcker, B. (2012). Potential of fragment recombination for rational design of proteins. *Journal of the American Chemical Society, 134*, 4019-4022. doi:10.1021/ja211657k

Eriksson, K., Urszula, Fischer, G., Friemann, R., Enkavi, G., Tajkhorshid, E., & Neutze, R. (2013). Subangstrom Resolution X-Ray Structure Details Aquaporin-Water Interactions. *Science, 340*(6138), 1346-1349.

Faham, S., Yang, D., Bare, E., Yohannan, S., Whitelegge, J. P., & Bowie, J. U. (2004). Side-chain contributions to membrane protein structure and stability. *Journal of Molecular Biology, 335*, 297-305. doi:10.1016/j.jmb.2003.10.041

Fleishman, S. J., Leaver-Fay, A., Corn, J. E., Strauch, E.-M., Khare, S. D., Koga, N., Ashworth, J., Murphy, P., Richter, F., Lemmon, G., Meiler, J., Baker, D. (2011). RosettaScripts: a scripting language interface to the Rosetta macromolecular modeling suite. *PLoS One, 6*, e20161. doi:10.1371/journal.pone.0020161

Fleishman, S. J., Whitehead, T. A., Ekiert, D. C., Dreyfus, C., & Jacob, E. (2012). Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science*, *332*, 816-821. doi:10.1126/science.1202617.Computational

Fleming, K. G., Ackerman, A. L., & Engelman, D. M. (1997). The effect of point mutations on the free energy of transmembrane alpha-helix dimerization. *J. Mol. Biol., 272*, 266-275.

Fleming, K. G., & Engleman, D. (2001). Specificity in transmembrane helix-helix interactions can define a hierarchy of stability for sequence variants. *Proc Natl Acad Sci U S A, 98*, 14340-14344.

Fleming, K.G. (2016). Applications of Analytical Ultracentrifugation to Membrane Proteins. *Analytical Ultracentrifugation*, doi:10.1007/978-4-431-55985-6_15

Folkman, L., Stantic, B., Sattar, A., & Zhou, Y. (2016). EASE-MM: Sequence-Based Prediction of Mutation-Induced Stability Changes with Feature-Based Multiple Models. *Journal of Molecular Biology, 428*, 1394-1405. doi:10.1016/j.jmb.2016.01.012

Forrest, L. R., & Rudnick, G. (2009). The rocking bundle: a mechanism for ion-coupled solute flux by symmetrical transporters. *Physiology (Bethesda, Md.), 24*, 377-386. doi:10.1152/physiol.00030.2009

Forrest, L. R., Zhang, Y.-W., Jacobs, M. T., Gesmonde, J., Xie, L., Honig, B. H., & Rudnick, G. (2008). Mechanism for alternating access in neurotransmitter transporters. *Proceedings of the National Academy of Sciences of the United States of America, 105*, 10338-10343. doi:10.1073/pnas.0804659105

Fortenberry, C., Bowman, E. A., Proffitt, W., Dorr, B., Combs, S., Harp, J., Mizoue, L., Meiler, J. (2011). Exploring symmetry as an avenue to the computational design of large protein domains. *Journal of the American Chemical Society, 133*, 18026-18029. doi:10.1021/ja2051217

Fosmo, A. L., & Skraastad, O. B. (2017). The Kv7 Channel and Cardiovascular Risk Factors. *Front Cardiovasc Med, 4*, 75. doi:10.3389/fcvm.2017.00075

Fu, D. (2000). Structure of a Glycerol-Conducting Channel and the Basis for Its Selectivity. *Science, 290*, 481-486. doi:10.1126/science.290.5491.481

Fu, D., Libson, A., Miercke, L. J., Wietzman, C., Nollert, P., Krucinski, J., & Stroud, R. M. (2000). Structure of a Glycerol-Conducting Channel and the Basis for Its Selectivity. *Science, 290*, 481-486. doi:10.1126/science.290.5491.481

Gerlt, J. a. (2000). New wine from old barrels. *Nature structural biology, 7*, 171-173. doi:10.1038/73249

Goujon, M., McWilliam, H., Li, W., Valentin, F., Squizzato, S., Paern, J., & Lopez, R. (2010). A new bioinformatics analysis tools framework at EMBL-EBI. *Nucleic Acids Res, 38*(Web Server issue), W695-699. doi:10.1093/nar/gkq313

Granseth, E. (2010). Dual-topology : one sequence , two topologies. Structural Bioinformatics of Membrane Proteins. 137-150 (2010).

Guerois, R., Nielsen, J. E., & Serrano, L. (2002). Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. *Journal of Molecular Biology, 320*(2), 369-387. doi:10.1016/s0022-2836(02)00442-4

Heijman, J., & Dobrev, D. (2017). Ion channels as part of macromolecular multiprotein complexes. *Herzschrittmacherther Elektrophysiol*. doi:10.1007/s00399-017-0542-y

Henrion, U., Renhorn, J., Borjesson, S. I., Nelson, E. M., Schwaiger, C. S., Bjelkmar, P., Wallner, B., Lindahl, E., Elinder, F. (2012). Tracking a complete voltage-sensor cycle with metal-ion bridges. *Proc Natl Acad Sci U S A, 109*(22), 8552-8557. doi:10.1073/pnas.1116938109

Hirai, T., Heymann, A. W., Maloney, P. C., & Subramaniam, S. (2003). Structural Model for 12-Helix Transporters Belonging to the Major Facilitator Superfamily. *JOURNAL OF BACTERIOLOGY, 185*, 1712-1718. doi:10.1128/JB.185.5.1712

Hoang, T. X., Trovato, A., Seno, F., Banavar, J. R., & Maritan, A. (2004). Geometry and symmetry presculpt the free-energy landscape of proteins. *Proceedings of the National Academy of Sciences of the United States of America, 101*, 7960-7964. doi:10.1073/pnas.0402525101

Hohl, M., Briand, C., Grütter, M. G., & Seeger, M. a. (2012). Crystal structure of a heterodimeric ABC transporter in its inward-facing conformation. *Nature structural & molecular biology, 19*, 395-402. doi:10.1038/nsmb.2267

Hong, H., Park, S., Flores Jimenez, R., Rinehart, D., & Tamm, L. (2007). Role of Aromatic Side Chains in the Folding and Thermodynamic Stability of Integral Membrane Proteins. *J. Am. Chem. Soc., 129*(26), 8320-8327.

Hong, H., Joh, N.H., Bowie, J.U., Tamm, L.K. (2009). Methods for Measuring the Thermodynamic Stability of Membrane Proteins. *Methods in Enzymology*, 455, 213-236

Hopf, T. A., Colwell, L. J., Sheridan, R., Rost, B., Sander, C., & Marks, D. S. (2012). Three-Dimensional Structures of Membrane Proteins from Genomic Sequencing. *Cell, 149*, 1607-1621. doi:10.1016/j.cell.2012.04.012

Huysmans, G. H., Baldwin, S. A., Brockwell, D. J., & Radford, S. E. (2010). The transition state for folding of an outer membrane protein. *Proc Natl Acad Sci U S A, 107*, 10174-10177.

Jefferson, R. E., Min, D., Corin, K., Wang, J. Y., & Bowie, J. U. (2017). Applications of Single-Molecule Methods to Membrane Protein Folding Studies. *J Mol Biol*. doi:10.1016/j.jmb.2017.05.021

Jensen, M., Jogini, V., Borhani, D., Leffler, A., Dror, R., & Shaw, D. (2012). Mechanism of Voltage Gating in Potassium Channels. *Science, 336*, 229-233.

Jiang, L., Althoff, E. A., Clemente, F. R., Doyle, L., Rothlisberger, D., Zanghellini, A., Gallaher, J.L., Betker, J.L., Tanaka, F., Barbas, C.F., Hilvert, D., Houk, K.N., Stoddard, B.L., Baker, D. (2008). De Novo Computational Design of Retro-Aldol Enzymes. *Science, 319*, 1387-1391. doi:10.1126/science.1152692

Joachimiak, L. A., Kortemme, T., Stoddard, B. L., & Baker, D. (2006). Computational Design of a New Hydrogen Bond Network and at Least a 300-fold Specificity Switch at a Protein-Protein Interface. *Journal of Molecular Biology, 361*, 195-208. doi:10.1016/j.jmb.2006.05.022

Joh, N. H., Min, A., Faham, S., Whitelegge, J. P., Yang, D., Woods, V. L., & Bowie, J. U. (2008). Modest stabilization by most hydrogen-bonded side-chain interactions in membrane proteins. *Nature, 453*(7199), 1266-1270. doi:10.1038/nature06977

Joh, N. H., Oberai, A., Yang, D., Whitelegge, J., & Bowie, J. U. (2009). Similar energetic contributions of packing in the core of membrane and water-soluble proteins. *J Am Chem Soc, 131*, 10846-10847.

Jura, N., Endres, N. F., Engel, K., Deindl, S., Das, R., & Lamers, M. H. (2009). Mechanism for Activation of the EGF Receptor Catalytic Domain by the Juxtamembrane Segment. *Cell, 137*, 1293-1307. doi:10.1016/j.cell.2009.04.025

Kaufmann, K. W., Lemmon, G. H., Deluca, S. L., Sheehan, J. H., & Meiler, J. (2010). Practically useful: what the Rosetta protein modeling suite can do for you. *Biochemistry, 49,* 2987-2998. doi:10.1021/bi902153g

Kellogg, E. H., Leaver-Fay, A., & Baker, D. (2011). Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins: Structure, Function and Bioinformatics, 79,* 830-838. doi:10.1002/prot.22921

Khafizov, K., Staritzbichler, R., Stamm, M., & Forrest, L. R. (2010). A study of the evolution of inverted-topology repeats from LeuT-fold transporters using AlignMe. *Biochemistry, 49*, 10702-10713. doi:10.1021/bi101256x

Khatun, J., Khare, S. D., & Dokholyan, N. V. (2004). Can contact potentials reliably predict stability of proteins? *J Mol Biol, 336*(5), 1223-1238. doi:10.1016/j.jmb.2004.01.002

King, N. P., Bale, J. B., Sheffler, W., McNamara, D. E., Gonen, S., Gonen, T., Yeates, T.O., Baker, D. (2014). Accurate design of co-assembling multi-component protein nanomaterials. *Nature, 510,* 103-108. doi:10.1038/nature13404

King, N. P., & Lai, Y.-t. (2013). Practical approaches to designing novel protein assemblies. *Current Opinion in Structural Biology, 23,* 632-638. doi:10.1016/j.sbi.2013.06.002

King, N. P., Sheffler, W., Sawaya, M. R., Vollmar, B. S., Sumida, J. P., André, I., Gonen, T., Yeates, T.O., Baker, D. (2012). Computational design of self-assembling protein nanomaterials with atomic level accuracy. *Science (New York, N.Y.), 336*, 1171-1174. doi:10.1126/science.1219364

Kintzer, A. F., & Stroud, R. M. (2016). Structure, inhibition and regulation of two-pore channel TPC1 from Arabidopsis thaliana. *Nature*, *531*(7593), 258-262. doi:10.1038/nature17194

Knowles, T. P., Vendruscolo, M., & Dobson, C. M. (2014). The amyloid state and its association with protein misfolding diseases. *Nat Rev Mol Cell Biol, 15*(6), 384-396. doi:10.1038/nrm3810

Konagurthu, A. S., Whisstock, J. C., Stuckey, P. J., & Lesk, A. M. (2006). MUSTANG: a multiple structural alignment algorithm. *Proteins, 64*(3), 559-574. doi:10.1002/prot.20921

Korkegian, A., Black, M., Baker, D., & Stoddard, B. L. (2012). Computational thermostabilization of an enzyme. *308,* 857-860. doi:10.1126/science.1107387.Computational

Kortemme, T., Joachimiak, L. a., Bullock, A. N., Schuler, A. D., Stoddard, B. L., & Baker, D. (2004). Computational redesign of protein-protein interaction specificity. *Nature structural & molecular biology, 11*, 371-379. doi:10.1038/nsmb749

Kozma, D., Simon, I., & Tusnády, G. E. (2013). PDBTM: Protein Data Bank of transmembrane proteins after 8 years. *Nucleic acids research, 41,* D524-529. doi:10.1093/nar/gks1169

Krishnamurthy, H., Piscitelli, C. L., & Gouaux, E. (2009). Unlocking the molecular secrets of sodium-coupled transporters. *Nature, 459*, 347-355. doi:10.1038/nature08143

Kroncke, B. M., Duran, A. M., Mendenhall, J. L., Meiler, J., Blume, J. D., & Sanders, C. R. (2016). Documentation of an Imperative to Improve Methods for Predicting Membrane Protein Stability. *Biochemistry, 55*, 5002-5009. doi:10.1021/acs.biochem.6b00537

Kroncke, B. M., Vanoye, C. G., Meiler, J., George, A. L., & Sanders, C. R. (2015). Personalized biochemistry and biophysics. *Biochemistry, 54,* 2551-2559. doi:10.1021/acs.biochem.5b00189

Kuhlman, B., & Baker, D. (2000). Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences, 97,* 10383-10388. doi:10.1073/pnas.97.19.10383

Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L., & Baker, D. (2003). Design of a novel globular protein fold with atomic-level accuracy. *Science (New York, N.Y.), 302*, 1364-1368. doi:10.1126/science.1089427

Labro, A. J., Boulet, I. R., Choveau, F. S., Mayeur, E., Bruyns, T., Loussouarn, G., Raes, A.L., Snyders, D. J. (2011). The S4-S5 linker of KCNQ1 channels forms a structural scaffold with the S6 segment controlling gate closure. *J Biol Chem, 286*(1), 717-725. doi:10.1074/jbc.M110.146977

Ladokhin, A.S., White, S.H. (1999). Folding of amphipathic alpha-helices on membrane: Energetics of helix formation by melittin. *J Mol Biol,* 285, 1363-9.

Leaver-Fay, A., Jacak, R., Stranges, P. B., & Kuhlman, B. (2011). A generic program for multistate protein design. *PLoS One, 6*(7), e20937. doi:10.1371/journal.pone.0020937

Leaver-Fay, A., O'Meara, M. J., Tyka, M., Jacak, R., Song, Y., Kellogg, E. H., Thompson, J., Davis, I.W., Pache, R.A., Lyskov, S., Gray, J.J., Kortemme, T., Richardson, J.S., Havranek, J.J., Snoeyink, J., Baker, D., Kuhlman, B. (2013). Scientific benchmarks for guiding macromolecular energy function improvement. *Methods in Enzymology, 523*, 109-143. doi:10.1016/B978-0-12-394292-0.00006-0

Lemmon, M. a., Flanagan, J. M., Hunt, J. F., Adair, B. D., Bormann, B. J., Dempsey, C. E., & Engelman, D. M. (1992). Glycophorin A dimerization is driven by specific interactions between transmembrane alpha-helices. *The Journal of biological chemistry, 267*, 7683-7689.

Lemmon, M. A., & Schlessinger, J. (2010). Cell Signaling by Receptor Tyrosine Kinases. *Cell, 141*, 1117-1134. doi:10.1016/j.cell.2010.06.011

Levin, E. J., Quick, M., & Zhou, M. (2009). Crystal structure of a bacterial homologue of the kidney urea transporter. *Nature, 462*, 757-761. doi:10.1038/nature08558

Levy, E. D., Boeri Erba, E., Robinson, C. V., & Teichmann, S. a. (2008). Assembly reflects evolution of protein complexes. *Nature, 453*, 1262-1265. doi:10.1038/nature06942

Li, Q., Wanderling, S., Paduch, M., Medovoy, D., Singharoy, A., McGreevy, R., Villalba-Galea, C.A., Hulse, R.E., Roux, B., Schulten, K., Kossiakoff, A., Perozo, E. (2014). Structural mechanism of voltage-dependent gating in an isolated voltage-sensing domain. *Nat Struct Mol Biol, 21*(3), 244-252. doi:10.1038/nsmb.2768

Lise, S., Archambeau, C., Pontil, M., & Jones, D. T. (2009). Prediction of hot spot residues at protein-protein interfaces by combining machine learning and energy-based methods. *BMC Bioinformatics, 10*, 365. doi:10.1186/1471-2105-10-365

Lolkema, J. S., Dobrowolski, A., & Slotboom, D.-J. (2008). Evolution of antiparallel two-domain membrane proteins: tracing multiple gene duplication events in the DUF606 family. *Journal of Molecular Biology, 378*, 596-606. doi:10.1016/j.jmb.2008.03.005

Loll, P. J. (2003). Membrane protein structural biology: The high throughput challenge. *Journal of Structural Biology, 142*, 144-153. doi:10.1016/S1047-8477(03)00045-5

Lomize, A. L., Pogozheva, I. D., Lomize, M. a., & Mosberg, H. I. (2006). Positioning of proteins in membranes: a computational approach. *Protein science : a publication of the Protein Society, 15*, 1318-1333. doi:10.1110/ps.062126106

Lomize, M. a., Pogozheva, I. D., Joo, H., Mosberg, H. I., & Lomize, A. L. (2012). OPM database and PPM web server: resources for positioning of proteins in membranes. *Nucleic acids research, 40*, D370-376. doi:10.1093/nar/gkr703

Lynch, M. (2000). The Evolutionary Fate and Consequences of Duplicate Genes. *Science, 290,* 1151-1155. doi:10.1126/science.290.5494.1151

MacKenzie, K. R., Prestegard, J. H., & Engleman, D. (1997). A transmembrane helix dimer: structure and implications. *Science, 276*(5309), 131-133.

Madej, M. G., Dang, S., Yan, N., & Kaback, H. R. (2013). Evolutionary mix-and-match with MFS transporters. doi:10.1073/pnas.1303538110/-/DCSupplemental.www.pnas.org/cgi/doi/10.1073/pnas.1303538110

Marshall, S. A., Vizcarra, C. L., & Mayo, S. L. (2005). One- and two-body decomposable Poisson-Boltzmann methods for protein design calculations. *Protein Science, 14*, 1293-1304. doi:10.1110/ps.041259105.4

McLachlan, A. D. (1972). Gene Duplication in Carp Muscle Calcium Binding Protein. *Nature New Biology, 240*, 83-85.

Meisenzahl, E. M., Schmitt, G. J., Scheuerecker, J., & Möller, H. (2007). The role of dopamine for the pathophysiology of schizophrenia. *International Review of Psychiatry, 19*, 337-345. doi:10.1080/09540260701502468

Moon, C. P., & Fleming, K. G. (2011). Side-chain hydrophobicity scale derived from transmembrane protein folding into lipid bilayers. *Proc Natl Acad Sci U S A, 108*(25), 10174-10177. doi:10.1073/pnas.1103979108

Morrison, E. a., DeKoster, G. T., Dutta, S., Vafabakhsh, R., Clarkson, M. W., Bahl, A., Kern, D., Ha, T., Henzler-Wildman, K. A. (2012). Antiparallel EmrE exports drugs by exchanging between asymmetric structures. *Nature, 481*, 45-50. doi:10.1038/nature10703

Moss, A. J., & Kass, R. S. (2005). Long QT syndrome : from channels to cardiac arrhythmias. *Journal of Clinical Investigation, 115*. doi:10.1172/JCI25537.2018

Neumann, J., Klein, N., Otzen, D. E., & Schneider, D. (2014). Folding energetics and oligomerization of polytopic alpha-helical transmembrane proteins. *Arch Biochem Biophys, 564*, 281-296. doi:10.1016/j.abb.2014.07.017

Norholm, M. H., Light, S., Virkki, M. T., Elofsson, A., von Heijne, G., & Daley, D. O. (2012). Manipulating the genetic code for membrane protein production: what have we learnt so far? *Biochim Biophys Acta, 1818*(4), 1091-1096. doi:10.1016/j.bbamem.2011.08.018

O'Meara, M. J., Leaver-Fay, A., Tyka, M. D., Stein, A., Houlihan, K., Dimaio, F., Bradley, P., Kortemme, T., Baker, D., Snoeyink, J., Kuhlman, B. (2015). Combined covalent-electrostatic model of hydrogen bonding improves structure prediction with Rosetta. *Journal of Chemical Theory and Computation, 11*, 609-622. doi:10.1021/ct500864r

Ortiz, A. R., & Strauss, C. E. M. (2002). MAMMOTH (Matching molecular models obtained from theory): An automated method for model comparison. *Protein Science, 11*, 2606-2621. doi:10.1110/ps.0215902.next

Otzen, D. E. (2011). Mapping the folding pathway of the transmembrane protein DsbB by protein engineering. *Protein Eng., Des. Sel., 24*, 139-149.

Overington, J., Al-Lazikani, B., & Hopkins, A. (2006). How many drug targets are there? *Nat Rev Drug Discov, 5*, 993-996.

Paslawski, W., Lillelund, O. K., Kristensen, J. V., Schafer, N. P., Baker, R. P., Urban, S., & Otzen, D. E. (2015). Cooperative folding of a polytopic alpha-helical membrane protein involves a compact N-terminal nucleus and nonnative loops. *Proc Natl Acad Sci U S A, 112*, 7978-7983.

Pautsch, A., & Schulz, G. E. (2000). High-resolution structure of the OmpA membrane domain. *J. Mol. Biol., 298*, 273-282.

Payandeh, J., Scheuer, T., Zheng, N., & Catterall, W. A. (2011). The crystal structure of a voltage-gated sodium channel. *Nature, 475*, 353-358. doi:10.1038/nature10238

Pellegrini-Calace, M., Maiwald, T., & Thornton, J. M. (2009). PoreWalker: a novel tool for the identification and characterization of channels in transmembrane proteins from their three-dimensional structure. *PLoS Comput Biol, 5*(7), e1000440. doi:10.1371/journal.pcbi.1000440

Perez-aguilar, J. M., & Saven, J. G. (2012). Computational Design of Membrane Proteins. *Structure/Folding and Design, 20*, 5-14. doi:10.1016/j.str.2011.12.003

Pettit, F. K., Bare, E., Tsai, A., & Bowie, J. U. (2007). HotPatch: a statistical approach to finding biologically relevant features on protein surfaces. *Journal of Molecular Biology, 369*, 863-879. doi:10.1016/j.jmb.2007.03.036

Pires, D. E. V., Ascher, D. B., & Blundell, T. L. (2014a). DUET: A server for predicting effects of mutations on protein stability using an integrated computational approach. *Nucleic Acids Research, 42*, 1-6. doi:10.1093/nar/gku411

Pires, D. E. V., Ascher, D. B., & Blundell, T. L. (2014b). MCSM: Predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics, 30*, 335-342. doi:10.1093/bioinformatics/btt691

Pokala, N., & Handel, T. M. (2004). Energy functions for protein design I : Efficient and accurate continuum electrostatics and solvation. *Protein Science, 13*, 925-936. doi:10.1110/ps.03486104.where

Popot, J.-L., & Engelman, D. M. (2000). Helical Membrane Protein Folding Stability and Evolution. *Annu. Rev. Biochem, 69*, 881-922.

The PyMol Molecular Graphics System. Version 1.8.6.0, Schrodinger, LLC.

Ramachandran, R., Tweten, R. K., & Johnson, A. E. (2004). Membrane-dependent conformational changes initiate cholesterol-dependent cytolysin oligomerization and intersubunit beta-strand alignment. *Nature structural & molecular biology, 11*, 697-705. doi:10.1038/nsmb793

Rapoport, T.A. (2007). Protein translocation across the eukaryotic endoplasmic reticulum and bacterial plasma membranes. *Nature*, 450, 663-9.

Rapp, M., Granseth, E., Seppälä, S., & von Heijne, G. (2006). Identification and evolution of dual-topology membrane proteins. *Nature structural & molecular biology, 13*, 112-116. doi:10.1038/nsmb1057

Rapp, M., Seppälä, S., Granseth, E., & von Heijne, G. (2007). Emulating membrane protein evolution by rational design. *Science (New York, N.Y.), 315*, 1282-1284. doi:10.1126/science.1135406

Reddy, V. S., Shlykov, M. a., Castillo, R., Sun, E. I., & Saier, M. H. (2012). The major facilitator superfamily (MFS) revisited. *The FEBS journal, 279*, 2022-2035. doi:10.1111/j.1742-4658.2012.08588.x

Remmert, M., Biegert, a., Linke, D., Lupas, a. N., & Söding, J. (2010). Evolution of outer membrane beta-barrels from an ancestral beta beta hairpin. *Molecular biology and evolution, 27*, 1348-1358. doi:10.1093/molbev/msq017

Rohl, C. a., Strauss, C. E. M., Misura, K. M. S., & Baker, D. (2004). Protein structure prediction using Rosetta. *Methods in enzymology, 383*, 66-93. doi:10.1016/S0076-6879(04)83004-0

Röthlisberger, D., Khersonsky, O., Wollacott, A. M., Jiang, L., DeChancie, J., Betker, J., Gallaher, J.L., Althoff, E.A., Zanghellini, A., Dym, O., Albeck, S., Houk, K.N., Tawfik, D.S., Baker, D. (2008). Kemp elimination catalysts by computational enzyme design. *Nature, 453*, 190-195. doi:10.1038/nature06879

Sanders, C. R., & Myers, J. K. (2004). Disease-related misassembly of membrane proteins. *Annu. Rev. Biophys Biomol Struct, 33*, 25-51.

Sanders, C. R., & Sönnichsen, F. (2006). Solution NMR of membrane proteins: Practice and challenges. *Magnetic Resonance in Chemistry, 44*, 24-40. doi:10.1002/mrc.1816

Schlebach, J. P., Woodall, N. B., Bowie, J. U., & Park, C. (2014). Bacteriorhodopsin folds through a poorly organized transition state. *Journal of the American Chemical Society, 136*, 16574-16581. doi:10.1021/ja508359n

Schmidt, C., & Peyronnet, R. (2018). Voltage-gated and stretch-activated potassium channels in the human heart : Pathophysiological and clinical significance. *Herzschrittmacherther Elektrophysiol*. doi:10.1007/s00399-017-0541-z

Schueler-furman, O. (2005). R EVIEW Progress in Modeling of Protein Structures and Interactions. *Science, 638*. doi:10.1126/science.1112160

Schuldiner, S. (2009). EmrE, a model for studying evolution and mechanism of ion-coupled transporters. *Biochimica et biophysica acta, 1794*, 748-762. doi:10.1016/j.bbapap.2008.12.018

Senes, A. (2011). Computational design of membrane proteins. *Current Opinion in Structural Biology, 21*, 460-466. doi:10.1016/j.sbi.2011.06.004

Senes, A., Chadi, D. C., Law, P. B., Walters, R. F. S., Nanda, V., & Degrado, W. F. (2007). E z , a Depth-dependent Potential for Assessing the Energies of Insertion of Amino Acid Side-chains into Membranes : Derivation and Applications to Determining the Orientation of Transmembrane and Interfacial Helices. *J. Mol. Biol., 366*, 436-448. doi:10.1016/j.jmb.2006.09.020

Sevy, A. M., Jacobs, T. M., Crowe, J. E., Jr., & Meiler, J. (2015). Design of Protein Multi-specificity Using an Independent Sequence Search Reduces the Barrier to Low Energy Sequences. *PLoS Comput Biol, 11*(7), e1004300. doi:10.1371/journal.pcbi.1004300

Shi, Z., & Moult, J. (2011). Structural and functional impact of cancer-related missense somatic mutations. *J Mol Biol, 413*(2), 495-512. doi:10.1016/j.jmb.2011.06.046

Shimizu, T., Mitsuke, H., Noto, K., & Arai, M. (2004). Internal gene duplication in the evolution of prokaryotic transmembrane proteins. *Journal of Molecular Biology, 339*, 1-15. doi:10.1016/j.jmb.2004.03.048

Siegel, J. B., Zanghellini, A., Lovick, H. M., Kiss, G., Lambert, A. R., St Clair, J. L. , Gallaher, J.L., Hilvert, D., Gelb, M.H., Stoddard, B.L., Houk, K.N., Michael, F.E., Baker, D. (2010). Computational Design of an Enzyme Catalyst for a Stereoselective Bimolecular Diels-Alder Reaction. *Science, 329*, 309-313.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Soding, J., Thompson, J.D., Higgins, D. G. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol Syst Biol, 7*, 539. doi:10.1038/msb.2011.75

Smith, J. A., Vanoye, C. G., George, A. L., Jr., Meiler, J., & Sanders, C. R. (2007). Structural Models for the KCNQ1 Voltage-Gated Potassium Channel. *Biochemistry, 46*, 14141-14152.

Snijder, H. J., Ubarretxena-Belandia, I., Blaauw, M., Kalk, K. H., Verheij, H. M., Egmond, M. R., & Dijkstra, B. W. (1999). Structural evidence for dimerization-regulated activation of an integral membrane phospholipase. *Nature, 401*, 717-721.

Söding, J., & Lupas, A. N. (2003). More than the sum of their parts: on the evolution of proteins from peptides. *BioEssays : news and reviews in molecular, cellular and developmental biology, 25*, 837-846. doi:10.1002/bies.10321

Song, Y., DiMaio, F., Wang, R. Y., Kim, D., Miles, C., Brunette, T., Thompson, J., Baker, D. (2013). High-resolution comparative modeling with RosettaCM. *Structure, 21*(10), 1735-1742. doi:10.1016/j.str.2013.08.005

Stamm, M., Staritzbichler, R., Khafizov, K., & Forrest, L. R. (2013). Alignment of helical membrane protein sequences using AlignMe. *PLoS One, 8*, e57731. doi:10.1371/journal.pone.0057731

Stefl, S., Nishi, H., Petukh, M., Panchenko, A. R., & Alexov, E. (2013). Molecular mechanisms of disease-causing missense mutations. *J Mol Biol, 425*(21), 3919-3936. doi:10.1016/j.jmb.2013.07.014

Stenson, P. D., Ball, E. V., Mort, M., Phillips, A. D., Shaw, K., & Cooper, D. N. (2012). The Human Gene Mutation Database (HGMD) and its exploitation in the fields of personalized genomics and molecular evolution. *Curr Protoc Bioinformatics, Chapter 1*, Unit1 13. doi:10.1002/0471250953.bi0113s39

Strauch, E.-M., Fleishman, S. J., & Baker, D. (2014). Computational design of a pH-sensitive IgG binding protein. *Proceedings of the National Academy of Sciences of the United States of America, 111*, 675-680. doi:10.1073/pnas.1313605111

Stroud, R. M., Miercke, L. J., O’Connell, J., Khademi, S., Lee, J. K., Remis, J., Harries, W., Robles, Y., Akhavan, D. (2003). Glycerol facilitator GlpF and the associated aquaporin family of channels. *Current Opinion in Structural Biology, 13*, 424-431. doi:10.1016/S0959-440X(03)00114-3

Sun, J., & MacKinnon, R. (2017). Cryo-EM Structure of a KCNQ1/CaM Complex Reveals Insights into Congenital Long QT Syndrome. *Cell, 169*, 1042-1050. doi:10.1016/j.cell.2017.05.019

Tan, S., Hwee, T. T., & Chung, M. C. M. (2008). Membrane proteins and membrane proteomics. *Proteomics, 8*, 3924-3932. doi:10.1002/pmic.200800597

Thornton, J. M., Orengo, C. a., Todd, a. E., & Pearl, F. M. (1999). Protein folds, functions and evolution. *Journal of Molecular Biology, 293*, 333-342. doi:10.1006/jmbi.1999.3054

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B, 58*(1), 267-288.

Tinberg, C. E., Khare, S. D., Dou, J., Doyle, L., Nelson, J. W., Schena, A., Jankowski, W., Kalodimos, C.G., Johnsson, K., Stoddard, B.L., Baker, D. (2013). Computational design of ligand-binding proteins with high affinity and selectivity. *Nature, 501*, 212-216. doi:10.1038/nature12443

Valastyan, J. S., & Lindquist, S. (2014). Mechanisms of protein-folding diseases at a glance. *Dis Model Mech, 7*(1), 9-14. doi:10.1242/dmm.013474

Viklund, H., & Elofsson, A. (2008). OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar. *Bioinformatics (Oxford, England), 24*, 1662-1668. doi:10.1093/bioinformatics/btn221

Vinogradova, O., Badola, P., Czerski, L., Sönnichsen, F. D., & Sanders, C. R. (1997). Escherichia coli diacylglycerol kinase: a case study in the application of solution NMR methods to an integral membrane protein. *Biophysical journal, 72*, 2688-2701. doi:10.1016/S0006-3495(97)78912-4

Vinothkumar, K. R., Strisovsky, K., Andreeva, A., Christova, Y., Verhelst, S., & Freeman, M. (2010). The structural basis for catalysis and substrate specificity of a rhomboid protease. *EMBO J., 29*, 3797-3809.

Vitrac, H., Bogdanov, M., & Dowhan, W. (2013). In vitro reconstitution of lipid-dependent dual topology and postassembly topological switching of a membrane protein. *Proceedings of the National Academy of Sciences*, 1-6. doi:10.1073/pnas.1304375110

Vizcarra, C. L., Zhang, N., Marshall, S. A., Wingreen, N. E. D. S., Zeng, C., & Mayo, S. L. (2007). An Improved Pairwise Decomposable Finite-Difference Poisson – Boltzmann Method for Computational Protein Design. doi:10.1002/jcc

von Heijne, G. (2006). Membrane-protein topology. *Nature reviews. Molecular cell biology, 7*, 909-918. doi:10.1038/nrm2063

von Heijne, G., & Gavel, Y. (1988). Topogenic signals in integral membrane proteins. *European journal of biochemistry / FEBS, 174*, 671-678.

Wagner, S., Bader, M. L., Drew, D., & de Gier, J.-W. (2006). Rationalizing membrane protein overexpression. *Trends in biotechnology, 24*, 364-371. doi:10.1016/j.tibtech.2006.06.008

Wagner,S., Baars, L., Ytterberg, A.J., Klussmeier, A., Wagner, C.S., Nord, O., Nygren, P., van Wijk, K.J., de Gier, J.-W. (2007). Consequences of Membrane Protein Overexpression in Escherichia coli. *Molecular & Cellular Proteomics*, 6, 1527-1550.

Walters, R. F. S., & Degrado, W. F. (2006). Helix-packing motifs in membrane proteins. *Proceedings of the National Academy of Sciences, 103*, 13658-13663.

Wang, Q., Curran, M. E., Splawski, I., Burn, T. C., Millholland, J. M., VanRaay, T. J., Shen, J., Timothy, K.W., Vincent, G.M., de Jager, T., Schwartz, P.J., Toubin, J.A., Moss, A.J., Atkinson, D.L., Landes, G.M., Connors, T.D., Keating, M. T. (1996). Positional cloning of a novel potassium channel gene: KVLQT1 mutations cause cardiac arrhythmias. *Nature genetics, 12*, 17-23.

Wang, Z., & Moult, J. (2001). SNPs, Protein Structure, and Disease. *Human Mutation, 17*, 263-270.

White, S. H. Membrane Proteins of Known 3D Structure. http://blanco.biomol.uci.edu/mpstruc/

White, S. H. (2004). The progress of membrane protein structure determination. *Protein Science, 13*, 1948-1949. doi:10.1110/ps.04712004.Figure

Wiener, M. C. (2004). A pedestrian guide to membrane protein crystallization. *Methods, 34*, 364-372. doi:10.1016/j.ymeth.2004.03.025

Willis, J. R., Briney, B. S., DeLuca, S. L., Crowe, J. E., Jr., & Meiler, J. (2013). Human germline antibody gene segments encode polyspecific antibodies. *PLoS Comput Biol, 9*(4), e1003045. doi:10.1371/journal.pcbi.1003045

Wimley W.C., Hristova, K., Ladokhin, K.S., Silvestro, L., Axelsen, P.H., White, S.H. (1998). Folding of beta-sheet membrane proteins: A hydrophobic hexapeptide model. *J Mol Biol,* 277, 1091-110.

Wolynes, P. G. (1996). Symmetry and the energy landscapes of biomolecules. *Proceedings of the National Academy of Sciences of the United States of America, 93*, 14249-14255.

