MODELING INSERTION OF TRANSMEMBRANE BETA-BARRELS FOR DESIGNING OUTER MEMBRANE PROTEINS

By

Daniel Hsieh

A Dissertation submitted to the Graduate School – New Brunswick,
Rutgers, the State University of New Jersey,
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Graduate Program in Computational Biology and Molecular Biophysics
Written under the direction of Dr. Vikas Nanda
and approved by

_____

_____

_____

_____

_____

Piscataway, New Jersey 08854

October 2012

Abstract of the Dissertation

MODELING INSERTION OF TRANSMEMBRANE BETA-BARRELS FOR DESIGNING OUTER MEMBRANE PROTEINS

by DANIEL HSIEH

Dissertation Director:

Dr. Vikas Nanda

Outer membrane proteins (OMPs) perform a range of important functions in the cell biology of Gram-negative bacteria, mitochondria and chloroplasts. These functions include biogenesis, virulence, signal transduction, nutrient transport and apoptosis. In contrast to their inner membrane counterparts, OMPs have been more difficult to study due to the relative paucity of crystal structures.

Although outer membrane proteins have been characterized and studied extensively by various structural and biophysical methods, our understanding of their folding, insertion and oligomerization is far behind that of inner membrane proteins. The goal of this study is to elucidate the folding and insertion mechanism of these transmembrane β-barrel proteins (TMBs) and ultimately to provide guidelines for computationally designing OMP sequences that fold and insert efficiently. Using a subset of amino acids from thirty-five outer membrane proteins from Gram-negative bacteria and mitochondria, a propensity vs depth in membrane profile for each residue was derived. Although results indicate similar trends between amino acids of inner and outer membrane proteins, there are also differences that can be explained by differences in factors such as environment, secondary structural preferences, and folding/insertion pathway.

The propensity profiles were converted into energies of insertion as a function of depth from the center of the membrane. This thesis explores the many ways of using the statistical potential to answer questions about OMP folding, insertion and oligomerization that could not have been framed due to the experimental limitations. We conclude with a discussion of ways to improve our potential, including the assumption of asymmetry of the lipid bilayer as well as incorporating homology model building.

Acknowledgements

I would like to first acknowledge my family: Hsiao-Chiu, my hardworking mother who let me freely explore my interests, from music to mathematics to the biological sciences; Ching-Hwa, my father who taught me the virtue of persistence; my sister Jackie, an extremely talented aspiring doctor who taught me her passion for service and selflessness; and the love of my life, Xue, for her unwavering support all these years we have known each other. I am especially amazed at your energy of being a graduate student as well as a full time employee. I thank you all for your unconditional love and appreciation in all my endeavors. This work and my transformation are dedicated to you.

I would like to express my sincere gratitude to my advisor, Dr. Vikas Nanda, for it is his patience, understanding, and guidance throughout my graduate career that has been transforming me into a mature scientist. I am continuing to learn a great deal from this extraordinary scientist, including effective leadership, professionalism, and most of all, his passion for science. I also wish to thank my friends and co-workers, old and new, who offered their advice, expertise and friendship. Kenneth, Sumana, Mihir, Doug and Teresita: I wish you all a road of excellence in your future endeavors.

I consider the post-docs of the Nanda lab as excellent role models. Dr. Fei Xu, Dr. James A Stapleton, Dr. Isaac John Khan, Dr. Agustina Rodriguez-Granillo, Dr. Avanish Parmar and Dr. Srinivas Annavarapu and their advice on pragmatism kept me on track towards completing a Ph.D. I also want to thank Alexander Davis, an incredibly intelligent and diligent undergraduate member of our lab who is interested in mathematical biology, for the wonderful discussions about science, math and music we will continue to have and for his contagious passion for learning.

I would like to thank my committee members, Dr. Ann Stock, Dr. Alexandre Morozov, and Dr. Ron Levy, for their career advice and encouragement on my work. I am grateful to Dr. Gregory Caputo for providing me a hands-on opportunity to conduct membrane protein experiments and to Dr. Alan Finkelstein for introducing me to his experimental setup of measuring conductance of ion channel proteins.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

OM	Outer membrane
OMP	Outer membrane protein
TMB	Transmembrane β-barrel
TMH	Transmembrane α-helices
TM	Transmembrane
VDW	Van der Waals
BAM	β-barrel assembly machinery
POTRA	Polypeptide transported-associated domain
CHAMP	Computed helical anti-membrane protein
SASA	Solvent-accessible surface area
MASA	Maximum accessible surface area
LPS	Lipopolysaccharides
ProtCAD	Protein computer-aided design
VBA	Visual Basic for Applications
PPI	Protein-protein interface
DSSP	Dictionary of Secondary Structure Prediction
MSA	Multiple Sequence Alignment

1. Introduction

Protein design is a field that seeks to address the protein folding problem: how does an unstructured polypeptide attain its final three-dimensional conformation? A protein designer could either screen combinatorial libraries for designs with desired properties or rationally design the sequence with predicted structural and functional characteristics. The latter strategy is useful because when a designed protein fails, we can question the fundamental knowledge with which we used to design the proteins.

Where there are cells, there exist proteins found lodged directly in the cell membrane that act as receptors, signal transducers and nutrient and waste management, and more generally, as an interface between the outside world and the components within the cell. In this thesis we study a subclass of these membrane proteins called outer membrane proteins (OMPs) for which limited but growing structural information exists. There has been extraordinary progress in obtaining crystal structures of these membrane proteins, understanding function and their folding and assembly pathway, prediction of secondary structure, topology and recently, oligomeric (Naveed et al. 2009) and exposure status (Park et al. 2007; Hayat et al. 2011). However, there are still some grey areas regarding biogenesis (assembly, targeting, insertion and folding) to address (Rigel et al. 2011) before we can carry out *de novo* design of outer membrane proteins that properly fold and insert. For this thesis, we seek to determine the rules behind insertion of this class of proteins. They are found in pathogenic bacteria that have an outer membrane (in addition to their plasma membrane) and therefore extra defense mechanisms against all kinds of drugs scientists have painstakingly developed. Bacteria with outer membranes are Gram-negative, while those with only a plasma membrane are Gram-positive. Through evolutionary divergence, these proteins and the outer membrane are also found in chloroplasts and mitochondriae and fulfill non-pathogenic functions. The rules we

have developed and the insights gained from them can be used to predict orientation and protein-protein interfaces and bring us closer to our goal of *de novo* design of outer membrane proteins.

All of the structural bioinformatics work on outer membrane proteins was carried out on Microsoft Excel, a spreadsheet software that is commonly found in academic institutions but that is underused in scientific practice in comparison to financial analysis. The work is automated using the Visual Basic for Applications (VBA) programming language. Excel VBA is a powerful tool and most commonly used in financial analysis because of its compatibility with users of a wide skill range. Automation unlocks Excel's potential, especially in the areas of visualization. Chapter 5 will demonstrate Excel's usefulness in parsing structures of membrane proteins, performing intricate computational experiments using the rules we have derived, building visualization tools to assess their energies of insertion in a visually comprehendible manner. Chapter 4 features a sample Excel add-in tool built by Microsoft Research that helped in building homology models for expanding our knowledge in improving our rules of inserting outer membrane proteins. Chapter 8 contains major excerpts of the code I used to develop these tools. Automating these tools unlocks the full potential of Excel's computational capacity, rendering it another avenue for solving problems in computational biology and bioinformatics.

1.1 The Bacterial Outer Membrane

1.1.1 Composition and Function

Cells are defined by biological membranes that help compartmentalize organelles and selectively determine the biomolecules that can or cannot enter the cytoplasm. Those biomolecules that can enter are nutrients, signals or forms of energy for the cell; those that are prohibited are usually registered as harmful by components of the cell. Biomembranes thus play a major role in the survival of bacteria. In addition to having an inner plasma membrane, Gram-negative bacteria are also surrounded by an outer membrane of a different composition. Both types of membrane are composed of inner and outer leaflets. While both leaflets of the inner membrane (IM) are composed of phospholipids such as cardiolipin, phosphatidylethanolamine and phosphatidylglycerol, (Raetz et al. 1990) the outer membrane (OM) is asymmetric, consisting of the same phospholipid composition in the inner leaflet, along with lipopolysaccharides (LPS) on its outer leaflet (Kamio and Nikaido 1976), rendering itself impermeable to harmful compounds such as antibiotics.

From the center of the bilayer and outwards, the LPS outer leaflet begins with Lipid A, a hydrophobic membrane anchor which is responsible for triggering a heavy immune response. The Lipid A domain is then extended by a core domain of oligosaccharides called the inner and outer cores. The inner core saccharides are negatively charged (Fig. 1).

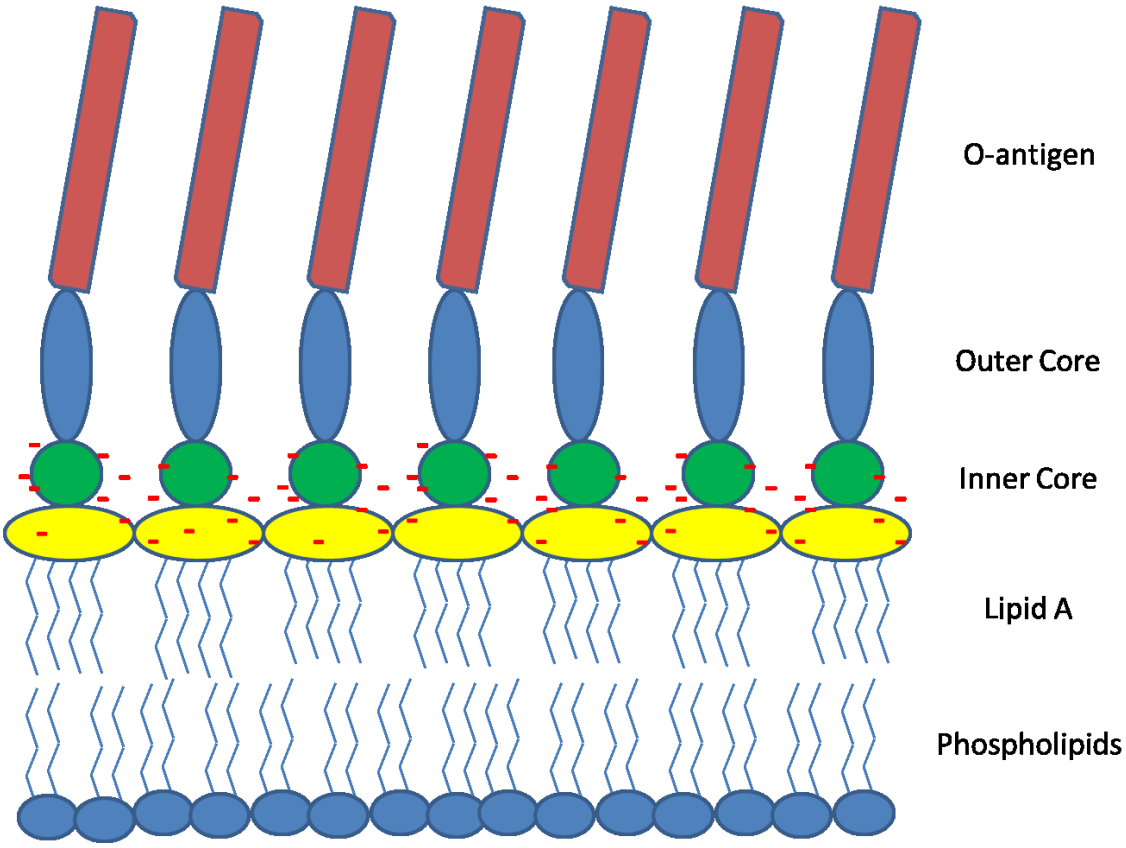


Figure 1 - Composition of the outer membrane

Phospholipids compose the inner leaflet of the outer membrane. Lipid A, inner and outer core, and O-antigen compose its outer leaflet. Lipid A and the inner core bear negative charges.