Worth, C. L., Preissner, R., & Blundell, T. L. (2011). SDM - A server for predicting effects of mutations on protein stability and malfunction. *Nucleic Acids Research, 39*, 215-222. doi:10.1093/nar/gkr363

Wu, D., Delaloye, K., Zaydman, M. A., Nekouzadeh, A., Rudy, Y., & Cui, J. (2010). State-dependent electrostatic interactions of S4 arginines with E1 in S2 during Kv7.1 activation. *J Gen Physiol, 135*(6), 595-606. doi:10.1085/jgp.201010408

Wu, H., Wang, C., Gregory, K. J., Han, G. W., Cho, H. P., Xia, Y., Niswender, C.M., Katritch, V., Meiler, J., Cherezov, V., Conn, P.J., Stevens, R. C. (2014). Structure of a Class C

GPCR Metabotropic Glutamate Receptor 1 Bound to an Allosteric Modulator. *Science, 344*, 58-65.

Wu, J., Ding, W., & Horie, M. (2016). Molecular pathogenesis of long QT syndrome type 1. *J Arrhythm, 32*, 381-388.

Yamashita, A., Singh, S. K., Kawate, T., Jin, Y., & Gouaux, E. (2005). Crystal structure of a bacterial homologue of Na+/Cl--dependent neurotransmitter transporters. *Nature, 437*, 215-223. doi:10.1038/nature03978

Yang, Y., Chen, B., Tan, G., Vihinen, M., & Shen, B. (2013). Structure-based prediction of the effects of a missense variant on protein stability. *Amino Acids, 44*, 847-855. doi:10.1007/s00726-012-1407-7

Yarov-Yarovoy, V., Schonbrun, J., & Baker, D. (2006). Multipass membrane protein structure prediction using Rosetta. *Proteins, 62*, 1010-1025. doi:10.1002/prot.20817

Yohannan, S., Yang, D., Faham, S., Boulting, G., Whitelegge, J., & Bowie, J. U. (2004). Proline substitutions are not easily accommodated in a membrane protein. *J Mol Biol, 341*(1), 1-6. doi:10.1016/j.jmb.2004.06.025

Yu, F., Yarov-Yarovoy, V., Gutman, G., & Catterall, W. A. (2005). Overview of Molecular Relationships in the Voltage-Gated Ion Channel Superfamily. *Pharmacol Rev, 57*, 387-395. doi:10.1124/pr.57.4.13

Yue, P., Li, Z., & Moult, J. (2005). Loss of protein structure stability as a major causative factor in monogenic disease. *Journal of Molecular Biology, 353*, 459-473. doi:10.1016/j.jmb.2005.08.020

Zanghellini, A., Jiang, L., Wollacott, A. M., Cheng, G., Meiler, J., Althoff, E. A., Rothlisberger, D., Baker, D. (2006). New algorithms and an in silico benchmark for computational enzyme design. *Protein Sci, 15*(12), 2785-2794. doi:10.1110/ps.062353106

Zaydman, M. A., & Cui, J. (2014). PIP2 regulation of KCNQ channels: biophysical and molecular mechanisms for lipid modulation of voltage-dependent gating. *Frontiers in Physiology, 5*(195), 1-11. doi:10.3389/fphys.2014.00195

Zaydman, M. A., Silva, J. R., Delaloye, K., Li, Y., Liang, H., Larsson, H. P., Shi, J., Cui, J. (2013). Kv7.1 ion channels require a lipid to couple voltage sensing to pore opening. *Proc Natl Acad Sci U S A, 110*(32), 13180-13185. doi:10.1073/pnas.1305167110

Zhou, Y., & Bowie, J. U. (2000). Building a thermostable membrane protein. *The Journal of biological chemistry, 275*, 6975-6979.

Zhou, Y., Cierpicki, T., Jimenez, R. H., Lukasik, S. M., Ellena, J. F., Cafiso, D. S., Kadokura, H., Beckwith, J., Bushweller, J. H. (2008). NMR solution structure of the integral membrane enzyme DsbB: functional insights into DsbB-catalyzed disulfide bond formation. *Mol Cell, 31*(6), 896-908. doi:10.1016/j.molcel.2008.08.028

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic-net. *Journal of the Royal Statistical Society, 67,* 301-320. doi:10.1111/j.1467-9868.2005.00503.x

APPENDIX A

PROTOCOL CAPTURE FOR CHAPTER 1

(Rosetta Design workshop protocol)

This protocol capture was included in the supplemental materials for the publication.

Bender*, Cisneros*, Duran*, Finn*, Fu*, Lokits*, Mueller*, Sangha*, Sauer*, Sevy*, Sliwoski, Sheehan, DiMaio, Meiler, and Moretti, 2016 (*Authors contributed equally)

Contribution: I developed the entire protocol capture for Protein Design that included cross-over with RosettaMembrane and Rosetta Symmetry in an XML environment. Additionally, I have included a modified protocol that uses RosettaMP.

Protein Design Protocol Capture

`fixed width text means you should type the command into your terminal`

If you want to try making files that already exist (e.g., input files), write them to a different directory! (mkdir my_dir) (NOTE: For many of the commands you will be using for this tutorial, remove 's before hitting enter. Otherwise you will get an error.)

Objective: In this exercise, we will examine the Rosetta design features by mutating user-specified residues. The membrane protein we will be using is a homo-dimer, so we will employ RosettaMembrane and Rosetta Symmetry to model the dimer during design. RosettaScripts will be used to combine the two applications. In the additional notes at the end, you will find an adaptation of the design protocol from Step 3 using RosettaMP.

Rosetta Applications: RosettaDesign, RosettaMembrane, RosettaMP, Rosetta Symmetry, RosettaScripts

Input and Analysis Scripts: `clean_pdb.py, get_fasta_from_pdb.py`

Tutorial

Preparation: Locate the necessary input PDB file.

```
    cd ~/rosetta_workshop/tutorials/protein_design
```

Included in this folder is a PDB file downloaded from the Protein Data Bank (www.rcsb.org ID:3UKM). Open this in pymol to familiarize yourself with the structure:

```
    pymol 3UKM.pdb
```

You should notice that this file shows two homo-dimers. We will focus on the dimer made from Chains A and B (lower dimer when loaded). This will be important when setting up the symmetry definition file in the next step. Close pymol and proceed to step 1.

1. Setting up the symmetric PDB
    1. Rosetta Symmetry. In this step, we will create the proper symmetry definition file for this particular protein structure. We will need the input structure from the preparation step.
    2. `cd Step1_symm`
    3.
       ```
       cp ../3UKM.pdb .
       ```

       (this copies the pdb file to the Step1 directory)

       Next, we will use a perl script in Rosetta to generate a symmetry file from the input crystal structure. First, if you'd like to display the available options for this script, simply enter:

       ```
       ~/rosetta_workshop/rosetta/main/source/src/apps/public/symmetry/m
       ake_symmdef_file.pl
       ```

       Next, we will use non-crystallographic mode (NCS), Chain A as the reference, Chain B as an interacting chain, and include the input structure. The output will be redirected, using the greater than sign, into a new file called 3UKM.symm.

       ```
       ~/rosetta_workshop/rosetta/main/source/src/apps/public/symmetry/m
       ake_symmdef_file.pl \
       -m NCS -a A -i B -p 3UKM.pdb > 3UKM.symm
       ```

       The perl script will generate a couple of outputs:

       - 3UKM_INPUT.pdb = chain A
       - 3UKM.kin
       - 3UKM_model_AB.pdb = model generated to show subunit interactions with the input
       - 3UKM_symm.pdb = model generated to show the full point group symmetry
       - 3UKM.symm = symmetry definition file that you just created

       Examine symmetry file equation. `gedit 3UKM.symm`

4. Next, we will use clean_pdb.py to prepare the input protomer for setting up symmetry.

   ```
   ~/rosetta_workshop/rosetta/tools/protein_tools/scripts/clean_pdb.
   py 3UKM A
   ```

   clean_pdb.py strips PDB code that Rosetta can not parse such as comments, anisotropic atom positions, unnatural amino acid types, and waters. The first argument in the script is the 4-letter PDB code and the second argument is a string containing the chains to return, in this case, only chain A.

5. Now, we will use the clean input structure to test the symmetry definition file. We will accomplish this through a very basic use of RosettaScripts. While still in the same directory:

   ```
   gedit setup_symm.xml
   ```

   And look at the contents of the file, which should look like this:

   ```
   <ROSETTASCRIPTS>
       <SCOREFXNS>
       </SCOREFXNS>
       <TASKOPERATIONS>
       </TASKOPERATIONS>
       <FILTERS>
       </FILTERS>
       <MOVERS>
         <SetupForSymmetry name="setup_symm" definition="3UKM.symm"
   />
       </MOVERS>
       <APPLY_TO_POSE>
       </APPLY_TO_POSE>
       <PROTOCOLS>
         <Add mover_name="setup_symm" />
       </PROTOCOLS>
   </ROSETTASCRIPTS>
   ```

   Next, run this protocol using RosettaScripts. We applied the setup_symm protocol to the input structure, 3UKM_A.pdb.

   ```
   ~/rosetta_workshop/rosetta/main/source/bin/rosetta_scripts.defaul
   t.linuxgccrelease \
   -parser:protocol setup_symm.xml -s 3UKM_A.pdb -out:prefix
   setupsymm_
   ```

   When Rosetta is finished, examine the output structure using pymol:

   ```
   pymol setupsymm_3UKM_A_0001.pdb
   ```

Does the resulting structure look as you would expect? Sometimes you have to make manual adjustments to the symmetry definition file by paying careful attention to the jumps. In this case, it looks great. Before we move forward, examine the score file generated from setting up symmetry:

```
gedit setupsymm_score.sc
```

The total energy score of the protein is the first number. For this protein, you will probably see a number in the positive 6000s. We know that this is not a good Rosetta score for a protein. Before moving on to an application such as design, it is recommended to energetically minimize the structure in some way to improve the imperfections in the crystal structure.

Additionally, this score is based on the default Rosetta scoring function. We will need to create a span file and add the membrane high resolution scoring function into our XML script.

6. We need to create a span file which will tell Rosetta where the membrane-spanning region is on our protein. Step 1.2 outputs a fasta file.

   ```
   cat 3UKM_A.fasta
   ```

   In a web browser, go to octopus.cbr.su.se and paste the fasta sequence into the form. Then click "Submit OCTOPUS" (There is also an option to use SPOCTOPUS which considers signal peptide sequences).

   When it's done running, near the top it will say "A text version of the topology prediction can be found in the OCTOPUS topology file (txt)" click on that link.

   Select all of the text and copy.

   `gedit 3UKM.topo` paste the text into this file and save.

   A script in Rosetta will take this topo file named 3UKM.topo and create a span file named 3UKM.span:

   ```
   ~/rosetta_workshop/rosetta/main/source/src/apps/public/membrane_a
   binitio/octopus2span.pl \
   3UKM.topo > 3UKM.span
   ```

2. Energy minimization of the input structure

Relax is a common protocol used in Rosetta to minimize protein structures. Typically, 100 relax models is sufficient to find a low-energy structure as an input model.

I have provided output for this step in the `Step2_relax` directory. I have also described the approaches to use for analysis of relaxed structures.

Please make sure you are in the `Step2_relax` directory. You can create relaxed structures in a similar way that we set up symmetry, using RosettaScripts. View this by opening symm_relax.xml:

```
 <ROSETTASCRIPTS>
 <SCOREFXNS>
   <mem_highres weights="membrane_highres_Menv_smooth.wts" symmetric=1
/>
 </SCOREFXNS>
 <TASKOPERATIONS>
   <InitializeFromCommandline name=ifcl/>
   <RestrictToRepacking name=rtr />
 </TASKOPERATIONS>
 <FILTERS>
 </FILTERS>
 <MOVERS>
   <SetupForSymmetry name=setup_symm definition=3UKM.symm />
   <FastRelax name=fast_rlx scorefxn=mem_highres repeats=8
task_operations=ifcl,rtr />
 </MOVERS>
 <APPLY_TO_POSE>
 </APPLY_TO_POSE>
 <PROTOCOLS>
   <Add mover_name=setup_symm />
   <Add mover_name=fast_rlx />
 </PROTOCOLS>
 </ROSETTASCRIPTS>
```

Then, run using the command-line:

```
~/rosetta_workshop/rosetta/main/source/bin/rosetta_scripts.default.linu
xgccrelease \
        -parser:protocol symm_relax.xml -s 3UKM_A.pdb \
        -in:file:spanfile 3UKM.span -membrane:no_interpolate_Mpair \
        -membrane:Membed_init -membrane:Menv_penalties \
        -score:weights membrane_highres_Menv_smooth.wts \
        -restore_pre_talaris_2013_behavior \
        -extra_res_fa
~/rosetta_workshop/rosetta/main/database/chemical/residue_type_sets/fa_
standard/residue_types/rosetta_specific/INV_VRT.params
```

Analyze the output. There are a few ways of going about this. Some may look at just the best scoring models. Others calculate the RMSD of the relaxed models to the input structure and plot the Score vs. RMSD to find the best (lowest) scoring model that is most similar to the input structure.

Ideally, the lowest scoring model would also have the lowest RMSD. This model should be used in all subsequent steps in redesign. Generally in design, we use an ensemble of structures accounting for the lowest cluster of RMSD's and scores.

3. Prepare files for protein design at user-specified residues.
    1. With an energy minimized input structure, we are almost ready to design our protein! In this step, we will first combine SetupForSymmetry and SymPackRotamers movers in another RosettaScripts protocol.

    2.
       ```
       cd ../Step3_design
       ls
       ```

       You should see several input files ready for you to use. Here you will find the symmetry definition file, an energy minimized input structure named `Best_rlx_3UKM_A.pdb`, and an XML file.

       ```
       gedit symm_design.xml
       ```

       Here I have provided the required XML file to complete this task:

       ```
       <ROSETTASCRIPTS>
           <SCOREFXNS>
             <mem_highres weights="membrane_highres_Menv_smooth.wts"
       symmetric="1" />
           </SCOREFXNS>
           <TASKOPERATIONS>
             <InitializeFromCommandline name="ifcl"/>
           </TASKOPERATIONS>
           <FILTERS>
           </FILTERS>
           <MOVERS>
             <SetupForSymmetry name="setup_symm" definition="3UKM.symm"
       />
             <SymPackRotamersMover name="sym_pack"
       scorefxn="mem_highres" task_operations="ifcl"/>
           </MOVERS>
           <APPLY_TO_POSE>
           </APPLY_TO_POSE>
           <PROTOCOLS>
             <Add mover_name="setup_symm" />
             <Add mover_name="sym_pack" />
           </PROTOCOLS>
       </ROSETTASCRIPTS>
       ```

       Notice under SCOREFXN, the membrane high-resolution weights are specified. Read the XML and see if you understand the different sections. Reference the lecture slides if you need to. Exit out of the file when you are done. Notice the command-line below has additional options `-restore_pre_talaris_2013_behavior` and `-extra_res_fa` along with a path. These are required because we are using RosettaMembrane which uses a scoring function based on pre-talaris score terms and weights. Modeling membrane proteins in Rosetta is currently in flux, so some protocols work best with RosettaMembrane, while others have transitioned to RosettaMP (see additional notes). Now run design. This step should take about 30 minutes.

```
~/rosetta_workshop/rosetta/main/source/bin/rosetta_scripts.defaul
t.linuxgccrelease \
-parser:protocol symm_design.xml -s Best_rlx_3UKM_A.pdb -
in:file:spanfile 3UKM.span \
-membrane:no_interpolate_Mpair -membrane:Membed_init \
-membrane:Menv_penalties -score:weights
membrane_highres_Menv_smooth.wts \
-restore_pre_talaris_2013_behavior \
-extra_res_fa
~/rosetta_workshop/rosetta/main/database/chemical/residue_type_se
ts/fa_standard/residue_types/rosetta_specific/INV_VRT.params \
-out:prefix full_design_ -nstruct 1
```

In the interest of time, we will only do one full design of the protein. In the output folder, I have included 20 output structures.

3. Now we will run design again, but this time we will guide design with a resfile. A resfile is a file that is read by RosettaScripts during design. The file specifies a residue number, chain ID, and a command associated with the particular residue. This command alerts the packer with how to deal with the residue. (see slides on resfile for more examples)

   Design is done on a fixed backbone. Today we will use a hypothetical situation where a number of residues will be simply re-packed (minimized side-chains) by the command NATAA. A small number of residues will have a specific group of amino acids to choose from during design, and two residues will consider all amino acid rotamers during design.

   ```
   gedit 3UKM.resfile
   ```

   The resfile should resemble this:

   ```
   NATAA
   start

   29 A ALLAA
   30 A PIKAA P
   31 A ALLAA
   39 A PIKAA LIY
   46 A PIKAA FL
   52 A PIKAA C
   58 A PIKAA LKIY
   ```

```
61 A PIKAA FLI
62 A APOLAR
65 A PIKAA VI
66 A PIKAA LVS
67 A POLAR
68 A PIKAA A
69 A APOLAR
70 A PIKAA NRGK
72 A PIKAA AGV
80 A ALLAA
84 A ALLAA
85 A ALLAA
86 A APOLAR
87 A APOLAR
88 A PIKAA AGVIL
94 A PIKAA TIV
95 A PIKAA TIV
96 A PIKAA AGV
97 A PIKAA YFLI
98 A PIKAA AGV
99 A PIKAA HNYD
100 A ALLAA
```

Based on sequence alignments from homologous proteins, we know that these positions prefer a certain type of amino acid. We are going to use a resfile to guide Rosetta during design. Look at your lecture slides and understand which amino acid rotamers will be allowed at each position. When you are comfortable with the format, exit the file.

Now, we will create an XML file that will read in the resfile. First, copy the current XML file and rename it symm_res_design.xml, then open the file

```
cp symm_design.xml symm_res_design.xml
gedit symm_res_design.xml
```

Next, find the section labeled TASKOPERATIONS. Insert this task operation underneath the `<TASKOPERATION>` line and before the `</TASKOPERATIONS>` line so that it is in line with the:

```
 <ReadResfile name="rrf" filename= "3UKM.resfile" />
```

Notice, we gave this task the name "rrf". Find the SymPackRotamersMover under and add "rrf" after the task operations tag so it resembles this:

```
<SymPackRotamersMover name="sym_pack" scorefxn="mem_highres"
task_operations="ifcl,rrf"/>
```

4. Design the protein at user-specified residues. We have a relaxed input structure, a symmetry definition file, a resfile to direct design, and an XML protocol to setup symmetry, and design according to a resfile. We are now ready to move forward with design! Run this command:
5. `~/rosetta_workshop/rosetta/main/source/bin/rosetta_scripts.default.linuxgccrelease \`
6. `-parser:protocol symm_res_design.xml -s Best_rlx_3UKM_A.pdb -in:file:spanfile 3UKM.span \`
7. `-membrane:no_interpolate_Mpair -membrane:Membed_init \`
8. `-membrane:Menv_penalties -score:weights membrane_highres_Menv_smooth.wts \`
9. `-restore_pre_talaris_2013_behavior \`
10. `-extra_res_fa ~/rosetta_workshop/rosetta/main/database/chemical/residue_type_sets/fa_standard/residue_types/rosetta_specific/INV_VRT.params \`
    `-out:prefix resfile_design_ -nstruct 2`

    Again, many, many more structures than just 2 should be made for production runs. In the interest of time, we will just run 2 for today. This should take about 2 minutes. This step will simply ensure that you can successfully run Rosetta Symmetry and Design. Use the output structures provided in the Step3_design/output folder for the analysis step. Note that this folder contains only 20 models. In your own experiments, you will likely want to make more than just 20 models.

11. Analysis of Designs. Now that we have a few design structures, we want to examine one of the regions we designed. First, we must sort the top five structures by score. You should still be in the Step3_design directory.

    ```
    cd ./output/resfile_design
    ls

    grep pose resfile_design*.pdb | sort -nk 23 | head
    ```

    This shows you the top 10 structures by best score. We can use awk to store the list of the top 10.

    ```
    grep pose resfile_design*.pdb | sort -nk 23 | head | \

    awk '{print(substr($1,1,length($1)-5))}' > best.list
    ```

    Next, we will use awk to automate generating fastas for each of our top models. (NOTE: Make this all one line and remove \'s before hitting enter for this command!)

    ```
    cat best.list | awk '{system( \
    ```

```
"python2.7
~/rosetta_workshop/rosetta/tools/protein_tools/scripts/get_fasta_
from_pdb.py \
"$1" A "substr($1,1,length($1)-3)"fasta")}'
```

Now we can cat all of the fastas and use WebLogo to generate a figure to show our designed residues.

```
cat *.fasta > all_fasta.txt
```

```
cat all_fasta.txt
```

(If you are running out of time, you can cd into ../Step4_analysis where the fastas of the top 10 models for each design experiment are included)

Now, copy and paste the text into the WebLogo server weblogo.berkeley.edu/logo.cgi

Under advanced logo options, choose Logo Range to be 80-100. Now Click Create Logo at the Bottom.

If you need to, you can re-open the resfile you used in the design step to see if Rosetta Design did what you expected.

For example: Residue 94 should be T, I, or V, and residue 86 could be any apolar residue.

Since we have restricted design a lot, we expect to see single identities for these positions in this sequence logo.

If you have enough time, you can go back and make a sequence logo over this same range for the full design output. Compare the logos. You should see quite a bit more variation in the full design sequence logo.

Additional notes:

Protein Design Analysis. A script named `Deep_Analysis` is available as an alternative to the WebLogo server. It is in `~/rosetta_workshop/rosetta/tools/protein_tools/scripts/deep_analysis`. There are many options such as using fastas or pdbs as your input. You can also pass a resfile to specify which regions you want to appear on the logo (instead of a single range).

Rosetta Design using the Rosetta Membrane Framework. The steps to setup Rosetta to use the Membrane Framework are slightly different than Membrane Mode. To properly use span information throughout the protocol, one must use the appropriate movers `<AddMembraneMover>` and `<MembranePositionFromTopologyMover>` before setting up `<PackRotamersMover>`. For simplicity, we will treat the protein as monomeric. In the future, symmetry and the membrane framework will be more compatible.

From the main `protein_design` directory change directories into `mpframework_design`

```
cd ./mpframework_design
```

Then open the file mpf_design.xml

```
<ROSETTASCRIPTS>
<SCOREFXNS>
  <memb_hires weights="mpframework_smooth_fa_2012.wts" />
</SCOREFXNS>
<TASKOPERATIONS>
  <InitializeFromCommandline name=ifcl/>
</TASKOPERATIONS>
<FILTERS>
</FILTERS>
<MOVERS>
  <AddMembraneMover name=add_memb />
  <MembranePositionFromTopologyMover name=init_pose />
  <PackRotamersMover name=pack scorefxn=memb_hires task_operations=ifcl />
</MOVERS>
<PROTOCOLS>
  <Add mover=add_memb />
  <Add mover=init_pose />
  <Add mover=pack />
</PROTOCOLS>
</ROSETTASCRIPTS>
```

To run, use the following command-line:

```
~/rosetta_workshop/rosetta/main/source/bin/rosetta_scripts.default.linuxgccre
lease \
    -parser:protocol mpf_design.xml -s 3UKM_A.pdb \
    -mp:setup:spanfiles 3UKM.span -mp:scoring:hbond -nstruct 1 \
    -in:ignore_unrecognized_res -packing:pack_missing_sidechains false \
    -score:weights mpframework_smooth_fa_2012.wts
```

APPENDIX B

SUPPLEMENTAL INFORMATION FOR CHAPTER 2

Table AB.1. Raw counts of amino acids in RosettaMembrane design for the monomeric set. Tabular data to support main text Figure 2.3. Total counts of amino acids represented at the surface, core, and total protein are reported for all native and selected RosettaMembrane designs analyzed from the monomeric set.

	Counts									Percentages					
	Surface			Core			Total			Surface		Core		Total	
	Correct	Native	Design	Correct	Native	Design	Correct	Native	Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design
ALA	32	525	35	417	1108	764	651	2394	1151	6.1	91.4	37.6	54.6	27.2	56.6
CYS	0	29	0	0	81	0	0	162	0	0.0	0.0	0.0	0.0	0.0	0.0
ASP	149	328	304	20	61	57	241	540	529	45.4	49.0	32.8	35.1	44.6	45.6
GLU	90	308	127	25	176	38	203	783	288	29.2	70.9	14.2	65.8	25.9	70.5
PHE	60	381	65	255	535	326	616	1836	860	15.7	92.3	47.7	78.2	33.6	71.6
GLY	288	407	358	727	1107	756	1360	2223	1481	70.8	80.4	65.7	96.2	61.2	91.8
HIS	11	77	40	50	108	187	139	351	382	14.3	27.5	46.3	26.7	39.6	36.4
ILE	269	682	502	336	630	509	1131	2223	2065	39.4	53.6	53.3	66.0	50.9	54.8
LYS	309	541	550	2	65	2	453	981	726	57.1	56.2	3.1	100.0	46.2	62.4
LEU	417	531	2173	503	751	849	2155	2826	5970	78.5	19.2	67.0	59.2	76.3	36.1
MET	1	141	24	195	422	366	272	882	841	0.7	4.2	46.2	53.3	30.8	32.3
ASN	145	293	223	148	246	605	433	837	1153	49.5	65.0	60.2	24.5	51.7	37.6
PRO	212	354	245	79	226	79	416	873	449	59.9	86.5	35.0	100.0	47.7	92.7
GLN	49	170	109	72	135	323	208	531	655	28.8	45.0	53.3	22.3	39.2	31.8
ARG	202	278	601	97	145	201	536	756	1475	72.7	33.6	66.9	48.3	70.9	36.3
SER	138	291	382	423	603	1414	753	1260	2368	47.4	36.1	70.1	29.9	59.8	31.8
THR	112	220	338	182	358	601	546	1035	1598	50.9	33.1	50.8	30.3	52.8	34.2
VAL	23	455	30	181	589	220	414	2079	516	5.1	76.7	30.7	82.3	19.9	80.2
TRP	44	160	171	74	112	263	266	576	1157	27.5	25.7	66.1	28.1	46.2	23.0
TYR	17	132	26	159	337	235	370	999	483	12.9	65.4	47.2	67.7	37.0	76.6

Table AB.2. Raw counts of amino acids in Talaris designs for the monomeric set. Tabular data to support main text Figure 2.3. Total counts of amino acids represented at the surface, core, and total protein are reported for all native and selected Talaris designs analyzed from the monomeric set.

	Counts									Percentages					
	Surface			Core			Total			Surface		Core		Total	
	Correct	Native	Design	Correct	Native	Design	Correct	Native	Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design
ALA	93	533	213	898	1104	1860	1297	2394	2788	17.4	43.7	81.3	48.3	54.2	46.5
CYS	0	27	11	1	86	5	1	162	29	0.0	0.0	1.2	0.0	0.6	0.0
ASP	102	337	518	28	58	116	209	540	962	30.3	19.7	48.3	24.1	38.7	21.7
GLU	149	313	756	55	168	244	347	783	2027	47.6	19.7	32.7	22.5	44.3	17.1
PHE	90	383	96	337	524	526	789	1836	1132	23.5	93.8	64.3	64.1	43.0	69.7
GLY	290	405	480	817	1100	858	1511	2223	1792	71.6	60.4	74.3	95.2	68.0	84.3
HIS	23	76	78	60	123	144	153	351	411	30.3	29.5	48.8	41.7	43.6	37.2
ILE	197	717	315	503	648	676	1082	2223	1647	27.5	62.5	77.6	74.4	48.7	65.7
LYS	199	542	1144	17	68	71	374	981	1986	36.7	17.4	25.0	23.9	38.1	18.8
LEU	184	524	405	534	762	814	1514	2826	2388	35.1	45.4	70.1	65.6	53.6	63.4
MET	12	148	66	198	427	288	293	882	605	8.1	18.2	46.4	68.8	33.2	48.4
ASN	113	304	366	108	257	182	323	837	761	37.2	30.9	42.0	59.3	38.6	42.4
PRO	329	366	435	221	229	224	819	873	966	89.9	75.6	96.5	98.7	93.8	84.8
GLN	48	179	212	59	144	137	222	531	728	26.8	22.6	41.0	43.1	41.8	30.5
ARG	85	293	333	85	149	272	345	756	1410	29.0	25.5	57.0	31.3	45.6	24.5
SER	111	308	268	165	604	283	430	1260	864	36.0	41.4	27.3	58.3	34.1	49.8
THR	79	203	311	165	389	275	418	1035	910	38.9	25.4	42.4	60.0	40.4	45.9
VAL	65	467	126	359	598	454	786	2079	1028	13.9	51.6	60.0	79.1	37.8	76.5
TRP	39	193	114	90	114	172	252	576	658	20.2	34.2	78.9	52.3	43.8	38.3
TYR	37	134	205	149	321	272	450	999	1055	27.6	18.0	46.4	54.8	45.0	42.7

Table AB.3. Raw counts of amino acids in RosettaMembrane designs for the homo-oligomeric set. Tabular data to support main text Figure 2.3. Total counts of amino acids represented at the surface, core, and total protein are reported for all native and selected RosettaMembrane designs analyzed from the homo-oligomeric set.

	Counts									Percentages					
	Surface			Core			Total			Surface		Core		Total	
	Correct	Native	Design	Correct	Native	Design	Correct	Native	Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design
ALA	59	1004	67	2121	4283	3608	2681	7182	4519	5.9	88.1	49.5	58.8	37.3	59.3
CYS	0	57	0	0	408	0	0	570	0	0.0	0.0	0.0	0.0	0.0	0.0
ASP	96	312	254	78	243	108	254	1014	531	30.8	37.8	32.1	72.2	25.0	47.8
GLU	150	618	226	148	472	171	527	1614	698	24.3	66.4	31.4	86.5	32.7	75.5
PHE	105	654	190	831	2039	1077	1306	4134	2029	16.1	55.3	40.8	77.2	31.6	64.4
GLY	697	1041	859	2669	3827	2765	3982	6102	4322	67.0	81.1	69.7	96.5	65.3	92.1
HIS	59	360	88	37	216	193	204	1044	530	16.4	67.0	17.1	19.2	19.5	38.5
ILE	261	570	886	1335	2101	2171	2446	4116	5334	45.8	29.5	63.5	61.5	59.4	45.9
LYS	316	717	871	0	228	3	412	1290	1048	44.1	36.3	0.0	0.0	31.9	39.3
LEU	738	1083	4102	2597	3698	3809	6058	8220	14127	68.1	18.0	70.2	68.2	73.7	42.9
MET	20	168	146	496	996	1346	623	1632	2311	11.9	13.7	49.8	36.8	38.2	27.0
ASN	132	416	253	554	668	1906	921	1650	2804	31.7	52.2	82.9	29.1	55.8	32.8
PRO	425	832	473	485	876	489	1181	2436	1233	51.1	89.9	55.4	99.2	48.5	95.8
GLN	51	289	150	203	402	642	492	1356	1306	17.6	34.0	50.5	31.6	36.3	37.7
ARG	395	668	1089	276	384	492	1296	1914	2900	59.1	36.3	71.9	56.1	67.7	44.7
SER	165	531	539	800	1464	4125	1291	2640	5636	31.1	30.6	54.6	19.4	48.9	22.9
THR	117	533	338	1080	1809	2643	1612	3126	4011	22.0	34.6	59.7	40.9	51.6	40.2
VAL	30	623	53	945	2292	1161	1271	4656	1627	4.8	56.6	41.2	81.4	27.3	78.1
TRP	77	298	513	227	476	545	615	1398	2232	25.8	15.0	47.7	41.7	44.0	27.6
TYR	14	354	31	448	975	603	593	1920	816	4.0	45.2	45.9	74.3	30.9	72.7

Table AB.4. Raw counts of amino acids in Talaris designs for the homo-oligomeric set. Tabular data to support main text Figure 2.3. Tabular Total counts of amino acids represented at the surface, core, and total protein are reported for all native and selected Talaris designs analyzed from the homo-oligomeric set.

	Counts									Percentages					
	Surface			Core			Total			Surface		Core		Total	
	Correct	Native	Design	Correct	Native	Design	Correct	Native	Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design
ALA	53	1014	162	3554	4285	6358	4428	7182	7849	5.2	32.7	82.9	55.9	61.7	56.4
CYS	0	60	25	19	408	56	19	570	116	0.0	0.0	4.7	33.9	3.3	16.4
ASP	175	306	1118	137	243	414	451	1014	2200	57.2	15.7	56.4	33.1	44.5	20.5
GLU	215	618	1488	273	448	824	839	1614	4472	34.8	14.4	60.9	33.1	52.0	18.8
PHE	170	648	188	1248	2050	1677	2034	4134	2778	26.2	90.4	60.9	74.4	49.2	73.2
GLY	772	1036	1014	3107	3825	3285	4623	6102	5144	74.5	76.1	81.2	94.6	75.8	89.9
HIS	70	360	103	109	210	312	346	1044	861	19.4	68.0	51.9	34.9	33.1	40.2
ILE	151	567	332	1536	2076	2187	2337	4116	3609	26.6	45.5	74.0	70.2	56.8	64.8
LYS	191	717	1520	74	233	304	453	1290	2937	26.6	12.6	31.8	24.3	35.1	15.4
LEU	354	1105	669	2593	3704	3279	4774	8220	6446	32.0	52.9	70.0	79.1	58.1	74.1
MET	7	174	26	553	990	849	626	1632	1196	4.0	26.9	55.9	65.1	38.4	52.3
ASN	126	416	685	404	654	803	804	1650	2103	30.3	18.4	61.8	50.3	48.7	38.2
PRO	750	828	987	790	870	833	2198	2436	2536	90.6	76.0	90.8	94.8	90.2	86.7
GLN	88	305	371	210	402	522	653	1356	1721	28.9	23.7	52.2	40.2	48.2	37.9
ARG	166	700	537	168	378	723	738	1914	3057	23.7	30.9	44.4	23.2	38.6	24.1
SER	171	523	595	378	1459	896	887	2640	2115	32.7	28.7	25.9	42.2	33.6	41.9
THR	127	508	561	817	1809	1205	1413	3126	2538	25.0	22.6	45.2	67.8	45.2	55.7
VAL	57	602	176	1534	2289	1847	2192	4656	2856	9.5	32.4	67.0	83.1	47.1	76.8
TRP	91	298	198	302	460	490	675	1398	1334	30.5	46.0	65.7	61.6	48.3	50.6
TYR	114	348	378	550	970	899	948	1920	2146	32.8	30.2	56.7	61.2	49.4	44.2

Table AB.5. Raw counts of amino acids in RosettaMembrane designs for the oligomeric set modeled as monomers. Tabular data to support main text Figure 2.3. Total counts of amino acids represented at the surface, core, and total protein are reported for all native and selected RosettaMembrane monomeric designs analyzed from the homo-oligomeric set.

	Counts									Percentages					
	Surface			Core			Total			Surface		Core		Total	
	Correct	Native	Design	Correct	Native	Design	Correct	Native	Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design	Correct/ Native	Correct/ Design
ALA	30	506	34	497	1087	843	724	2334	1153	5.9	88.2	45.7	59.0	31.0	62.8
CYS	0	18	0	0	114	0	0	168	0	0.0	0.0	0.0	0.0	0.0	0.0
ASP	37	168	116	12	55	24	64	324	178	22.0	31.9	21.8	50.0	19.8	36.0
GLU	43	246	75	35	120	43	146	534	206	17.5	57.3	29.2	81.4	27.3	70.9
PHE	47	294	95	187	470	241	363	1326	604	16.0	49.5	39.8	77.6	27.4	60.1
GLY	276	463	319	829	1135	862	1261	1986	1357	59.6	86.5	73.0	96.2	63.5	92.9
HIS	6	180	8	12	65	49	44	336	123	3.3	75.0	18.5	24.5	13.1	35.8
ILE	132	290	420	218	399	378	702	1326	1763	45.5	31.4	54.6	57.7	52.9	39.8
LYS	123	278	430	1	30	1	134	408	456	44.2	28.6	3.3	100.0	32.8	29.4
LEU	454	627	2158	506	805	737	2015	2802	5441	72.4	21.0	62.9	68.7	71.9	37.0
MET	0	85	88	98	216	318	144	510	685	0.0	0.0	45.4	30.8	28.2	21.0
ASN	32	175	93	131	162	482	217	510	752	18.3	34.4	80.9	27.2	42.5	28.9
PRO	178	383	202	116	227	119	361	804	388	46.5	88.1	51.1	97.5	44.9	93.0
GLN	22	169	53	33	66	180	134	438	378	13.0	41.5	50.0	18.3	30.6	35.4
ARG	170	306	552	75	107	131	420	648	1120	55.6	30.8	70.1	57.3	64.8	37.5
SER	66	247	214	224	360	1079	398	834	1638	26.7	30.8	62.2	20.8	47.7	24.3
THR	66	250	199	266	440	627	474	1002	1143	26.4	33.2	60.5	42.4	47.3	41.5
VAL	22	300	35	197	509	255	317	1452	413	7.3	62.9	38.7	77.3	21.8	76.8
TRP	37	140	199	45	102	151	174	456	772	26.4	18.6	44.1	29.8	38.2	22.5
TYR	15	185	20	95	206	155	156	612	240	8.1	75.0	46.1	61.3	25.5	65.0

Table AB.6. Tabular representation of percent difference in sequence composition for designs. Each dataset was designed using RosetttaMembrane and Talaris scoring functions. Support for figure 2.8

	Monomers		Homo-oligomers		Homo-oligomers as Monomers
	Membrane	Talaris	Membrane	Talaris	Membrane
ALA	-4.65	1.64	-4.69	0.79	-6.63
ARG	15.50	22.13	1.97	2.02	12.18
ASN	1.34	-0.29	1.87	0.91	1.21
ASP	-0.22	1.67	-0.65	2.07	-0.68
CYS	-0.74	-0.59	-0.90	-0.70	-0.90
GLN	0.41	0.77	-0.02	0.67	-0.42
GLU	-1.99	5.35	-1.51	5.46	-1.69
GLY	-3.03	-1.71	-3.06	-1.65	-3.27
HIS	0.09	0.30	-0.84	-0.41	-1.28
ILE	-0.64	-2.49	2.38	-0.80	2.86
LEU	13.81	-2.04	9.76	-3.94	15.07
LYS	-1.33	4.17	-0.18	3.33	0.45
MET	-0.29	-1.34	1.44	-0.47	1.11
PHE	-4.08	-2.98	-3.41	-2.20	-3.52
PRO	-2.55	-1.53	-2.03	0.16	-3.18
SER	3.86	-1.57	4.62	-0.78	3.58
THR	2.24	-0.36	1.39	-1.20	0.58
TRP	2.44	0.24	1.23	-0.28	1.18
TYR	-2.11	0.42	-2.08	0.31	-2.16
VAL	-6.32	-4.26	-5.28	-3.32	-5.83
Average absolute deviation	±3.4	±2.8	±2.5	±1.6	±3.4

Table AB.7. Tabular representation of percent difference in amino acid property recovery. Support for figure 2.8

	Monomers		Homo-oligomers		Homo-oligomers as Monomers
Property	Membrane	Talaris	Membrane	Talaris	Membrane
ALIPHATIC	-2.83	-9.85	-1.52	-9.25	1.16
AROMATIC	-3.66	-2.04	-5.13	-2.60	-5.81
BETA-BRANCHED	9.73	-6.66	5.86	-8.47	9.83
CHARGED	-0.58	13.94	-0.35	12.92	0.57
LONG-FLEXIBLE	-0.30	11.66	1.70	11.03	1.92
NEGATIVE	-2.22	7.03	-2.16	7.54	-2.36
POLAR&CHARGED	6.59	12.17	5.78	11.42	3.35
POLAR	7.18	-1.77	6.15	-1.48	2.80
POSITIVE	1.62	6.91	1.79	5.37	2.93
SMALL	-3.84	-1.67	-3.14	-1.63	-6.34
Average absolute deviation	±3.9	±7.4	±3.4	±7.3	±3.7

APPENDIX C

PROTOCOL CAPTURE FOR CHAPTER 2

The protocol capture provides the generalized steps that were necessary to carry out the benchmark experiments described Duran and Meiler, 2017, and was included in the supplemental materials. The Rosetta software suite is free for academic users.

Part 1a: The effects of various modes of relax on the sequence recovery for membrane protein design (monomers)

Step 1 - Set up files for monomeric proteins

A. Spanfiles- an input file that describe the start and ends of transmembrane spans. The structure based approach used in this protocol was to obtain span information from the PDBTM XML file specific to each PDBID. XMLs were parsed for the start and end of transmembrane regions and spanfiles were created using the standard Rosetta spanfile format.

B. Option file-an input file with common options:

```
-in:file:fullatom
-in:file:s myprot.pdb
-in:file:spanfile myfile.span
-membrane:no_interpolate_Mpair
-membrane:Membed_init
-membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts
-ex1
-ex2
-ex2aro
-use_input_sc
-out:file:fullatom
-out:path:pdb
-database /path/to/Rosetta/main/database/
```

Step 2- Various relaxation strategies to prepare for membrane protein design

Steps A-E are the different approaches to prepare structural models for design

A. Repack only: Only side-chains are minimized (no backbone minimization). Repack uses a resfile to restrict to repacking only.

Input resfile:
```
NATAA
start
```

To repack using a RosettaScripts XML set-up:

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
@myoptions -parser:protocol myrpkprotocol.xml
```

Input XML “myrpkprotocol.xml” :

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
         <mem_hres weights="membrane_highres_Menv_smooth.wts"/>
        </SCOREFXNS>
        <TASKOPERATIONS>
         <InitializeFromCommandline name=”ifcl”/>
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
          <PackRotamersMover name=”pack” scorefxn=”mem_hres”
task_operations=”ifcl” />
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
          <Add mover_name=”pack”/>
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To repack using the design application:

```
/path/to/rosetta/fixbb.linuxgcc.release @myoptions
-resfile my.resfile
```

B. Constrained to start coordinates: constrained relax using option available in combination with FastRelax

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
@myoptions -parser:protocol myrlxprotocol.xml
-relax:constrain_relax_to_start_coords true
```

Input XML “myrlxprotocol.xml”:

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
          <mem_hres weights="membrane_highres_Menv_smooth.wts"/>
        </SCOREFXNS>
        <TASKOPERATIONS>
          <InitializeFromCommandline name="ifcl"/>
          <IncludeCurrent name="ic" />
          <RestrictToRepacking name="rtr" />
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
          <FastRelax name="relax" scorefxn="mem_hres"
task_operations="ifcl,ic,rtr" />
```

```
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
          <Add mover_name="relax"/>
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To relax using the relax application:

```
/path/to/rosetta/main/source/bin/relax.default.linuxgccrelease
@myoptions
-relax:constrain_relax_to_start_coords true
```

C. FastRelax: This is the standard minimization protocol in Rosetta.

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
@myoptions
-parser:protocol myprotocol.xml
```

Input XML: Same as Part1a: Step 2B

Generated 250-300 models of each protein

To relax using the relax application:

```
/path/to/rosetta/main/source/bin/relax.default.linuxgccrelease
@myoptions
```

D. Dualspace: This is a protocol that combines internal coordinate relax and Cartesian coordinate relax focusing on ideal bond lengths

```
/path/to/Rosetta/main/source/bin/rosetta_script.linuxgccrelease
@myoptions -parser:protocol myprotocol.xml -relax:dualspace
-relax:minimize_bond_angles -set_weights cart_bonded 0.5 pro_close
0 -default_max_cycles 200
```

Input XML: Same as Part1a: Step 2B

To perform dualspace relax using the relax application:

```
/path/to/rosetta/main/source/bin/relax.default.linuxgccrelease
@myoptions -relax:dualspace -relax:minimize_bond_angles -set_weights
cart_bonded 0.5 pro_close 0 -default_max_cycles 200
```

E. Minimize with constraints: This protocol is the first step of the standard ddg_monomer protocol

```
/path/to/Rosetta/main/source/bin/minimize_with_cst.linuxgccrelease
-in:file:l mypdbs.list -in:file:fullatom -ignore_unrecognized_res
-fa_max_dis 9.0 -database /path/to/Rosetta/main/database/
-ddg::harmonic_ca_tether 0.5
-score:weights membrane_highres_Menv_smooth.wts
-membrane:no_interpolate_Mpair -membrane:Membed_init
-membrane:Menv_penalties -ddg::constraint_weight 1.0
-ddg::out_pdb_prefix min_cst_0.5 -ddg::sc_min_only false
```

Skip step 3- only one model to start from

Step 3 - Selecting models for design

To select the best relaxed models for design, evaluate the total score and full atom RMSD to starting structure
Calculate the RMSD with respect to the starting structure by using a script available in Rosetta:

```
/path/to/rosetta/tools/protein_tools/scripts/score_vs_rmsd.py
--native=mynative.pdb --table=mytable.txt my*.pdbs
```

where the wildcard represents all models in a location

For each protein, the three models with the lowest total score and lowest RMSD were selected as templates for design. Approximately 300 models were generated for each protein. Three models were chosen to generate designs on multiple possible conformation of each protein.

Step 4 - Full design of monomeric proteins

The top relaxed models from step 3 are used as input to replace pdbs in the options file:

```
-in:file:pdb myrlxprot.pdb
```

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease @myoptions
-parser:protocol myrlxprotocol.xml -linmem_ig 10
```

Input XML "mydesprotocol":

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
          <mem_hres weights="membrane_highres_Menv_smooth.wts"/>
        </SCOREFXNS>
        <TASKOPERATIONS>
          <InitializeFromCommandline name=”ifcl”/>
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
          <PackRotamersMover name=”pack” scorefxn=”mem_hres”
task_operations=”ifcl” />
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
          <Add mover_name=”pack”/>
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To design using the design application:

```
/path/to/rosetta/main/source/bin/fixbb.default.linuxgccrelease @myoptions
```

Step 5 - Selecting design models for analysis

Select top 10% by score as the backbone is fixed during design

```
grep pose *.pdb | sort -nk 23 | head -n
```

Where n is 1/10th of the total number of designs generated for a protein (approximately 75 designs generated for each protein)

Step 6 – Calculating sequence recovery

Calculating the ability of the top 10% of models by score to recover the native sequence

I made a file containing a list of the top 10% of models by score for each protein.

```
/path/to/rosetta/main/source/bin/sequencerecovery.linuxgccrelease
-native_pdb_list mynatives.list -redesign_pdb_list designs.list
-ignore_unrecognized_res
```

Calculated the average and standard deviation for the top 10% of models

Protocol for part 1b: The effects of various modes of relax on the sequence recovery for membrane protein design (homo-oligomers)

Step 1 – Setting up files for oligomeric membrane proteins

A. Create spanfile – see Part 1a; note that this will contain starts and ends of transmembrane helices for a single protomer

B. Option file-an input file with common options: note that the input pdb file is of a single protomer

```
-in:file:fullatom
-in:file:s myprot.pdb
-in:file:spanfile myfile.span
-membrane:no_interpolate_Mpair
-membrane:Membed_init
-membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts
-ex1
-ex2
-ex2aro
-use_input_sc
-out:file:fullatom
-out:path:pdb
-database /path/to/Rosetta/main/database/
```

C. Symmetry definition file
Structures of the oligomeric complex were downloaded from PDBTM. These full structures were used to generate the symmetry definition file.

```
/path/to/rosetta/main/source/apps/public/symmetry/
make_symmdef_file.pl -m NCS -a A -i B -p myprotein.pdb > myfile.sym
```

Note: parameters for generating symfiles vary for oligomeric assembly and chainids

Step 2- Various relaxation strategies to prepare for symmetric membrane protein design

A. Repack only: Only side-chains are minimized (no backbone minimization). Repack uses a resfile to restrict to repacking only.

Input resfile:

```
NATAA
start
```

To repack using a RosettaScripts XML set-up:

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
@myoptions -parser:protocol myrpkprotocol.xml
```

Input XML “mysymrpkprotocol.xml” :

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
         <mem_hres weights="membrane_highres_Menv_smooth.wts"
         symmetric="1" />
        </SCOREFXNS>
        <TASKOPERATIONS>
          <InitializeFromCommandline name="ifcl" />
         <ReadResfile name=”rrf” />
        </TASKOPERATIONS>
        <MOVERS>
         <SetupForSymmetry name="setup_symm" definition=myfile.sym
      />
         <SymPackRotamersMover name="sym_pack" scorefxn="mem_hres"
      task_operations="ifcl,rrf” />
        </MOVERS>
        <PROTOCOLS>
          <Add mover="setup_symm" />
          <Add mover="sym_pack" />
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To repack using the design application:

```
/path/to/rosetta/fixbb.linuxgcc.release @myoptions
–resfile my.resfile –symmetry –symmetry_definition myfile.sym
```

B. Constrained to start coordinates: constrained relax using option available in combination with FastRelax

```
/path/to/rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
-parser:protocol mysymrlxprotocol.xml
-relax:constrain_relax_to_start_coords true
```

```
-relax:jump_move true
```

Input XML ‘mysymrlxprotocol.xml’:

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
          <mem_hres_sym
weights="membrane_highres_Menv_smooth.wts" symmetric="1" />
        </SCOREFXNS>
        <TASKOPERATIONS>
          <InitializeFromCommandline name="ifcl"/>
          <RestrictToRepacking name="rtr" />
          <IncludeCurrent name="ic" />
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
          <SetupForSymmetry name="setup_symm"
definition="myfile.sym" />
          <FastRelax name="relax" scorefxn="mem_hres_sym"
repeats="1" task_operations="ifcl,rtr,ic" />
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
          <Add mover_name="setup_symm" />
          <Add mover_name="relax" />
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To relax using the relax application:

```
/path/to/rosetta/main/source/bin/relax.default.linuxgccrelease
@myoptions -relax:constrain_relax_to_start_coords true
–symmetry –symmetry_definition myfile.sym -relax:jump_move
```

C. FastRelax: This is the standard minimization protocol in Rosetta.

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
@myoptions -parser:protocol myprotocol.xml -relax:jump_move true
```

Note that instead of the standard eight (at the time) rounds of relax, one round resulted in decently scored models and required much less time resources

To relax using the relax application:

```
/path/to/rosetta/main/source/bin/relax.default.linuxgccrelease
@myoptions –symmetry –symmetry_definition myfile.sym
-relax:jump_move
```

D. Dualspace: This is a protocol that combines internal coordinate relax and Cartesian coordinate relax focusing on ideal bond lengths

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
```

```
-parser:protocol myprotocol.xml -relax:dualspace
-relax:minimize_bond_angles -set_weights cart_bonded 0.5 pro_close
0 -default_max_cycles 200
```

Input XML: Same as Part1b: Step 2b

To perform dualspace relax using the relax application:

```
/path/to/rosetta/main/source/bin/relax.default.linuxgccrelease
@myoptions -relax:dualspace -relax:minimize_bond_angles -
set_weights cart_bonded 0.5 pro_close 0 -default_max_cycles 200 –
symmetry –symmetry_definition myfile.sym
```

E. Minimize with constraints: This protocol is the first step of the standard ddg_monomer protocol

```
/path/to/rosetta/main/source/bin/
minimize_with_cst.linuxgccrelease -in:file:l mypdbs.list
-in:file:fullatom -ignore_unrecognized_res -fa_max_dis 9.0
-database /path/to/Rosetta/main/database/
-ddg::harmonic_ca_tether 0.5
-score:weights membrane_highres_Menv_smooth.wts
-membrane:no_interpolate_Mpair -membrane:Membed_init
-membrane:Menv_penalties -ddg::constraint_weight 1.0
-ddg::out_pdb_prefix min_cst_0.5 -ddg::sc_min_only false
```

Note that this minimizes only in the context of the single protomer
Skip step 3- only one model to start from

Step 3 - Selecting models for design

To select the best relaxed models for design, evaluate the total score and full atom RMSD to starting structure
Calculate the RMSD with respect to the starting structure by using a script available in Rosetta:

```
/path/to/rosetta/tools/protein_tools/scripts/score_vs_rmsd.py
--native=mynative.pdb --table=mytable.txt my*.pdbs
```

Where the wildcard represents all models in a location
Note that the input native should be the symmetric assembly of the starting conformation

For each protein, the three models with the lowest total score and lowest RMSD were selected as templates for design. Three models were chosen to generate designs on multiple possible conformation of each protein.

Step 4 - Full design of oligomeric proteins

Chain A of the top relaxed models from step 3 are used as input to replace pdbs in the options file:

```
-in:file:pdb myrlxprot.pdb
```

However, one must align the relaxed models to the coordinates of the input structure in order to use the same symmetry definition file in the design protocol

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease @myoptions
-parser:protocol myrlxprotocol.xml -linmem_ig 10
```

Input XML "mysymdesprotocol":

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
         <mem_hres weights="membrane_highres_Menv_smooth.wts" symmetric="1" />
        </SCOREFXNS>
        <TASKOPERATIONS>
           <InitializeFromCommandline name="ifcl" />
        </TASKOPERATIONS>
        <MOVERS>
          <SetupForSymmetry name="setup_symm" definition=myfile.sym />
          <SymPackRotamersMover name="sym_pack" scorefxn="mem_hres”
task_operations="ifcl" />
        </MOVERS>
        <PROTOCOLS>
          <Add mover="setup_symm" />
          <Add mover="sym_pack" />
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To design using the design application:

```
/path/to/rosetta/main/source/bin/fixbb.default.linuxgccrelease @myoptions
-linmem_ig 10 –symmetry –symmetry_definition myfile.sym
```

Step 5 - Selecting design models for analysis
See Part1a: Step 5

Step 6 - Sequence Recovery
See Part1a: Step 6

Protocol for part 2a: The effects of various scoring functions on the sequence recovery and composition for membrane protein design (monomers)

Step 1 – Preparation of files necessary for design

A. Minimization of starting structure- selected the top three models of each protein by score and RMSD of relaxed structures from the constrained to start coordinates relax protocol from Part 1a: Step 2b

B. Option file-an input file with common options:

```
        -in:file:fullatom
        -in:file:s myprot.pdb
        -ex1
        -ex2
        -ex2aro
        -use_input_sc
        -linmem_ig 10
        -out:file:fullatom
        -out:path:pdb
        -database /path/to/Rosetta/main/database/
```

Step 2 – Full Design of Monomers

A. Design of monomers using RosettaMembrane

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
@myoptions -parser:protocol memdes.xml
-in:file:spanfile myfile.span -membrane:no_interpolate_Mpair
-membrane:Membed_init -membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts
```

Input XML 'memdes.xml':

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
          <mem_highres weights="membrane_highres_Menv_smooth.wts"/>
        </SCOREFXNS>
        <TASKOPERATIONS>
          <InitializeFromCommandline name=ifcl/>
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
           <PackRotamersMover name=pack scorefxn=mem_highres
task_operations=ifcl/>
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
          <Add mover_name=pack/>
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To perform full design using the design application:

```
/path/to/rosetta/main/source/bin/fixbb.default.linuxgccrelease
@myoptions -in:file:spanfile myfile.span
-membrane:no_interpolate_Mpair -membrane:Membed_init
-membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts
```

B. Design of monomers using soluble scoring function Talaris

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
@myoptions -parser:protocol taldes.xml
```

Input XML taldes.xml:

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
            <talaris weights="talaris2013.wts"/>
        </SCOREFXNS>
        <TASKOPERATIONS>
            <InitializeFromCommandline name=ifcl/>
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
            <PackRotamersMover name=pack scorefxn=talaris
            task_operations=ifcl />
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
            <Add mover_name=pack/>
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To perform full design using the design application:

```
 /path/to/rosetta/main/source/bin/fixbb.default.linuxgccrelease
@myoptions -score:weights talaris2013.wts
```

Step 3 - Selecting design models for analysis

See Part1a: Step 5

Step 4 – Calculating Sequence Recovery

The sequence recovery application can give information for total sequence recovery (focus in Part 1); however the Rosetta Sequence Recovery Application gives sequence recovery for subgroups core >= 24 neighbors within c-beta distance of 10 Angstroms; surface <= 16 neighbors within c-beta distance of 10 Angstroms; and overall sequence recovery

```
/path/to/Rosetta/main/source/bin/sequencerecovery.linuxgccrelease
-native_pdb_list mynatives.list -redesign_pdb_list designs.list
-ignore_unrecognized_res
```

Inputs: designs.list = top10% of the respective protein
Two outputs sequencerecovery.txt and a submatrix.txt are produced. Only sequencerecovery.txt was used for analysis where numbers for core, overall, and surface recovery were extracted and converted into percentages

Step 5 – Calculating Sequence Composition

An in-house script was used that calculates the percentage of each amino acid identity in the protein as well as properties such as polar&charged, polar, charged, aromatic, aliphatic, positive, negative, small, beta branched, long flexible.

For a comparison, the amino acid composition of the native structures was calculated. Then the top 10% of designs by score were analyzed for amino acid composition.
The composition of both amino acid identities and properties were analyzed.

Step 6 - Hotpatch Analysis-Hotpatch server

Submitted representative models for each structure and submitted for lipid-interacting hotpatch analysis

Step 7 - OPM Analysis-Orientation of Proteins in the Membrane server

http://opm.phar.umich.edu/server.php

The output gives information about the thickness of the protein well as the tilt angle, transfer energy of the protein from water to lipid bilayer, and membrane embedded residues along with a pdb of the protein in the bilayer; however, only transfer energy was a metric for analysis

Protocol for part 2b: The effects of various scoring functions on the sequence recovery and composition for membrane protein design (homo-oligomers)

Step 1 – Preparation of files necessary for symmetric design

A. Minimization of starting structure- selected the top three models of each protein by score and RMSD of relaxed structures from the constrained to start coordinates relax protocol from Part 1b: Step 2b

B. Option file-an input file with common options:

```
-in:file:fullatom
-in:file:s myprot.pdb
-ex1
-ex2
-ex2aro
-use_input_sc
-linmem_ig 10
-out:file:fullatom
-out:path:pdb
-database /path/to/Rosetta/main/database/
```

Step 2-Full design of homo-oligomers

A. Design of homo-oligomers : RosettaMembrane

Use RosettaMembrane to fully design the protein as an oligomer

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
-parser:protocol symmemdes.xml -membrane:no_interpolate_Mpair
-membrane:Membed_init -membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts
```

Input XML symmemdes.xml:

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
          <mem_hres weights="membrane_highres_Menv_smooth.wts"
symmetric=1 />
        </SCOREFXNS>
        <TASKOPERATIONS>
          <InitializeFromCommandline name=ifcl/>
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
          <SetupForSymmetry name=setup_symm definition= myfile.sym />
          <SymPackRotamersMover name=sym_pack scorefxn=mem_hres
task_operations=ifcl />
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
          <Add mover_name=setup_symm />
          <Add mover_name=sym_pack/>
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To perform full design using the design application:

```
 /path/to/rosetta/main/source/bin/fixbb.default.linuxgccrelease
@myoptions -in:file:spanfile myfile.span
-membrane:no_interpolate_Mpair -membrane:Membed_init
-membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts -symmetry
-symmetry_definition myfile.sym
```

B. Design of homo-oligomers using soluble scoring function Talaris

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease
-parser:protocol symtaldes.xml –symmetry
–symmetry_definition myfile.sym
```

Input XML ‘symtaldes.xml’:

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
          <talaris weights="talaris2013.wts" symmetric=1 />
        </SCOREFXNS>
        <TASKOPERATIONS>
          <InitializeFromCommandline name=ifcl/>
        </TASKOPERATIONS>
        <FILTERS>
        </FILTERS>
        <MOVERS>
          <SetupForSymmetry name=setup_symm definition= myfile.sym />
          <SymPackRotamersMover name=sym_pack scorefxn=talaris
task_operations=ifcl />
        </MOVERS>
        <APPLY_TO_POSE>
        </APPLY_TO_POSE>
        <PROTOCOLS>
          <Add mover_name=setup_symm />
          <Add mover_name=sym_pack/>
        </PROTOCOLS>
</ROSETTASCRIPTS>
```

To perform full design using the design application:

```
 /path/to/rosetta/main/source/bin/fixbb.default.linuxgccrelease @myoptions
-score:weights talaris2013.wts -symmetry  -symmetry_definition myfile.sym
```

Step 3 - Selecting design models for analysis
See Part1a: Step 5

Step 4 – Calculating Sequence Recovery
See Part2a:Step4

Step 5 - Calculating Sequence Composition
See Part2a:Step5

Step 6 - Hotpatch Analysis – Hotpatch Server
See Part2a:Step6

Step 7 - OPM Analysis-Orientation of Proteins in the Membrane Server
See Part2a:Step7

Step 3: Design of homo-oligomers as monomers

From the single chain input for Part 1b: Step 2b, run through the monomeric version of the protocol at Part1a:Step 2b, then continue through the rest of the protocol using the monomeric versions in Part1a and Part 2a.

APPENDIX D

PROTOCOL CAPTURES FOR CHAPTER 3

The implementation for soluble proteins (high-resolution)

Step 1. Generate Constraints File

To run this application, you have to input a list (not a path to a single file). So if you have one pdb, create a 'list' with a single row.

```
 ~myrosetta/main/source/bin/minimize_with_cst.linuxgccrelease -in:file:l pdb.list -
 in:file:fullatom -ignore_unrecognized_res -fa_max_dis 9.0 -ddg::harmonic_ca_tether 0.5 -
 score:weights score12.wts -ddg::constraint_weight 1.0 -ddg::out_pdb_prefix min_cst_0.5 -
 ddg::sc_min_only false > mincst.log
```

and then use:

```
 ~myrosetta/main/source/src/apps/public/ddg/convert_to_cst_file.sh
```

Step 2. Run the application

To run the application, you must include a number of options (shown below)

```
 \~myrosetta/main/source/bin/ddg_monomer.linuxgccrelease @ddG_monomer.options \-s
 mystruct.pdb \-resfile mymut.resfile
```

```
-ddg:weight_file soft_rep_design # Use soft-repulsive weights for the
initial sidechain optimization stage
-ddg:minimization_scorefunction score12 # optional -- the weights file to
use, if not given, then "score12" will be used (score12 = standard.wts +
score12.wts_patch)
-restore_pre_talaris_2013_behavior # essential for versions of Rosetta 2013
and beyond
#-ddg::minimization_patch <weights patch file > # optional -- the weight-
patch file to apply to the weight file; does not have to be given
-database /path/to/rosetta/main/database #the full oath to the database is
required
-fa_max_dis 9.0 # optional -- if not given, the default value of 9.0
Angstroms is used.
-ddg::iterations 50 # 50 is the recommended number of iterations
-ddg::dump_pdbs true # write out PDB files for the structures, one for the
wildtype and one for the pointmutant for each iteration
-ignore_unrecognized_res # optional -- if there are residues in the input
PDB file that Rosetta cannot recognize, ignore them instead of quitting with
an error message
-ddg::local_opt_only false # recommended: local optimization restricts the
sidechain optimization to only the 8 A neighborhood of the mutation
(equivalent to row 13)
-ddg::min_cst true # use distance restraints (aka constraints) during the
backbone minimization phase
-constraints::cst_file 1py6.cst # the set of constraints to use during
minimization which should reflect distances in the original (non-pre-
relaxed) structure
-ddg::suppress_checkpointing true # don't checkpoint
```

```
-in::file::fullatom # read the input PDB file as a fullatom structure
-ddg::mean true # do not report the mean energy
-ddg::min true # report the minimum energy
-ddg::sc_min_only false # do not minimize only the backbone during the
backbone minimization phase
-ddg::ramp_repulsive true # perform three rounds of minimization (and not
just the default 1 round) where the weight on the repulsive term is
increased from 10% to 33% to 100%
-mute all # optional -- silence all of the log-file / stdout output
generated by this protocol
-unmute core.optimization.LineMinimizer # optional -- unsilence a particular
tracer
-ddg::output_silent true # write output to a silent file
```

For the membrane protein high-resolution protocol:

Step 1. Generate Constraints File

To run this application, you have to input a list (not a path to a single file). So if you have one pdb, create a 'list' with a single row.

```
~myrosetta/main/source/bin/minimize_with_cst.linuxgccrelease -in:file:l pdb.list -
in:file:fullatom -ignore_unrecognized_res -fa_max_dis 9.0 -ddg::harmonic_ca_tether 0.5 -
score:weights membrane_highres_Menv_smooth.wts -ddg::constraint_weight 1.0 -
ddg::out_pdb_prefix min_cst_0.5 -ddg::sc_min_only false -membrane:no_interpolate_Mpair -
membrane:Membed_init -membrane:Menv_penalties -in:file:spanfile myspan.span >
mincst.log
```

and then use:

```
~myrosetta/main/source/src/apps/public/ddg/convert_to_cst_file.sh
```

Step 2. Run the application

To run the application, you must include a number of options (shown below)

```
\~myrosetta/main/source/bin/ddg_monomer.linuxgccrelease @ddG_monomer.options \-s
mystruct.pdb \-resfile mymut.resfile
```

Note: since there are no 'soft rep' weights for rosetta membrane, rather than cutting and pasting parts of each scorefunction into a soft rep for membrane, I used the original soft_rep_design as the part for repacking, while using the membrane_highres_Menv_smooth.wts for minimization. It's not a great way, but I feel like it is better than pasting together a soft rep membrane weights file that hasn't been trained

```
-ddg:weight_file soft_rep_design # Use soft-repulsive weights for the
initial sidechain optimization stage
-ddg:minimization_scorefunction membrane_highres_Menv_smooth.wts # optional
-- the weights file to use, if not given, then "score12" will be used
(score12 = standard.wts + score12.wts_patch)
-restore_pre_talaris_2013_behavior
#-ddg::minimization_patch <weights patch file > # optional -- the weight-
patch file to apply to the weight file; does not have to be given
-database /path/to/rosetta/main/database #the full oath to the database is
required
-fa_max_dis 9.0 # optional -- if not given, the default value of 9.0
Angstroms is used.
-ddg::iterations 50 # 50 is the recommended number of iterations
-ddg::dump_pdbs true # write out PDB files for the structures, one for the
wildtype and one for the pointmutant for each iteration
```