Depending on the species of Gram-negative bacteria, further linkage is established between either the inner (Pieretti et al. 2012) or outer core domain and a lengthy polysaccharide called O-antigen. X-ray diffraction shows the total thickness of the LPS outer leaflet, including lipid A, core and O-antigen combined is about 5.6 nm, whereas lipid A itself measured approximately 2.6nm (Labischinski et al. 1985). Lipid A is an endotoxin because once it is identified by the immune system, its response escalates, an event which triggers septic (toxic) shock as well as a fatal overproduction of clotting factors.

1.1.2 Modeling the Lipid Bilayer

Modeling the lipid bilayer is difficult because of the constraints of computational power used in simulations involving membranes. The most costly type of simulation would be an all-atom one, followed by course-grained modeling (Bennun et al. 2009), where the amount of course-graining macromolecules can now be determined computationally (Wang and Cheung 2012). Another way to course-grain the simulation of lipids is through an implicit solvent or water-free model, where a potential of mean force captures the behavior of solute-solvent interactions (Lazaridis 2003).

Because the lipid bilayer has shown dynamic and material properties, it is also possible to study these characteristics of the membrane. The fluidity of the lipids (Lenaz 1987), or viscosity, shows that lipids may distort to match the hydrophobicity of the membrane protein (Mitra et al. 2004; Engelman 2005). Elasticity is another well-studied parameter in modeling lipid bilayers (Andersen and Koeppe 2007) because deformations may cause hydrophobic mismatch between bilayer thickness and the TM region of the membrane proteins (Ellena et al. 2011) and even induce curvature, which all affect the folding and insertion of membrane proteins (Hong and Tamm 2004; Burgess et al. 2008).

Aside from dynamics and material properties of the fluid bilayers, X-ray and neutron diffraction studies suggest a model of the bilayer of Gaussian type distributions of lipid components (Wiener and White 1991). This water-lipid continuum model (Fig. 2) has been used to study (Senes et al. 2007) and design (Yin et al. 2007) depth-dependent insertion and orientation of inner membrane helical proteins (TMHs) that target and compete for TMH dimerization. This model is the basis of a similar approach we are taking in order to study outer membrane proteins.

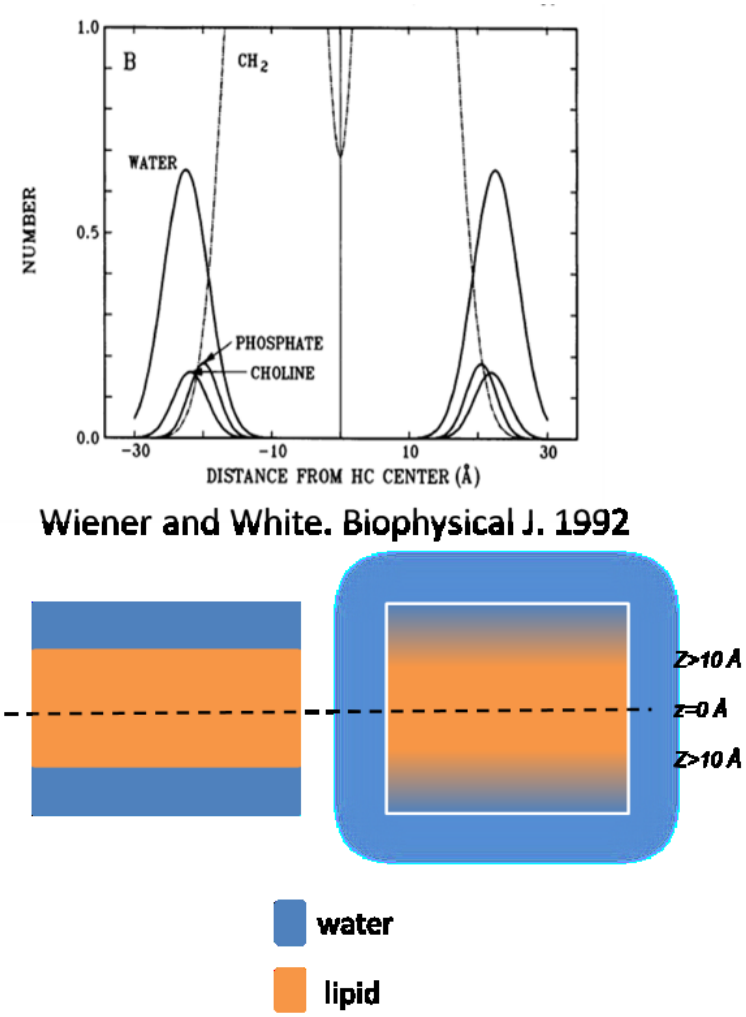


Figure 2 - Lipid Bilayer as a Continuum Model

Top: Distributions of major components of phospholipids distributed along the depth of the lipid bilayer, as determined by X-ray and neutron scattering.

Bottom left: Model of lipid bilayer as a hydrophobic slab and (bottom right) as a continuum from water (blue) to lipid (light tan)

1.2 Outer Membrane Proteins

The membrane proteins of Gram-negative bacteria, chloroplasts and mitochondria are found in two different groups: inner membrane proteins, which are α-helical, and the outer membrane proteins which are exclusively β-barrels. Our current biophysical understanding of membrane protein folding and function lags significantly behind that of water-soluble proteins. Such a gap is even more evident for outer membrane proteins. As of 2009, there were around 20 structures of β-barrel membrane proteins (Meng et al. 2009) deposited into the Orientations of Proteins in Membranes database (A.L. Lomize et al. 2006); there are currently a few more than 100 crystal structures available, yet so many of them are structurally similar (Jiménez-Morales et al. 2008). These TM β-barrel (TMB) proteins fulfill many important functions from nutrient uptake to cell signaling to virulence, in Gram-negative bacteria, mitochondria (Schein et al. 1976) and chloroplasts (Schleiff et al. 2003) and are thus studying their structure and function is highly medically pertinent. This section will review the structure, function and biogenesis of TMBs and discuss possible applications in outer membrane protein design.

1.2.1 A Brief History of Outer Membrane Proteins

The general diffusion porin is one of the first major proteins identified in the outer membrane of *Escherichia coli*. This name "porin" was first coined (Pugsley and Schnaitman 1978) when it was found that, upon introducing this class of "matrix proteins" into membranes, solutes as large as 600 Da were able to penetrate through the membranes. Three of the earliest general diffusion porins identified are OmpC, OmpF and PhoE (Inokuchi et al., 1982; Mizuno et al., 1983; Overbeeke et al., 1983). Trypsin digestion experiments have shown the OM to be densely populated by porins (Braun and Rehn 1969; Rosenbusch 1974), on the order of 10^5 monomers per cell. Another one of the major outer membrane proteins, OmpA, has multiple purposes such as being a phage recognition site by bacteriophages (Morona et al. 1984), conjugation (Schweizer and Henning 1977; Ried and Henning 1987), and maintaining structure of the outer membrane (Sonntag et al. 1978).

Studies from different species *Salmonella* and *E. coli* showed that porins have quite polar composition despite being situtated in membrane (Tokunaga et al. 1979; Rosenbusch et al. 1974). Furthermore, proteins in the outer membrane with diffusive properties that could accommodate solutes like amino acids and sugars have been reported (Heuzenroeder et al. 1981; Ishi et al. 1981). Diffuse X-ray diffraction work from Kleffel, Rosenbusch and co-workers indicated the secondary structure of porins in *E. coli* consisted mainly of antiparallel β-sheets (Kleffel et al. 1985).

1.2.2 Structure and Function of Outer Membrane Proteins

Outer membrane proteins (OMPs) are almost exclusively of β-barrel topology. β-barrels are formed by wrapping many β-strands around to form a cylinder. In the context of TM β-barrels (TMBs), their constituent β-strands span the membrane, with the exception of the α-helical lipoprotein WzA (Dong et al. 2006). There are new observations ever since Georg Schulz first described a set of rules about TMB structures:

"1) All β-strands are antiparallel and locally connected to their next neighbors

2) Both the N- and C- termini are at the periplasmic [edge of the] barrel restricting the strand number n to even values

3) Upon trimerization, a nonpolar core is formed at the molecular threefold axis of the porins so that the central part of the trimer resembles a water-soluble protein

4) The external β-strand connections are long loops named L1, L2, etc., whereas the periplasmic strand connections are generally minimum-length turns named T1, T2, etc.

5) [Unrolling] the barrel […] and placing the periplasmic end at the bottom, the chain runs from right to the left.

6) In all porins, the constriction at the barrel center is formed by an inserted long loop L3.

7) The β-barrel surface contacting the nonpolar membrane interior is coated with aliphatic side chains forming a nonpolar ribbon. The two rims of this ribbon are lined by girdles of aromatic side chains.

8) The sequence variability in TMBs is higher than in water-soluble proteins and exceptionally high in the external loops" (Schulz 2002).

The first rule describes the observation of a simpler up-down topology where neighboring β-strands are antiparallel than other topologies found in β-barrel structures such as the Greek motif and the topology of green fluorescent protein (Jackson, Craggs, and Huang 2006). TMBs consist of 8 to 24 β-strands, with the number of strands being even. However, solution NMR (Hiller and Wagner 2009) and X-ray crystallography revealed the human voltage-dependent anion channel (VDAC) to possess 19-strands while maintaining the up-down topology. This breaks the second rule of the TMB having an even number of strands. Schulz also noted that N- and C- termini are coincidental at the periplasmic region. Because the number of strands, *n* for VDAC is 19, strands 1 and 19 are parallel instead of antiparallel. However, the N-terminal helix of VDAC is still found closer to the periplasmic side of the barrel and coinciding with the C-terminal strand 19.

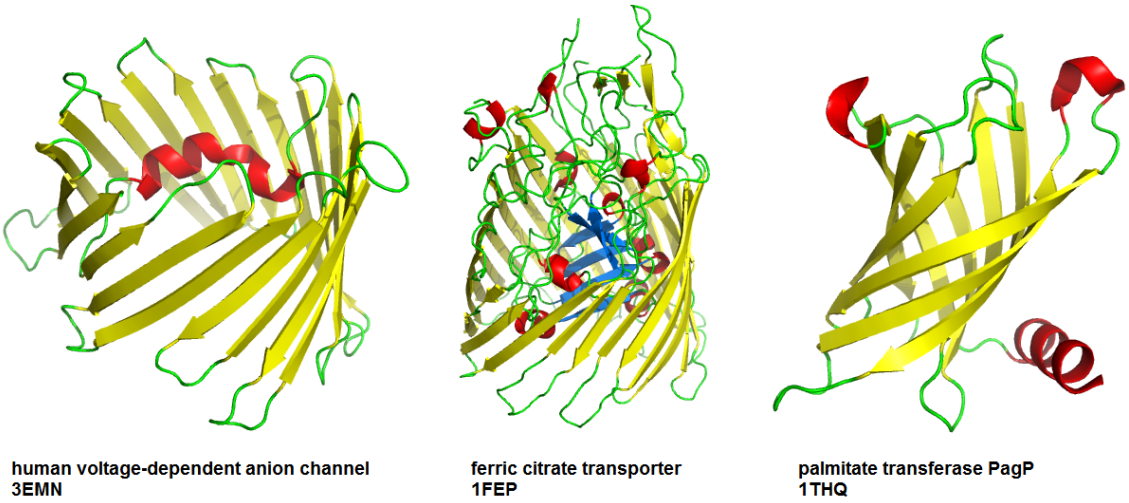


Figure 3 - TMBs with secondary structural stabilizing regions

Constriction loops (red, shown on left), in-plugs (blue, shown on middle) and out-clamps (red, shown on right) highlighted

Extracellular loops have a tendency to be longer and more flexible than periplasmic loops for many functional reasons. A group of small (8-stranded) TMBs called adhesins extend their extracellular loops in -sheet conformation to offer non-specific binding regions (Vogt and Schulz 1999). In larger TMBs such as trimeric porins, one of the extracellular loops called the "constriction" loop L3 folds back into the barrel while another, indexed as L2, "latches" onto a neighboring barrel lumen (Phale et al. 1998). The existence of short periplasmic and long variable extracellular loops is likely the result of the evolutionary amplification of short ββ hairpins (Remmert et al. 2010).