```
-ignore_unrecognized_res # optional -- if there are residues in the input
PDB file that Rosetta cannot recognize, ignore them instead of quitting with
an error message
-ddg::local_opt_only false # recommended: local optimization restricts the
sidechain optimization to only the 8 A neighborhood of the mutation
(equivalent to row 13)
-ddg::min_cst true # use distance restraints (aka constraints) during the
backbone minimization phase
-constraints::cst_file 1py6.cst # the set of constraints to use during
minimization which should reflect distances in the original (non-pre-
relaxed) structure
-ddg::suppress_checkpointing true # don't checkpoint
-in::file::fullatom # read the input PDB file as a fullatom structure
-ddg::mean true # do not report the mean energy
-ddg::min true # report the minimum energy
-ddg::sc_min_only false # do not minimize only the backbone during the
backbone minimization phase
-ddg::ramp_repulsive true # perform three rounds of minimization (and not
just the default 1 round) where the weight on the repulsive term is
increased from 10% to 33% to 100%
-mute all # optional -- silence all of the log-file / stdout output
generated by this protocol
-unmute core.optimization.LineMinimizer # optional -- unsilence a particular
tracer
-ddg::output_silent true # write output to a silent file
-membrane:no_interpolate_Mpair
-membrane:Membed_init
-membrane:Menv_penalties
```

APPENDIX E

DATASET FROM CHAPTERS 3 AND 4

Table AE.1 Complete dataset from Chapters 3 and 4 of experimentally derived and predicted mutation stabilities for membrane proteins. Entries contain the wild-type amino acid identity, residue number, mutation amino acid identity, and thermostability predictions from various programs.

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
E	9	A	-0.1	-0.05	-2.828	-0.81	-3.64677	-0.912	0.66	-0.937	-0.609	-1.054	-0.822	-0.88198	-0.3768	INT	BUR	1PY6	BR	AH	0.59	0.578078
L	13	A	-1.8	-4.781	-4.312	-2.44	-2.62844	-3.079	-2.34	-3.409	-3.842	-3.179	-1.536	-1.77341	-1.2538	INT	BUR	1PY6	BR	AH	3.03	0.415865
A	39	P	-0.6	-7.406	-53.695	-0.98	-3.98559	0.099	-4.55	-0.144	-1.03	-1.816	-1.659	-0.25733	-0.3017	INT	BUR	1PY6	BR	AH	-1	1.25584
F	42	A	-2	-5.05	-5.515	-2.02	-3.7489	-3.149	-2.6	-3.386	-4.976	-2.174	-2.701	-1.85417	-2.4333	INT	BUR	1PY6	BR	AH	4.79	0.485153
Y	43	A	-2.1	-3.315	-4.469	-2.44	-3.14197	-3.179	-0.94	-3.321	-3.704	-2.492	-3.981	-0.8211	-2.5306	INT	BUR	1PY6	BR	AH	6.02	1.76937
Y	43	F	-1.7	0.844	0.445	-1.05	0.840668	-1.324	1.79	-1.277	-1.085	-1.018	-1.439	0.260715	-0.5496	INT	BUR	1PY6	BR	AH	6.02	1.76937
Y	43	P	0.1	-11.273	-311.519	-1.78	-7.40469	-2.342	-2.76	-2.572	-0.817	-2.23	-4.637	-1.77421	-1.2835	INT	BUR	1PY6	BR	AH	6.02	1.76937
M	60	A	-1	-3.332	-2.876	-1.4	-2.95278	-2.289	-2.33	-2.406	-2.833	-2.119	-3.416	-1.3509	-2.1913	INT	BUR	1PY6	BR	AH	9	2.77871
Y	79	F	-0.1	-0.845	-1.687	-0.62	-2.20774	-1.251	0.41	-1.209	-1.078	-0.822	-1.281	0.017982	-0.2625	INT	BUR	1PY6	BR	AH	3.24	0.980699
Y	83	A	-1.7	-5.851	-6.68	-1.79	-3.11418	-2.63	-0.8	-2.67	-4.241	-1.695	-6.341	-1.28492	-2.803	INT	BUR	1PY6	BR	AH	9.14	3.6986
Y	83	F	-1	0.525	-0.26	-0.82	-0.00156	-0.778	1.62	-0.185	-1.2	-0.832	-2.818	-0.65566	-0.7002	INT	BUR	1PY6	BR	AH	9.14	3.6986
L	100	A	-3.2	-4.28	-3.875	-2.86	-3.46776	-2.554	-2.34	-2.87	-3.186	-1.692	-4.061	-1.81036	-2.0536	INT	BUR	1PY6	BR	AH	3.23	0.31928
T	170	A	-0.9	-0.633	-0.512	-1.98	0.006492	-1.216	2.34	-1.063	-1.238	-1.322	-2.017	-0.20936	-0.3074	INT	BUR	1PY6	BR	AH	0	-0.04189
W	189	F	1	-3.356	-3.227	-1.09	-2.29241	-1.621	-1.6	-1.8	-1.217	-1.225	-9.51	-1.09945	-1.7769	INT	BUR	1PY6	BR	AH	9.73	3.60209
S	193	A	0.1	0.207	0.155	-1.1	-0.61281	-0.706	1.37	-0.849	-1.586	-1.111	-0.522	-0.06511	-0.7632	INT	BUR	1PY6	BR	AH	0.23	-0.01375
E	204	A	-1.85	2.032	0.97	-1.62	2.13103	-2.346	0.96	-2.2	-2.382	-1.501	-3.966	0.860896	-0.7345	INT	BUR	1PY6	BR	AH	4.51	0.144358

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
S	226	A	-0.9	0.604	-0.488	-0.67	-0.50007	-0.649	1.18	-0.674	-1.518	-0.418	-0.554	0.297906	-0.06	INT	BUR	1PY6	BR	AH	1.09	-0.07189
T	97	A	-2.05	-1.352	-1.465	-1.28	-1.3821	-1.151	1.65	-1.03	-1.624	-1.152	-4.93	NA	-0.9522	INT	BUR	2XOV	GLPG	AH	7.49	1.59288
M	111	A	-0.68	-3.344	-3.19	-1.27	-2.85398	-2.002	-2.3	-2.102	-2.624	-1.911	-4.02	-2.94741	-1.0837	INT	BUR	2XOV	GLPG	AH	5.09	1.00544
Q	112	A	-0.81	-0.527	-1.407	-0.27	-0.58268	-0.66	0.5	-0.474	-0.149	-0.097	-2.804	-0.57452	-0.4095	INT	BUR	2XOV	GLPG	AH	1.2	0.448319
M	120	A	-0.47	-2.206	-2.266	-0.69	-2.3283	-2.603	-1.03	-2.603	-0.42	-1.152	-3.7	-1.52395	-1.6077	INT	BUR	2XOV	GLPG	AH	2	0.205404
R	137	A	-0.88	-5.051	-5.152	-0.42	-4.59401	-1.971	-0.63	-2.156	-0.155	-0.636	-5.986	-0.56947	-2.3884	INT	BUR	2XOV	GLPG	AH	8.02	2.01486
H	141	A	-1.12	-2.448	-2.155	-0.87	-0.8004	-1.836	-1.96	-2.19	-1.219	-1.1	-5.003	-1.4529	-1.3266	INT	BUR	2XOV	GLPG	AH	2.12	1.14554
L	143	A	-0.14	-3.621	-3.123	-2.07	-2.13838	-2.393	-2.34	-2.676	0.683	-2.457	-4	-1.5085	-2.6615	INT	BUR	2XOV	GLPG	AH	0.36	3.67207
M	144	A	-0.42	-4.088	-3.633	-1.21	-3.64952	-3.022	-2.61	-3.228	-2.742	-1.407	-3.416	-2.36853	-2.05	INT	BUR	2XOV	GLPG	AH	3.33	0.639248
H	145	A	-1.71	-6.731	-5.706	-0.91	-1.71689	-2.128	-0.95	-2.35	-1.38	-1.125	-9.743	0.674099	-2.2558	INT	BUR	2XOV	GLPG	AH	11.71	7.79326
H	150	A	-0.8	-4.075	-2.15	-0.38	0.631722	-2.042	-0.79	-2.113	-0.986	-0.912	-9.043	0.360235	-1.516	INT	BUR	2XOV	GLPG	AH	11.64	7.51144
E	166	A	-5.46	-3.903	-6.905	-1.01	-5.90595	-0.491	0.96	-0.491	-0.361	-1.022	-5.998	0.313757	-0.6514	INT	BUR	2XOV	GLPG	AH	8.11	2.03573
L	169	A	-1.08	-1.744	1.001	-2.62	-3.24802	-1.822	-1.45	-1.967	0.747	-2.911	-4.17	-2.19087	-1.0248	INT	BUR	2XOV	GLPG	AH	3.51	1.06039
G	170	A	-0.75	-8.56	-8.156	-1.33	-3.24081	-0.673	-2.21	-0.646	-0.453	-2.034	-5.882	-0.46648	-0.8988	INT	BUR	2XOV	GLPG	AH	7.72	1.68637
S	171	A	0.1	-1.015	-0.478	-0.11	-0.36648	-0.727	2.45	-0.507	-0.637	-0.339	-1.056	0.161168	-0.5549	INT	BUR	2XOV	GLPG	AH	2.98	0.799608
K	173	A	-0.76	-0.535	-0.877	-1.45	-1.6414	-1.079	0.96	-1.069	-0.178	-1.407	-4.765	-0.07561	-0.3115	INT	BUR	2XOV	GLPG	AH	3.05	0.763084
L	174	A	-1.88	-4.875	-4.354	-2.71	-3.59217	-2.395	-2.34	-2.705	-3.24	-2.977	-4.865	-2.5175	-0.9183	INT	BUR	2XOV	GLPG	AH	1.71	2.55451
G	194	V	-1.2	-19.262	-123.735	-0.26	-5.22042	-0.374	-0.17	-0.226	-0.442	-0.347	-8.431	-2.06291	-1.0002	INT	BUR	2XOV	GLPG	AH	2.47	0.448202
F	197	A	-0.93	-6.586	-8.427	-1.11	-3.77177	-3.191	-3.82	-3.442	-4.805	-1.458	-7.998	-1.8904	-2.6406	INT	BUR	2XOV	GLPG	AH	3.58	0.3658
G	198	V	-0.64	-5.801	-110.911	0	-11.0418	0.095	2	0.343	-0.132	-0.82	-8.997	-1.58175	0.2437	INT	BUR	2XOV	GLPG	AH	2.02	0.276188
G	199	V	-1.2	-28.632	-197.59	-0.08	-14.5222	-0.387	-1.02	-0.472	-0.313	-0.976	-8.997	0.715402	0.3104	INT	BUR	2XOV	GLPG	AH	7.85	2.044
G	199	A	-1.65	-8.434	-9.655	-0.47	-4.84136	-1.024	-2.19	-1.239	-0.652	-1.05	-5.998	-1.34649	-0.5188	INT	BUR	2XOV	GLPG	AH	7.85	2.044

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
R	214	A	-0.07	-0.431	1.121	-1.49	-2.6945	-1.531	0.99	-1.581	-0.723	-1.407	-1.29	-2.04949	-1.8914	INT	BUR	2XOV	GLPG	AH	-0.44	0.662798
G	215	V	-0.8	-7.661	-252.105	-0.53	-6.67194	-0.851	2.72	-0.222	-0.342	-1.675	-8.51	-1.82131	0.0027	INT	BUR	2XOV	GLPG	AH	1.89	0.062846
D	218	A	-1.42	0.316	-1.388	-1.09	-0.3578	-0.279	0.51	-0.401	-2.818	-3.005	-0.546	-0.63337	-0.2326	INT	BUR	2XOV	GLPG	AH	0.2	0.40706
W	241	A	-0.88	-1.76	-2.42	-0.64	-1.56166	-2.022	0.45	-1.772	-2.632	-1.54	-4.655	-1.15392	-1.9197	INT	BUR	2XOV	GLPG	AH	2.15	0.720551
M	247	A	-0.19	-3.942	-2.389	-0.84	-2.4663	-2.699	-3.16	-2.913	-2.29	-1.323	-3.243	-1.60968	-0.5791	INT	BUR	2XOV	GLPG	AH	-1.03	0.420906
A	253	V	-1.43	-10.54	-31.141	-0.04	-5.43618	-0.491	-0.18	-0.362	-1.039	-0.046	-3.897	-1.10838	-0.3846	INT	BUR	2XOV	GLPG	AH	6.47	1.04775
S	269	V	-1.41	-1.583	-14.054	0.09	0.107241	-0.277	2.94	0.329	0.481	-0.055	-2.243	-0.63473	-0.4547	INT	BUR	2XOV	GLPG	AH	0.99	0.121279
L	75	A	-1.3	-1.504	-0.781	-1.09	-1.04807	-1.547	1.13	-1.245	-2.035	-2.473	-4	-0.06106	-2.1201	INT	BUR	1AFO	GLYA	AH	1.43	-0.21112
I	76	A	-1.8	-1.1543	-0.841	-0.57	-1.90674	-2.389	1.2	-2.171	-0.668	-0.339	-5	-0.67316	-1.6811	INT	BUR	1AFO	GLYA	AH	5.73	0.448366
G	79	A	-1.7	1.58	1.268	-0.72	0.618656	-1.141	4.34	-0.628	-0.314	-1.013	-2.688	-0.30954	0.417	INT	BUR	1AFO	GLYA	AH	4.97	0.061509
V	80	A	-0.4	-0.3663	-0.30867	-0.72	1.94038	-1.438	1.57	-1.098	0.25	-0.349	-3.769	0.504461	-0.5081	INT	BUR	1AFO	GLYA	AH	5.28	0.197526
D	24	N	-0.6	2.959	1.717	-0.66	-0.31891	-1.25	-0.22	-1.154	-0.034	-0.814	-4.811	-0.84255	0.146	INT	BUR	3GP6	PAGP	BB	4.31	0.548193
S	58	A	-1.2	-0.419	-1.213	-0.78	0.263235	-2.106	1.85	-2.116	-0.685	-0.786	-2.912	-0.41841	-0.545	INT	BUR	3GP6	PAGP	BB	1.87	0.156614
E	90	A	-0.7	0.394	-1.315	-0.5	-1.86516	-3.403	1.52	-3.135	0.564	-0.917	-5.574	0.145003	-0.2512	INT	BUR	3GP6	PAGP	BB	1.96	0.177297
R	94	A	-1.5	-1.246	0.852	-0.18	-1.82855	-0.801	0.45	-0.978	-0.659	-0.106	-4.4	0.00139	-1.0818	INT	BUR	3GP6	PAGP	BB	1.01	0.094195
K	30	M	0.3	1.46	1.098	0.22	0.337787	-0.326	1.92	-0.36	1.496	-0.349	-0.506	0.524433	0.3342	INT	EXP	1PY6	BR	AH	-1.04	1.1385
K	40	A	-0.3	-0.043	-0.164	-0.34	-1.04477	-0.863	0.96	-0.733	0.119	-0.426	-1.953	-0.3714	-1.3626	INT	EXP	1PY6	BR	AH	3.04	1.28817
K	40	P	-1	-20.169	-262.387	-0.33	-6.39931	-0.175	-1.64	-0.256	-1.048	-1.328	-2.354	-1.04875	-1.0942	INT	EXP	1PY6	BR	AH	3.04	1.28817
K	41	A	-1.4	-0.363	-1.009	-0.32	-1.50099	-0.777	1.46	-0.566	0.03	-0.629	-1.223	-0.33748	-1.107	INT	EXP	1PY6	BR	AH	-0.19	0.161854
K	41	P	-0.6	-12.212	-200.973	-0.3	-5.25026	-0.059	-1.1	-0.013	-0.786	-1.239	-1.767	-1.16114	-1.1208	INT	EXP	1PY6	BR	AH	-0.19	0.161854
L	58	A	0.3	-2.141	-2.911	-1.52	-2.65041	-1.889	-0.71	-1.904	-1.189	-2.58	-1.744	-0.44781	-2.1872	INT	EXP	1PY6	BR	AH	2.3	1.2341

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
S	59	A	-0.1	1.056	1.079	-0.17	0.829447	-0.822	2.3	-0.616	-1.618	-1.013	0.258	0.401934	0.2583	INT	EXP	1PY6	BR	AH	0.94	0.110075
L	61	A	0.7	-0.551	-1.494	-1.5	-1.55472	-1.743	-0.71	-1.744	-1.697	-2.588	0.937	-0.95892	-1.2857	INT	EXP	1PY6	BR	AH	-1.22	0.638659
L	62	A	0.5	-1.779	-1.478	-1.32	-3.02636	-1.649	-0.28	-1.595	-0.898	-3.148	-0.266	0.21736	-0.9735	INT	EXP	1PY6	BR	AH	1.01	-0.10796
A	62	G	-1.3	-1.138	-1.721	-1.11	-0.77373	-0.899	-2.4	-1.172	-0.727	-1.28	-2.693	-0.50931	-1.605	INT	EXP	2K73	DSBB	AH	1.96	0.671679
P	95	A	-0.49	-0.142	-0.889	-0.69	-1.56944	-0.738	2.21	-0.327	-1.012	-0.922	-7.977	NA	0.1259	INT	EXP	2XOV	GLPG	AH	5.83	1.86678
E	134	A	-0.4	-0.15	-1.481	-0.21	-0.84826	0.512	0.63	0.481	-1.709	-0.903	-4.23	-0.87803	-0.7376	INT	EXP	2XOV	GLPG	AH	5.67	1.10225
W	136	A	-0.51	-2.446	-2.358	-0.59	-2.29546	-2.544	0.67	-2.247	-2.534	-1.243	-13.928	0.467498	-4.0956	INT	EXP	2XOV	GLPG	AH	12.35	7.66927
Y	138	F	-2.07	-0.818	-0.665	-0.19	-0.67328	-0.531	0.72	-0.311	0.484	-0.686	-1.49	-0.64244	-0.4298	INT	EXP	2XOV	GLPG	AH	0.57	0.95745
F	146	A	-0.59	-4.715	-3.832	-1.11	-2.15454	-2.278	-0.34	-2.334	-1.778	-1.45	-7.981	-1.55666	-2.1513	INT	EXP	2XOV	GLPG	AH	4.17	0.278983
S	147	A	-1.2	-0.067	-0.794	-0.34	-1.44422	-0.521	-0.72	-0.526	-0.939	-0.607	-2.873	-0.95767	-0.6013	INT	EXP	2XOV	GLPG	AH	2.81	0.68463
R	168	A	0.85	-0.803	1.125	-0.91	-1.51479	-0.458	0.35	-0.371	-0.843	-0.532	-3.247	-1.36601	0.0186	INT	EXP	2XOV	GLPG	AH	1.65	0.0425
Q	190	A	-0.17	1.532	0.331	-0.02	-0.769	-0.859	2.81	-0.655	-1.464	-0.38	0.558	0.494176	-0.0647	INT	EXP	2XOV	GLPG	AH	-2.86	0.803091
P	219	A	0.46	-1.5	-2.199	-1.22	-2.10626	-0.447	2.21	-0.03	-0.891	-1.389	-7.996	-1.46481	-0.8103	INT	EXP	2XOV	GLPG	AH	3.62	1.16513
S	221	A	-0.69	-0.717	-0.687	-1.02	-0.22563	-0.61	3.09	-0.399	-1.016	-1.232	-0.456	-0.56495	-0.7578	INT	EXP	2XOV	GLPG	AH	-0.06	0.229632
Q	226	A	-0.25	-0.187	-0.283	-0.09	-0.24012	-0.457	0.76	-0.403	-0.004	0.078	-0.677	-0.03163	-0.2544	INT	EXP	2XOV	GLPG	AH	0.48	0.141805
D	243	A	-0.51	0.651	-0.185	0.44	-0.06883	-0.343	-0.03	-0.275	-0.854	0.077	-4.727	-0.35111	0.0306	INT	EXP	2XOV	GLPG	AH	0.35	0.101607
D	268	A	-1.1	0.019	0.081	-1.59	-1.80475	0.229	2.18	0.481	-2.426	-2.98	-7.615	0.130846	-0.5171	INT	EXP	2XOV	GLPG	AH	-0.66	1.02971
I	77	A	-0.1	-2.2853	-1.042	-0.98	-0.34364	-1.625	-0.26	-1.481	-0.842	-0.945	-4.923	-0.342	-1.5816	INT	EXP	1AFO	GLYA	AH	4.81	0.274534
F	78	A	0.1	0.3507	-1.457	-1.12	-0.80652	-1.577	-0.86	-1.525	-2.663	-1.123	-4.94	-0.50368	-1.6037	INT	EXP	1AFO	GLYA	AH	6.08	1.03481
W	7	A	-3.6	-3.244	-6.192	-0.58	-1.73668	-2.061	0.44	-1.816	-0.896	-0.725	-9.46	-1.91537	-3.5398	INT	EXP	1QJP	OMPA	BB	7.48	2.91973
W	15	A	-2	-0.374	-3.234	-0.77	-2.18887	-2.253	-0.25	-2.122	-0.964	-1.375	-11.82	-1.32924	-3.2831	INT	EXP	1QJP	OMPA	BB	4.75	1.67298
Y	43	A	-3.8	-3.992	-6.205	-0.98	-3.00761	-2.36	-1.2	-2.62	-0.325	-0.858	-8.163	-1.92287	-2.5719	INT	EXP	1QJP	OMPA	BB	9.61	3.68364

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
W	57	A	-2	-4.157	-5.996	-0.95	-2.00093	-2.56	-0.25	-2.395	-0.979	-1.598	-7.505	-2.04251	-3.2328	INT	EXP	1QJP	OMPA	BB	5.42	1.55394
F	123	A	-2.1	-2.609	-6.269	-0.93	-2.60065	-2.495	-1.92	-2.784	-0.784	-1.125	-0.668	-1.04852	-1.963	INT	EXP	1QJP	OMPA	BB	2.25	0.686893
Y	129	A	-2.7	-2.771	-5.041	-1.17	-2.14551	-1.967	-1.43	-2.085	-0.126	-0.763	-8.71	-2.28002	-2.3268	INT	EXP	1QJP	OMPA	BB	8.86	3.31672
Y	141	A	-3.3	-2.752	-6.053	-0.9	-2.29389	-2.109	-0.91	-2.152	-0.087	-1.312	-8.734	-2.57521	-2.5331	INT	EXP	1QJP	OMPA	BB	9.39	4.20551
W	143	A	-3.1	-6.193	-5.754	-0.69	-3.06563	-2.495	-0.25	-2.338	-0.965	-1.087	-11.039	-2.84753	-3.4715	INT	EXP	1QJP	OMPA	BB	2.16	2.59991
Y	168	A	-0.2	-3.072	-6.272	-0.62	-2.75947	-1.987	-1.43	-2.107	-0.129	-0.094	-8.528	-2.08809	-2.3658	INT	EXP	1QJP	OMPA	BB	9.55	4.29754
F	170	A	-2.4	-1.669	-4.975	-0.94	-1.10341	-1.772	-1.06	-1.824	-0.753	-0.558	-7.715	-1.32334	-2.2804	INT	EXP	1QJP	OMPA	BB	9.78	5.20867
Y	214	A	-2.4	-4.533	-4.409	-1	1.56585	-1.959	-1.43	-2.066	-3.492	-0.986	-5.364	-1.42968	-1.9024	INT	EXP	1QD6	OMPLA	BB	2.79	0.766949
Y	214	L	1.2	-2.459	-2.801	-0.02	-0.04169	-1.064	-0.62	-1.03	0.129	-0.409	-4.583	-0.55606	-0.5698	INT	EXP	1QD6	OMPLA	BB	2.79	0.766949
Y	214	R	-0.6	-3.461	-4.231	-0.36	0.95489	-0.395	-1.2	-0.356	-0.914	-0.955	-4.73	-0.52236	-0.817	INT	EXP	1QD6	OMPLA	BB	2.79	0.766949
W	17	A	-1	-0.194	-0.45	-0.79	0.342552	-1.603	0.87	-1.358	-1.091	-1.82	-13.788	-0.40647	-3.7881	INT	EXP	3GP6	PAGP	BB	9.64	3.41313
Y	23	A	-0.9	-3.156	-2.377	-1.06	-1.83179	-1.915	-1.2	-2.128	-0.718	-1.765	-4.622	-0.61106	-1.8353	INT	EXP	3GP6	PAGP	BB	3.11	1.85142
W	51	A	-0.4	-1.016	-0.521	-0.4	-0.6627	-1.095	0.48	-0.848	-0.997	-0.669	-13.504	-0.21259	-3.9838	INT	EXP	3GP6	PAGP	BB	9.29	3.25137
Y	153	A	-0.6	-5.168	-5.98	-1.16	-2.8321	-2.501	-1.43	-2.703	-0.737	-1.376	-5.427	-0.21259	-1.866	INT	EXP	3GP6	PAGP	BB	1.1	-0.0234
M	20	A	-2.8	-2.339	-2.001	-1.62	-3.10244	-2.149	-2.33	-2.26	-3.194	-1.99	-1.884	-1.78721	-1.1776	MID	BUR	1PY6	BR	AH	8.51	2.75225
T	24	A	0.6	0.628	0.271	-0.93	0.577373	-1.381	2.29	-1.237	-1.455	-0.667	-1.094	0.118989	-0.2299	MID	BUR	1PY6	BR	AH	3.43	0.153928
T	24	S	-0.2	-0.895	-1.265	-0.48	-0.83707	-1.419	-0.15	-1.383	-0.842	-0.588	-1.315	-0.04158	-0.6573	MID	BUR	1PY6	BR	AH	3.43	0.153928
T	24	V	0.3	-0.989	-9.692	0.09	0.555354	-0.802	2.13	-0.608	1.055	0.107	-0.623	0.621022	0.2634	MID	BUR	1PY6	BR	AH	3.43	0.153928
F	27	A	-2.1	-5.707	-6.35	-1.85	-3.90452	-2.86	-2.6	-3.113	-4.814	-2.132	-4.281	-1.92578	-1.7097	MID	BUR	1PY6	BR	AH	7.84	1.98627
T	46	A	-2.2	0.198	0.087	-1.12	-0.29372	-1.491	1.65	-1.379	-1.303	-1.06	-2.223	-0.53416	-0.9121	MID	BUR	1PY6	BR	AH	2.52	0.109583
T	46	P	-1.1	-15.209	-124.423	-0.85	-4.30328	-0.652	-1.74	-0.918	-0.828	-1.094	-2.824	-1.36443	-0.6476	MID	BUR	1PY6	BR	AH	2.52	0.109583

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
T	46	S	-0.1	-0.817	-0.831	-0.81	-0.20398	-1.582	-1.21	-1.742	-1.5	-0.857	-1.812	-0.65796	-1.4476	MID	BUR	1PY6	BR	AH	2.52	0.109583
T	47	A	-1.1	0.612	-0.401	-1.17	-0.90532	-1.589	2.29	-1.466	-1.342	-1.085	-0.657	-0.40993	-0.5891	MID	BUR	1PY6	BR	AH	0.64	0.358364
T	47	P	-0.9	-5.922	-307.676	-0.76	-5.53322	-0.968	-0.28	-1.02	-1.095	-1.387	-2.051	-1.26146	-0.5106	MID	BUR	1PY6	BR	AH	0.64	0.358364
V	49	A	-0.3	-2.414	-2.75	-1.21	-2.36722	-2.253	-1.53	-2.499	-1.948	-1.548	-2.602	0.02032	-1.8325	MID	BUR	1PY6	BR	AH	4.71	0.549385
P	50	A	0.6	0.057	-2.793	-0.88	-1.34234	-2.408	2.74	-2.26	-1.369	-1.13	-2.166	0.288419	-0.9665	MID	BUR	1PY6	BR	AH	2.27	0.864056
Y	57	A	-3.7	-3.871	-3.795	-1.75	-3.69287	-3.778	-0.94	-3.885	-3.898	-1.789	-5.918	-2.1885	-2.7688	MID	BUR	1PY6	BR	AH	9.39	3.65032
T	90	A	-1.3	0.432	0.435	-1.61	0.134111	-0.937	1.65	1.65	-1.234	-1.457	-4.316	0.766321	-1.0957	MID	BUR	1PY6	BR	AH	4.88	1.88375
L	94	A	-3.1	-4.369	-3.941	-1.84	-3.73194	-2.85	-2.34	-3.179	-3.675	-1.967	-4.588	-1.36395	-2.5967	MID	BUR	1PY6	BR	AH	4.29	0.425871
D	96	A	-1.5	3.844	3.776	-0.79	1.34407	2.349	2.29	2.348	-2.25	-1.003	-3.485	1.43065	-0.0484	MID	BUR	1PY6	BR	AH	1.12	0.072252
L	97	A	-2.9	-5.071	-4.811	-2.27	-3.47551	-2.522	-2.34	-2.839	-3.886	-1.964	-4.424	-1.98634	-2.8043	MID	BUR	1PY6	BR	AH	5.57	0.595715
L	111	A	-1.7	-3.532	-4.087	-2.13	-4.03529	-2.389	-2.34	-2.693	-3.791	-2.958	-2.991	-1.81036	-2.1197	MID	BUR	1PY6	BR	AH	3.33	-0.0754
D	115	A	0.5	5.061	4.842	-0.07	2.99274	-0.877	2.29	-0.41	-2.255	-0.454	-7.347	1.8039	-0.0227	MID	BUR	1PY6	BR	AH	4.39	0.331542
I	148	A	-2.3	-5.237	-5.041	-2.13	-3.53554	-2.685	-2.25	-3.006	-3.645	-2.205	-3.395	-1.36622	-1.0292	MID	BUR	1PY6	BR	AH	3.44	0.056015
I	148	V	-0.2	-2.205	-2.056	-0.58	-0.87824	-1.462	-0.98	-1.596	-1.156	-0.94	-0.088	-0.48356	-0.5838	MID	BUR	1PY6	BR	AH	3.44	0.056015
L	152	A	-1.9	-4.448	-3.021	-2.87	-3.86642	-2.567	-2.34	-2.887	-4.068	-2.413	-4.576	-1.59037	-1.2206	MID	BUR	1PY6	BR	AH	4.22	0.490358
F	171	A	-1.1	-6.372	-6.845	-2.28	-3.92959	-3.315	-2.6	-3.552	-4.808	-2.371	-6.302	-1.09659	-2.9789	MID	BUR	1PY6	BR	AH	5.83	1.99131
L	174	A	-1.8	-5.598	-5.295	-2.94	-4.2928	-2.462	-2.34	-2.776	-4.068	-1.972	-4.458	-1.7667	-2.6417	MID	BUR	1PY6	BR	AH	4.15	0.399443
Y	185	A	-4.2	-3.137	-4.971	-2.28	-4.25098	-0.462	1.79	-0.158	-4.35	-1.809	-8.948	-1.22082	-2.6555	MID	BUR	1PY6	BR	AH	8.79	2.84939
Y	185	F	-0.4	0.014	-0.874	-1.01	-0.49256	-0.462	1.79	-0.158	-0.567	-0.973	-3.951	0.565192	-0.5816	MID	BUR	1PY6	BR	AH	8.79	2.84939
P	186	A	-0.9	-0.933	-2.116	-0.97	-3.09477	-1.353	2.74	-1.103	-0.787	-1.116	-7.987	-0.12288	-1.2163	MID	BUR	1PY6	BR	AH	1.49	0.013079
D	212	A	-1.2	0.429	-1.745	-1.06	-2.91971	0.764	2.29	1.247	-2.133	-1.116	-7.442	0.210193	-0.7508	MID	BUR	1PY6	BR	AH	8.41	2.49604
A	19	G	0.4	-2.414	-1.989	-1.11	-1.49573	-1.204	-3.48	-1.439	-0.804	-0.891	-2.64	-1.09652	-1.6761	MID	BUR	2K73	DSBB	AH	2.17	0.943208

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
E	26	L	2.3	4.837	3.312	0.74	1.81949	0.044	0.18	0.091	-0.047	0.667	-6.901	2.53069	0.2966	MID	BUR	2K73	DSBB	AH	0.38	0.090519
A	29	G	0.4	-2.208	-1.746	-1.25	-0.60918	-0.939	-3.54	-1.056	-0.315	-1.436	-3.439	-0.33592	-1.5893	MID	BUR	2K73	DSBB	AH	5.66	0.201734
A	57	G	-0.7	-3.153	0.565	-1.26	-0.36225	-0.903	-4.24	-1.195	-1.021	-1.555	-3.583	-1.10924	-2.1296	MID	BUR	2K73	DSBB	AH	0.97	0.148829
A	157	G	1.1	-1.151	-2.164	-1.9	-1.81567	-1.064	-4.24	-1.361	-0.937	-2.25	-1.481	-1.07299	-1.0985	MID	BUR	2K73	DSBB	AH	2.1	0.365544
M	100	A	-1.67	-2.286	-1.79	-1.76	-3.05308	-2.779	-2.3	-2.91	0.399	-1.934	-1.7	-1.67456	-0.8574	MID	BUR	2XOV	GLPG	AH	1.16	1.02854
C	104	A	-0.93	0.088	-1.361	-1.17	-0.20833	-1.904	-2.04	-2.12	-1.269	-1.37	-7.999	-0.88949	-0.1284	MID	BUR	2XOV	GLPG	AH	7.27	2.97312
C	104	V	-0.89	2.646	0.22	-0.36	1.40725	-1.206	0.23	-1.113	-0.001	-0.941	-6.721	0.535398	-1.1013	MID	BUR	2XOV	GLPG	AH	7.27	2.97312
N	154	A	-1.5	-2.36	-1.96	-0.61	-0.75476	-1.859	2.73	-1.744	-1.374	-1.108	-7.981	0.339105	-1.2673	MID	BUR	2XOV	GLPG	AH	9.32	3.90977
L	155	A	-1.03	-4.987	-4.33	-2.14	-3.07127	-2.688	-1.45	-2.918	-1.353	-2.062	-4.722	-2.06768	-1.0756	MID	BUR	2XOV	GLPG	AH	1.14	3.84093
W	158	F	-0.02	-2.065	-2.861	-0.83	-1.64257	-2.221	-1.69	-2.469	-1.411	-0.848	-9.581	-1.93696	-0.9502	MID	BUR	2XOV	GLPG	AH	4.91	1.28589
G	162	V	-3.07	-13.835	-46.571	-0.36	-5.88372	0.545	2.82	0.896	-0.228	-0.908	-7.366	0.715402	0.2543	MID	BUR	2XOV	GLPG	AH	7.52	1.4351
I	177	A	-2.17	-5.718	-4.373	-1.77	-3.90319	-2.786	-2.25	-3.111	-2.765	-1.958	-4.131	-2.61936	-0.8263	MID	BUR	2XOV	GLPG	AH	4.74	0.707237
T	178	A	-0.95	-0.408	-0.246	-1.21	-0.45828	-0.836	1.65	-0.866	-1.159	-0.665	-1.418	-0.65941	-0.5167	MID	BUR	2XOV	GLPG	AH	-1.76	3.65551
L	200	A	-0.83	-6.171	-6.143	-1.78	-4.01096	-2.737	-2.77	-3.215	-3.331	-1.407	-4.932	-2.6618	-2.3889	MID	BUR	2XOV	GLPG	AH	0.32	0.55758
S	201	A	-0.27	-0.236	-0.247	-0.29	0.019565	-0.547	2.85	-0.339	-1.197	-0.859	-2.898	-0.18387	-0.6078	MID	BUR	2XOV	GLPG	AH	7.66	1.47001
G	202	A	-0.73	-0.808	-15.307	-0.12	-1.10093	-0.847	2.72	-0.172	-0.063	-0.81	-5.998	-0.83231	0.0796	MID	BUR	2XOV	GLPG	AH	7.37	1.31781
V	203	A	-1.55	-3.929	-2.542	-1.05	-2.12148	-2.037	-1.53	-2.265	-2.002	-1.379	-3.821	-1.54225	-1.5965	MID	BUR	2XOV	GLPG	AH	2.13	0.352724
Y	205	A	-0.58	-2.802	-3.26	-1.16	-3.23556	-2.46	-0.94	-2.613	-1.597	-1.277	-8.597	-2.53327	-2.5374	MID	BUR	2XOV	GLPG	AH	5.76	2.50673
A	206	G	-1.05	-3.49	-2.508	-1.38	-1.769	-2.184	-4.24	-2.591	-1.345	-1.653	-3.899	-1.2774	-1.6149	MID	BUR	2XOV	GLPG	AH	2.23	1.17171
L	207	A	-0.92	-6.914	-5.033	-1.75	-3.48723	-2.817	-2.34	-3.148	-3.182	-1.753	-4.465	-2.70335	-2.8191	MID	BUR	2XOV	GLPG	AH	5.16	0.711694
G	209	V	-0.57	-9.637	-76.864	-0.29	-4.34245	-0.542	2.72	-0.231	-0.446	-0.484	-8.181	-0.75352	0.4212	MID	BUR	2XOV	GLPG	AH	7.07	0.797192
Y	210	F	-1.65	-0.323	-0.996	-1.02	-1.19076	-1.505	1.79	-1.29	-1.077	-0.764	-3.999	-0.45914	-0.5159	MID	BUR	2XOV	GLPG	AH	3.22	0.492314

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
W	236	G	-0.46	-9.684	-9.753	-1.35	-6.62841	-2.161	-5.46	-2.161	-1.829	-1.5	-12.322	-3.05154	-2.4533	MID	BUR	2XOV	GLPG	AH	8.4	2.63503
G	257	V	-2.49	-21.589	-396.853	-0.02	-9.49234	-0.599	2.82	-0.289	-0.145	-0.561	-8.992	0.824236	0.2608	MID	BUR	2XOV	GLPG	AH	7.95	2.88688
G	261	V	-4.68	-36.175	-835.01	-0.35	-16.2964	-0.36	2.82	-0.034	-0.242	-0.665	-8.992	0.96582	0.2574	MID	BUR	2XOV	GLPG	AH	8.04	3.09784
G	83	A	-3.2	1.3267	0.944	-0.75	-2.00278	-0.968	3.88	-0.374	0.032	-1.573	-5.269	0.393422	0.162	MID	BUR	1AFO	GLYA	AH	5.38	0.025808
V	84	A	-1	-0.5433	-0.416	-0.97	-1.03898	-1.408	0.38	-1.164	0.037	-1.353	-3.809	0.184811	-0.4959	MID	BUR	1AFO	GLYA	AH	3.21	0.255775
T	87	A	-0.9	0.2424	0.01	-1.02	1.03008	-1.121	2.21	-0.823	-1.13	-0.333	-3.538	0.304209	-0.107	MID	BUR	1AFO	GLYA	AH	4.14	0.092334
Y	87	F	-1.4	-0.492	-0.657	-0.29	-0.46217	-0.746	1.01	-0.49	-0.39	0.094	-3.909	-0.2457	-0.6863	MID	BUR	3GP6	PAGP	BB	6.85	1.04334
A	44	P	-0.5	-9.15	-269.912	-0.51	-3.4298	-0.414	-2.22	-0.199	-1.885	-1.474	-1.546	-0.58931	-0.6761	MID	EXP	1PY6	BR	AH	0.58	0.689062
I	45	A	-1.9	-2.609	-2.489	-1.63	-1.7936	-2.239	-1.07	-2.4	-1.469	-1.294	-0.823	-0.40913	-2.1408	MID	EXP	1PY6	BR	AH	3.25	0.328107
L	48	A	-0.1	-1.344	-1.176	-1.53	-0.81949	-1.62	-0.71	-1.466	-3.049	-1.817	-1.876	-0.57879	-2.1983	MID	EXP	1PY6	BR	AH	2.43	0.30973
A	51	P	-2.4	-32.895	-217.741	-0.31	-6.01279	-0.329	-3.24	-0.366	-0.808	-1.442	-1.835	-0.17236	-0.9428	MID	EXP	1PY6	BR	AH	1.28	-0.04798
I	52	A	-1.5	-3.662	-3.97	-1.71	-1.64766	-2.219	-0.26	-2.263	-0.871	-1.715	-3.027	-1.46487	-2.1269	MID	EXP	1PY6	BR	AH	4.82	1.05321
F	54	A	-0.4	-2.522	-3.605	-1.46	-2.87125	-2.581	-0.86	-2.655	-3.373	-2.042	0.188	-0.67756	-1.803	MID	EXP	1PY6	BR	AH	3.39	0.568897
T	55	A	-0.1	0.691	-0.035	-0.58	-0.34088	-0.956	2.73	-0.653	-2.076	-0.595	-0.017	-0.09193	-0.3402	MID	EXP	1PY6	BR	AH	0	0.597079
M	56	A	1.6	-1.206	-0.931	-1.38	-0.9327	-1.262	-0.38	-1.05	-0.378	-2.281	0.938	-0.6817	-1.2435	MID	EXP	1PY6	BR	AH	0.44	0.438823
P	91	A	-1.3	-0.009	-1.576	-1	-2.29059	-1.865	2.28	-1.666	-0.793	-1.111	-7.153	1.34378	-1.6209	MID	EXP	1PY6	BR	AH	8.43	2.31823
F	82	A	-1.9	2.895	-0.565	-1.2	-0.82157	-1.176	1.01	-0.971	-0.885	-2.369	-2.946	-0.28509	-1.3783	MID	EXP	2K73	DSBB	AH	1.12	0.179947
R	83	A	-1	0.385	0.488	-0.2	-0.02065	-0.655	0.79	-0.422	-0.768	-0.987	-2.803	-0.33368	-0.2061	MID	EXP	2K73	DSBB	AH	-1.82	0.319151
Y	89	A	1.3	-0.482	-0.602	-0.69	-0.52896	-1.029	1.03	-0.787	-0.209	-1.48	-2.679	-0.60467	-1.3288	MID	EXP	2K73	DSBB	AH	0.52	0.196719
G	148	A	-0.2	0.921	1.459	-0.38	-0.02869	-0.176	4.34	0.34	0.376	-0.297	-0.275	0.343948	0.8243	MID	EXP	2K73	DSBB	AH	-2.12	0.312067
A	152	G	0	-1.654	-1.551	-1.6	-1.00596	-0.669	-3.88	-0.684	0.006	-2.155	-1.811	-0.38507	-1.2174	MID	EXP	2K73	DSBB	AH	1.64	0.213919

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
F	139	A	-1.12	-4.386	-4.519	-1.41	-3.46594	-2.318	-0.86	-2.386	-2.162	-2.14	-5.244	-2.38461	-2.9732	MID	EXP	2XOV	GLPG	AH	4.1	0.885554
T	140	A	-0.615	-0.489	-0.618	-0.7	-0.30758	-0.926	1.65	-0.784	-1.271	-0.805	-2.554	0.592648	-1.4052	MID	EXP	2XOV	GLPG	AH	7.55	1.46167
F	153	A	-0.682	-1.808	-2.162	-1.03	-2.2802	-2.28	-1.26	-2.443	-2.041	-1.669	-7.514	-2.07771	-1.4495	MID	EXP	2XOV	GLPG	AH	6.6	1.35198
Y	160	F	-0.54	-0.778	-0.007	-0.17	0.353031	-0.704	2.11	-0.392	0.733	-0.424	-3.499	-0.02169	-0.0068	MID	EXP	2XOV	GLPG	AH	2.53	0.541518
L	179	A	-0.37	-3.185	-1.73	-2.14	-1.204	-1.528	-0.71	-1.393	-1.053	-3.37	-3.971	-1.31004	-1.1371	MID	EXP	2XOV	GLPG	AH	4.68	0.713875
I	180	A	-0.43	-2.475	-1.64	-1.59	-1.31705	-1.509	-0.26	-1.333	-0.719	-2.568	-3.649	-1.24929	-0.8028	MID	EXP	2XOV	GLPG	AH	2.87	0.217636
L	184	A	0.41	-2.49	-1.475	-1.49	-1.71096	-1.689	-0.71	-1.559	-0.527	-2.213	-3.235	-1.35843	-1.1636	MID	EXP	2XOV	GLPG	AH	0.27	0.316792
L	229	A	0.2	-2.581	-2.378	-2.03	-1.48717	-1.785	-1.45	-1.954	0.974	-3.685	-3.94	-2.36507	-1.1845	MID	EXP	2XOV	GLPG	AH	2.72	0.010317
M	81	A	0.2	-0.3613	-0.761	-0.65	-1.16247	-1.143	0.21	-0.715	-3.049	-0.956	-5.971	-0.46672	-0.7949	MID	EXP	1AFO	GLYA	AH	6.74	1.14724
I	85	A	0.4	-1.2056	-0.741	-1.15	-1.23335	-0.932	1.2	-0.588	-2.271	-2.212	-4.846	-0.59136	-0.819	MID	EXP	1AFO	GLYA	AH	4.49	0.208342
G	86	A	0.1	1.5117	1.318	-0.88	0.752139	-0.44	4.34	0.093	-0.853	-2.072	-5.308	0.517997	0.2816	MID	EXP	1AFO	GLYA	AH	5.98	0.102439
F	51	A	-1.2	-1.753	-3.159	-1.09	-2.35883	-2.115	-1.96	-2.348	-0.791	-1.572	-2.897	-1.7261	-2.1667	MID	EXP	1QJP	OMPA	BB	0.5	0.591185
Y	55	A	-2.5	-2.107	-5.115	-0.67	-2.31728	-1.508	-1.43	-1.7	-0.26	-1.095	-7.482	-1.74505	-2.5737	MID	EXP	1QJP	OMPA	BB	7.6	1.87744
L	120	A	2.2	-3.918	-2.383	-2.21	2.71066	-1.692	-1.69	-1.749	-3.731	-2.402	-3.342	-1.8172	-2.4846	MID	EXP	1QD6	OMPLA	BB	2.44	0.929372
L	120	R	-2.4	-3.743	-2.646	-1.36	1.53336	-0.927	-0.63	-0.674	-1.006	-1.755	-4.563	-0.88535	-2.0069	MID	EXP	1QD6	OMPLA	BB	2.44	0.929372
A	164	L	1.2	-0.481	0	-0.19	-2.32023	-0.007	1.69	0.276	0.593	-0.304	-1.308	1.30216	0.3019	MID	EXP	1QD6	OMPLA	BB	0.69	1.79905
A	164	R	-0.8	0.034	-0.269	-0.52	-1.21824	-1.241	1.36	-0.925	-0.95	-0.781	-2.158	0.344995	-0.8499	MID	EXP	1QD6	OMPLA	BB	0.69	1.79905
A	210	C	-0.5	-1.438	0.256	-0.64	-0.35713	-0.873	0.33	-0.66	-2.033	-1.01	-1.161	0.185007	0.3262	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	D	-3	-2.018	-2.589	-0.8	0.86337	-0.472	-1.47	-0.221	-0.875	-0.875	-3.024	-0.41689	-1.607	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	E	-1.6	-2.016	-0.464	-0.65	-0.45145	-0.543	0.02	-0.155	-0.948	-0.91	-2.297	0.255094	-1.3907	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	F	2.2	0.112	0.596	-0.18	-2.34466	-0.891	1.97	-0.726	-0.07	-0.581	-0.464	1.3018	0.5628	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	G	-1.7	-2.341	-1.806	-1.19	1.22987	-0.873	-1.64	-0.797	-2.692	-0.83	-2.467	-0.55555	-1.6618	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
A	210	H	-4.8	-0.752	-0.267	-0.83	-0.41622	-0.982	-0.87	-0.841	-1.652	-1.08	-2.403	0.299952	-0.1788	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	I	1.6	-0.304	-0.308	0.14	-1.81841	-0.281	2.23	0.121	0.835	-0.533	0.57	1.6301	0.7194	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	K	-5.4	0.228	0.441	-0.7	-1.34381	-0.889	-0.54	-0.522	-1.658	-0.573	-1.825	0.598039	-0.5267	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	L	1.8	0.874	0.049	-0.06	-1.68256	-0.281	1.69	0.096	0.593	-0.304	1.097	1.58062	0.5073	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	M	0.8	-0.374	0.259	-0.07	-2.12617	-0.29	0.93	-0.116	-0.53	-0.359	0.825	1.3761	0.2188	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	N	-3.5	-0.477	-0.873	-0.8	-0.2909	-0.548	-1.79	-0.334	-0.875	-1.617	-2.192	-0.15397	-0.8139	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	P	1.5	-7.506	-186.46	-0.41	2.87929	-0.369	-2.27	-0.285	-1.86	-1.033	-2.401	-1.04221	-0.1028	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	Q	-3	-0.813	0.53	-0.52	-0.58435	-0.73	0.67	-0.34	-0.941	-0.926	-1.698	0.260014	-1.1102	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	R	-3.7	-0.861	-0.493	-0.36	-1.07447	-0.579	0.97	-0.223	-0.95	-0.781	-1.774	0.586816	-0.3238	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	S	-1.8	-0.756	0.017	-0.66	-0.02916	-0.951	-0.8	-0.634	-1.187	-1.047	-1.14	-0.14613	-0.9549	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	T	-1.8	-2.623	-1.278	-0.56	-0.52205	-0.92	0.17	-0.562	-1.538	-0.844	-0.573	0.230315	-0.1369	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	V	0.8	-0.871	-0.403	0.33	-1.1425	-0.369	1.99	0.016	-0.139	-0.266	0.618	0.62513	0.6407	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	W	0.4	3.235	2.715	-0.17	-2.68044	-1.137	0.25	-1.002	-1.189	-0.486	-2.044	0.344373	0.2636	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	210	Y	1.1	0.416	0.743	-0.25	-1.85424	-0.749	1.44	-0.589	-2.082	-0.807	-1.201	0.480749	0.1272	MID	EXP	1QD6	OMPLA	BB	1.99	0.183404
A	223	L	1.8	1.254	1.579	0.15	-1.62926	-0.359	1.69	0.039	0.593	-0.304	-1.283	-0.00474	0.4269	MID	EXP	1QD6	OMPLA	BB	0.07	0.109011
A	223	R	-2.1	-0.722	-0.309	-0.14	-1.05495	-0.399	0.52	-0.047	-0.95	-0.781	-3.257	0.240482	-0.3984	MID	EXP	1QD6	OMPLA	BB	0.07	0.109011
F	55	A	-1	-0.915	-1.846	-0.98	-1.63834	-2.054	-2.21	-2.252	-2.021	-1.69	-4.618	-0.43958	-2.6556	MID	EXP	3GP6	PAGP	BB	3.3	0.431507
A	85	G	-0.9	-1.842	-1.457	-1.08	-0.47481	-0.651	-2.07	-0.731	0.452	-1.223	1.847	-0.46868	-1.2555	MID	EXP	3GP6	PAGP	BB	3.44	-0.04167
L	105	A	-0.8	-1.324	-1.106	-0.84	-0.73295	-0.972	-1.69	-0.925	-2.447	-1.477	-3.058	-0.43958	-1.8131	MID	EXP	3GP6	PAGP	BB	2.04	-0.20961
M	157	A	-0.8	-2.312	-1.654	-0.94	-2.11672	-2.298	-0.93	-2.218	0.489	-1.413	-2.25	-0.07447	-1.5348	MID	EXP	3GP6	PAGP	BB	2.09	-0.08888
M	72	A	-0.3	-3.854	-3.825	-0.95	-3.56948	-1.967	-2.75	-2.092	-1.292	-1.471	-5.733	0.210493	-2.3319	SOL	BUR	3GP6	PAGP	BB	5.08	0.554691
T	108	A	-0.4	-0.218	-2.136	-0.83	-2.02631	-1.074	0.77	-1.008	-0.359	-0.453	-4.41	-0.2457	-1.3591	SOL	BUR	3GP6	PAGP	BB	3.98	0.220512

wt	id	mt	Exp ddG	Rosetta high	Rosetta low	Imutant 3	foldx	mcsm	sdm	duet	ddgm8	ddgm47	provean	elaspic	easemm	locat	expo state	pdb	prot	type	BLAST	SHAN
S	130	A	-2.1	-0.536	-0.842	0	-0.27711	-0.27	1.85	-0.283	-0.37	-0.627	-2.856	-0.46868	-0.4184	SOL	BUR	3GP6	PAGP	BB	4.06	0.057511
T	137	A	-0.5	-0.758	0.729	-0.74	-0.08341	-1.048	0.77	-0.85	-0.571	-0.284	-3.146	-0.43958	-1	SOL	BUR	3GP6	PAGP	BB	1.17	-0.04511
Q	139	A	-1	-2.577	-3.478	-0.38	-3.94389	-1.719	-0.85	-1.899	0.492	-1.004	-5.779	-0.04888	-1.2518	SOL	BUR	3GP6	PAGP	BB	2.67	0.672341
Q	160	A	0.2	-1.442	0.187	-0.61	-0.65472	-0.314	-0.67	-0.337	-0.472	-0.582	-3.954	0.639614	-0.6037	SOL	BUR	3GP6	PAGP	BB	3.52	0.315637
S	35	A	-0.3	-0.362	-0.23	-0.17	-0.08517	-0.183	0.16	0.051	-0.607	-0.421	-0.032	-0.09293	-0.4294	SOL	EXP	1PY6	BR	AH	1.23	0.19119
D	36	A	-0.9	0.605	-1.527	-0.71	-3.48369	-0.548	1.45	-0.589	-1.803	-1.612	-2.288	-0.75981	-0.2095	SOL	EXP	1PY6	BR	AH	1.44	0.048988
P	37	A	-0.2	0.018	-1.082	-0.9	-0.832	-0.786	2.21	-0.359	-1.222	-0.872	-1.83	-0.27013	0.0885	SOL	EXP	1PY6	BR	AH	4.63	0.476736
D	38	A	-0.5	1.923	0.138	-0.34	0.926906	-0.51	2.55	-0.233	-1.417	-0.795	-1.219	0.464045	-0.1009	SOL	EXP	1PY6	BR	AH	-0.66	-0.01885
P	40	A	-1	-0.743	-2.669	-1.57	-1.19132	-1.033	0.22	-1.015	-1.253	-1.867	-7.716	-0.418	-1.7124	SOL	EXP	2K73	DSBB	AH	8.75	2.85538
G	212	A	0.6	1.339	1.4	-1.35	0.109814	-0.755	1.65	-0.505	-1.352	-1.792	-1.523	-0.42583	-0.2111	SOL	EXP	1QD6	OMPLA	BB	0.67	0.009111
G	212	L	2.6	1.431	2.674	-0.3	-0.89847	-0.275	2.17	0.091	0.354	-1.353	-2.792	0.202924	0.2779	SOL	EXP	1QD6	OMPLA	BB	0.67	0.009111
G	212	R	-3.1	-0.15	0.633	-0.68	-0.94531	-1.078	0.78	-0.75	-1.038	-1.684	-3.1	0.054348	0.3998	SOL	EXP	1QD6	OMPLA	BB	0.67	0.009111

APPENDIX F

RAW TABLES FROM CHAPTER 4 REGRESSION ANALYSES

Table AF.1. Weights for all terms in the RosettaMembrane energy function obtained from Ridge regression and cross validation by protein backbone

Pearson	Spearman	AUC	fa_atr	fa_rep	fa_intra_rep	fa_mbenv	fa_mbsolv	pro_close	fa_pair	hbond_sr_bb	hbond_lr_bb	hbond_bb_sc	hbond_sc	dslf_ss_dst	dslf_cs_ang	dslf_ss_dih	dslf_ca_dih	rama	omega	fa_dun	p_aa_pp	ref	Menv_smooth
0.13	0.03	0.56	0.17	0.08	-0.04	0.14	0.19	-0.66	0.56	0.04	0.35	0.45	0.48	82	-613	-145	73	0.00	0.03	0.12	-0.12	0.07	0.09
0.15	0.32	0.65	0.13	0.07	-0.06	0.14	0.19	-2.05	0.75	0.11	0.44	0.39	0.46	-9	-658	-105	164	0.02	0.08	0.11	-0.12	0.03	0.01
0.34	0.35	0.74	0.14	0.09	-0.02	0.06	0.09	-0.42	0.19	0.06	-0.04	0.40	0.24	41	-726	10	62	-0.01	0.02	0.03	-0.06	0.11	-0.06
-0.06	-0.06	0.52	0.17	0.07	-0.03	0.15	0.20	-0.56	0.41	-0.07	0.10	0.44	0.54	116	-612	-133	45	0.00	0.05	0.14	-0.11	0.32	0.05
0.50	0.30	0.60	0.17	0.07	-0.03	0.15	0.20	-0.56	0.41	-0.07	0.10	0.44	0.54	116	-612	-133	45	0.00	0.05	0.14	-0.11	0.32	0.05
0.43	0.28	0.60	0.19	0.07	-0.03	0.15	0.20	-0.55	0.52	-0.06	0.50	0.38	0.39	122	-609	-155	-19	-0.01	0.15	0.09	-0.07	0.05	0.07
-0.08	-0.21	0.64	0.15	0.09	-0.04	0.18	0.14	-0.57	0.47	0.06	0.34	0.45	0.49	100	-622	-157	251	0.00	0.05	0.09	-0.10	0.10	-0.03

Table AF.2. Weights for the selected terms in the RosettaMembrane energy function obtained after Ridge regression using cross validation by protein backbone. First score terms that were contributing noise to the calculation were identified and removed. Ridge regression was performed again using the limited weight set.

Pearson	Spearman	AUC	fa_atr	fa_rep	fa_mbenv	fa_mbsolv	fa_pair	hbond_bb_sc	hbond_sc	omega	fa_dun
0.21	0.12	0.52	0.18	0.06	0.15	0.20	0.53	0.42	0.43	0.04	0.08
0.36	0.41	0.73	0.16	0.07	0.15	0.21	0.81	0.31	0.38	0.08	0.07

0.31	0.31	0.74	0.14	0.08	0.05	0.07	0.21	0.36	0.24	0.02	0.00
-0.05	-0.04	0.59	0.16	0.06	0.14	0.19	0.41	0.43	0.51	0.05	0.09
0.23	0.21	0.54	0.17	0.06	0.14	0.19	0.57	0.40	0.42	0.08	0.08
0.33	0.25	0.58	0.19	0.03	0.16	0.21	0.51	0.36	0.32	0.18	0.06
-0.07	-0.21	0.64	0.15	0.07	0.18	0.12	0.49	0.41	0.43	0.07	0.06

Table AF.3. Weights for the selected terms in the RosettaMembrane energy function obtained after Elastic Net regression using cross validation by protein backbone. First score terms that were contributing noise to the calculation were identified and removed. Elastic Net regression was performed again using the limited weight set.

Missing	Alpha	Pearson	Spearman	AUC	fa_atr	fa_rep	fa_mbenv	fa_mbsolv	fa_pair	hbond_bb_sc	hbond_sc	omega	fa_dun
none	0	0.45	0.45	0.72	0.12	0.04	0.08	0.09	0.45	0.36	0.28	0.09	0.07
none	0.1	0.48	0.48	0.73	0.14	0.05	0.11	0.13	0.48	0.38	0.32	0.09	0.07
none	0.2	0.47	0.47	0.73	0.13	0.05	0.10	0.12	0.47	0.36	0.30	0.09	0.07
none	0.3	0.48	0.48	0.73	0.14	0.05	0.11	0.14	0.48	0.37	0.32	0.09	0.06
none	0.4	0.49	0.49	0.73	0.15	0.05	0.12	0.16	0.50	0.38	0.34	0.09	0.06
none	0.5	0.48	0.48	0.73	0.14	0.05	0.11	0.13	0.48	0.35	0.30	0.08	0.06
none	0.6	0.48	0.48	0.73	0.14	0.05	0.11	0.14	0.48	0.34	0.30	0.08	0.06
none	0.7	0.47	0.47	0.73	0.13	0.05	0.10	0.12	0.46	0.32	0.28	0.08	0.06
none	0.8	0.47	0.47	0.73	0.13	0.05	0.10	0.12	0.46	0.31	0.28	0.07	0.06
none	0.9	0.49	0.49	0.73	0.15	0.05	0.12	0.16	0.50	0.36	0.33	0.08	0.06
none	1	0.49	0.49	0.73	0.15	0.05	0.12	0.15	0.49	0.34	0.31	0.08	0.06
1AFO	0	0.51	0.17	0.54	0.11	0.04	0.04	0.04	0.37	0.29	0.21	0.08	0.06
1AFO	0.1	0.38	0.18	0.53	0.14	0.05	0.07	0.09	0.42	0.32	0.28	0.06	0.07
1AFO	0.2	0.38	0.18	0.53	0.14	0.05	0.07	0.09	0.42	0.31	0.27	0.06	0.07
1AFO	0.3	0.36	0.19	0.53	0.14	0.05	0.08	0.10	0.42	0.32	0.28	0.06	0.07
1AFO	0.4	0.39	0.18	0.53	0.13	0.05	0.07	0.08	0.40	0.29	0.24	0.06	0.07
1AFO	0.5	0.37	0.19	0.53	0.14	0.05	0.07	0.09	0.41	0.29	0.25	0.05	0.07

Missing	Alpha	Pearson	Spearman	AUC	fa_atr	fa_rep	fa_mbenv	fa_mbsolv	fa_pair	hbond_bb_sc	hbond_sc	omega	fa_dun
1AFO	0.6	0.38	0.18	0.53	0.13	0.05	0.07	0.08	0.40	0.27	0.23	0.05	0.06
1AFO	0.7	0.32	0.19	0.53	0.15	0.05	0.09	0.11	0.43	0.31	0.29	0.05	0.07
1AFO	0.8	0.35	0.19	0.53	0.14	0.05	0.08	0.10	0.41	0.29	0.26	0.05	0.07
1AFO	0.9	0.33	0.19	0.53	0.14	0.05	0.08	0.10	0.42	0.29	0.27	0.05	0.07
1AFO	1	0.34	0.19	0.53	0.14	0.05	0.08	0.10	0.41	0.28	0.26	0.04	0.06
1PY6	0	0.26	0.34	0.71	0.08	0.04	0.02	0.03	0.60	0.24	0.13	0.10	0.05
1PY6	0.1	0.32	0.40	0.73	0.11	0.06	0.07	0.09	0.69	0.25	0.21	0.08	0.06
1PY6	0.2	0.32	0.40	0.73	0.11	0.06	0.07	0.09	0.69	0.24	0.20	0.08	0.06
1PY6	0.3	0.33	0.40	0.73	0.12	0.06	0.08	0.11	0.71	0.24	0.23	0.08	0.06
1PY6	0.4	0.31	0.39	0.73	0.11	0.06	0.06	0.08	0.68	0.21	0.18	0.07	0.06
1PY6	0.5	0.33	0.40	0.73	0.12	0.06	0.08	0.12	0.72	0.23	0.23	0.07	0.06
1PY6	0.6	0.32	0.40	0.73	0.11	0.06	0.07	0.10	0.70	0.21	0.20	0.07	0.06
1PY6	0.7	0.32	0.39	0.73	0.11	0.06	0.07	0.09	0.69	0.20	0.18	0.07	0.05
1PY6	0.8	0.33	0.40	0.73	0.12	0.06	0.08	0.11	0.71	0.21	0.20	0.07	0.06
1PY6	0.9	0.33	0.40	0.73	0.12	0.06	0.08	0.10	0.71	0.20	0.20	0.07	0.05
1PY6	1	0.33	0.40	0.73	0.12	0.06	0.07	0.10	0.71	0.19	0.19	0.06	0.05
1QD5	0	0.00	-0.01	0.54	0.08	0.04	0.00	-0.01	0.16	0.28	0.14	0.09	-0.01
1QD5	0.1	0.02	0.01	0.55	0.09	0.04	0.00	0.00	0.14	0.26	0.12	0.08	0.00
1QD5	0.2	0.02	0.02	0.55	0.09	0.05	0.00	0.00	0.12	0.23	0.10	0.07	0.00
1QD5	0.3	0.02	0.02	0.55	0.10	0.05	0.00	0.00	0.10	0.22	0.09	0.06	0.00
1QD5	0.4	0.02	0.02	0.56	0.09	0.05	0.00	0.00	0.08	0.18	0.07	0.05	0.00
1QD5	0.5	0.03	0.02	0.56	0.10	0.05	0.00	0.00	0.07	0.17	0.07	0.04	0.00
1QD5	0.6	0.03	0.03	0.56	0.09	0.05	0.00	0.00	0.03	0.12	0.03	0.02	0.00
1QD5	0.7	0.03	0.03	0.56	0.10	0.05	0.00	0.00	0.03	0.13	0.04	0.01	0.00
1QD5	0.8	0.03	0.02	0.56	0.11	0.06	0.00	0.00	0.04	0.14	0.05	0.01	0.00
1QD5	0.9	0.03	0.02	0.56	0.11	0.06	0.00	0.00	0.02	0.11	0.03	0.00	0.00
1QD5	1	0.03	0.03	0.56	0.10	0.06	0.00	0.00	0.00	0.07	0.01	0.00	0.00

Missing	Alpha	Pearson	Spearman	AUC	fa_atr	fa_rep	fa_mbenv	fa_mbsolv	fa_pair	hbond_bb_sc	hbond_sc	omega	fa_dun
1QJP	0	0.03	0.08	0.69	0.10	0.04	0.04	0.04	0.22	0.29	0.28	0.09	0.08
1QJP	0.1	0.01	0.04	0.67	0.11	0.04	0.05	0.06	0.24	0.31	0.32	0.07	0.08
1QJP	0.2	0.00	0.03	0.65	0.12	0.05	0.06	0.08	0.25	0.31	0.34	0.07	0.08
1QJP	0.3	-0.01	0.02	0.64	0.13	0.05	0.07	0.09	0.27	0.33	0.36	0.06	0.09
1QJP	0.4	-0.01	0.02	0.63	0.13	0.05	0.07	0.09	0.26	0.32	0.35	0.06	0.08
1QJP	0.5	-0.02	0.01	0.63	0.13	0.05	0.08	0.10	0.28	0.33	0.36	0.06	0.08
1QJP	0.6	-0.02	0.00	0.62	0.13	0.05	0.08	0.11	0.29	0.33	0.38	0.06	0.08
1QJP	0.7	-0.02	0.00	0.62	0.13	0.05	0.08	0.11	0.29	0.33	0.38	0.06	0.08
1QJP	0.8	0.00	0.03	0.64	0.12	0.05	0.06	0.08	0.24	0.29	0.32	0.06	0.08
1QJP	0.9	-0.01	0.02	0.63	0.13	0.05	0.07	0.09	0.26	0.30	0.34	0.06	0.08
1QJP	1	0.00	0.03	0.64	0.12	0.05	0.06	0.08	0.24	0.28	0.32	0.05	0.08
2K73	0	0.00	-0.05	0.55	0.10	0.04	0.03	0.03	0.39	0.27	0.19	0.11	0.05
2K73	0.1	0.11	0.01	0.53	0.12	0.05	0.05	0.06	0.43	0.29	0.23	0.09	0.06
2K73	0.2	0.14	0.04	0.52	0.13	0.05	0.06	0.08	0.45	0.30	0.25	0.09	0.07
2K73	0.3	0.13	0.03	0.52	0.12	0.05	0.05	0.06	0.42	0.27	0.22	0.09	0.06
2K73	0.4	0.13	0.02	0.52	0.12	0.05	0.05	0.06	0.42	0.26	0.21	0.08	0.06
2K73	0.5	0.19	0.08	0.51	0.14	0.05	0.08	0.11	0.47	0.31	0.29	0.08	0.07
2K73	0.6	0.18	0.07	0.50	0.13	0.05	0.07	0.09	0.45	0.29	0.25	0.08	0.06
2K73	0.7	0.18	0.06	0.50	0.13	0.05	0.07	0.09	0.45	0.28	0.25	0.08	0.06
2K73	0.8	0.19	0.08	0.50	0.14	0.05	0.07	0.09	0.46	0.29	0.26	0.08	0.06
2K73	0.9	0.19	0.07	0.50	0.14	0.05	0.07	0.09	0.45	0.28	0.26	0.08	0.06
2K73	1	0.19	0.08	0.51	0.13	0.05	0.07	0.09	0.45	0.27	0.25	0.07	0.06
2XOV	0	0.22	0.12	0.52	0.11	0.01	0.03	0.02	0.35	0.22	0.11	0.16	0.03
2XOV	0.1	0.26	0.17	0.54	0.15	0.02	0.07	0.09	0.40	0.28	0.17	0.18	0.06
2XOV	0.2	0.23	0.14	0.53	0.14	0.02	0.06	0.07	0.38	0.25	0.13	0.18	0.05
2XOV	0.3	0.24	0.15	0.54	0.15	0.02	0.07	0.08	0.39	0.25	0.15	0.18	0.05
2XOV	0.4	0.24	0.16	0.54	0.15	0.02	0.08	0.09	0.40	0.25	0.16	0.18	0.05

Missing	Alpha	Pearson	Spearman	AUC	fa_atr	fa_rep	fa_mbenv	fa_mbsolv	fa_pair	hbond_bb_sc	hbond_sc	omega	fa_dun
2XOV	0.5	0.23	0.15	0.53	0.15	0.02	0.07	0.09	0.40	0.24	0.15	0.18	0.05
2XOV	0.6	0.25	0.18	0.55	0.16	0.02	0.09	0.12	0.42	0.27	0.19	0.18	0.05
2XOV	0.7	0.23	0.16	0.54	0.15	0.02	0.08	0.10	0.40	0.24	0.16	0.18	0.05
2XOV	0.8	0.20	0.13	0.52	0.15	0.02	0.07	0.08	0.38	0.21	0.13	0.18	0.04
2XOV	0.9	0.23	0.16	0.54	0.16	0.02	0.09	0.11	0.41	0.24	0.17	0.18	0.04
2XOV	1	0.24	0.16	0.54	0.16	0.02	0.09	0.12	0.41	0.24	0.17	0.18	0.04
3GP6	0	-0.04	-0.15	0.63	0.09	0.05	0.12	0.00	0.38	0.33	0.27	0.11	0.03
3GP6	0.1	-0.04	-0.17	0.63	0.09	0.05	0.12	0.00	0.37	0.31	0.26	0.10	0.03
3GP6	0.2	-0.04	-0.18	0.63	0.09	0.05	0.12	0.00	0.36	0.27	0.22	0.09	0.02
3GP6	0.3	-0.04	-0.19	0.63	0.09	0.05	0.11	0.00	0.35	0.25	0.21	0.09	0.02
3GP6	0.4	-0.04	-0.19	0.64	0.10	0.06	0.14	0.02	0.38	0.30	0.27	0.08	0.03
3GP6	0.5	-0.04	-0.20	0.64	0.09	0.06	0.12	0.00	0.36	0.25	0.22	0.08	0.02
3GP6	0.6	-0.04	-0.20	0.64	0.10	0.06	0.13	0.00	0.36	0.27	0.23	0.07	0.02
3GP6	0.7	-0.04	-0.21	0.64	0.11	0.07	0.14	0.02	0.39	0.30	0.28	0.07	0.03
3GP6	0.8	-0.04	-0.21	0.63	0.09	0.06	0.12	0.00	0.35	0.22	0.20	0.07	0.01
3GP6	0.9	-0.04	-0.21	0.64	0.10	0.07	0.13	0.00	0.36	0.26	0.23	0.06	0.02
3GP6	1	-0.04	-0.21	0.63	0.10	0.07	0.13	0.00	0.36	0.23	0.21	0.06	0.01

Table AF.4. Weights for all terms in the RosettaMembrane energy function obtained from Elastic net regression and leave one out cross validation.

α	Pearson	Spearman	AUC	fa_atr	fa_rep	fa_intra_rep	fa_mbenv	fa_mbsolv	pro_close	fa_pair	hbond_sr_bb	hbond_lr_bb	hbond_bb_sc	hbond_sc	dslf_ss_dst	dslf_cs_ang	dslf_ss_dih	dslf_ca_dih	rama	omega	fa_dun	p_aa_pp	ref	Menv_smooth
0	0.50	0.48	0.73	0.08	0.04	-0.04	0.03	0.03	-0.50	0.46	0.06	-0.20	0.35	0.315	82	-469	-92	-87	0.03	0.11	0.08	-0.15	-0.27	0.17
0.1	0.52	0.50	0.74	0.11	0.06	-0.04	0.07	0.08	-0.61	0.51	0.02	-0.15	0.39	0.367	72	-488	-110	-29	0.02	0.10	0.10	-0.15	-0.18	0.16
0.2	0.52	0.50	0.74	0.11	0.06	-0.04	0.06	0.08	-0.57	0.50	0.00	-0.12	0.37	0.357	48	-441	-94	-30	0.01	0.10	0.09	-0.14	-0.18	0.16
0.3	0.52	0.50	0.74	0.11	0.06	-0.04	0.07	0.08	-0.56	0.50	0.00	-0.10	0.37	0.355	33	-413	-83	-33	0.01	0.09	0.09	-0.13	-0.18	0.15
0.4	0.52	0.50	0.74	0.11	0.06	-0.04	0.07	0.09	-0.56	0.51	0.00	-0.08	0.37	0.360	27	-408	-81	-28	0.01	0.09	0.09	-0.12	-0.16	0.14
0.5	0.52	0.50	0.74	0.11	0.06	-0.03	0.07	0.09	-0.54	0.50	0.00	-0.06	0.36	0.354	12	-378	-69	-36	0.00	0.09	0.09	-0.11	-0.16	0.14
0.6	0.52	0.50	0.74	0.11	0.06	-0.03	0.07	0.09	-0.54	0.51	0.00	-0.04	0.36	0.353	4	-364	-64	-38	0.00	0.09	0.08	-0.10	-0.15	0.14
0.7	0.52	0.50	0.74	0.11	0.06	-0.03	0.07	0.09	-0.53	0.50	0.00	-0.03	0.35	0.350	0	-346	-56	-32	0.00	0.08	0.08	-0.10	-0.15	0.14
0.8	0.52	0.50	0.74	0.12	0.06	-0.03	0.08	0.10	-0.53	0.51	0.00	-0.02	0.36	0.355	0	-349	-57	-21	0.00	0.08	0.08	-0.10	-0.14	0.13
0.9	0.52	0.50	0.74	0.12	0.06	-0.03	0.08	0.09	-0.52	0.51	0.00	-0.01	0.35	0.349	0	-328	-48	-16	0.00	0.08	0.08	-0.10	-0.14	0.14
1	0.52	0.50	0.74	0.12	0.06	-0.03	0.08	0.10	-0.52	0.51	0.00	0.00	0.35	0.353	0	-331	-49	-7	0.00	0.08	0.08	-0.10	-0.13	0.13

Table AF.5. Weights for the selected terms in the RosettaMembrane energy function obtained after Elastic Net regression using leave one out cross validation. First score terms that were contributing noise to the calculation were identified and removed. Elastic Net regression was performed again using the limited weight set.

Alpha	Pearson	Spearman	AUC	fa_atr	fa_rep	fa_mbenv	fa_mbsolv	fa_pair	hbond_bb_sc	hbond_sc	omega	fa_dun
0	0.44	0.44	0.72	0.12	0.04	0.07	0.09	0.45	0.36	0.28	0.10	0.07
0.1	0.48	0.48	0.73	0.14	0.05	0.11	0.13	0.48	0.38	0.32	0.09	0.07
0.2	0.48	0.47	0.73	0.14	0.05	0.11	0.13	0.48	0.37	0.31	0.09	0.07
0.3	0.48	0.48	0.73	0.14	0.05	0.11	0.14	0.48	0.37	0.32	0.09	0.06
0.4	0.48	0.48	0.73	0.14	0.05	0.11	0.13	0.48	0.35	0.31	0.08	0.06
0.5	0.48	0.48	0.73	0.14	0.05	0.11	0.13	0.48	0.35	0.30	0.08	0.06
0.6	0.48	0.48	0.73	0.14	0.05	0.11	0.14	0.48	0.34	0.30	0.08	0.06
0.7	0.48	0.48	0.73	0.14	0.05	0.11	0.14	0.48	0.34	0.30	0.08	0.06
0.8	0.48	0.48	0.73	0.14	0.05	0.11	0.14	0.48	0.33	0.30	0.08	0.06
0.9	0.48	0.48	0.73	0.14	0.05	0.11	0.14	0.48	0.33	0.30	0.07	0.06
1	0.48	0.48	0.73	0.14	0.05	0.11	0.14	0.47	0.32	0.29	0.07	0.05

Table AF.6. Weights for all terms in the RosettaMembrane energy function obtained from Ridge regression and 5-fold cross-validation. Training was done on 80% of the data and testing on 20% of the data. Replicates were in the same group. This shows the metrics obtained from consistently sized groups; however, this also means that in some cases, backbones were seen in both the training and test sets.

Pearson	Spearman	AUC	fa_atr	fa_rep	fa_intra_rep	fa_mbenv	fa_mbsolv	pro_close	fa_pair	hbond_sr_bb	hbond_lr_bb	hbond_bb_sc	hbond_sc	dslf_ss_dst	dslf_cs_ang	dslf_ss_dih	dslf_ca_dih	rama	omega	fa_dun	p_aa_pp	ref	Menv_smooth
0.475	0.409	0.70	0.16	0.07	-0.05	0.15	0.20	-1.35	0.59	0.29	0.27	0.48	0.50	60	-91	-65	372	0.03	0.11	0.09	-0.17	0.09	0.11
0.457	0.455	0.64	0.17	0.08	-0.05	0.18	0.25	-0.66	0.69	-0.03	-0.14	0.51	0.48	191	-666	-210	152	-0.03	0.11	0.11	-0.17	0.01	0.04
0.402	0.407	0.80	0.17	0.08	-0.04	0.15	0.21	-0.71	0.63	-0.07	-0.26	0.60	0.50	89	-1306	-217	352	-0.01	0.17	0.11	-0.07	0.04	0.08
0.541	0.533	0.78	0.16	0.07	-0.05	0.15	0.20	-0.63	0.57	-0.01	0.28	0.51	0.42	51	-649	-166	-126	0.01	0.09	0.12	-0.16	0.12	0.05
0.547	0.513	0.68	0.17	0.07	-0.04	0.15	0.21	-0.64	0.64	0.03	-0.07	0.34	0.43	106	-484	-112	-220	0.04	0.06	0.12	-0.19	0.11	0.08

APPENDIX G

PROTOCOL CAPTURE FOR CHAPTER 5

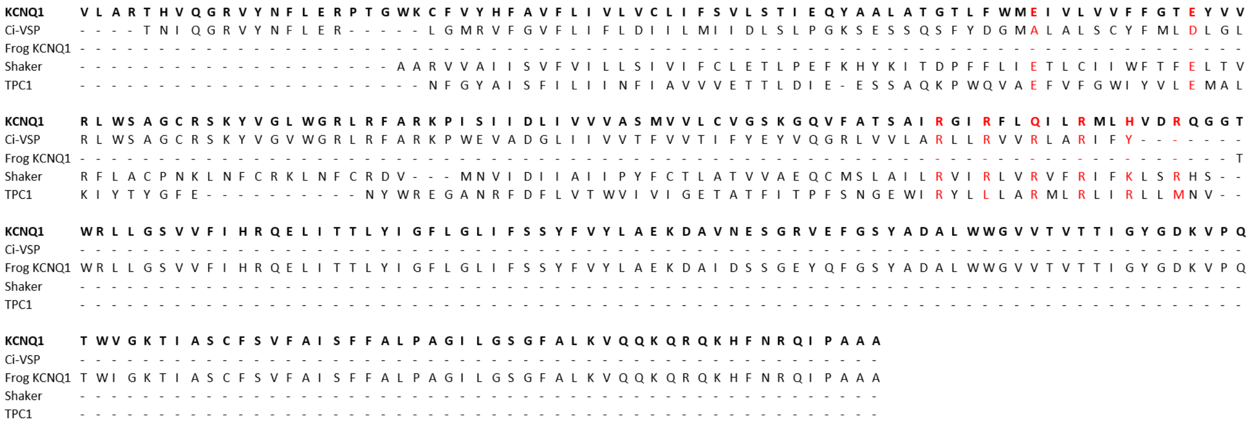
Mustang structural alignment:

When running mustang, the first pdb listed will be the coordinate frame to what all of the other template pdbs will be aligned. In the final protocol, I used the Frog KCNQ1 structure as the initial template.