In addition to extracellular and intracellular loops, there exist other secondary structures inside and outside of the TMB that provide added stability and function. For example, large TMBs such as FecA transport metals like iron as well as larger molecules such as vitamin B and carbohydrates. The transport process is selective and requires energy from a cascade of signals transduced from a complex in the inner membrane (Ferguson et al. 2002). Such transporters have secondary structure rich components situated within the barrel called plug domains, which undergo conformational changes upon binding with the appropriate signal (Carter et al. 2006). Another example is PagP, an enzyme that assists in reinforcing E coli outer membrane defense. PagP is stabilized by a post-assembly N-terminal helix that packs against an exterior side of the TM region. Despite this amphipathic helix's added stability to PagP, deletion of that region did not hinder the protein's ability to fold and insert (Huysmans et al. 2007). Together, these in-plugs and out-clamps (Fig. 3) have been shown to provide add stability to "weakly stable" regions of the TMB (Naveed et al. 2009).

Due to the antiparallel up-down motif of β-strands in TMBs, Schulz implies the β-strands run in an anti-clockwise direction. As shown in Figure 4A, there is a network of hydrogen bonds between the backbone amide nitrogen and carbonyl oxygen atoms. Figure 4B shows a hypothetical unrolled β-barrel where all of its β-strands lay on a plane and remain hydrogen-bonded with each other. Hydrogen bonds are represented by dashed lines in both parts of the figure.

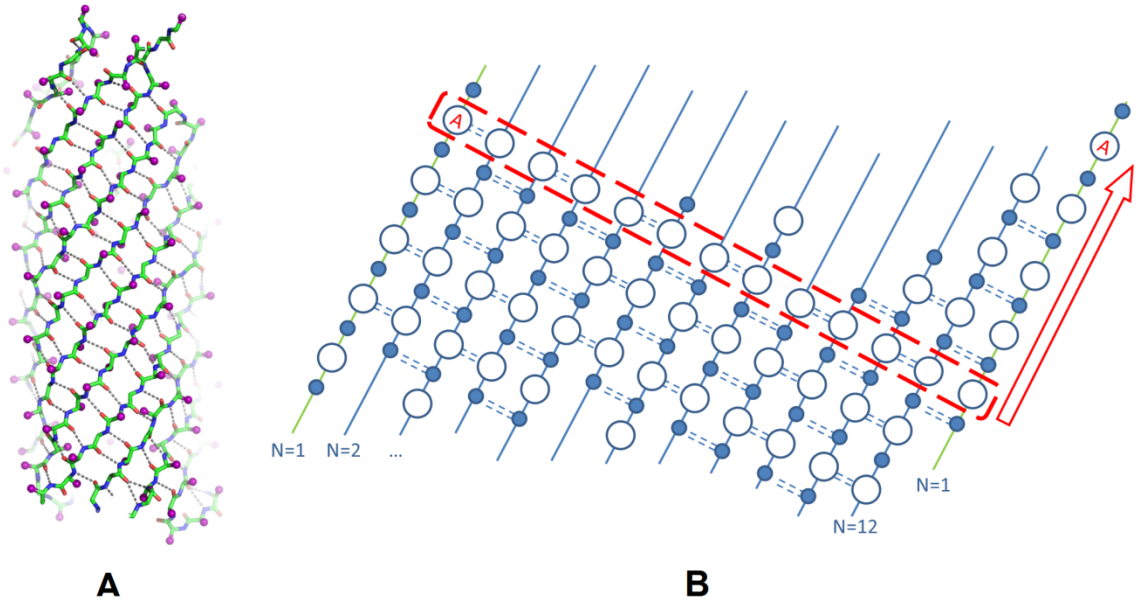


Figure 4 - Backbone hydrogen bond networks

A) Backbone hydrogen bond network of autotransporter NalP, view facing extracellular cap from side. PDB ID 1UYN **(Oomen et al. 2004)**. Atoms shown in purple spheres are Cαs of the polypeptide sequence. B) Hypothetical schematic of TM strands of TMB of n = 12 strands with a shear number S = 8 as adapted from Murzin et al 2004 **(Murzin et al. 1994)**. Each node, small filled and large open circles represent the alternating interior and exterior facing Cαs along the β-barrel structure. To count the shear number of the β-barrel, choose a residue on a starting strand, traverse along the hydrogen bond network until the start strand is reached. The shear is the difference between your residue position before and after interstrand traversing.

This regularity in the hydrogen bond network allows us to formulate a parameterization of an unrolled β-pleated sheet for identifying and predicting the three-dimensional coordinate along the β-barrel structure. Let the intrastrand distance between two consecutive residues be denoted *a* and the interstrand distance between any two residues connected by a hydrogen bond be *b*. The parameters *a* and *b* are given as 3.3Å and 4.4 Å, respectively. Optimal values for these parameters have been determined as *a* = 3.48Å and *b* = 4.83Å (Reboul et al. 2012). Another important parameter is the shear number *s*, which signifies the amount of staggering the strands have with respect to each other given the hydrogen bond patterning (Fig. 4b). The relationship between the radius of the β-barrel *R*, the number of strands *n*, the shear number *s*, and the two distance parameters can be described by the circumference of the cylindrical base of the β-barrel being equivalent to the hypotenuse of a right triangle whose sides are given by the orthogonal intra- and interstrand distances *Sa* and *nb:*

$$2\pi R = \sqrt{(Sa)^2 + (nb)^2} \tag{1}$$

The angle α between the strand and the barrel axis is therefore:

$$\tan\alpha = \frac{Sa}{nb} \tag{2}$$

The seventh rule describes general motifs. Aromatic residues are well known to contribute to the folding and insertion of TMBs, which will be discussed in the next section. They are usually found at the water-lipid interface and lined up as a girdle around the rim of the barrel structure (Fig. 5). Pairwise motifs/antimotifs involving aromatics on neighboring strands found by the Liang lab (Jackups and Liang 2005; Jackups et al. 2006), strand registration dependent rotameric conformation of tyrosine, and aromatics involved in chaperone-binding motifs (Bitto and McKay 2003) all add to the complexity of predicting TMB attributes.

While membrane proteins have been generalized as having almost alternating polarities between buried and lipid-exposed residues (Rees et al. 1989), scanning the hydrophobicities along the sequence of oligomeric proteins lead to a slight complication: The interior residues can be either hydrophobic or hydrophilic (Schulz 2002), and the interfacial regions of these oligomers are where the average hydrophobicity values drop (Seshadri et al. 1998), implying polar residues at these sites may be functionally important.

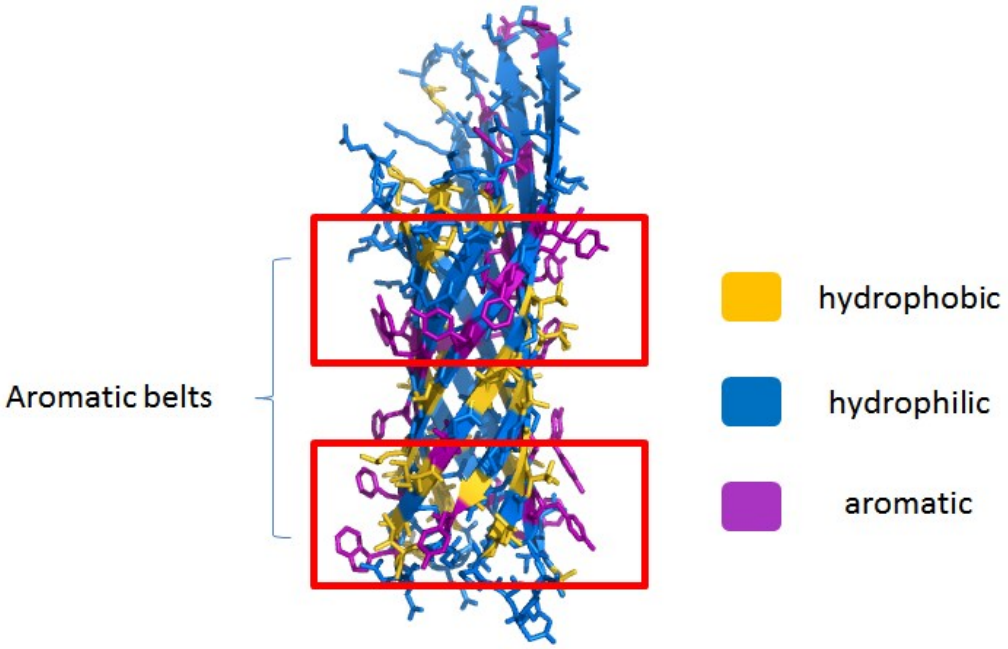


Figure 5 - Lipid-facing aromatic residues form girdles on TMBs

The "positive-outside" rule was recently posed, where the basic residues in the extracellular cap regions have more than twice the propensity than in the periplasmic cap region. On the contrary, acidic residues such as Asp and Glu seem to have a reversed preference for the periplasmic over extracellular region (Jackups and Liang 2005). They reason this asymmetric behavior is due to the different compositions of the inner and outer leaflet of the outer membrane. The outer leaflet is composed of LPS, which have negatively charged components whereas the inner leaflet contains primarily phosphatidylethanolamine. Early attraction between the negatively charged LPS and the positively charged residues (Tamm et al. 2004) of the extracellular cap region of TMBs may be responsible for proper orientation of TMBs.

Also, certain amino acids tend to form β-sheets and some prevent its formation. Through a mutagenesis study on a globular β-sheet protein, the residues with the highest ΔΔG (in kcal/mol) relative to alanine were threonine and isoleucine (Minor and Kim 1994). Because threonine is polar, the hydrophobic isoleucine is expected to be in the lipid-facing TM region of OMPs. Glycine, on the other hand, has been shown to have low preference at β-sheet regions (Minor Jr. and Kim 1994). At the interior of TMBs, this anti-motif can be "rescued" (Merkel and Regan 1998) by nearby tyrosines because glycine may be useful in relieving curvature stress on the β-barrel (Jackups and Liang 2005).

The eighth rule implies that fold space is very limited yet allows for such sequence diversity amongst TMBs. A randomization mutagenesis study by Koebnik (Koebnik 1999) showed that residues on strand 8 facing the interior of TMBs can tolerate changes without affecting the assembly of the protein. Randomizing the lipid-facing residues on strands 4, 6 and 8, however, did not yield as much phage sensitivity. As the reader will soon see, we will describe a computational sequence randomization experiment in section 2.2.5 of chapter 2.

1.2.3 Geometry of TM and water-soluble β-barrels

De novo protein design begins with understanding local and macroscopic geometric properties of the protein. Water-soluble β-barrel proteins such as TIM-barrels, green fluorescent protein, and lipocalin (Fig. 6) have been parameterized (Fig. 7) by equations describing surfaces such as catenoids (Koh and Kim 2005), elliptical cylinders, and the twisted hyperboloid of one sheet (Novotny et al. 1984, Lasters et al. 1988; Stec and Kreinovich 2005; Nava and Kreinovich 2012). The structures of some outer membrane proteins (OMPs) need to accommodate their function of being selectively porous.

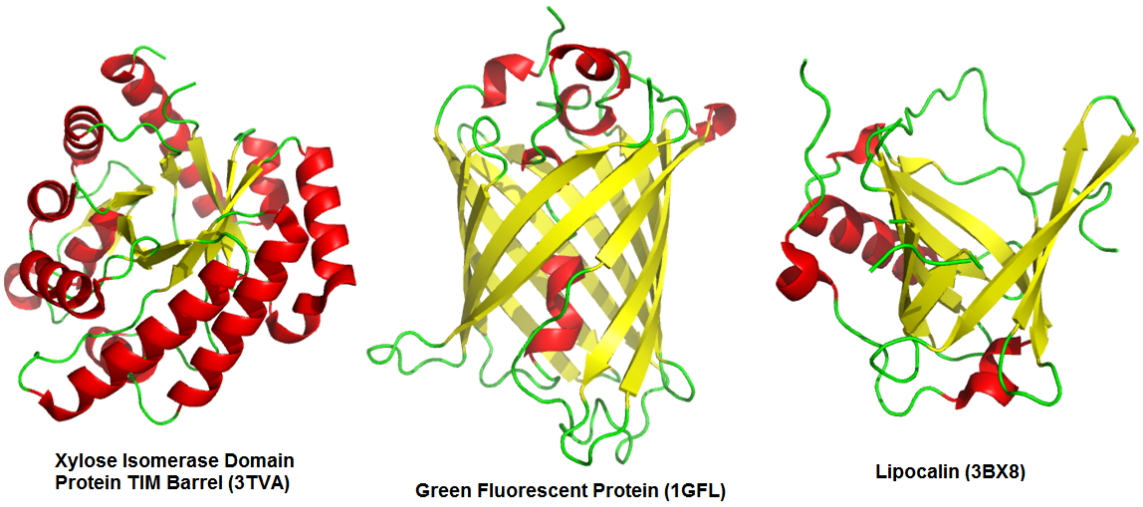


Figure 6 - Water-soluble β-barrels

The cylindrical β-barrel (McLachlan et al. 1979), in which the β-strands and their hydrogen bonds run approximately along a cylindrical surface, is a biophysically more feasible structure because internal plug sites in some OMPs and permeation of solutes introduce a VDW force on the interior (Gu and Li 1999), electrostatic interactions impart additional stability of the barrel (Irbäck and Mitternacht 2008), protein-lipid interactions may affect the barrel structure (Botelho et al. 2006) and the hydrophobic effect and tight packing within water-soluble β-barrels. According to a principle component analysis study on flexibility of β-sheets, it is found that bending is not the dominant mode and antiparallel sheets are less rigid than parallel ones possibly because of the length of the connecting loops and the difference in hydrogen bonding network (Ho and Curmi 2002; Emberly et al. 2004).

Naveed and coworkers have used this geometrical model in addition to strand registration prediction methods (Naveed et al. 2009) to generate accurate structures of β-barrel membrane proteins (Naveed et al. 2011) as well as predict contact sites (Geula et al. 2012). They do concede that accounting for lipid-protein interactions (Botelho et al. 2006; Adamian et al. 2011) may improve prediction of membrane protein structure. Chapter 2 of this thesis describes an ongoing effort in understanding oligomerization of porins using a different biophysical approach.

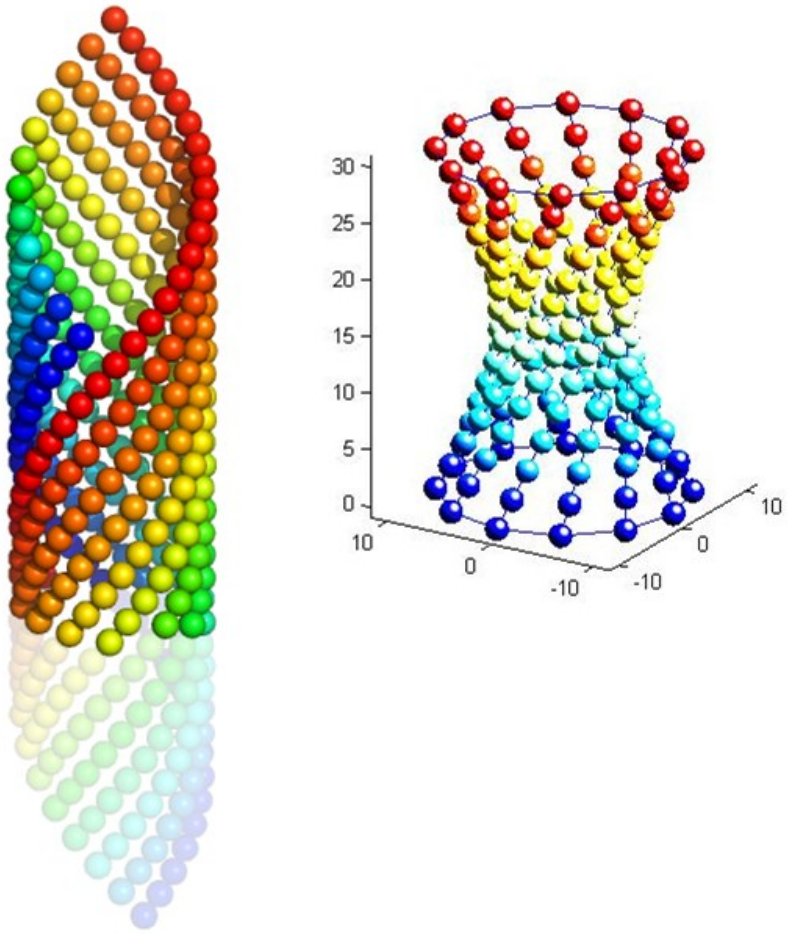


Figure 7 - McLachlan vs hyperbolic ruled surface models of the β-barrel.

(Left) A β-barrel can be generated given three arguments: the number of strands, shear number, and length of one strand (McLachlan 1979, Murzin et al. 1994). Code provided in Appendix 7.1.

(Right) A theoretical model of a β-barrel in which the strands are fully rigid and straight could be parameterized as a ruled surface function of base polygon, length and number of strands, and twist angle. Code provided in Appendix 7.2.

1.2.4 Assembly and biogenesis of outer membrane proteins

Outer membrane protein synthesis *in vivo* is followed by post-translational targeting to the secretion machinery Sec by the cytoplasmic chaperone SecB. After transiting Sec, their signal sequences are cleaved by a signal peptidase (Zwizinski and Wickner 1980), are received by binding to periplasmic chaperones to the β-barrel assembly machinery BAM. The major periplasmic chaperone of OMPs is the SurA, whereas the alternative pathway involves two chaperones, Skp and DegP (Rizzitello et al. 2001; Sklar et al. 2007). Such redundancy in function explains why any single gene deletion mutant yields only lower but not fully degenerate OMP levels (Sklar et al. 2007). As BAM is a complex, consisting of one TM β-barrel protein, a polypeptide transport-associated domain abbreviated POTRA, and four other lipoproteins BamB, C, D, and E which extend into the periplasmic space (See the work of KH Kim et al for crystal structures). Recently, a minimal *in vitro* system that was constructed from subsets of these components and SurA, successfully folded and inserted OmpT, an outer membrane β-barrel protease (Hagan et al. 2011). While we still have limited understanding of the folding and insertion of OMPs, this work further suggests no additional input of energy is required for the β-barrel assembly process. Furthermore, BAM can recognize the OMPs of mitochondrial outer membranes, and similarly, the assembly machinery of mitochondria can recognize bacterial OMPs (Walther et al. 2009; Walther et al. 2009; Tommassen 2010), suggesting sequence and/or structural homology between the two classes of OMPs.

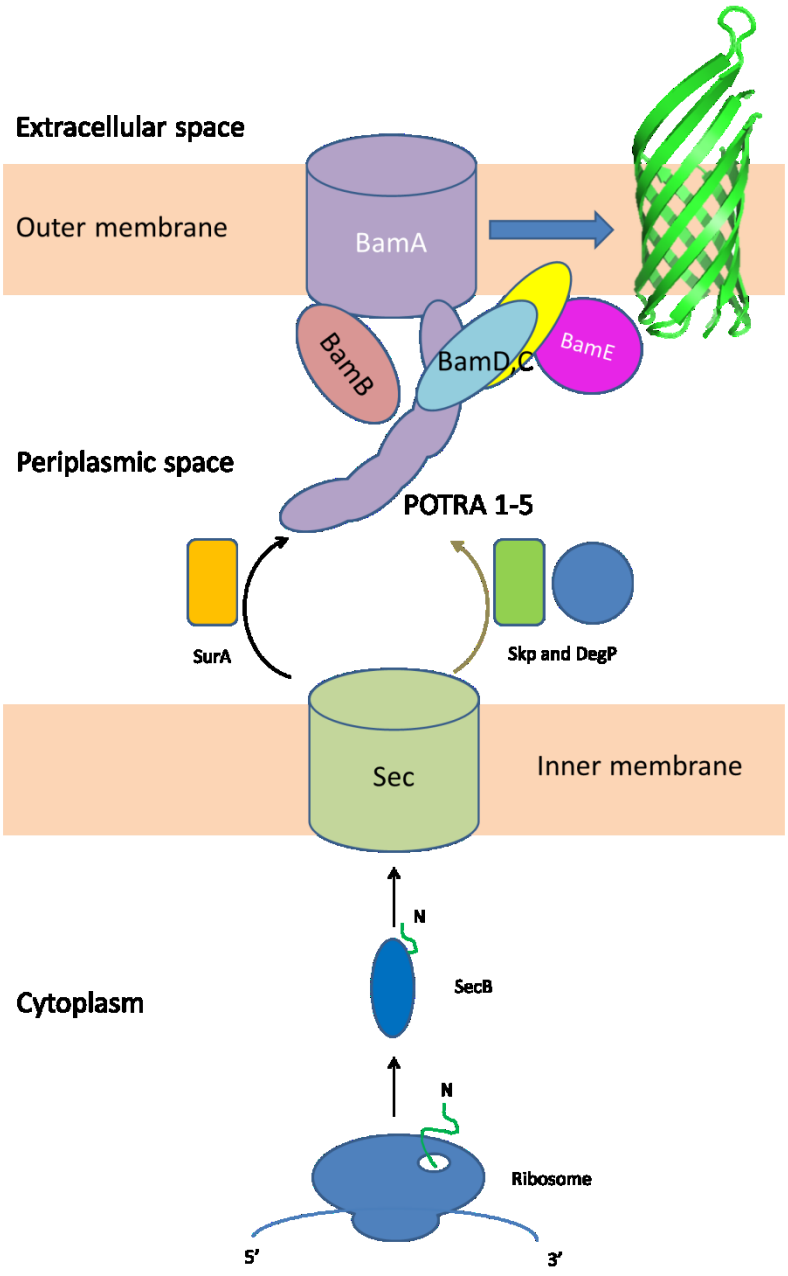


Figure 8 - The Folding/Insertion Pathway of TMBs

Modified from **(Hagan et al. 2011)**. From cytoplasm to extracellular space: TMB sequence is post-translationally translocated through the inner membrane Sec machinery via the cytoplasmic chaperone SecB upon recognition and cleavage of N-terminal signal sequence. SurA is the major periplasmic chaperone that assists in the

transit of the OMP into BAM complex (BamA, B, C, D, E) for folding and insertion into the outer membrane.