```
mustang -i frog-kcnq1.pdb other.pdbs -F fasta
```

Sequence alignment:

The fasta multiple sequence alignment output from mustang was used as the input for the server Clustal Omega. The output alignment was then manually adjusted so that gaps in secondary structured regions were removed, starts and ends of helices were of relatively similar length, conserved residues were aligned and on the same face of the helix. The resulting alignment can be found in Figure AG.1



```
KCNQ1       VLARTHVQGRVYNFLERPTGWKCFVYHFAVFLIVLVCLIFSVLSTIEQYAALATGTLFWMEIVLVVFFGTEYVV
Ci-VSP      ----TNIQGRVYNFLER-----LGMRVFGVFLIFLDIILMIIDLSLPGKSESSQSFYDGMALALSCYFMLDLGL
Frog KCNQ1  --------------------------------------------------------------------------
Shaker      --------------------AARVVAIISVFVILLSIVIFCLETLPEFKHYKITDPFFLIETLCIIWFTFELTV
TPC1        ----------------------NFGYAISFILIINFIAVVVETTLDIE-ESSAQKPWQVAEFVFGWIYVLEMAL

KCNQ1       RLWSAGCRSKYVGLWGRLRFARKPISIIDLIVVVASMVVLCVGSKGQVFATSAIRGIRFLQILRMLHVDRQGGT
Ci-VSP      RLWSAGCRSKYVGVWGRLRFARKPWEVADGLIIVVTFVVTIFYEYVQGRLVVLARLLRVVRLARIFY-------
Frog KCNQ1  -------------------------------------------------------------------------T
Shaker      RFLACPNKLNFCRKLNFCRDV---MNVIDIIAIIPYFCTLATVVAEQCMSLAILRVIRLVRVFRIFKLSRHS--
TPC1        KIYTYGFE----------NYWREGANRFDFLVTWVIVIGETATFITPFSNGEWIRYLLLARMLRLIRLLMNV--

KCNQ1       WRLLGSVVFIHRQELITTLYIGFLGLIFSSYFVYLAEKDAVNESGRVEFGSYADALWWGVVTVTTIGYGDKVPQ
Ci-VSP      --------------------------------------------------------------------------
Frog KCNQ1  WRLLGSVVFIHRQELITTLYIGFLGLIFSSYFVYLAEKDAIDSSGEYQFGSYADALWWGVVTVTTIGYGDKVPQ
Shaker      --------------------------------------------------------------------------
TPC1        --------------------------------------------------------------------------

KCNQ1       TWVGKTIASCFSVFAISFFALPAGILGSGFALKVQQKQRQKHFNRQIPAAA
Ci-VSP      ---------------------------------------------------
Frog KCNQ1  TWIGKTIASCFSVFAISFFALPAGILGSGFALKVQQKQRQKHFNRQIPAAA
Shaker      ---------------------------------------------------
TPC1        ---------------------------------------------------
```

Figure AG.1 Manually adjusted structure-based sequence alignment of templates and target (bold). Residues that are key for distinguishing VSD states are highlighted in red. The only pore domain template input for the final model generation was from Frog KCNQ1.

Partial threading:

For each alignment and template, the follow command was run to create a threaded template

```
/dors/meilerlab/apps/rosetta/rosetta-3.6/main/source/bin/
partial_thread.default.linuxgccrelease -in:file:fasta target_sequence.fasta
-in:file:alignment my_template.aln -in:file:template_pdb my_template.pdb
-ignore_zero_occupancy false
```

Each alignment file was in the Grishin format

Hybridization:

```
/dors/meilerlab/apps/rosetta/rosetta-3.6/main/source/bin/
rosetta_scripts.default.linuxgccrelease @my.options -parser:protocol
rosetta_cm.xml
```

The hybridization mover was employed with the following options (my.options):

```
# i/o
-in:file:fasta /dors/sanderslab/data/kchannels/duranam/kcnq1-closed/add-
frog/target_sequence.fasta
#-parser:protocol pore.xml
-out:pdb
#-out:file:silent_struct_type binary
#-in:detect_disulf true

# relax options
-relax:minimize_bond_angles
-relax:minimize_bond_lengths
-relax:jump_move true
-default_max_cycles 200
-relax:min_type lbfgs_armijo_nonmonotone
-relax:constrain_relax_to_start_coords true
-score:weights stage3_rlx_membrane.wts
-use_bicubic_interpolation
-hybridize:stage1_probability 1.0
-sog_upper_bound 15

# membrane options
-membrane
-in:file:spanfile kcnq1.span
#-in:file:lipofile ../t.lips4
-membrane:no_interpolate_Mpair
-membrane:Menv_penalties
-rg_reweight .1

# reduce memory footprint
-chemical:exclude_patches LowerDNA  UpperDNA Cterm_amidation SpecialRotamer
VirtualBB ShoveBB VirtualDNAPhosphate VirtualNTerm CTermConnect sc_orbitals
pro_hydroxylated_case1 pro_hydroxylated_case2 ser_phosphorylated
thr_phosphorylated  tyr_phosphorylated tyr_sulfated lys_dimethylated
lys_monomethylated  lys_trimethylated lys_acetylated glu_carboxylated
cys_acetylated tyr_diiodinated N_acetylated C_methylamidated
MethylatedProteinCterm
```

The XML file for hybridization (`rosetta_cm.xml`):

```
<ROSETTASCRIPTS>
    <TASKOPERATIONS>
    </TASKOPERATIONS>
    <SCOREFXNS>
        <stage1 weights=stage1_membrane symmetric=1>
            <Reweight scoretype=atom_pair_constraint weight=1/>
        </stage1>
        <stage2 weights=stage2_membrane symmetric=1>
            <Reweight scoretype=atom_pair_constraint weight=0.5/>
        </stage2>
        <fullatom weights=stage3_rlx_membrane symmetric=1>
            <Reweight scoretype=atom_pair_constraint weight=0.5/>
        </fullatom>
    </SCOREFXNS>
    <MOVERS>
       <Hybridize name=hybridize stage1_scorefxn=stage1 stage2_scorefxn=stage2 fa_scorefxn=fullatom
batch=1 stage1_increase_cycles=1.0 stage2_increase_cycles=1.0 linmin_only=1 realign_domains=0>
           <DetailedControls start_res="7" stop_res="15" sample_template="1" sample_abinitio="0"/>
           <DetailedControls start_res="88" stop_res="96" sample_template="1" sample_abinitio="0"/>
            <Fragments 3mers="kcnq1_frags.200.3mers" 9mers="kcnq1_frags.200.9mers"/>
            <Template pdb="Ci-VSP-frogKCNQ1-hybrid.pdb.pdb" cst_file="AUTO" weight=   1.000
symmdef="/dors/meilerlab/apps/rosetta/rosetta-3.6/main/database/symmetry/cyclic/C4_Z.sym"/>
            <Template pdb="TPC1-IIB-vs.pdb.pdb" cst_file="AUTO" weight=   0
symmdef="/dors/meilerlab/apps/rosetta/rosetta-3.6/main/database/symmetry/cyclic/C4_Z.sym"/>
            <Template pdb=”Shaker-c3-closed-vs.pdb.pdb" cst_file="AUTO" weight=   0
symmdef="/dors/meilerlab/apps/rosetta/rosetta-3.6/main/database/symmetry/cyclic/C4_Z.sym"/>
        </Hybridize>
    </MOVERS>
    <APPLY_TO_POSE>
    </APPLY_TO_POSE>
    <PROTOCOLS>
        <Add mover=hybridize/>
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

Then I relaxed using dualspace and constraint to start coordinates using this command:

```
/dors/meilerlab/apps/rosetta/rosetta-3.6/main/source/bin/
relax.linuxgccrelease @cds.options -nstruct 25 -s my.pdb
```

Options file (cds.options) contents:

```
-in:file:silent_struct_type binary
-out:file:silent_struct_type binary
-out:prefix cds_
-relax:dualspace
-relax:minimize_bond_angles
-relax:jump_move true
-relax:constrain_relax_to_start_coords true
-relax:ramp_constraints false
-set_weights cart_bonded .5 pro_close 0
-default_max_cycles 200
-flip_HNQ
-no_optH false
-symmetry
-symmetry_definition /dors/meilerlab/apps/rosetta/rosetta-3.6/main/database/
symmetry/cyclic/C4_Z.sym
-in:file:spanfile kcnq1.span
-membrane:no_interpolate_Mpair
-membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts
```

Then I relaxed a final time using dualspace and constrain to start coordinates with the ramping of repulsive terms turned off using this command:

```
/dors/meilerlab/apps/rosetta/rosetta-3.6/main/source/bin/
relax.linuxgccrelease @ds.options -nstruct 25 -s my.pdb
```

Option file (ds.options) contents:

```
-in:file:silent_struct_type binary
-out:file:silent_struct_type binary
-out:prefix ds_
-relax:dualspace
-relax:minimize_bond_angles
-relax:jump_move true
-relax:constrain_relax_to_start_coords true
-set_weights cart_bonded .5 pro_close 0
-default_max_cycles 200
-flip_HNQ
-no_optH false
-symmetry
-symmetry_definition /dors/meilerlab/apps/rosetta/rosetta-3.6/main/database/
symmetry/cyclic/C4_Z.sym
-in:file:spanfile kcnq1.span
-membrane:no_interpolate_Mpair
-membrane:Menv_penalties
```

Notes on other approaches:

Initially, the approach was taken to model the voltage sensor and pore domain separately and then tack them together using the S4-S5 linker from Smith. However, this consistently left frustrations in the helix and resulted in subunits being oriented at multiple interfaces. With no experimental restraints to specify which orientation was favored over others, I used the old KCNQ1 model from Smith et al. as my template for the orientation of domains. However, after the frog structure was realized, we noticed that the orientation of the subunits was like that of the orientation of the MLotiK1 protein structure – not a true voltage sensor, but a S1-S4 domain with the pore in the closed state. This was sufficient evidence to move forward with this approach.

APPENDIX H

SUPPLEMENTAL INFORMATION FOR CHAPTER 7

For Figure 7.13 in the main text, these are individual plots for each of the 20 symmetric backbones designed. Backbones were designed symmetrically and the number of mutations is plotted along with the normalized energy contribution (ΔREU/amino acid, here REU is Rosetta energy units) from these pairs of mutations. Please note that axis are not the same scale.

For 1FX8:

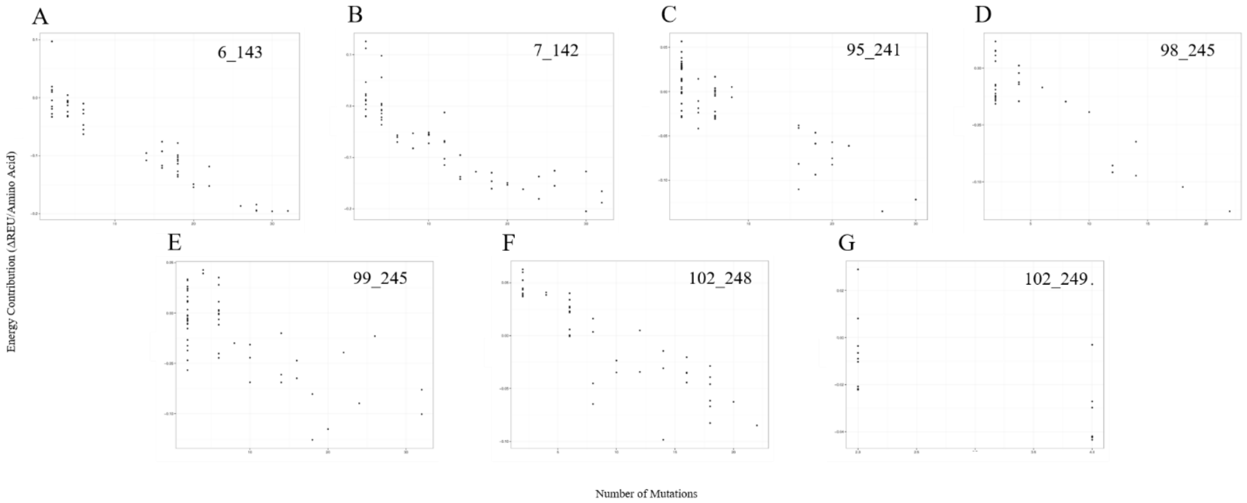


Figure AH.1 Individual plots for symmetric variants of 1FX8 of the number of mutations and their normalized energy contribution (ΔREU/amino acid). Note that panel G only has mutants for 2 and 4 mutations. The general trend tends to be that with each stepwise increase in mutations, the energy is lowered.

For 2D57:

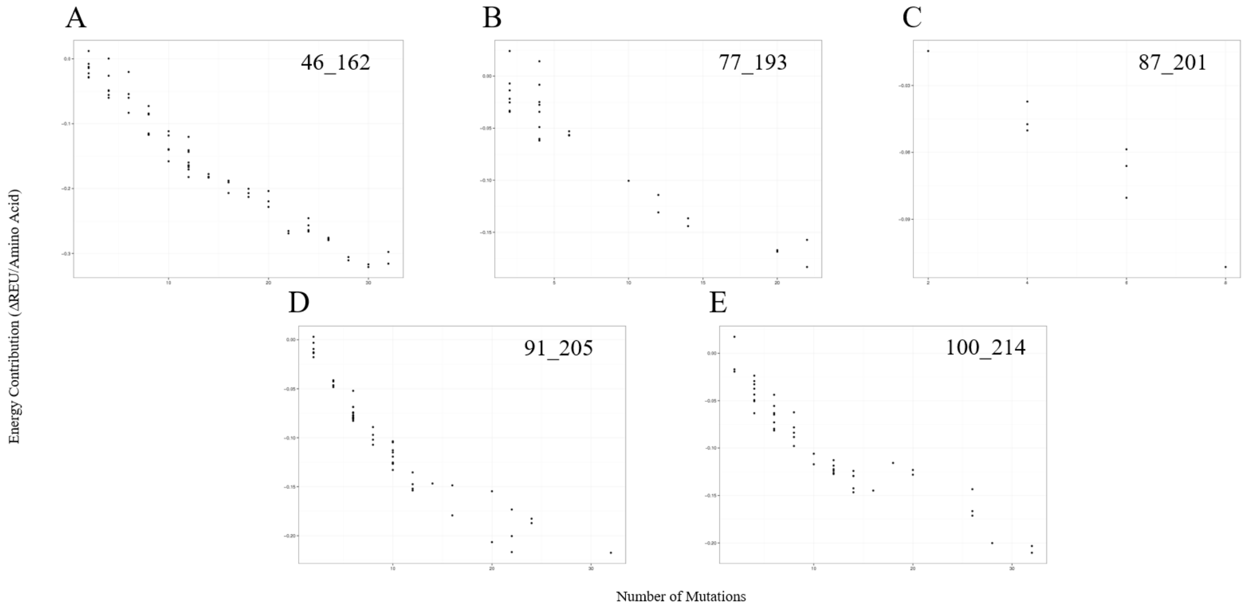


Figure AH.2 Individual plots for symmetric variants of 2D57 of the number of mutations and their normalized energy contribution. Note that panel C only has mutants for 2, 4, 6 and 8 mutations. The general trend tends to be that with each stepwise increase in mutations, the energy is lowered.

For 3ZOJ:

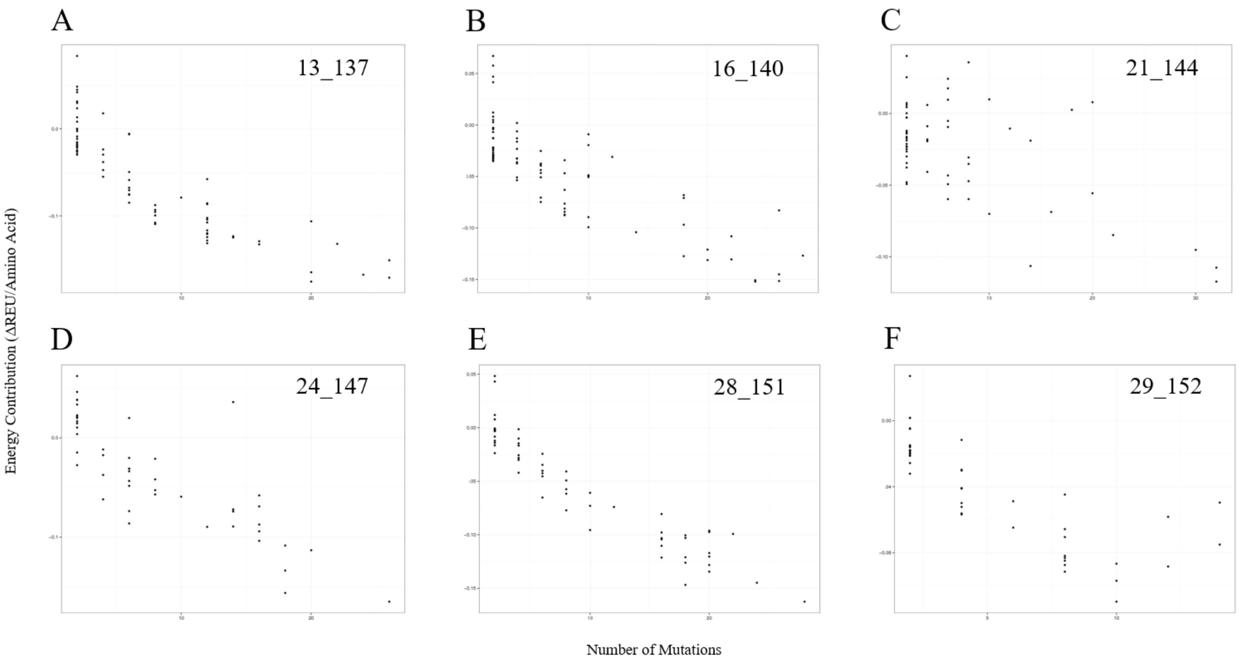


Figure AH.3 Individual plots for symmetric variants of 3ZOJ of the number of mutations and their normalized energy contribution. The general trend tends to be that with each stepwise increase in mutations, the energy is lowered.

From these plots of mutations and energetic contributions, I was able to identify symmetric variant backbones that were more stable by which backbones did not acquire many mutations for a large drop in energy. I looked for backbones that required several mutations in order to see a lowered energy. Then I selected the final symmetric variant backbones. The pI was calculated using the software ExPasy to determine whether this would be an issue during expression, in particular for the nickel affinity column.

Table AH.1. The calculated pI determined from the sequence of the protein symmetric variants.

Symmetric Variant	**pI**
1FX8 99_245	5
2D57 77_193	9.36
1FX8 102_249	5.8
2D57 100_214	6.4
3ZOJ 13_137	9.93
1FX8 95_241	5
2D57 91_205	6.4
1FX8 6_143	6.1
2D57 46_162	8.56
1FX8 102_248	5.8
3ZOJ 28_151	8.9
3ZOJ 16_140	9.93
3ZOJ 19_143	9.93
2D57 87_202	6.4
1FX8 98_245	5
2D57 87_201	6.4
3ZOJ 24_147	9.93
1FX8 7_142	6.01
3ZOJ 21_144	9.93
3ZOJ 29_152	8.9

APPENDIX I

PROTOCOL CAPTURE FOR CHAPTER 7

Rosetta revision number 57232 was used for this protocol.

I created a python script that calculated distances between symmetric counterparts on the wildtype and inverted structures. The user may input a threshold and if the c-alpha distance is lower than the threshold, the pair of residues is printed as a cut-point. I also accounted for the possibility that an error in the placement of the backbone or structural alignment may result in fewer cut-points by also sampling each residue before and after a proposed cut-point.

With the fragment start and end points, I was able to create a python script that constructed a symmetric backbone in a similar way to the protein engineering strategy circular permutation. Because the symmetric backbones were pasted in cartesian space as fragments, I utilized the dualspace relaxation protocol in Rosetta. Dualspace allows for both cartesian and internal coordinate relaxation. The first relax protocol below enables dualspace relax.

Protocols for various relaxations:

These were all run through the relax application using this command and the appropriate options files listed below:

```
~myrosetta/main/source/bin/relax.default.linuxgcc.release @my.options
-in:file:spanfile my.span
```

first relax:

```
-relax:dualspace
-relax:minimize_bond_angles #setting used with dualspace relax (from
Amanda's protocol)
```

```
-set_weights cart_bonded .5 pro_close 0 #setting used with dualspace
relax (from Amanda's protocol)

-default_max_cycles 200

-out:file:fullatom #output file will be fullatom

-out:pdb

-out:prefix rlx_

-membrane:no_interpolate_Mpair # membrane scoring specification

-membrane:Menv_penalties # turn on membrane penalty scores

-score:weights membrane_highres_Menv_smooth.wts
```

To obtain a consistent score for evaluation purposes:

```
-out:file:fullatom #output file will be fullatom

-out:pdb

-out:prefix sc_mc_

-membrane:no_interpolate_Mpair # membrane scoring specification

-membrane:Menv_penalties # turn on membrane penalty scores

-center_search true

-center_max_delta 1

-score:weights membrane_highres_Menv_smooth.wts

-hbond_bb_per_residue_energy
```

Second relax (to create more conformational diversity): (40x)

```
-out:file:fullatom #output file will be fullatom

-out:pdb

-out:prefix rlx_

-constrain_relax_to_start_coords true

-membrane:no_interpolate_Mpair # membrane scoring specification

-membrane:Menv_penalties # turn on membrane penalty scores

-score:weights membrane_highres_Menv_smooth.wts
```

Approaches for redesign of symmetric variants:

All approaches for sampling mutations involved the FavorSymmetricSequence mover (FSS). The moved can be used in RosettaScripts with the following syntax:

```
<FavorSymmetricSequence penalty="(&real)" name="sym_csts" symmetric_units="(&size)"/>
```

Where the penalty is the unit of REU added to each residue when the symmetric counterpart does not match the mutation tested, and the size is the how many internal symmetric units are in a single chain. For the purposes of this protocol, I chose the symmetric units to be 2. I tested a combination of FSS penalties along with Favor Native Residue (FNR) bonuses to see the effect on sequence recovery for both the monomeric and tetrameric model of the symmetric variants.

Table AI.1. The percent native sequence recovery resulting from FNR bonuses and FSS penalties applied to design experiments. The monomeric and tetrameric versions of symmetric variants were modeled and various bonuses and penalties were evaluated.

FNR	**Monomer**	**Tetramer**
0.5	40.1	45.4
1	63.3	67.7
1.25	78.6	81.4
1.5	90	90
FSS	**Monomer**	**Tetramer**
0.5	37	n/a
1	38.7	n/a
2	39.8	n/a
5	24.1	n/a

Table AI.2. The percent native sequence recovery from combinations of a FSS penalty and a FNR bonus applied to design experiments.

FSS	**FNR**	**Monomer**	**Tetramer**
0.5	1	67.7	n/a
1	1	69.6	63
1	1.5	90.5	86.4

Ideally, fewer mutations is better. The combination of an FSS penalty of 1 and an FNR bonus of 1.5 performs where there would only be a few mutations. An FSS penalty of higher numbers, while forcing symmetry, appears to constrain it in such a way that skews the relevance of mutations. The first attempt to redesign focused on creating symmetric space-filling mutations. I used FSS along with FNR to identify positions that would benefit from mutations.

I analyzed each half of the protein to identify mutations seen most frequently regardless of half. I noticed 4 pairs of consensus mutations that showed up in 100% of models. I list them as original residue, symmetric pair position, mutation. A13_160I; V17_164L; C109_256V; A112_259F. I redesigned using only these residues and evaluated the energy differences between the original symmetric backbone, 97_243, and the mutations (Figure AI.1)

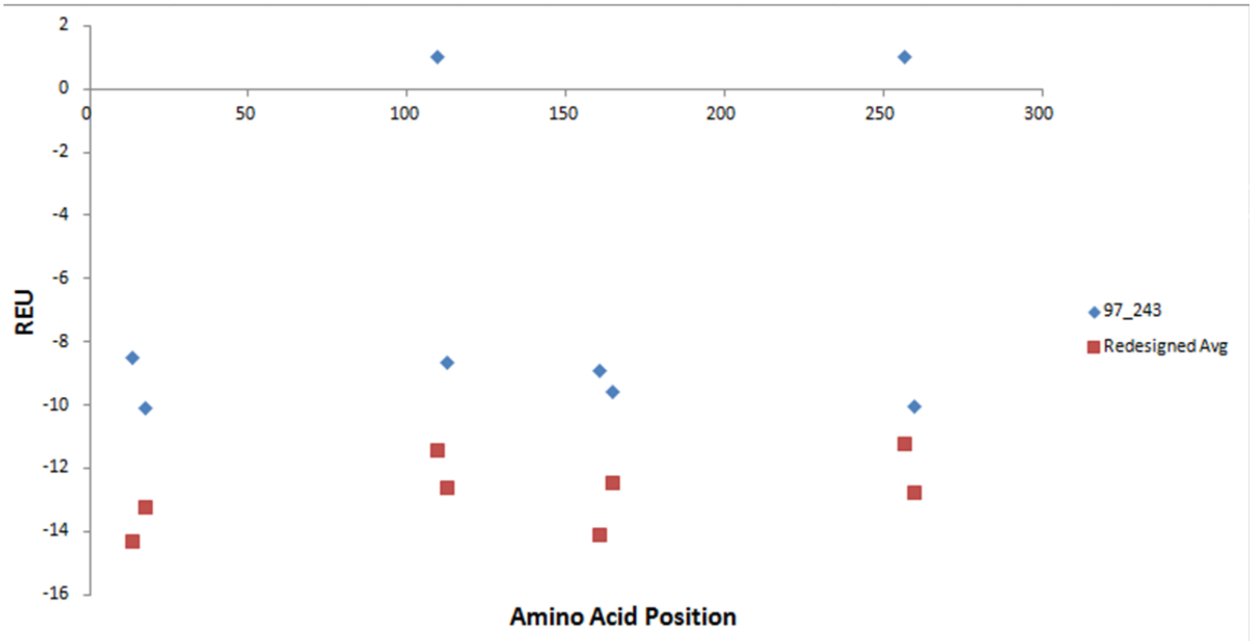


Figure AI.1. Rosetta energy unit contribution of each consensus mutation. The mutations seen in all designs were modeled as four pairs of consensus mutations only and compared energetically to the original symmetric backbone 97_243.

Next, I mapped these positions onto the structure to better understand the reason for the improved energies. However, all mutations appear on the surface (Figure AI.2) and it appears to be driven by reference energies and fa_mbenv.

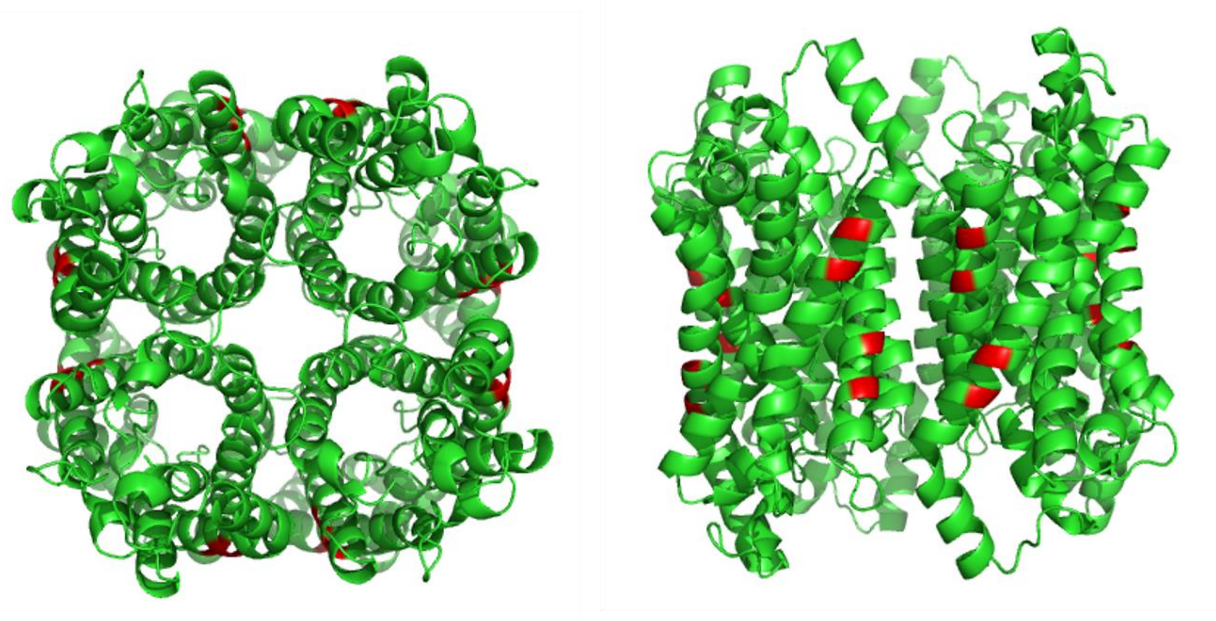


Figure A.I.2. Consensus mutations mapped onto the symmetric variant in the tetrameric form. The mutations seen in all design experiments (red) were mapped onto the model of the symmetric variants of 97_243.

These mutations appear on the surface of the protein, likely interacting with the lipid bilayer. Next, from the analysis of symmetric mutations, I identified several additional possible mutations that were favored most of the time (Table AI.3).

Table AI.3. Additional possible mutations from FSS and FNR experiments. The percent preference for each side of the protein is reported.

'native' AA	**'A' position**	**% pref: AA**	**'B' position**	**% pref:AA**
L	25	90:F	172	100:L
T	30	100:T	177	100:V
D	32	100:A	179	100:D
I	50	70:V	197	70:I
M	57	98:M	204	100:T
L	100	100:I	247	100:L
G	116	100:G	263	70:G

From this, I chose to force asymmetric mutations L25_172F and M57_204T to be symmetric. Because the consensus mutations were all on the surface of the protein, these were an interesting pair of mutations because they occur at the interface of the homo-tetrameric assembly (Figure AI.3).

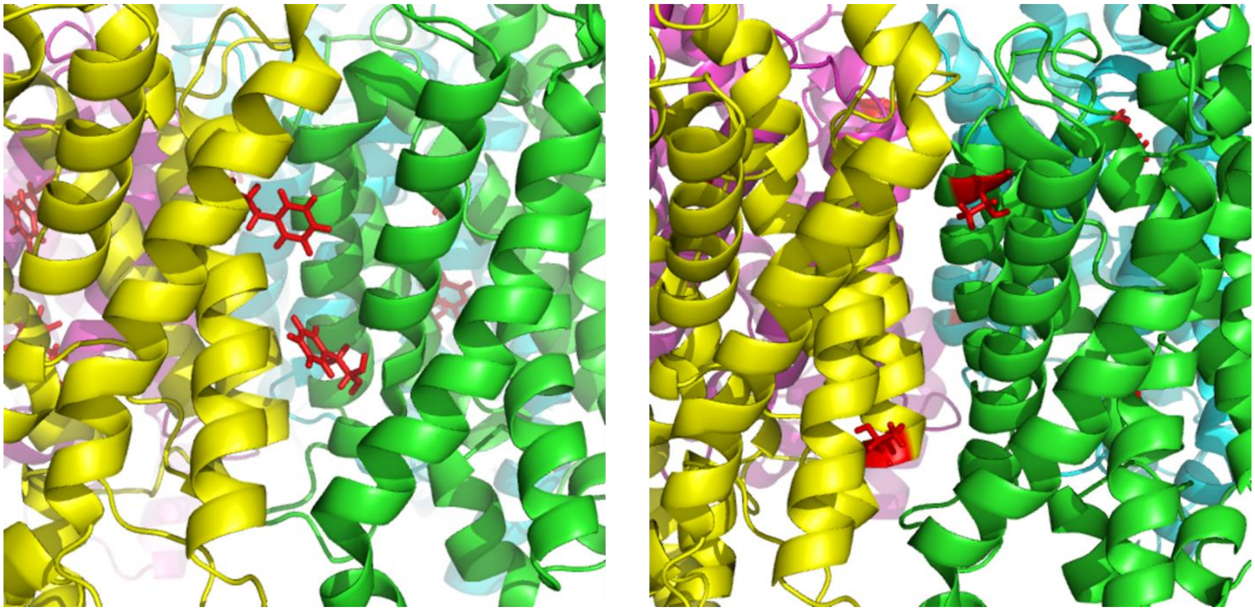


Figure AI.3. Forced mutations are at an interface for the homo-tetrameric assembly. The proposed mutations for L25_172F (left) and M57_204T (right) are shown at the interface. Positions of mutations are in red while the subunits are represented as different colors.

Next, I then performed some relaxation studies where I ultimately sorted the energies of resulting models from a relax protocol to visualize the range of sampling. I relaxed the original symmetric backbone 97_243, each forced symmetric mutations, and both forced symmetric mutations (Figure AI.4).

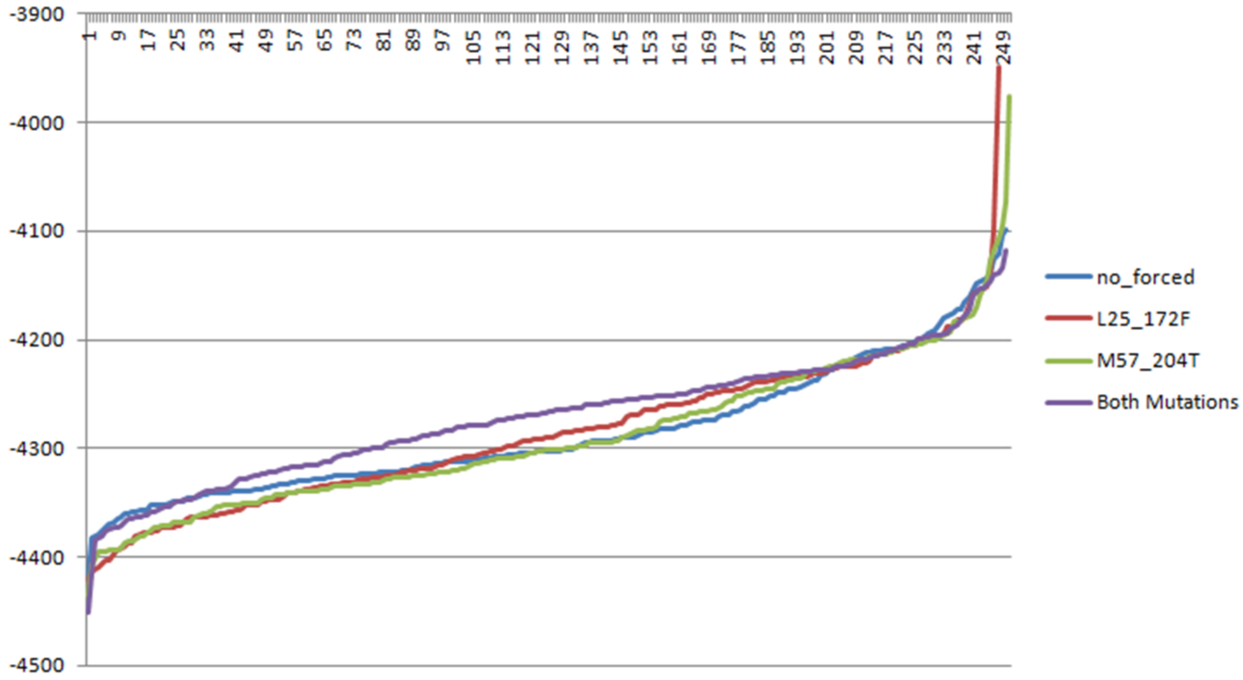


Figure AI.4. Sorted total energies of relaxed models of symmetric variant designs. While most of the models for both mutations have a worse score than the others, the lowest scoring model is one with both mutations.