1.2.5 Folding and insertion of TMHs differ from that of TMBs

Popot and Engelman's work with the bacteriorhodopsin, a TMH, laid the foundation of studying insertion of membrane proteins by suggesting a two-stage folding model (Popot and Engelman 1990). In the first stage, individually stable helices in the bilayer are formed because secondary structure formation via hydrogen bonding contributes to more stability, especially in medium of low dielectric constant (Engelman et al. 1986). Because hydrophobic residues tend to appear more frequently in transbilayer regions (Wallin et al. 1997), the hydrophobic effect further drives a TMH away from interacting with water and into the membrane environment. The free energy cost of TMH insertion is around 30-40 kcals/mol per helix (Engelman, Steitz, and Goldman 1986). The formation of hydrogen bonds adds 50-70 kcals/mol of stability. In the second stage, dimerization of the helices in membrane is driven by helix-specific motifs (Senes et al. 2004; Walters and DeGrado 2006), weak hydrogen bond interactions (Senes et al. 2001) and conserved polar residues (Dawson et al. 2003).

Secondary structure context may also influence the evolutionary selection and placement of residues in TMHs and TMBs. For example, Lys-flanked poly-Leu TM peptide sequences have been shown to form TMHs (Davis et al. 1983; Huschilt et al. 1989; Zhang et al. 1992). Extending and curtailing TMH sequences have influence on the re-orientation as well as kinking of the TMHs due to hydrophobic mismatch (Cordes et al. 2002; Caputo et al 2003; Anderson and Koeppe 2007), whereas introducing hydrophobic mismatch (Mouritsen et al. 1984) and plying apart β-sheets through proline substitutions is less forgiving to TMB folding and insertion (Yohannan et al. 2004). Furthermore, the repelling of flanked Lys residues on poly-Leu TM peptides promotes oligomerization of TMHs (Lew et al. 2000). Long polar residues are

capable of snorkeling (Chamberlain et al. 2004), a property which their aliphatic moieties of their sidechains can bury themselves to allow the polar portion to interact with the headgroup region of the bilayer. Amino acid preferences for α-helices and β-barrels secondary (Blaber et al. 1993; Minor Jr. and Kim 1994) and tertiary structures (Seshadri et al. 1998; Senes et al. 2004) are also different.

The insertion for outer membrane proteins, which are β-barrel, is also purported to be different due to secondary structure effect on the geometry of intermolecular forces. Insertion of individual β-strands is not favorable because the hydrogen bond network forms in an interstrand rather than an intrastrand fashion. Instead, studies with the OmpA support a concerted insertion and folding model (Kleinschmidt 2006). This model does not rule out the possibility of the association of fragments of TMBs upon insertion (Debnath et al. 2010), a feat which a heptameric β-barrel toxin, α-hemolysin, (Gouaux, Hobaugh, and Song 1997) is able to achieve.

1.2.6 Motivation and tools behind studying folding, insertion, and oligomerization of TMBs

In order to approach the level of *de novo* designability of TMBs, we need to understand TMB folding and insertion processes, a subject in membrane protein science that lacks clear detail. Certain general properties of membrane proteins allow us to study the biophysics of their folding and insertion. The work of Hessa and coworkers first provided major headway in our understanding of insertion of membrane proteins: the apparent free energies of inserting a peptide with a combination of leucines vary with the depth of their placement and is additive (Hessa et al. 2005; Jaud et al. 2009). Combined with the observation that the membrane is not a binary slab, but rather a water-to-lipid continuum, this leads to modeling insertion energetics more as a gradient of insertion energies due to depth-dependent preferences of individual residues along the normal axis to the lipid bilayer (Senes et al. 2007). We shall discuss the inner membrane protein depth-dependent potential in

Chapter 2. The distributions of amino acids in TMBs are crucial in understanding TMB folding and insertion. As mentioned before, aromatic residues line up around the rim of the TMBs forming a girdle at the headgroup region. Unlike TMHs, phenylalanine shows aromatic more so than hydrophobic behavior. This phenomenon may be due to the added stability through π-stacking interactions in large TMBs, whereas clustered aromatic residues are more suited for helix packing in TMHs (Hong et al. 2007).

There are properties of TMBs that help us study their specific folding and insertion behavior. Aromatics are also useful in probing TMB folding due to their ability to absorb UV light. Kleinschmidt and others studied OmpA using tryptophan fluorescence. They found that OmpA appears to have folding intermediates (Surrey and Jähnig 1995) that have not been incorporated into the lipid bilayer when the folding proceeds in a low-temperature setting (below ~10°C). Also, *in vitro* folding and insertion begins with the anchoring of key tryptophans while other tryptophans transition from one leaflet of the lipid bilayer to form stable TM β-strands (Kleinschmidt and Tamm 1996). Using fluorescence quenching of tryptophans by neighboring amino acids mutated to spin-labeled cysteines, they also determined that strands 1, 2, and 8 of OmpA associate to finalize the wrapping of the barrel (Kleinschmidt et al. 2011). Another property is the ability for the TMBs to refold upon diluting denaturant (urea) concentration (Bowie 2004). This property was first discovered by Surrey and Jahnig (Surrey and Jähnig 1992).

TMBs require proper folding and insertion in order to perform their function. Phage recognition is one tool for determining proper orientation of the OmpAs by binding to their extracellular loops (Morona et al. 1984). Some TMBs have enzymatic capabilities, and one such example is the OmpT. This is quite useful information because one can study insertion of TMBs by using the OmpT for its ability to cleave peptides between two consecutive basic residues and such activity can be monitored by fluorescence. Hagan et al used this OmpT folding/insertion model to identify key components of the BAM machinery (Hagan et al. 2010).

1.3 Potential Benefits of Redesigning OMPs

Because OMPs participate in a wide range of functions including bridging signaling pathways between extracellular and intracellular space, virulence, biogenesis and even apoptosis (Zalk et al. 2005), they make excellent drug targets (Tusnády et al. 2004). However, unlike their α-helical inner membrane counterparts, designing drugs that target the outer membrane protein is particularly challenging. While the outer membrane does not feature a thick peptidoglycan layer, it does have a very impermeable LPS layer, which when attacked by the immune response, triggers septic shock and even death. Another problem is the relative paucity of crystal structures which help in computational studies of additional motifs, preferences, and higher-order structure prediction. In the meanwhile, protein engineering can lead us to indirectly solving the membrane protein folding problem for OMPs and possibly design novel folds and function.

1.3.1 Redesign of OMPs

We have seen a rising interest and progress in studying OMPs over the past 20 years. We have only recently begun to design OMPs to test the extent of sequence malleability. Successful re-engineerings of OMPs indicate that the sequence tolerates multiple mutations (Koebnik 1999), deletions (Mohammad et al. 2011), insertions (Chen et al. 2008; Muhammad et al. 2011), and even synthetic modulation with (Krewinkel et al. 2011) and without (Reitz et al. 2009) disrupting core structural features. Thus, even though additional crystal structures OMPs along with more structural bioinformatics analyses will edge us closer to a fuller understanding of OMP folding and insertion, we are concurrently making great strides to engineer these OMPs into proteins with novel function and biochemical properties.

1.3.2 de novo design and structure prediction of OMPs

Another method to solving the outer membrane protein folding problem is to attempt *de novo* design using rudimentary principles. Given a large dataset of structures, one way to do this is to parameterize the structure in order to design a scaffold for careful placement of amino acids. So far, many parameterizations have been offered (see 1.2.2 for references). Naveed and the Liang lab revisited the cylindrical barrel model (Naveed et al. 2011) coupled with statistical tools (Jackups and Liang 2005; Jackups et al. 2006; Naveed et al. 2009) for three-dimensional structural prediction of TMBs. Because loops and strand lengths are variable, it is currently impossible to parameterize the entire protein structure including TM and water-soluble regions.

The alternative, if not complement, to *de novo* designing by geometric parameterization is the development of a knowledge-based potential for rational design. These statistical potentials can capture subtle features of the protein that multi-term energy functions fail to incorporate (Nanda et al. 2009). The next chapter will introduce statistical potentials in the context of studying membrane proteins, then discusses recent findings from a knowledge based potential derived from outer membrane proteins.

2. Amino Acid Depth Propensities within Transmembrane β-Barrels (TMBs) and α-Helices (TMHs)

2.1 Amino acid propensities are useful in protein design

2.1.1 Propensities, Odds Ratio, and the Log Transformation

In order to improve the success rate of designing a certain class of proteins, it would be beneficial to extract as many sequence or structural patterns about that class of proteins as possible. Amino acids in proteins rich in secondary structural content, such as α-helices and β-sheet have position-dependent preferences that improve chances of the primary sequence to fold into a desired conformation. Contiguous segments or hotspots of sequences that appear in similar positions along the primary structure more often than expected at random are generally called sequence motifs. Similarly, there may exist positions along the sequence where we know a particular amino acid or combinations of amino acids cannot exist. The antithesis of a motif is aptly termed an anti-motif. Patterns that occur in secondary and higher-order structures are termed structural motifs.

For example, the sequence of collagen can be condensed into a sequence motif – a repeat of triplets $[Gly\text{-}X\text{-}Y]_n$ where X and Y each could be any of the 20 amino acids. Through a set of host-guest peptides, frequencies of all possible amino acid pairs at those positions were determined (Persikov et al. 2000). Because different relative frequencies of different X-Y positioned amino acid pairs may be computed from different sample sizes and lead to biased conclusions, a commonly used metric called the propensity can be calculated by asking the question: what is the probability of finding a certain amino acid at a certain position compared to probability of expecting it at random? Because the propensity metric essentially normalizes data with respect to its background random noise, it is one of the accepted forms of quantifying

preferences of amino acids at positions. The amino acid pair with highest-marked propensity is proline at the X position, followed by hydroxyproline at the Y position. Another commonly used metric used to control for imbalances between groups is called the odds ratio or logistic regression, which has some advantages in certain cases (Cepeda et al. 2003). In this type of study, calculating either logistic regression or propensity is a valid method of quantifying preference.

Both propensity and odds ratio are restricted to the non-negative range. Taking the logarithm of these metrics gives a symmetric interpretation of the significance of a value like 1/100 compared to its inverse 100/1. Now that the resulting transformations of the two metrics give values -2 and 2, respectively, a person interpreting the log odds ratio can now see that the odds ratio of one event's probability p/(1-p) occurring has the same magnitude as that of its complement (1-p)/p.

2.1.2 Glycosylation Mapping - an Experimental Approach to Assessing Insertion of Engineered TMHs

One approach to studying insertion of TMHs required a complex translocon system of proteins that incorporates these guest peptides in membranes. A bitopic (passing the bilayer twice) membrane protein called the leader peptidase (also known as *lep*) (Wolfe et al. 1993) is one of the components of this system and is used to assess positioning changes of an inserted helical peptide due by site-directed mutagenesis. Residues in the loop regions of *lep* are engineered in such a way that insertion of peptides can be detected by the efficiency of glycosylation (Nilsson et al. 1998).

2.1.3 Depth-dependent Propensities of Amino Acids in Transmembrane Helical Proteins

The lipid bilayer is commonly envisioned as a hydrophobic slab with water outside. A membrane protein could be sectioned accordingly into a hydrophobic, water-lipid interface, and aqueous regions. A variety of hydrophobicity scales reflecting free energies of transfer of individual amino acids from water to near center of the lipid bilayer have been determined. Current knowledge supports the water-lipid interface being represented by a continuum instead of a binary model.

Separately, work involving a *lep* construct has shown that varying number of leucines in a TM helical segment inserted with varied rates of success (Hessa et al. 2005). It is therefore reasoned that depth is an important factor in calculating free energy of transfer of an amino acid. Instead of estimating apparent free energies of transfer from water to a lipid solvent, Senes et al. investigated the effects of water-lipid continuum (if any) on the presence of amino acids throughout the continuum (Senes et al. 2007). The metric that gauges presence of amino acids is the propensity as a function of its depth. Because aromatic residues tend to be found in the headgroup region, the shape of the propensity distribution profile would be different and can be expected to be of Gaussian-like nature. Unlike aromatics, polar and aliphatic residues tend to prefer one environment over the other and would therefore be modeled with a sigmoidal type distribution of propensities. The average translocon-mediated insertion probability conforms to the Boltzmann distribution (Hessa et al. 2005; Hessa et al. 2007; Jaud et al. 2009). Furthermore, the number of transmembrane Leu's in the guest peptide is proportional with the apparent free energy, suggesting the total depth-dependent energy could be calculated as the sum of individual depth-dependent energies of amino acids. Senes' propensity P_z, a function of depth and residue type, is calculated by eqn. 4.

Yin and the DeGrado lab put the $E_z\alpha$ propensity to the test by designing peptides that insert and compete for binding to the GxxxG motif of TM helices of integrins, causing the TM helices to dissociate into a conformation that recruits blood clotting factors. In the absence of ADP, which is a platelet stimulating factor, CHAMP shifts the equilibrium of integrins from non-active to active form (Fig. 9). Thus, a knowledge-based depth-dependent potential could be useful for designing membrane-penetrating peptides.

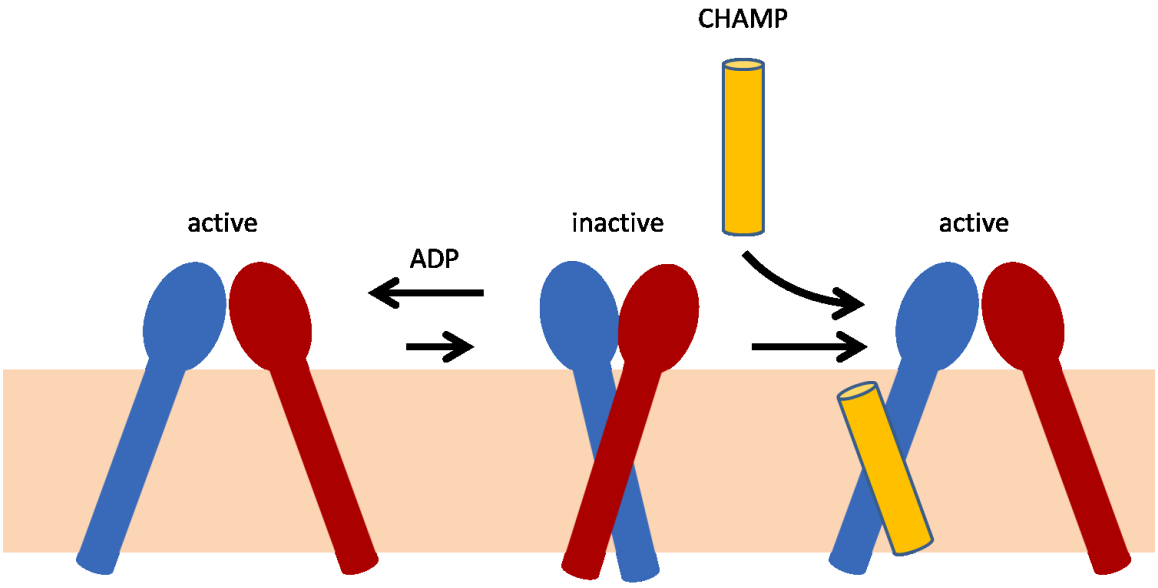


Figure 9 - Computed Helical Anti-Membrane Peptide (CHAMP) binds to integrins to recruit blood clot factors

2.2 Determination of depth-dependent propensities of individual amino acids in transmembrane β-barrel proteins

This sub-chapter is an adaption from the following published manuscript which is the joint work with Alexander Davis and Vikas Nanda:

Hsieh D, Davis A, Nanda V. 2012. Protein Science. A knowledge-based potential highlights unique features of membrane α-helical and β-barrel protein insertion and folding. 21(1):50-62.

In this sub-chapter, a derivation of a TMB-specific knowledge-based potential called $E_z\beta$ is presented. This potential is then compared to one previously derived ($E_z\alpha$) from a dataset of TMHs. Computational experiments in this work show four contributions to our understanding of membrane protein folding and insertion: (1) evidence supporting separate potentials for TMHs and TMBs are necessary, (2) $E_z\beta$ has the ability to reorient and center TMBs in and outside of training dataset (3) lipid facing residues (not buried residues) are the ones driving insertion and (4) $E_z\beta$ can be used for predicting higher-order structure from sequence alone.

The first two sections of this sub-chapter describe the training set of 35 β-barrel membrane proteins and a grid-search algorithm that determined each protein's z-alignment. The parameters that characterize each of two functional forms of a statistical potential, along with the biophysical constants used, are all presented in the third section. Chapter 4 of this thesis shows one possible setup using spreadsheets such as Microsoft Excel one can consider if interested in performing such calculations. The parameters of $E_z\beta$ are explained in the context of each class of residues (polar/non-polar, aromatic) in the fourth section of this sub-chapter.

Computational experiments that validate the necessity of $E_z\beta$ are described. First, we ask: is $E_z\alpha$ sufficient to assess the energy of insertion for all membrane

proteins? Due to differences in secondary structure, environment, and proposed mechanisms of folding and insertion between OMPs and their helical counterparts in the inner membrane, our dataset required an extra step of partitioning into solvent-exposed and buried residues. This part of the work is not about optimizing the classification method but rather validating that such classification is necessary for TMBs whereas no such step is required for TMHs. In the fifth section, we also describe a computational experiment involving thirty trials of randomly swapping residues of each partition and calculate a measure of the sensitivity of $E_z\beta$ for each partition.

Because $E_z\beta$ can properly place amino acids along the depth of the water-lipid continuum, the specificity of $E_z\beta$ in designing TMBs implies its uses in determining position of OMPs in its environment. The potential is then used to re-align the dataset proteins as well as proteins outside of the dataset. Furthermore, in the seventh section of this sub-chapter, the preferences of amino acids are calculated and compared across the partitions (buried or solvent-exposed), secondary structure ($E_z\alpha$ and $E_z\beta$), and residue class. This data provides strong evidence validating our choice of partitioning. In sections 2.2.8 to 2.2.10, a method of visually assessing the ability of a region of residues of oligomeric OMPs to form protein-protein interactions (PPIs) with other subunits is described. The final section discusses the importance of depth-dependence in detecting PPIs by comparing a method of computing an $E_z\beta$ moment vector that points to the interfacial sites with a previously determined depth-independent hydrophobicity moment.

2.2.1 TMB data set

A set of 35 crystallized protein structures of TMBs (Table 1) was first compiled from a larger list of 67 available structures (circa 2010) in the Orientations of Proteins in Membranes database, filtering for a maximum of 26% pairwise sequence homology using EMMA, a ClustalW (Thompson, Higgins, and Gibson 1994) interface included within the software suite EMBOSS (Rice et al. 2000).

2.2.2 Geometric alignment of TMBs along the z-axis

Since the TMBs have preferred orientations in the outer membrane in addition to native conformation (Lomize et al. 2006; Booth and Clarke 2010) it is necessary to determine a geometric alignment of these protein structures for proper studies. A script was constructed using protCAD (Summa 2002), a set of in-house libraries for protein design, to align the β-barrel axis to the z-axis (i.e. normal to the membrane bilayer). The best-fit Euler rotation parameters were determined using a grid-search algorithm that maximized the projection of the transmembrane segments of β-strands to the z-axis. The ends of transmembrane segments were specified by Orientations of Proteins in Membranes (Lomize et al. 2006). The "origin" and thus center of rotation was determined preliminarily by the center of mass of all C_α atoms of the TM segments.

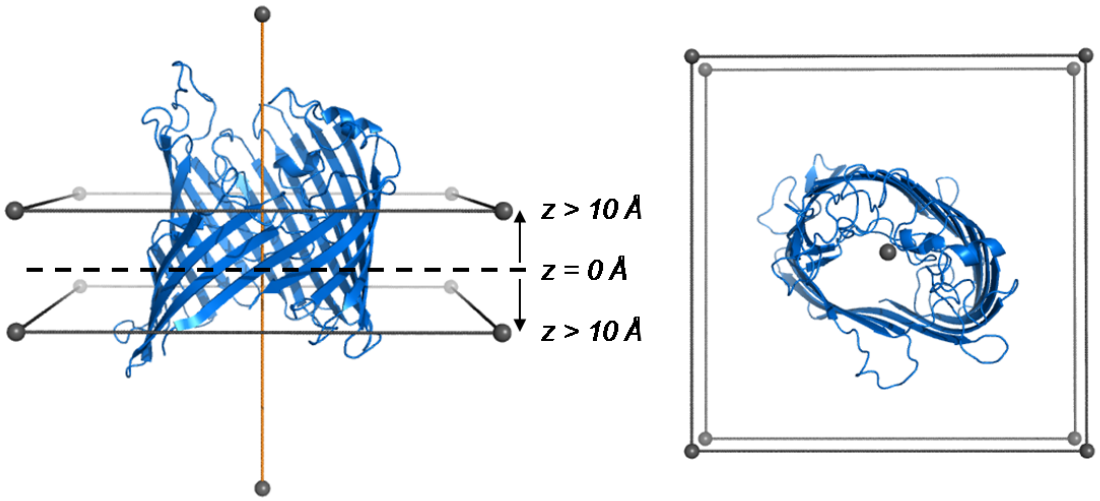


Figure 10 - Z-aligned TMBs

Sucrose porin (PDB 1A0S) aligned to the z-axis using a grid-search algorithm as viewed from side (left) and top view (right).

2.2.3 Calculating $E_z\beta$ parameters

Let the center of the bilayer be defined as z = 0 Å, and assume symmetry over the inner and outer leaflets (Fig. 10). The coordinates of the C_β of an amino acid were used to specify its distance from the bilayer center (glycine is the only exception). Only residues with a fraction of maximal solvent accessible surface area (SASA) greater than 20% were considered (Glyakina et al. 2007). SASA was calculated using DSSP (Kabsch and Sander 1983). The final dataset of amino acids considered for the propensity study was primarily composed of lipid-facing and extra-membrane residues.