Finally, my last design approach was to take the two halves of each aquaporin and create a resfile with a list of positions with both the native amino acid identity at that position and its counterpart identity. Because I restricted design to only identities seen in evolution at that position of the backbone, I refer to these as the AQP proteins. I compared the energies of the original symmetric variant to the calculated energies of each of the aforementioned designed proteins from the several rounds of design. The comparison showed a huge improvement in the sampling from the AQP design (Figure AI.5). Unfortunately, attempts to express any of these variants failed for unknown reasons.

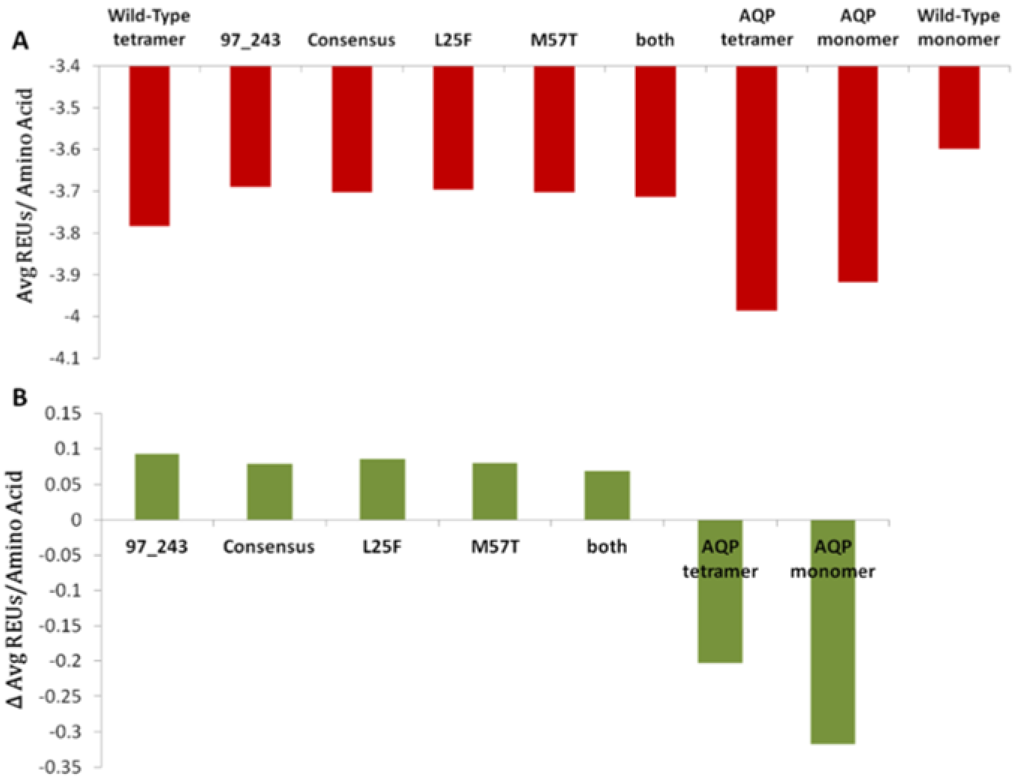


Figure AI.5. Energy evaluation of designs with evolutionary information. A shows normalized per residue energy for all models. B calculates the difference between the designs and their respective wild-type models. The engineered symmetric backbone (97_243) was initially redesigned using symmetric counterpart mutations. These mutations (consensus, L25F, M57T) did not improve energy. However, design guided by an AQP sequence alignment predicts symmetric mutations for 97_243 that result in a much lower energy than both designs and wild type.

APPENDIX J

PREVIOUS STUDIES ON SYMMETRIC VARIANTS OF GLPF

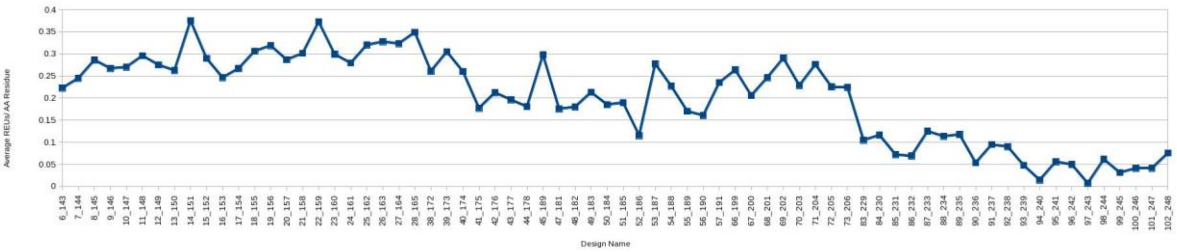


Figure AJ.1 The average, normalized REU was plotted for each symmetric variant. Symmetric variants constructed from the C-terminal side of 1FX8 tend to have a much lower energy than compared to the N-terminal side.

The first approach for symmetric designs focused on the symmetric variant 1FX8 100_243. I extracted GlpF genomic DNA and optimized the expression conditions using protocols based on the Stroud lab's protocols. Those are seen below. Before I tried expression the symmetric variants, I had to be sure that the native protein would express in in our CSB vector pBG100.

Sequences of Wild Type and Designed Half of GlpF

Nucleotide sequence of Wild Type GlpF from E. coli BL21 (DE3) (K12 Derivative)

```
atgAGTCAAACATCAACCTTGAAAGGCCAGTGCATTGCTGAATTCCTCGGTACCGGGTTGTTG

ATTTTCTTCGGTGTGGGTTGCGTTGCAGCACTAAAAGTCGCTGGTGCGTCTTTTGGTCAGTG

GGAAATCAGTGTCATTTGGGGACTGGGGGTGGCAATGGCCATCTACCTGACCGCAGGGGTT

TCCGGCGCGCATCTTAATCCCGCTGTTACCATTGCATTGTGGCTGTTTGCCTGTTTCGACAAG

CGCAAAGTTATTCCTTTTATCGTTTCACAAGTTGCCGGCGCTTTCTGCGCTGCGGCTTTAGTTT

ACGGGCTTTACTACAATTTATTTTTCGACTTCGAGCAGACTCATCACATTGTTCGCGGCAGCGT
```

TGAAAGTGTTGATCTGGCTGGCACTTTCTCTACTTACCCTAATCCTCATATCAATTTTGTGCAGG

CTTTCGCAGTTGAGATGGTGATTACCGCTATTCTGATGGGGCTGATCCTGGCGTTAACGGACG

ATGGCAACGGTGTACCACGCGGCCCTTTGGCTCCCTTGCTGATTGGTCTACTGATTGCGGTC

ATTGGCGCATCTATGGGCCCATTGACGGGTTTTGCCATGAACCCAGCGCGTGACTTCGGTCC

GAAAGTCTTTGCCTGGCTGGCGGGCTGGGGCAATGTCGCCTTTACCGGCGGCAGAGACATT

CCTTACTTCCTGGTGCCGCTTTTTGGCCCTATCGTTGGCGCGATTGTAGGTGCATTTGCCTACC

GCAAACTGATTGGTCGCCATTTGCCTTGCGATATCTGTGTTGTGGAAGAAAAGGAAACCACAA

CTCCTTCAGAACAAAAAGCTTCGCTGtaa

Amino Acid Sequence of Wild Type GlpF from E. coli BL21 (DE3) (K12 Derivative)

MSQTSTLKGQ CIAEFLGTGL LIFFGVGCVA ALKVAGASFG QWEISVIWGL GVAMAIYLTA

GVSGAHLNPA VTIALWLFAC FDKRKVIPFI VSQVAGAFCA AALVYGLYYN LFFDFEQTHH

IVRGSVESVD LAGTFSTYPN PHINFVQAFA VEMVITAILM GLILALTDDG NGVPRGPLAP

LLIGLLIAVI GASMGPLTGF AMNPARDFGP KVFAWLAGWG NVAFTGGRDI PYFLVPLFGP

IVGAIVGAFA YRKLIGRHLP CDICVVEEKE TTTPSEQKAS L

Nucleotide sequence of Designed Half GlpF

TATCCGAATCCGCATATTAACTTTGTTCAAGCGTTTGCCGTGGAAATGGTTATTACCGCAATCCT

GATGGGTCTGATCCTGGCTCTGACCGATGACGGCAACGGTGTGCCGCGTGGTCCGCTGGC

ACCGCTGCTGATTGGTCTGCTGATTGCCGTTATCGGCGCAAGTATGGGTCCGCTGACCGGCT

TTGCTATGAACCCGGCGCGTGATTTTGGTCCGAAAGTTTTCGCTTGGCTGGCGGGCTGGGGT

AATGTCGCCTTCACGGGCGGTCGCGACATTCCGTATTTTCTGGTCCCGCTGTTCGGTCCGATT

GTCGGCGCAATCGTGGCGGCCGCACTGGTTTACGGCCTGTATTACAACCTGTTTTTCGATTTTG

AACAGACGCATCACATCGTGCGCGGTAGCGTGGAAAGCGTGGACCTGGCGGGCACCTTCAGCACGTAA

Amino Acid Sequence of Designed Half GlpF

YPNPHINFVQAFAVEMVITAILMGLILALTDDGNGVPRGPLAPLLIGLLIAVIGASMGPLTGFAMNPARDFGPKVFA

WLAGWGNVAFTGGRDIPYFLVPLFGPIVGAIVAAALVYGLYYNLFFDFEQTHHIVRGSVESVDLAGTFST

wt GlpF MW: 30 kD

Designed GlpF MW: 31 kD

Primer Sequences

Primers used to target and extract wild type glycerol facilitator protein from E. coli BL21 (DE3) genomic DNA:

FWD: TCCCGTAGTCATATTACAGCGAAGC

REV: TCAGGATCCGATTATGAGTCAAACA

Primers used to SLIC clone wild type GlpF into a Bam I Not I double cut pBG100:

FWD: ctggaagttctgttccaggggcccGGATCCatgagtcaaacatcaaccttg

REV: gctagcccgtttgatctcgagtGCGGCCGCttacagcgaagctttttgttc

Primers used to SLIC clone and assemble designed half into Bam 1 Not 1 double cut pBG100:

P1: tggaagttctgttccaggggcccGGATCCTATCCGAATCCGCATATTAAC

P2: atatgcggattcggatacgtgctgaaggtgcccgccaggt

P3: acctggcgggcaccttcagcacgtatccgaatccgcatat

P4: gctagcccgtttgatctcgagtGCGGCCGCttaCGTGCTGAAGGTGCCCG

Purification Protocol for E. coli BL21(DE3) GlpF (wt)

Adapted from: Biochemistry 2008, 47, 3513-3524

See Amanda Duran Lab Notebook 1

Transform pBG100*+GlpF in to BL21 (DE3) pLysS

O/N culture of single transformant (~10 mL/L) (37C, 235 RPM)

Inoculate Terrific Broth w/ starter; 37C, 235 RPM

Induce @ OD600 0.6-0.7 (up to 1 ended up fine see Lab Notebook: Amanda Duran 1), 1 mM IPTG (final) (still 37C, 235RPM)

2 hrs induction, harvest

(can store pellet in -80C until proceed)

Resuspend pellet: Buffer “A”=25 mM Phosphate, 200 mMNaCl, 2 mM BME, (Lysozyme, DNase, RNase and Magnesium Acetate)

Sonicate @40%, 5 sec on; 5 sec off for 5 min (total time 10 min; process for 5)

Centrifuge down cells @ 20K, 20 min, 4C

Supernatant ultra-centrifuged (30K using Ti45, 60 min, 4C)

Solubilize resulting membrane pellet in:

25 mM Phosphate
200 mM NaCl
2 mM BME
20 mM LMPC (or 30 mM DDM)

Use glass homogenizer to fully solubilize

Equilibrate Ni-NTA (~2-5 mL resin if 1 g) with 10 CV of EB.

Equilibration Buffer:
25 mM Phosphate

200 mM NaCl

2 mM BME

2 mM LMPC (or 3 mM DDM)

pH=7.5

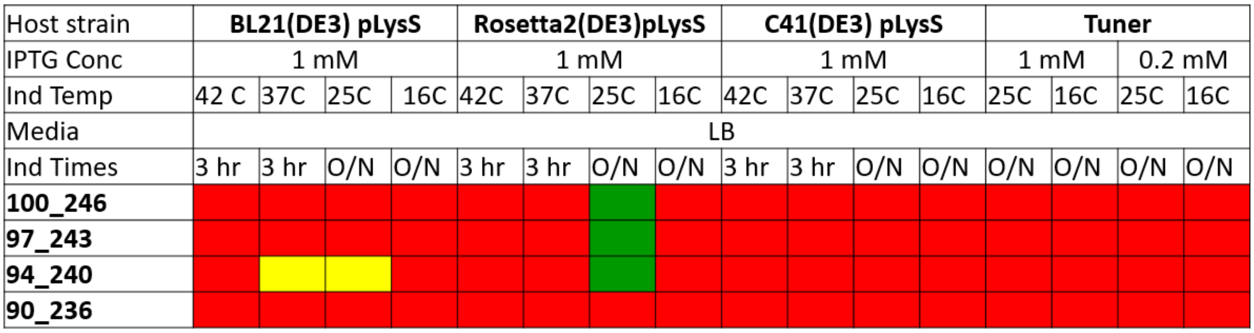
Wash 1: (until A 280 is <0.01 ~ 50 mL for 4 mL Resin)

Equilibration buffer + 50 mM imidazole

Eluted with (~20mL depending on starting amount)

Equilibration buffer + 250 mM imidazole

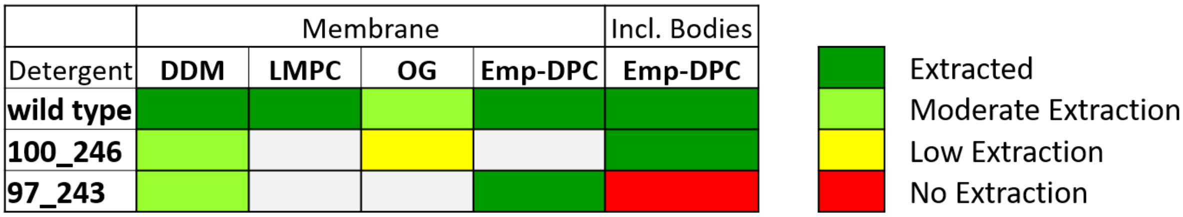
Additional constructs 97_245, 94_240 and 90_236 were subjected to extensive expression screens which included the following conditions:



Host strain	**BL21(DE3) pLysS**				**Rosetta2(DE3)pLysS**				**C41(DE3) pLysS**				**Tuner**			
IPTG Conc	1 mM				1 mM				1 mM				1 mM		0.2 mM	
Ind Temp	42 C	37C	25C	16C	42C	37C	25C	16C	42C	37C	25C	16C	25C	16C	25C	16C
Media	LB															
Ind Times	3 hr	3 hr	O/N	O/N	3 hr	3 hr	O/N	O/N	3 hr	3 hr	O/N	O/N	O/N	O/N	O/N	O/N
100_246	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
97_243	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
94_240	🟥	🟨	🟨	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
90_236	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

Figure AJ.2. Results of the expression screen for constructs. Conditions included varying host strain, induction temperature, and induction time. Only Rosetta2(DE3)pLysS at 25C overnight induction expressed proteins of interest sufficiently.

Then, the protein expressed from the one expression condition above was purified and screened for detergent extraction



	Membrane				Incl. Bodies
Detergent	**DDM**	**LMPC**	**OG**	**Emp-DPC**	**Emp-DPC**
wild type	🟩	🟩	🟢	🟩	🟩
100_246	🟢		🟨		🟩
97_243	🟢			🟩	🟥

🟩	Extracted
🟢	Moderate Extraction
🟨	Low Extraction
🟥	No Extraction

Figure AJ.3. Results of the detergent screen for constructs. The engineered proteins tended to go to the membrane, so extraction from the membrane was ideal.

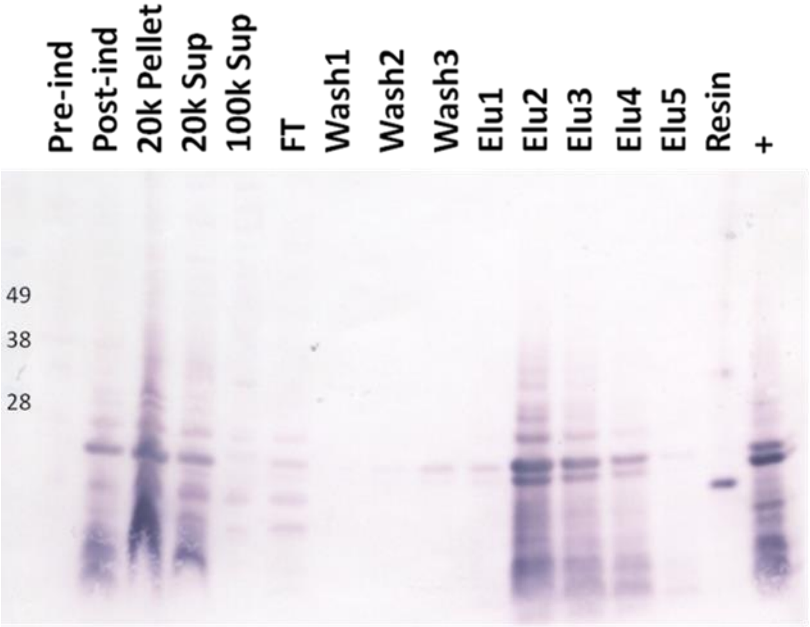
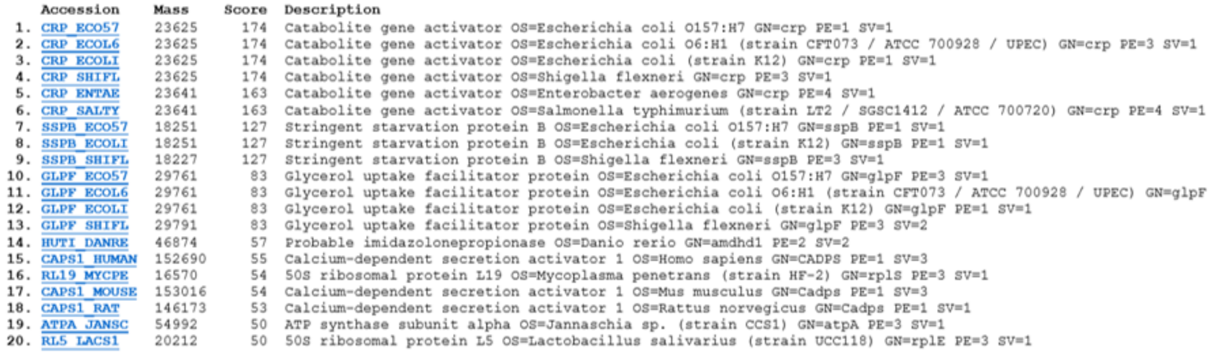


Figure AJ.4. Representative western blot of the expression and purification of a HIS-tagged symmetric variant of GlpF. This particular variant is from the symmetric counterpart mutations in 97_243.

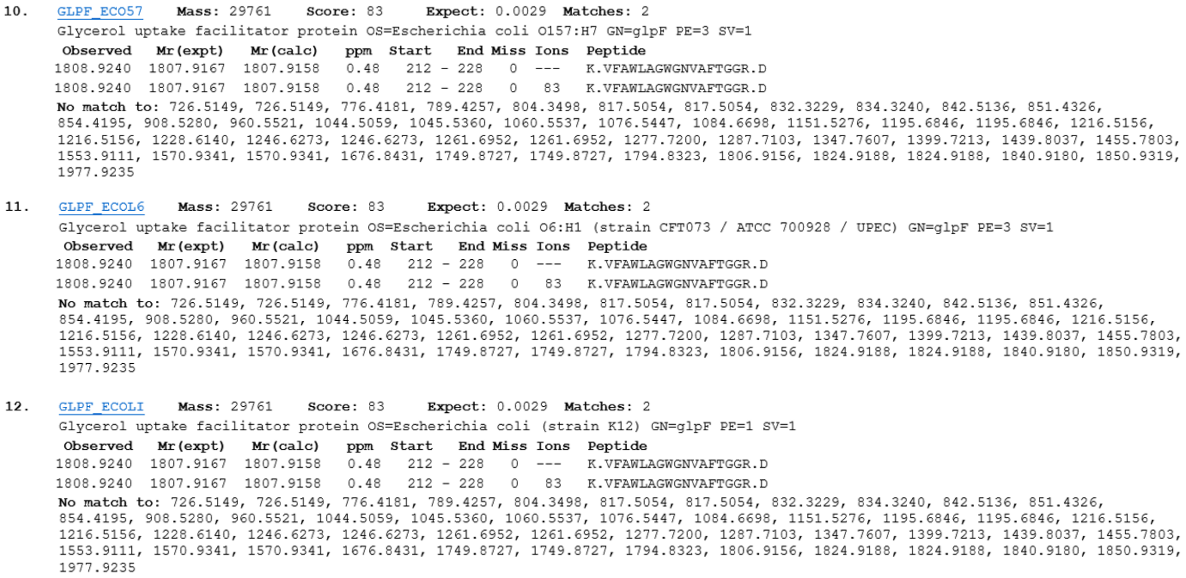
Mass spectrometry was used to sequence in order to verify its identity. While GlpF was identified in the spectra, it was at a lower concentration than other proteins in the same band. However, the top band was analyzed separately from the bottom bands and GlpF was not found in the bottom band, so this indicates it was not degradation. Unfortunately, no trypsin cuts sites were in our construct in the first 60 residues, so we could not identify whether it was the intact symmetric protein or not. Table AJ.1 shows the results in order of the most significant. A score of above 70 is significant (p<0.05)

Table AJ.1. Results of the mass spectrometry results for identification of a symmetric variant of GlpF. The top band from expression and purification studies was extracted for analysis. The band appears to be a mixture of proteins.



	Accession	Mass	Score	Description
1.	CRP_ECO57	23625	174	Catabolite gene activator OS=Escherichia coli O157:H7 GN=crp PE=1 SV=1
2.	CRP_ECOL6	23625	174	Catabolite gene activator OS=Escherichia coli O6:H1 (strain CFT073 / ATCC 700928 / UPEC) GN=crp PE=3 SV=1
3.	CRP_ECOLI	23625	174	Catabolite gene activator OS=Escherichia coli (strain K12) GN=crp PE=1 SV=1
4.	CRP_SHIFL	23625	174	Catabolite gene activator OS=Shigella flexneri GN=crp PE=3 SV=1
5.	CRP_ENTAE	23641	163	Catabolite gene activator OS=Enterobacter aerogenes GN=crp PE=4 SV=1
6.	CRP_SALTY	23641	163	Catabolite gene activator OS=Salmonella typhimurium (strain LT2 / SGSC1412 / ATCC 700720) GN=crp PE=4 SV=1
7.	SSPB_ECO57	18251	127	Stringent starvation protein B OS=Escherichia coli O157:H7 GN=sspB PE=1 SV=1
8.	SSPB_ECOLI	18251	127	Stringent starvation protein B OS=Escherichia coli (strain K12) GN=sspB PE=1 SV=1
9.	SSPB_SHIFL	18227	127	Stringent starvation protein B OS=Shigella flexneri GN=sspB PE=3 SV=1
10.	GLPF_ECO57	29761	83	Glycerol uptake facilitator protein OS=Escherichia coli O157:H7 GN=glpF PE=3 SV=1
11.	GLPF_ECOL6	29761	83	Glycerol uptake facilitator protein OS=Escherichia coli O6:H1 (strain CFT073 / ATCC 700928 / UPEC) GN=glpF
12.	GLPF_ECOLI	29761	83	Glycerol uptake facilitator protein OS=Escherichia coli (strain K12) GN=glpF PE=1 SV=1
13.	GLPF_SHIFL	29791	83	Glycerol uptake facilitator protein OS=Shigella flexneri GN=glpF PE=3 SV=2
14.	HUTI_DANRE	46874	57	Probable imidazolonepropionase OS=Danio rerio GN=amdhd1 PE=2 SV=2
15.	CAPS1_HUMAN	152690	55	Calcium-dependent secretion activator 1 OS=Homo sapiens GN=CADPS PE=1 SV=3
16.	RL19_MYCPE	16570	54	50S ribosomal protein L19 OS=Mycoplasma penetrans (strain HF-2) GN=rplS PE=3 SV=1
17.	CAPS1_MOUSE	153016	54	Calcium-dependent secretion activator 1 OS=Mus musculus GN=Cadps PE=1 SV=3
18.	CAPS1_RAT	146173	53	Calcium-dependent secretion activator 1 OS=Rattus norvegicus GN=Cadps PE=1 SV=1
19.	ATPA_JANSC	54992	50	ATP synthase subunit alpha OS=Jannaschia sp. (strain CCS1) GN=atpA PE=3 SV=1
20.	RL5_LACS1	20212	50	50S ribosomal protein L5 OS=Lactobacillus salivarius (strain UCC118) GN=rplE PE=3 SV=1

Table AJ.2. Origin of fragments identified by mass spectrometry as GlpF. The sequences of all of the identified fragments are aligned to the sequence of the organism.

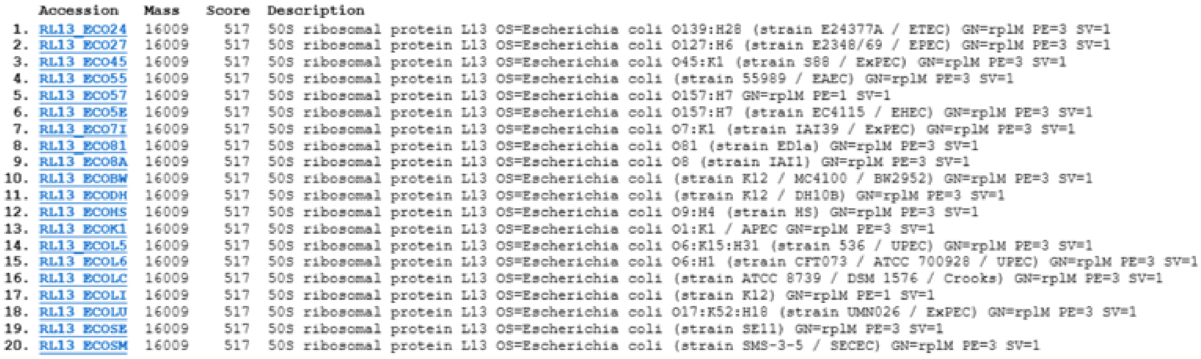


```
10.    GLPF_ECO57     Mass: 29761     Score: 83     Expect: 0.0029  Matches: 2
       Glycerol uptake facilitator protein OS=Escherichia coli O157:H7 GN=glpF PE=3 SV=1
        Observed   Mr(expt)   Mr(calc)    ppm  Start    End Miss Ions  Peptide
       1808.9240  1807.9167  1807.9158   0.48   212 - 228   0  ---   K.VFAWLAGWGNVAFTGGR.D
       1808.9240  1807.9167  1807.9158   0.48   212 - 228   0   83   K.VFAWLAGWGNVAFTGGR.D
       No match to: 726.5149, 726.5149, 776.4181, 789.4257, 804.3498, 817.5054, 817.5054, 832.3229, 834.3240, 842.5136, 851.4326,
       854.4195, 908.5280, 960.5521, 1044.5059, 1045.5360, 1060.5537, 1076.5447, 1084.6698, 1151.5276, 1195.6846, 1195.6846, 1216.5156,
       1216.5156, 1228.6140, 1246.6273, 1246.6273, 1261.6952, 1261.6952, 1277.7200, 1287.7103, 1347.7607, 1399.7213, 1439.8037, 1455.7803,
       1553.9111, 1570.9341, 1570.9341, 1676.8431, 1749.8727, 1749.8727, 1794.8323, 1806.9156, 1824.9188, 1824.9188, 1840.9180, 1850.9319,
       1977.9235

11.    GLPF_ECOL6     Mass: 29761     Score: 83     Expect: 0.0029  Matches: 2
       Glycerol uptake facilitator protein OS=Escherichia coli O6:H1 (strain CFT073 / ATCC 700928 / UPEC) GN=glpF PE=3 SV=1
        Observed   Mr(expt)   Mr(calc)    ppm  Start    End Miss Ions  Peptide
       1808.9240  1807.9167  1807.9158   0.48   212 - 228   0  ---   K.VFAWLAGWGNVAFTGGR.D
       1808.9240  1807.9167  1807.9158   0.48   212 - 228   0   83   K.VFAWLAGWGNVAFTGGR.D
       No match to: 726.5149, 726.5149, 776.4181, 789.4257, 804.3498, 817.5054, 817.5054, 832.3229, 834.3240, 842.5136, 851.4326,
       854.4195, 908.5280, 960.5521, 1044.5059, 1045.5360, 1060.5537, 1076.5447, 1084.6698, 1151.5276, 1195.6846, 1195.6846, 1216.5156,
       1216.5156, 1228.6140, 1246.6273, 1246.6273, 1261.6952, 1261.6952, 1277.7200, 1287.7103, 1347.7607, 1399.7213, 1439.8037, 1455.7803,
       1553.9111, 1570.9341, 1570.9341, 1676.8431, 1749.8727, 1749.8727, 1794.8323, 1806.9156, 1824.9188, 1824.9188, 1840.9180, 1850.9319,
       1977.9235

12.    GLPF_ECOLI     Mass: 29761     Score: 83     Expect: 0.0029  Matches: 2
       Glycerol uptake facilitator protein OS=Escherichia coli (strain K12) GN=glpF PE=1 SV=1
        Observed   Mr(expt)   Mr(calc)    ppm  Start    End Miss Ions  Peptide
       1808.9240  1807.9167  1807.9158   0.48   212 - 228   0  ---   K.VFAWLAGWGNVAFTGGR.D
       1808.9240  1807.9167  1807.9158   0.48   212 - 228   0   83   K.VFAWLAGWGNVAFTGGR.D
       No match to: 726.5149, 726.5149, 776.4181, 789.4257, 804.3498, 817.5054, 817.5054, 832.3229, 834.3240, 842.5136, 851.4326,
       854.4195, 908.5280, 960.5521, 1044.5059, 1045.5360, 1060.5537, 1076.5447, 1084.6698, 1151.5276, 1195.6846, 1195.6846, 1216.5156,
       1216.5156, 1228.6140, 1246.6273, 1246.6273, 1261.6952, 1261.6952, 1277.7200, 1287.7103, 1347.7607, 1399.7213, 1439.8037, 1455.7803,
       1553.9111, 1570.9341, 1570.9341, 1676.8431, 1749.8727, 1749.8727, 1794.8323, 1806.9156, 1824.9188, 1824.9188, 1840.9180, 1850.9319,
       1977.9235
```

Table AJ.3. Results of the mass spectrometry results for identification of a symmetric variant of GlpF. The lower band from expression and purification studies was extracted for analysis. It appears to be a homogenous sample of a ribosomal factor.



	Accession	Mass	Score	Description
1.	RL13_ECO24	16009	517	50S ribosomal protein L13 OS=Escherichia coli O139:H28 (strain E24377A / ETEC) GN=rplM PE=3 SV=1
2.	RL13_ECO27	16009	517	50S ribosomal protein L13 OS=Escherichia coli O127:H6 (strain E2348/69 / EPEC) GN=rplM PE=3 SV=1
3.	RL13_ECO45	16009	517	50S ribosomal protein L13 OS=Escherichia coli O45:K1 (strain S88 / ExPEC) GN=rplM PE=3 SV=1
4.	RL13_ECO55	16009	517	50S ribosomal protein L13 OS=Escherichia coli (strain 55989 / EAEC) GN=rplM PE=3 SV=1
5.	RL13_ECO57	16009	517	50S ribosomal protein L13 OS=Escherichia coli O157:H7 GN=rplM PE=1 SV=1
6.	RL13_ECO5E	16009	517	50S ribosomal protein L13 OS=Escherichia coli O157:H7 (strain EC4115 / EHEC) GN=rplM PE=3 SV=1
7.	RL13_ECO7I	16009	517	50S ribosomal protein L13 OS=Escherichia coli O7:K1 (strain IAI39 / ExPEC) GN=rplM PE=3 SV=1
8.	RL13_ECO81	16009	517	50S ribosomal protein L13 OS=Escherichia coli O81 (strain ED1a) GN=rplM PE=3 SV=1
9.	RL13_ECO8A	16009	517	50S ribosomal protein L13 OS=Escherichia coli O8 (strain IAI1) GN=rplM PE=3 SV=1
10.	RL13_ECOBW	16009	517	50S ribosomal protein L13 OS=Escherichia coli (strain K12 / MC4100 / BW2952) GN=rplM PE=3 SV=1
11.	RL13_ECODH	16009	517	50S ribosomal protein L13 OS=Escherichia coli (strain K12 / DH10B) GN=rplM PE=3 SV=1
12.	RL13_ECOHS	16009	517	50S ribosomal protein L13 OS=Escherichia coli O9:H4 (strain HS) GN=rplM PE=3 SV=1
13.	RL13_ECOK1	16009	517	50S ribosomal protein L13 OS=Escherichia coli O1:K1 / APEC GN=rplM PE=3 SV=1
14.	RL13_ECOL5	16009	517	50S ribosomal protein L13 OS=Escherichia coli O6:K15:H31 (strain 536 / UPEC) GN=rplM PE=3 SV=1
15.	RL13_ECOL6	16009	517	50S ribosomal protein L13 OS=Escherichia coli O6:H1 (strain CFT073 / ATCC 700928 / UPEC) GN=rplM PE=3 SV=1
16.	RL13_ECOLC	16009	517	50S ribosomal protein L13 OS=Escherichia coli (strain ATCC 8739 / DSM 1576 / Crooks) GN=rplM PE=3 SV=1
17.	RL13_ECOLI	16009	517	50S ribosomal protein L13 OS=Escherichia coli (strain K12) GN=rplM PE=1 SV=1
18.	RL13_ECOLU	16009	517	50S ribosomal protein L13 OS=Escherichia coli O17:K52:H18 (strain UMN026 / ExPEC) GN=rplM PE=3 SV=1
19.	RL13_ECOSE	16009	517	50S ribosomal protein L13 OS=Escherichia coli (strain SE11) GN=rplM PE=3 SV=1
20.	RL13_ECOSM	16009	517	50S ribosomal protein L13 OS=Escherichia coli (strain SMS-3-5 / SECEC) GN=rplM PE=3 SV=1

APPENDIX K

COMPUTATIONAL AND EXPERIMENTAL FEEDBACK FROM A2A RECEPTOR TRAFFICKING

This appendix is based on a paper entitled “Engineering the adenosine A2a receptor for site-directed interrogation of structure and dynamics: transmembrane cysteines are not required for stability and function” in preparation by:

Nikki Schonenbach and Amanda Duran

In this study, I contributed computational design and modeling of mutants of the A2a receptor. Our collaborators performed the trafficking and FACS experiments. Of the figures presented here, I contributed data in Figures AK.1, AK.2, AK.3 A-B, AK.4, AK.5, and AK. 6.

Abstract

Membrane proteins are responsible for many functions that involve large conformational changes. Using traditional methods, such as X-ray crystallography, for structure determination often require homogenous samples and are a static snapshot for may be expected to happen biologically. Electron paramagnetic resonance (EPR) is one technique that can provide insight regarding structural dynamics. EPR requires tagged of a single position in a protein using a paramagnetic label. Typically, this involves attaching a nitroxide spin label to a particular cysteine-which should be the only cysteine exposed to the solvent. G protein coupled receptors (GPCRs) are structures that often have several cysteines in their sequence, many of which can be freely accessible to the solvent. GPCRS often have several accessible cysteines and usually involve mutagenesis by trial-and-error or variants containing unnatural amino acids. This study used a strategy to combine the high-throughput screening power of a fluorescence-activated cell

sorting (FACS) ligand binding assay with Rosetta, a macromolecular modeling suite, to engineer a properly folded adenosine A2a receptor variant produced from canonical amino acids and void of solvent exposed free cysteine.