The set of propensity data points were gathered by dividing the model environment into discrete 3 Å bins, offset by 1.5 Å (i.e. 0 – 3 Å, 1.5 – 4.5 Å, 3 – 6 Å, and so on). The propensity of an amino acid in a bin is denoted $P_{res,bin}$, and is defined as the observed frequency of the amino acid over its expected frequency in a certain bin:

$$P_{res,bin} = \frac{freq(observed)}{freq(expected)} = \frac{n_{res,bin}}{n_{tot} f_{res} f_{bin}} \quad (3)$$

Here, $n_{res,bin}$ represented the observed number of a particular residue found in a bin. By this definition and the assumption of symmetry of the outer membrane bilayer, $n_{Arg,4.5Å}$ was the number of arginines observed at both 4.5 ± 1.5 Å and -4.5 ± 1.5 Å away from the center of the bilayer. n_{tot} was defined as the total number of residues in the dataset; f_{res} was the frequency of the residue in the entire dataset, and f_{bin} was the frequency in a certain bin. Once all $P_{res,bin}$ were calculated, they were fit using a nonlinear least squares method (Lasdon et al. 1973) to a continuous function P(z).

$$P(z) = P_{aq} e^{-\frac{\Delta E(z)}{RT}} \quad (4)$$

$\Delta E(z) = E(\infty) - E(z)$ is the energetic cost of transferring an amino acid from solvent to a particular depth z in the membrane. R and T are the gas constant and absolute temperature in $\frac{kcal}{mol \cdot K}$ and Kelvin, respectively. The absolute room temperature of 298 K is used for this calculation. The following parameters are solved for as a result of the nonlinear fitting: P_{aq} is the propensity for the amino acid to partition in the aqueous phase. The propensities were fit using either a sigmoidal or Gaussian distribution. The sigmoidal-fit propensity represented the proclivity of an amino acid to partition into either the hydrophobic or aqueous phase.

$$\Delta E_z = \frac{\Delta E_0}{1 + \left(\frac{z}{z_{mid}}\right)^n} \tag{5}$$

ΔE(0) was the energy of transferring a residue from water to the center of the bilayer: E(∞) – E(0). z_{mid} was defined as the depth at which transfer energy is half-maximal. For polar and nonpolar residues, z_{mid} gives an estimate of how deeply a group prefers to be situated from the center of the membrane. n was the steepness of transition, and reflects how tightly coupled the position of an amino acid on the surface of a TMB is with the change in hydrophobicity from a polar aqueous to nonpolar environment.

The amino acids that tend to partition only in the headgroup region of the bilayer are modeled with a Gaussian distribution. The associated functional form is described by the following key parameters: ΔE_{min}, the free energy change between partitioning the residue from water to the headgroup region; z_{min} is the associated depth at which the amino acid attains its lowest energy of insertion; σ is the width of the transition.

$$\Delta E_z = \Delta E_{min} e^{-\frac{(z-z_{min})^2}{2\sigma^2}} \quad (6)$$

Each residue is described by either of these functional forms and individual fits are presented in Table 1. To obtain statistical error on our parameters, a jackknife (leave-one-out) method was applied to a dataset of 35 TMBs to obtain standard errors. Due to low count on the TMB exteriors, Met and Cys were removed from the analysis. (68 Met and one Cys)

Table 1 - Parameters of the of the $E_z\beta$ potential function determined by leave-one-out analysis

Parameters of the $E_z\beta$ potential				
Functional Form 1				
Residue	P_{aq}	$\Delta E(0)$	Z_{mid}	n
Ala	0.7 ± 0.0	-0.8 ± 0.0	6.0 ± 0.1	7.1 ± 0.9
Leu	0.1 ± 0.0	-2.0 ± 0.0	17 ± 0.0	2.9 ± 0.1
Ile	0.3 ± 0.0	-1.0 ± 0.1	15 ± 0.1	18 ± 1.8
Val	0.1 ± 0.0	-1.5 ± 0.0	15 ± 0.1	23 ± 4.2
Asp	3.0 ± 0.0	1.3 ± 0.0	15 ± 0.1	7.2 ± 0.3
Glu	2.9 ± 0.3	1.1 ± 0.1	15 ± 0.3	8.0 ± 1.9
Lys	3.0 ± 0.1	1.3 ± 0.0	14 ± 0.3	3.7 ± 0.2
Asn	1.7 ± 0.0	0.7 ± 0.0	13 ± 0.1	29 ± 1.9
Pro	1.7 ± 0.2	0.8 ± 0.1	11 ± 0.3	8.8 ± 5.6
Gln	1.6 ± 0.3	0.7 ± 0.1	11 ± 0.4	7.7 ± 1.5
Arg	3.0 ± 0.0	1.4 ± 0.1	12 ± 1.0	1.7 ± 0.2
His	1.3 ± 0.0	1.2 ± 0.0	8.0 ± 0.2	14 ± 1.4
Ser	2.4 ± 0.2	0.9 ± 0.1	17 ± 0.0	3.6 ± 1.5
Functional Form 2				
Residue	P_{aq}	ΔE_{min}	Z_{min}	σ
Phe	0.0 ± 0.0	-3.0 ± 0.0	9.5 ± 0.1	10 ± 0.1
Trp	0.1 ± 0.0	-2.1 ± 0.2	11 ± 0.1	6.2 ± 0.4
Tyr	0.3 ± 0.0	-1.3 ± 0.0	9.3 ± 0.0	3.6 ± 0.1
Gly	1.4 ± 0.0	0.7 ± 0.0	9.9 ± 0.1	2.9 ± 0.1

P_{aq} is the limiting propensity in water ($z = \infty$); $\Delta E(z)$ is the free energy change by transferring the amino acid from water to membrane depth z. ΔE_{min} is interpreted as $\Delta E(z_{min})$, where z_{min} is the depth z at which an aromatic residue attains most favorable insertion energy; z_{mid} is where a nonaromatic, non-glycine residue reaches its half-maximal energy; n and σ are the steepness and width of transition, respectively.

Table 2 - Training set of TMBs used

PDB_ID	Name	Organism	Resolution (Å)
1KMO	OM transporter FecA	*E. coli*	2
3EFM	OM receptor FauA	*B. pertussis*	2.33
1QFG	Ferric hydroxamate uptake receptor FhuA	*E. coli*	2.5
1K24	OM adhesin/invasin OpcA	*N. meningitidis*	2.03
2VQI	P pilus usher PapC translocation domain	*E. coli*	3.2
1QD6	OM phospholipase A (OmpLA)	*E. coli*	2.1
3FHH	OM heme transporter ShuA	*S. dysenteriae*	2.6
3CSL	Hemophore receptor HasR	*S. marcescens*	2.7
2ERV	Lipid A deacylase	*P. aeruginosa*	2
2GUF	OM cobalamin transporter BtuB, meso form	*E. coli*	1.95
1FEP	Ferric enterobactin receptor FepA	*E. coli*	2.4
2O4V	Porin OprP	*P. aeruginosa*	1.94
1QJ8	OM protein X (OmpX)	*E. coli*	1.9
1I78	OM protease OmpT	*E. coli*	2.6
3DZM	Major OM protein from T. thermophiles	*T. thermophilus*	2.8
1P4T	OM protein NspA	*N. meningitidis*	2.55
2J1N	Osmoporin OmpC	*E. coli*	2
3PRN	Porin	*R. blastica*	1.9
2POR	Porin	*R. capsulatus*	1.8
1E54	Anion-selective porin	*C. acidovorans*	2.1
2F1C	OM protein G (OmpG)	*E. coli*	2.3
1AF6	Maltoporin	*E. coli*	2.4
1A0S	Sucrose-specific porin	*S. enterica*	2.4
2WJR	Acidic sugar-specific porin NanC	*E. coli*	1.8
2QDZ	Filamentous hemagglutinin transporter FhaC	*B. pertussis*	3.15
1UYN	Autotransporter NalP	*N. meningitidis*	2.6
3JTY	BenF-like porin	*P. fluorescens*	2.58
3BS0	Toluene transporter TodX	*P. putida*	2.6
1THQ	Lipid A acylase PagP	*E. coli*	1.9
2F1V	OM protein W	*E. coli*	2.7
3DWO	FadL homologue	*P. aeruginosa*	2.2
1T16	Fatty acid transporter FadL	*E. coli*	2.6
1TLY	Bacterial Nucleoside Transporter Tsx	*E. coli*	3
1QJP	OM protein A (OmpA)	*E. coli*	1.65
3EMN	Voltage-dependent anion channel (VDAC-1)	*M. musculus**	2.3

*3EMN is the only TMB in the dataset from mitochondrial outer membrane.

2.2.4 Comparison of $E_z\beta$ with $E_z\alpha$

While determining $E_z\alpha$ utiltized all residues of TMH proteins, $E_z\beta$ only considered the residues of lipid-facing and extra-membrane regions. The dataset of thirty-five high-resolution TMB crystal structures considered a subset composed of 4710 of the 12,886 total residues. Due to limited dataset size, the differential partitioning into inner and outer leaflets of the outer membrane was ignored. Parameters for cysteine and methionine were not calculated due to insufficient counts. Only absolute distance from the membrane center was taken into account.

Most amino acids exhibited similar distributions in $E_z\alpha$ and $E_z\beta$. As expected, polar residues preferred the outside of the membrane, while hydrophobic residues had the reverse preference (Fig. 11). Aromatic residues were predominantly situated in the headgroup region. Values of the parameter ΔE_0 for $E_z\beta$ strongly correlated with those of $E_z\alpha$ (R^2 = 0.78) and with an experimentally derived hydrophobicity scale (R^2= 0.68), indicating general physio-chemical behavior was conserved across both classes of membrane proteins (Fig. 12).

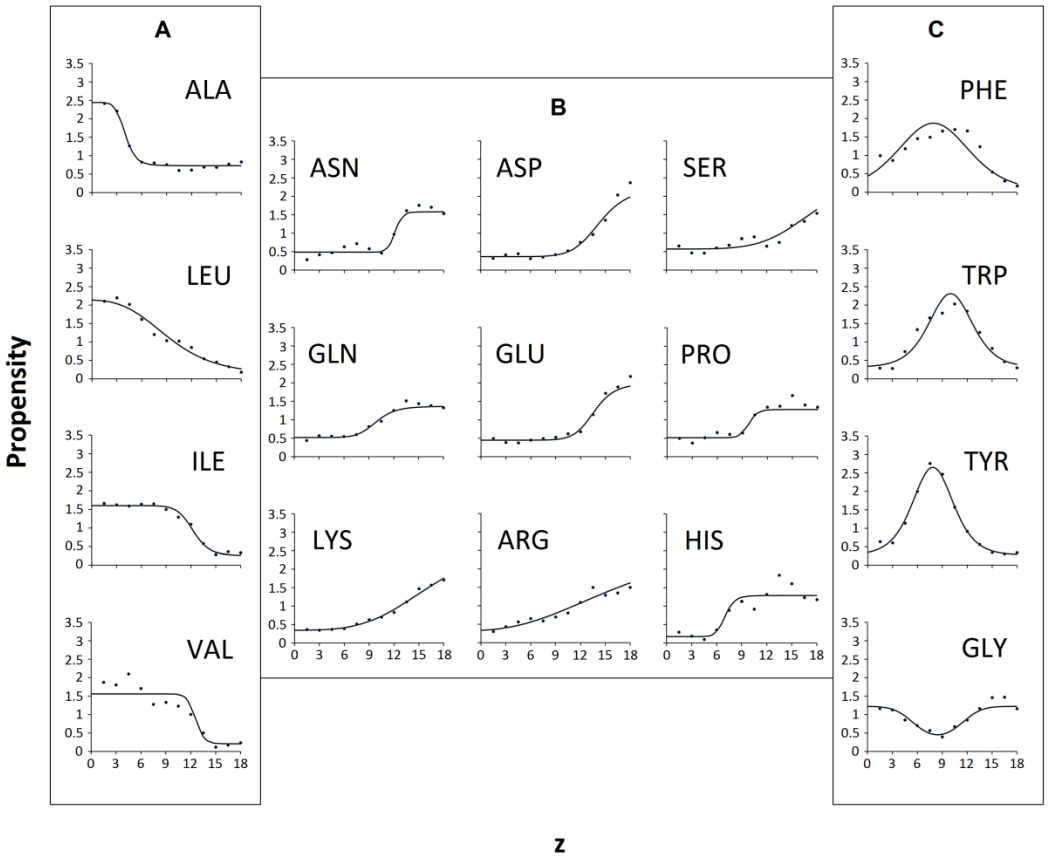


Figure 11 - Amino acid propensity profiles.