Statement of Significance

Overall, the methods developed and the subsequent findings in this study offer two advances to the GPCR community. The methods examine the combined power of computational modeling and high throughput screening of a GPCR variant library toward engineering a specific construct ideal for EPR experiments. Specifically, the effort was directed at identifying A2a receptor variants void of all non-critical free cysteines that express well in yeast *Saccharomyces cerevisiae* suitable for biophysical characterization by site-directed spin labeling. Additionally, the results provide insight into the role of transmembrane cysteines on the structure and function of the adenosine A2a receptor through computational modeling and membrane trafficking assays.

The results presented in this appendix are centered around the computational studies. These studies involved both the prediction of possible mutants of the A2a receptor using Rosetta Design and the energetic evaluation of mutants proposed from FACS library screening through modeling of mutations.

Rosetta was used to propose a number of mutations away from cysteine. Design allowing all amino acid identities, except for cysteine, to be sampled was performed at six positions of interests on multiple backbones for A2a (2YDO, 2YDV, 3EML, 3PWH, 3REY, 3RFM, 4EIY). The top 10 percent of models were analyzed for the favored mutations at each position. Figure AK.1 shows sequence logos that show the mutations seen in the top 10 percent for each backbone. Then mutations seen frequently in the sequence logos were modeled on multiple backbones for A2a (2YDO, 2YDV, 3EML, 3PWH, 3REY, 3RFM, 4EIY) and the total energy of the wild-type protein was subtracted from the energy of the mutant protein to assess the stability of the individual mutations. Then consensus models were created for combinations of the mutations based on the overall consensus of the sequence profile.

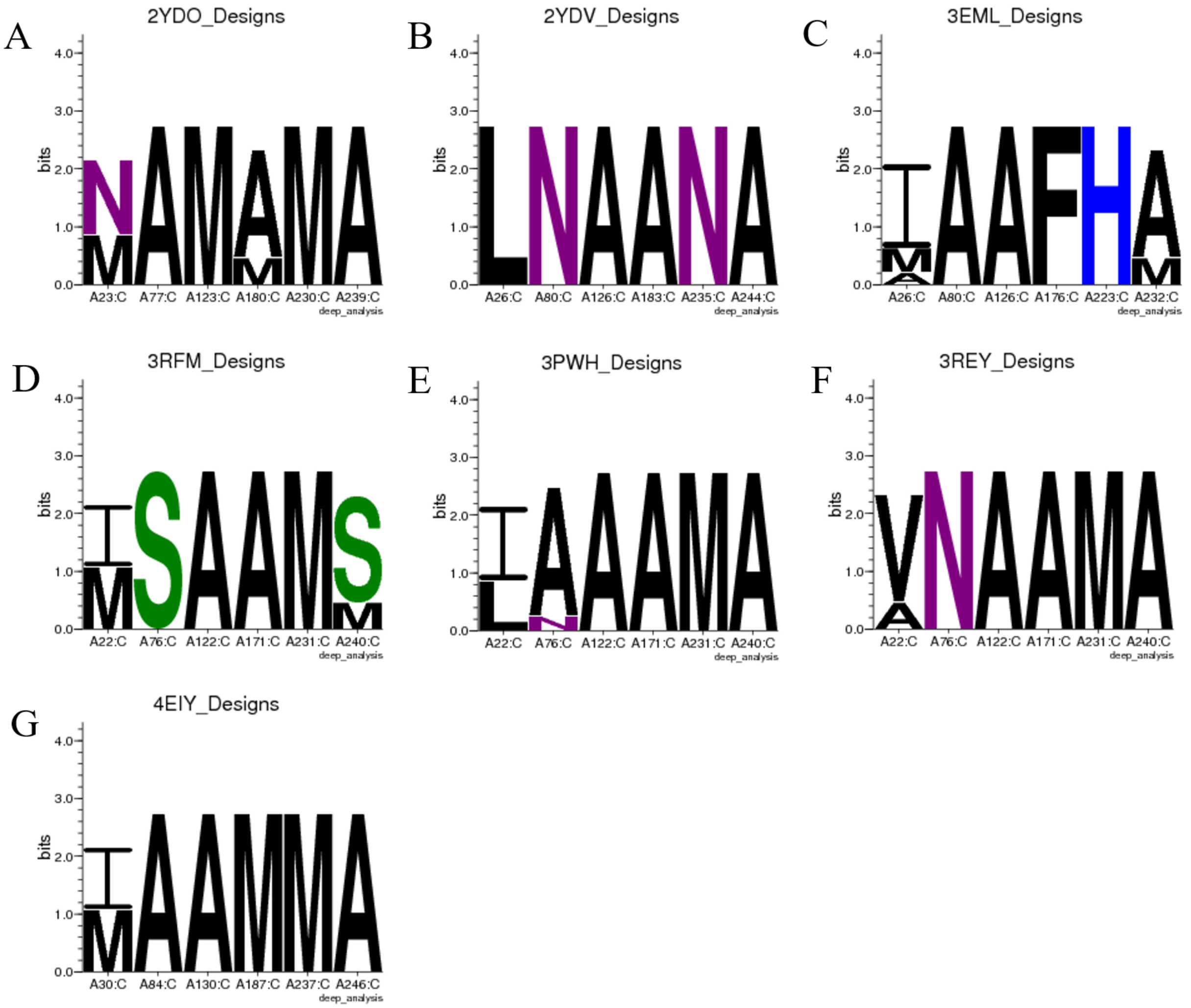


Figure AK.1. Sequence logos of the top models from single state design of the A2a receptor. The top ten percent of models by score were selected for each PDBID for analysis. Logos cover designed residues in the place of six of the native cysteines. Letters that make up most of the bits score for each position are seen the most frequently.

Experiments that sampled either Alanine and Serine at the six positions were also performed in parallel to the design experiments that sampled all amino acids except for cysteine. These experiments sampled all six positions at the same time. The most frequently seen mutations for each position were then modeled individually. The motivation is that the mutations seen most frequently are tolerated in a variety of combined mutants, thus are the most likely to be stabile independent of the identities of the other five positions. The energy differences between the proposed single-point mutants and wild-type proteins are shown in Figure AK.2. The error bar shows the range of energy differences seen among the seven different backbones. The positions of cysteines to be redesigned are denoted as C1-C6. From the sequence logos and energetic analysis, I proposed C1A, C1I, C1M, C2A, C2N, C3A, C4A, C4N, C5M, C5S, C5T, C6A, and C6M.

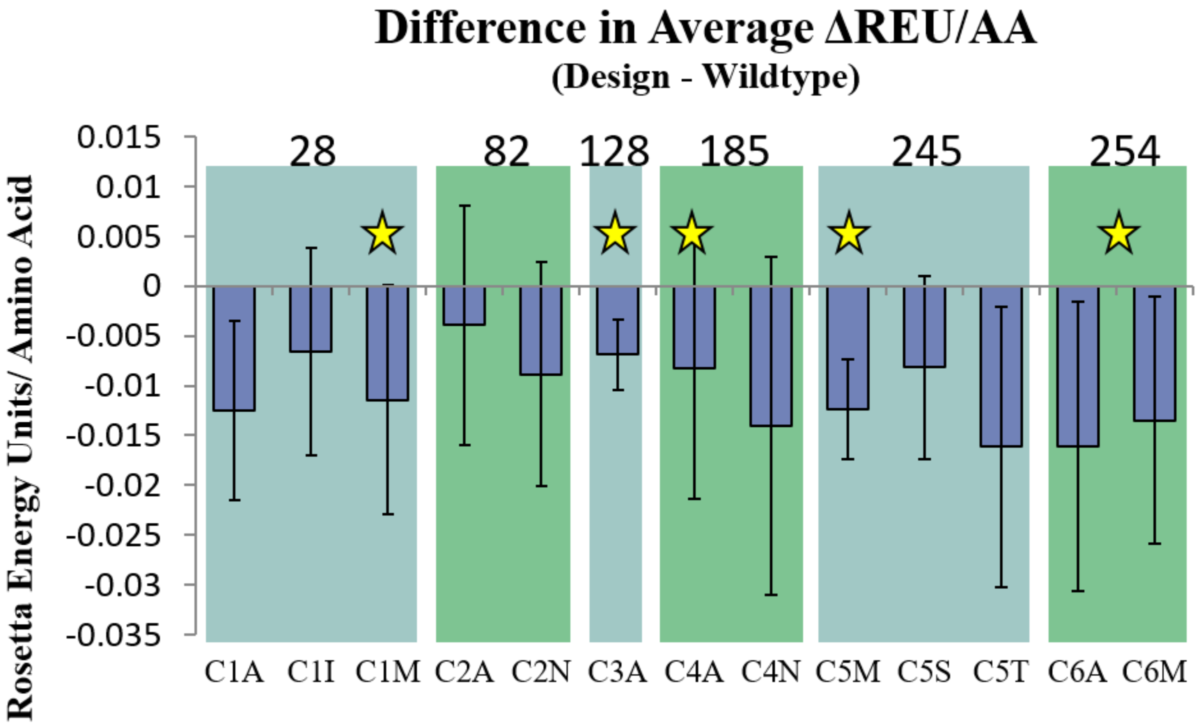


Figure AK.2. Evaluation of Rosetta energy score and whether the mutant trafficked to the membrane. Proposed mutations were gathered from experiments that design all six positions in parallel. Single mutations were modeled in Rosetta and the difference in energy between the wild-type and mutants was calculated and normalized (Rosetta Energy Units / Amino Acid). The stars represent the mutants that trafficked best. For C6, both C6A and C6M trafficked well.

Hexamutants were proposed based on the results from the single point mutation design experiments. The INANTM hexamutant was created from non-Alanine and non-Serine mutations, when available. Whereas AAAASA was created from a combination of only Alanine and Serine mutations. Unforunately, despite having an improved Rosetta energy compared to wile-type for all seven backbones, these hexamutants did not traffick to the membrane (Figure AK.3).

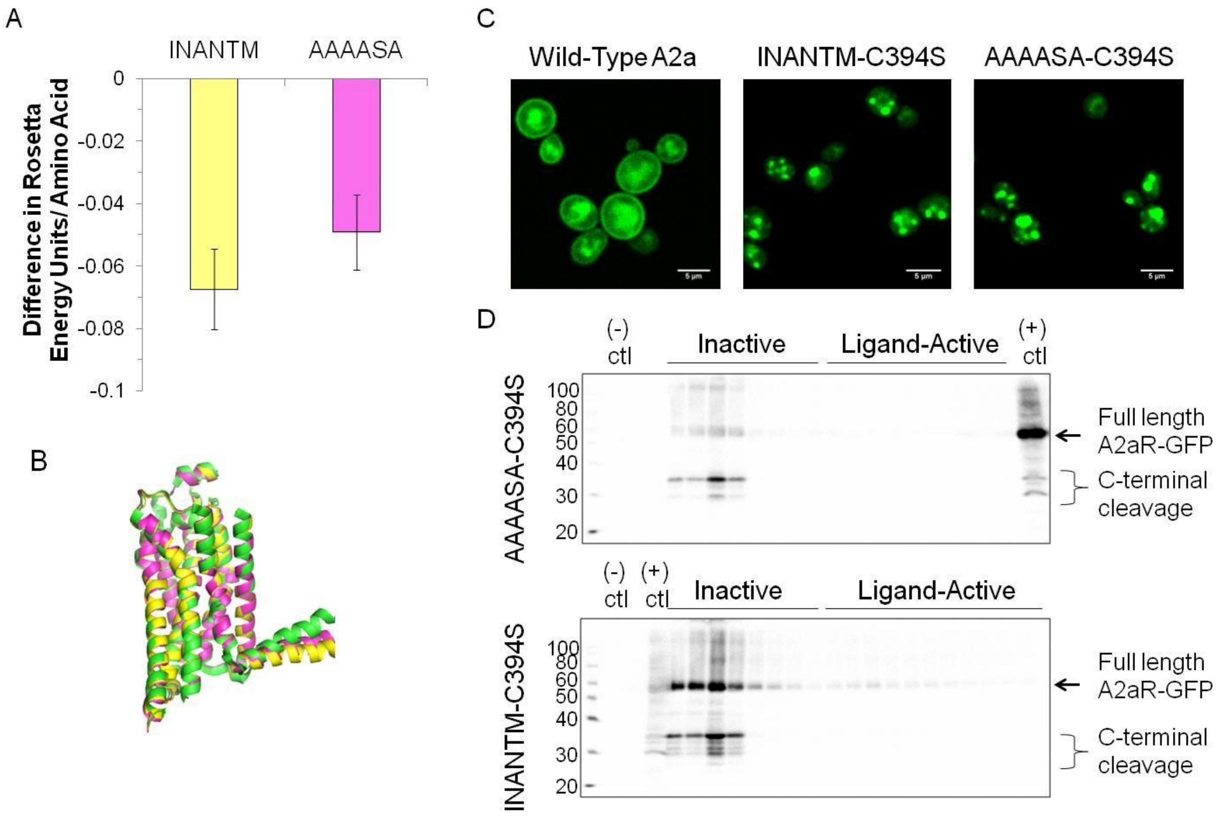


Figure AK.3. Rosetta proposed hexamutants did not traffick to the membrane. A shows the energy of the mutants compared to the normalized improvement in Rosetta energy units. B shows the respective models with green as wild-type, yellow as INANTM, and pink as AAAASA. C shows that these hexamutants did not traffick to the membrane.

While the hexamutants proposed did not traffick well despite improved Rosetta energies, the single point mutants did fair quite well in trafficking experiments. To show a measure of this success, mutants that trafficked well were treated as true positive while mutants that did not traffick well were false positives. A receiver operating characteristic (ROC) curve was generated with an area under the curve (AUC) of 0.73 (Figure AK. 4).

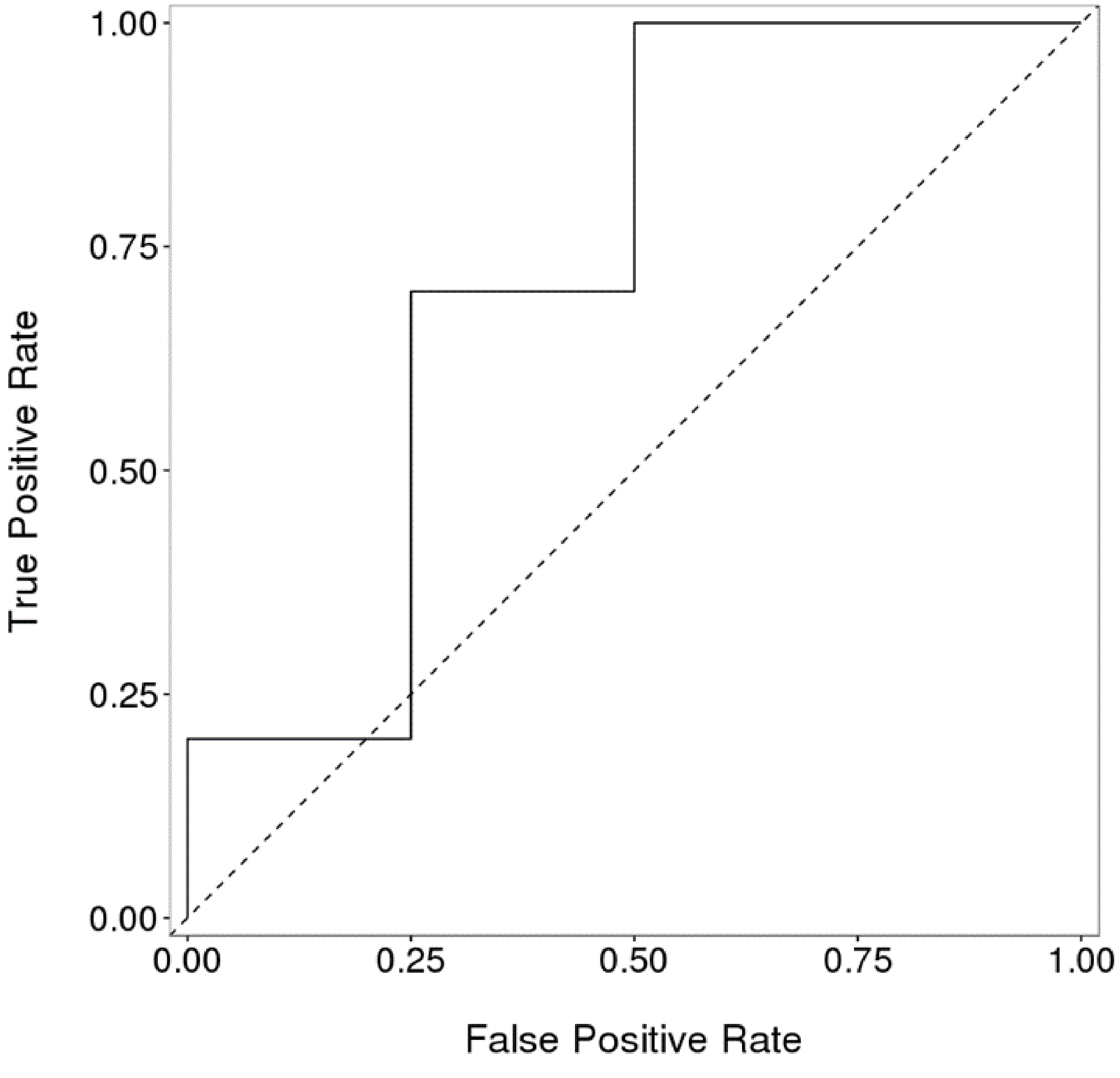


Figure AK.4. ROC curve generated for A2a mutants predicted to be stable by Rosetta and their ability to traffick to the membrane. For each single point mutants from the first part of the study, a mutant that trafficked to the membrane was deemed successful and treated as a true positive, whereas proposed mutants that did not traffick to the membrane were given a false positive rate. The area under the curve was 0.73.

FACS was used to screen a library of mutants as possible candidates. The sequences of the mutants were then used to create models in Rosetta. The energies of the mutant models were compared to the wild-type models to determine whether Rosetta would have predicted these mutants as successful. Figure AK.5 shows the correlation of the normalized change in Rosetta energy units and the percent of wild-type gating. The correlations showed a negative correlation which is expected as because a lower REU/AA is desired along with a higher percentage of wild-type gating. The Pearson R correlation was -0.49 and the Spearman R was -0.53.

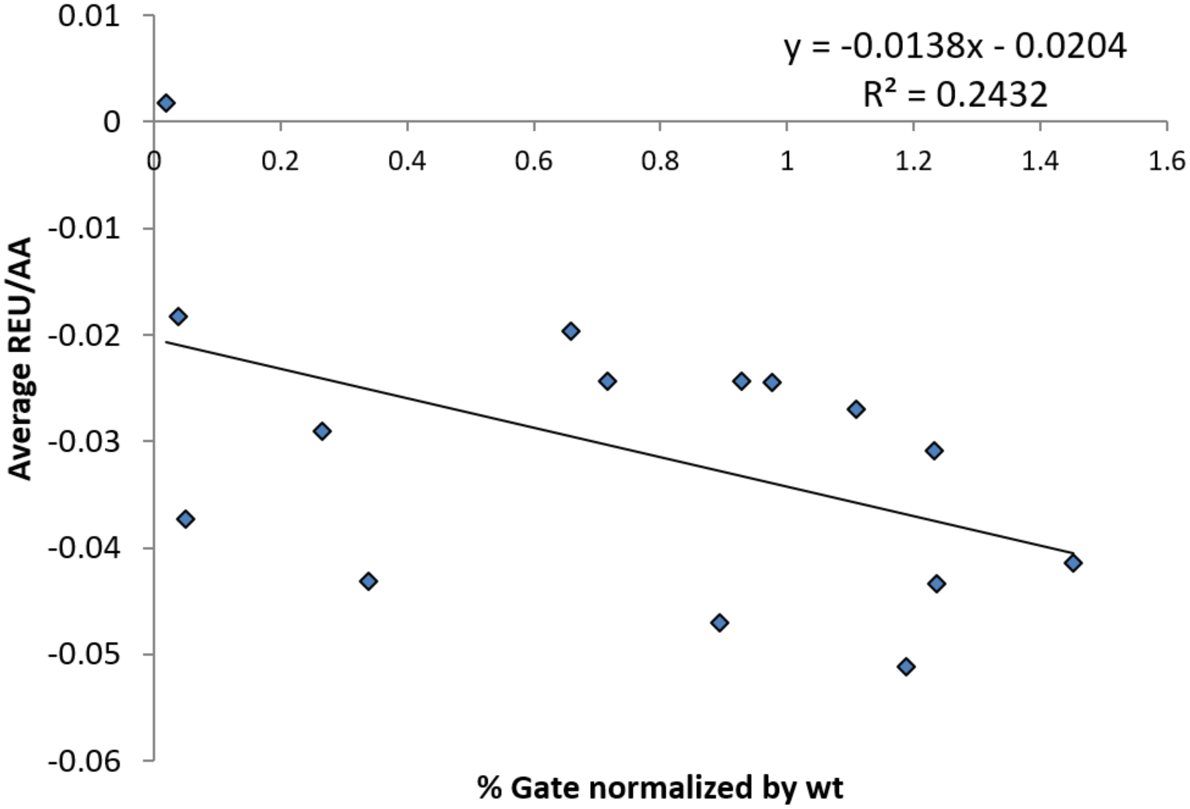


Figure AK.5. Correlation of the normalized changed in Rosetta energy units for mutant proteins to the percentage of gating in flow cytometry experiments. There is a negative Pearson R of -0.49 and a negative Spearman R of -0.53. In this case negative is good because the desired effect is a lower REU/AA and higher percent of gating.

The performance of Rosetta energy predictions was evaluated using the percentage of wild-type gating metric obtained from FACS experiments. I classified good from bad mutants by using a threshold of 80% for the percentage of wild-type gating. Mutants that were calculated by Rosetta to have an improved energy and were seen to have met the 80% threshold were treated as true positives. Mutants that were calculated to have an improved Rosetta energy and were classified as bad binders were treated as false positives. Mutants that were calculated to have a worse Rosetta energy and classified as good binders were treated as false positives. Mutants that were calculated to have a worse Rosetta energy and were classified as bad binders were treated as true positives.

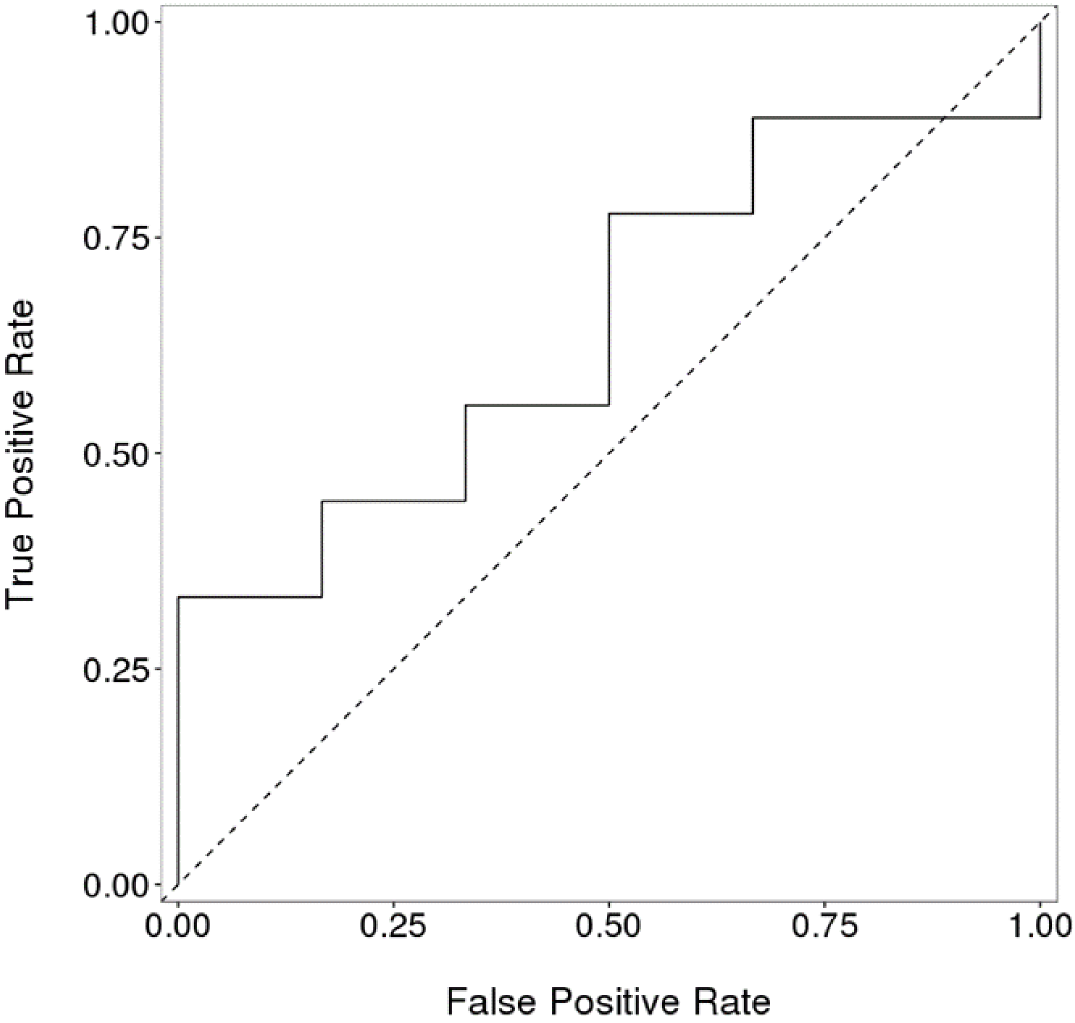


Figure AK.6. ROC curve generated for A2a mutants selected from FACS library screen. The mutants were analyzed based on the percentage of wild-type ligand binding. A threshold of 80% of wild-type ligand binding was used to classify mutants as good. The mutants that met this threshold and had an improved Rosetta energy were treated as true positives.

APPENDIX L

PROTOCOL CAPTURE FOR APPENDIX K

For these particular protocols mentioned in this Appendix, I used Rosetta revision number 57698. In brief, I used two approaches to sample possible mutations at six positions in the A2a receptor. The first was ensemble design where all seven backbones were taken into the protocol to hopefully suggest mutations for six positions that were agreeable with all seven backbones. The second was single state design of all seven backbones independent of each other. The top scoring models were then analyzed for suggested mutations that were seen in most of the backbones. I found that relying on the ensemble design protocol to converge on mutations at these six positions was too restricting. I moved forward with the single state design approach that uses the top models by score to suggest mutations that are seen in most of the models across all seven backbones.

Suggestions for hexamutants were created based on a combination of the favored mutations seen from the single state design experiments. To create models of these hexamutants, I used the protocol for modeling defined mutations. Upon FACS sorting of a library, our collaborators suggested mutants to model in Rosetta. These were also modeled using the protocol for modeling defined mutations.

Sampling possible mutations using an ensemble design protocol:

First, input structures were prepared by minimization using the relax application. Ten models were generated for each PDBID (2YDO, 2YDV, 3EML, 3PWH, 3REY, 3RFM, 4EIY) and the top model was selected for further analysis.

```
/path/to/Rosetta/main/source/bin/relax.default.linuxgccrelease
-s myfile.pdb -in:file:spanfile myfile.span -out:file:fullatom -out:pdb
-out:prefix rlx_ -membrane:no_interpolate_Mpair -membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts -nstruct 10
```

The top model of each was then used for the setup of ensemble design. Ensemble design was run using the following XML file , command, and options:

Contents of hybrid_min.xml:

```
<ROSETTASCRIPTS>
      <SCOREFXNS>
            <mem_highres weights=membrane_highres_Menv_smooth.wts >
                  <Reweight scoretype=res_type_constraint weight=1.0 />
            </mem_highres>
      </SCOREFXNS>
      <TASKOPERATIONS>
            <InitializeFromCommandline name=ifcl />
            <RestrictToRepacking name=rtr />
      </TASKOPERATIONS>
      <MOVERS>
            <PackRotamersMover name=design scorefxn=mem_highres
task_operations=ifcl />
            <MSDMover name=msd1 design_mover=design constraint_weight=0.5
resfiles=%%resfiles%% />
            <MSDMover name=msd2 design_mover=design constraint_weight=1
resfiles=%%resfiles%% />
            <MSDMover name=msd3 design_mover=design constraint_weight=1.5
resfiles=%%resfiles%% />
            <MSDMover name=msd4 design_mover=design constraint_weight=2
resfiles=%%resfiles%% />
            <FindConsensusSequence name=finish scorefxn=mem_highres
resfiles=%%resfiles%% />
            <PackRotamersMover name=repack scorefxn=mem_highres
task_operations=ifcl,rtr />
            <TaskAwareMinMover name=min tolerance=0.001 task_operations=ifcl
type=lbfgs_armijo_nonmonotone chi=1 bb=1 jump=1 scorefxn=mem_highres />
            <FastRelax name=relax scorefxn=mem_highres
task_operations=ifcl,rtr repeats=1 />
      </MOVERS>
      <FILTERS>
      </FILTERS>
      <APPLY_TO_POSE>
      </APPLY_TO_POSE>
      <PROTOCOLS>
            <Add mover=msd1 />
            <Add mover=min />
            <Add mover=msd2 />
            <Add mover=min />
            <Add mover=msd3 />
            <Add mover=min />
            <Add mover=msd4 />
            <Add mover=min />
            <Add mover=finish />
            <Add mover=relax />
      </PROTOCOLS>
```

```
</ROSETTASCRIPTS>
```

```
/path/to/Rosetta/main/source/bin/rosetta_scripts.default.linuxgccrelease
@ensemble_design.options -parser:protocol hybrid_min.xml -in:file:spanfile
2YDV.span -parser:script_vars
resfiles=2YDO.resfile,2YDV.resfile,3EML.resfile,3PWH.resfile,3REY.resfile,3RF
M.resfile,4EIY.resfile -s rlx_al_2YDO_A1_0005.pdb rlx_al_2YDV_A2_0006.pdb
rlx_al_3EML_rtrim3_0003.pdb rlx_al_3PWH_A4_0010.pdb rlx_al_3REY_A5_0003.pdb
rlx_al_3RFM_A6_0004.pdb rlx_al_4EIY_rtrim7_0009.pdb -out:path:pdb ./randomize
-nstruct 10 -out:prefix mem_hybridmin_ -run:msd_randomize
```

Contents of ensemble_design.options:

```
-in:file:fullatom
-out:file:fullatom
-linmem_ig 10
-ex1
-ex2
-run:msd_job_dist
-score:weights membrane_highres_Menv_smooth.wts
-membrane:no_interpolate_Mpair
-membrane:Menv_penalties
-fixed_membrane true
-membrane_center -22.751  16.194 -28.472
-membrane_normal -0.708  -0.691   0.145
```

Sampling possible mutations at multiple positions using RosettaMembrane and Rosetta Design:

First, input structures were prepared by minimization using the relax application. Ten models were generated for each PDBID (2YDO, 2YDV, 3EML, 3PWH, 3REY, 3RFM, 4EIY) and the top model was selected for further analysis.

```
/path/to/Rosetta/main/source/bin/relax.default.linuxgccrelease
-s myfile.pdb -in:file:spanfile myfile.span -out:file:fullatom -out:pdb
-out:prefix rlx_ -membrane:no_interpolate_Mpair -membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts -nstruct 10
```

The top model by score of each backbone was then used to setup modeling the mutant. The mutations must be made using a resfile where the points of mutation are allowed to be sampled and sidechains are repacked.

```
/path/to/Rosetta/main/source/bin/fixbb.default.linuxgccrelease
-s rlx_myfile.pdb -in:file:spanfile myfile.span -out:file:fullatom -out:pdb
-out:prefix des_ -membrane:no_interpolate_Mpair -membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts -nstruct 10 -resfile
myfile.resfile
```

Example of a resfile: (NATAA = repack; ALLAAxC = sample all amino acids except cysteine)

```
NATAA
start
23 A ALLAAxC
77 A ALLAAXC
123 A ALLAAxC
180 A ALLAAxC
230 A ALLAAxC
239 A ALLAAxC
```

100 mutation models are created and the top 10 models by score for each PDBID is used for analysis of possible mutations and combinations of mutations at these sites. The sequences were extracted and a sequence logo of the six designed positions for each PDBID was created to visualize which mutations were generally tolerated among the difference starting backbones. These mutants were then modeled using the next protocol.

Modeling defined mutations using RosettaMembrane:

First, input structures were prepared by minimization using the relax application. Ten models were generated for each PDBID (2YDO, 2YDV, 3EML, 3PWH, 3REY, 3RFM, 4EIY) and the top model was selected for further analysis.

```
/path/to/Rosetta/main/source/bin/relax.default.linuxgccrelease
-s myfile.pdb -in:file:spanfile myfile.span -out:file:fullatom -out:pdb
-out:prefix rlx_ -membrane:no_interpolate_Mpair -membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts -nstruct 10
```

The top model by score of each backbone was then used to setup modeling the mutant. The mutations must be made using a resfile where the points of mutation are allowed to be sampled and sidechains are repacked.

```
/path/to/Rosetta/main/source/bin/fixbb.default.linuxgccrelease
-s rlx_myfile.pdb -in:file:spanfile myfile.span -out:file:fullatom -out:pdb
-out:prefix des_ -membrane:no_interpolate_Mpair -membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts -nstruct 10 -resfile
myfile.resfile
```

Example of a resfile: (NATAA = repack; PIKAA = sample only the specified amino acid)

```
NATAA
start
23 A PIKAA A
77 A PIKAA T
123 A PIKAA C
180 A PIKAA G
230 A PIKAA C
239 A PIKAA C
```

Ten models were created for each PDBID backbone. The top model by score was selected for the final minimization step using the relax application.

```
/path/to/Rosetta/main/source/bin/relax.default.linuxgccrelease
-s des_rlx_myfile.pdb -in:file:spanfile myfile.span -out:file:fullatom
-out:pdb -out:prefix csc_rlx_ -membrane:no_interpolate_Mpair
-membrane:Menv_penalties -score:weights membrane_highres_Menv_smooth.wts
-constrain_relax_to_start_coords true
```

For each PDBID, 100 relaxed models were created from the mutant. The top ten models by score were then used for Rosetta energy analysis. Rosetta energy of the mutant models was compared to the wild-type models. The next section covers the protocol for the correct way to model wild-type in parallel with the previous models.

Creating wild-type models in parallel for comparison of energies:

First, input structures were prepared by minimization using the relax application. The same models that were generated from the initial relax step towards building mutant models can be used for the sake of consistency.

The top model by score of each backbone from the relaxation step was then used to setup modeling the wild-type at the ‘mutation’ stage. Since some sampling is performed during the mutation stage for modeling the mutant, we create a resfile that simply repacks all residues and create the same number of models.

```
/path/to/Rosetta/main/source/bin/fixbb.default.linuxgccrelease
-s rlx_myfile.pdb -in:file:spanfile myfile.span -out:file:fullatom -out:pdb
-out:prefix rpk_ -membrane:no_interpolate_Mpair -membrane:Menv_penalties
-score:weights membrane_highres_Menv_smooth.wts -nstruct 10 -resfile
myfile.resfile
```

Example of a resfile for modeling wild-type: (NATAA = repack; PIKAA = sample only the specified amino acid)

```
NATAA
start
```

Ten models were created for each PDBID backbone. The top model by score was selected for the final minimization step using the relax application.

```
/path/to/Rosetta/main/source/bin/relax.default.linuxgccrelease
-s des_rlx_myfile.pdb -in:file:spanfile myfile.span -out:file:fullatom
-out:pdb -out:prefix csc_rlx_ -membrane:no_interpolate_Mpair
-membrane:Menv_penalties -score:weights membrane_highres_Menv_smooth.wts
-constrain_relax_to_start_coords true
```

For each PDBID, 100 relaxed models were created from the mutant. The top ten models by score were then used for Rosetta energy analysis. Rosetta energy of the wild-type models can now be compared to the mutant models.