Profiles are plotted as a function of depth along membrane depth, separated into three categories: (a) hydrophobic (b) polar and (c) aromatic + glycine. Threonine showed no depth-dependent bias and could not be fit either functional form.

However, a few differences between $E_z\alpha$ and $E_z\beta$ parameters pointed to unique features of TMBs. First, values of the parameters z_{min} for Trp, Tyr, and Phe in $E_z\beta$ were smaller than the corresponding values in $E_z\alpha$, consistent with the proposal that the inner membrane has a thicker hydrophobic core than the outer membrane due to differences in lipid composition (Kleinschmidt and Tamm 2002). The average value of z_{min} for aromatic residues in TMHs (12.4Å) versus TMBs (10.1Å) corresponds to a predicted difference in hydrophobic thickness of 5Å. Second, $E_z\alpha$ and $E_z\beta$ differed in values of the parameter n for aliphatic amino acids. This parameter represents the

steepness of the transition from one environment to another and was previously interpreted as the cooperativity of the interaction with the environment (Senes et al. 2007). In the case of $E_z\alpha$, Leu, Val, and Ile had similar values of n, suggesting similar cooperativities in TMH proteins. In contrast, we observed values of n that were 5- to 6-fold greater for Ile and Val relative to Leu in TMB proteins. In many structures in our TMB training set, the β-sheet ends at the lipid/water interface. Therefore, the steep change in Ile and Val propensities may be due to a favorable β-sheet preference for β-branched amino acids (Minor Jr. and Kim 1994; Regan 1994), rather than cooperativity from sidechain-lipid interactions. The third difference was Phe and Gly required different functional forms of E(z) [Eqs. 5 and 6] to describe their partitioning behavior. In TMBs, phenylalanine partitioned like the other two aromatic residues, localizing to the headgroup region, in contrast to its preference for the bilayer center in TMHs (Wallin et al. 1997; Killian et al. 2000) (Fig. 13).

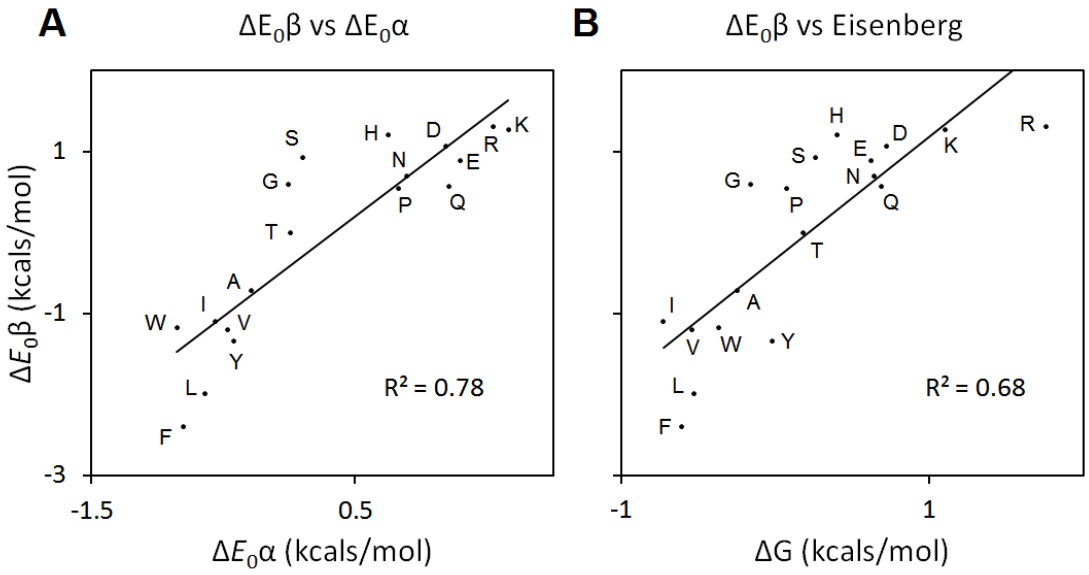


Figure 12 - Correlation with other hydrophobicity scales

(A) Comparison of ΔE(0) for $\Delta E_z\beta$ and $\Delta E_z\alpha$. (B) Comparison of ΔE(0) for $E_z\beta$ with an experimental hydrophobicity scale **(Eisenberg, Weiss, and Terwilliger 1984)**.

Tyr and Trp presumably localize in the headgroup region due to hydrophobic interactions with lipids and hydrogen bonding between the sidechain and water. Phenylalanine lacks a polar moiety on the sidechain but can still form nonconventional hydrogen bonds with water through interaction with the aromatic π-electron cloud (Brandl et al 2001). In TMHs, phenylalanine behaves mostly like a hydrophobic amino acid and preferentially localizes in the center of the membrane; it has the largest $E_z\alpha$ z_{mid} of all the amino acids, suggesting some affinity for the headgroup region as well. In TMBs, affinity for the headgroup region is more pronounced presumably due to favorable π-stacking interactions between aromatic residues in adjacent β-strands, a general feature in β-sheet proteins (Russell and Cochran 2000).

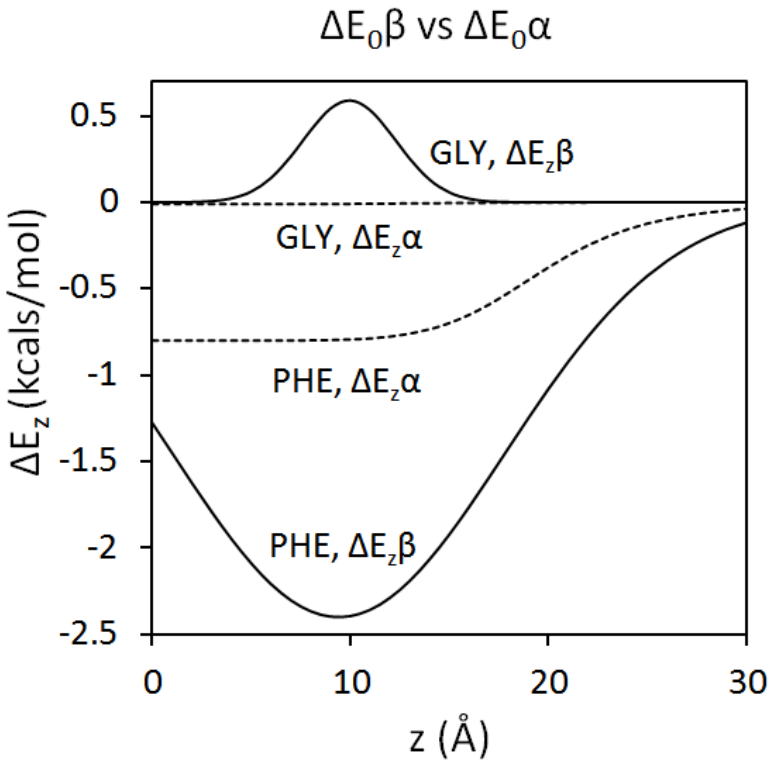


Figure 13 - Insertion energy profiles of glycine and phenylalanine

Energies of insertion of glycine and phenylalanine with respect to membrane depth compared between $E_z\alpha$ (dashed) and $E_z\beta$ (solid).

Relative structural properties of aromatic groups were reflected in the $E_z\beta$ parameters. The optimal $E_z\beta$ propensities of aromatic residues in the headgroup region (P_{min}: Tyr = 2.7, Trp = 2.3, and Phe = 1.9) were consistent with experimentally derived stabilities of substitutions in OmpA (Tyr = -2.6, Trp = -2.0, and Phe = -1.0 kcal/mol) (Hong et al. 2007).

Glycine in $E_z\alpha$ was poorly fit to a sigmoidal function (ΔE_0 = -0.01 kcals/mol), indicating no depth preference. In TMBs, one would expect Gly to be uniformly destabilizing due to its inability to shield the backbone from competing interactions with solvent (Smith et al. 1994). Surprisingly, glycine was found at a higher-than-expected frequency at z = 0 and was unfavorable only in the headgroup region; the distribution best fit a Gaussian with a positive ΔE_{min}. The center of the bilayer is only minimally hydrated, (Wiener and White 1991) mitigating the competing solvent interactions. The headgroup region has significant water content, making the presence of glycine destabilizing to cross-strand hydrogen bonds in this region. Glycine is again found in extra-membrane loops due its flexibility and the absence of secondary structure. Glycine is thus uniquely unfavorable at the headgroup region, the only location with both secondary structure and hydration. In some β-structures, the presence of glycine is compensated by cross-strand pairing with aromatic residues through an interaction called aromatic rescue (Merkel and Regan 1998). However, in the headgroup region, aromatic rescue must compete with solvent hydrogen bonding (i.e., snorkeling (Chamberlain et al. 2004)) and π-stacking interactions. Thus, very few instances of aromatic rescue were observed in this region in contrast with other locations within TMBs (Jackups and Liang 2005).

2.2.5 Computational mass mutagenesis hints confirmation of the insertion model

Information from three-dimensional coordinates (PDB) as well as secondary structural data (Kabsch and Sander 1983) was used to generate a hybrid file (see Chapter 5). This spreadsheet file is the basic unit behind all computational experiments. Random sequence swapping, one of the earliest computations applied to the PDB hybrid file, was performed to investigate whether the amino acids were sensitive to a mass mutagenesis, a feat which cannot be analyzed *in vitro*. For each protein, we considered interior and exterior partitions of the protein using a 20% surface area solvent accessibility cutoff. We also considered partitions of the protein using the geometric criteria of Cα-Cβ projections from the aligned protein structure onto the X-Y plane.

The amino acids within the specified partition were swapped simultaneously and the resulting $E_z\beta$ energies of the partitions were calculated. Thirty swapping trials were performed, and the mean μ and standard deviation σ of these energies were used to calculate a z-score given by the following equation:

$$Z = \frac{x - \mu}{\sigma} \tag{7}$$

The z-score, also known as the standard score, is a transformation that normalizes the distribution to mean = 0 and standard deviation = 1. In our case, using the z-score allows us to compare $E_z\beta$ energies across different proteins in our dataset. The following figure depicts all z-scores for each partition for each TMB in our dataset (Fig. 14).

	1KMO	3EFM	1QFG	1K24	2VQI	1QD6	3FHH
SASA EXT	9.6464	7.7095	9.3579	7.6252	3.2556	7.9357	5.4274
SASA INT	0.0773	0.3792	1.0324	2.7788	4.7279	0.0743	0.2129
VP EXT	7.7874	11.2227	9.1901	7.5725	6.78427	5.7061	7.5919
VP INT	0.3771	3.4783	2.416	4.9902	0.9621	1.2315	0.3408
	3CSL	2ERV	2GUF	1FEP	2O4V	1QJ8	1I78
SASA EXT	8.7898	6.795	6.3286	3.6145	5.5735	6.4656	6.0276
SASA INT	0.7874	2.0721	0.2617	3.1321	0.533	0.5251	0.5369
VP EXT	9.7733	5.0407	6.1049	7.8729	7.2264	5.0192	8.4267
VP INT	4.4462	2.2739	1.36	1.6065	2.6394	0.615	0.7294
	3DZM	1P4T	2J1N	3PRN	2POR	1E54	2F1C
SASA EXT	7.5054	6.4683	7.724	5.5866	4.1397	0.4253	7.2501
SASA INT	0.4847	0.8547	0.0767	0.0215	0.572	4.745	0.5864
VP EXT	7.4173	7.2226	6.6507	7.8466	4.8281	6.6666	6.5024
VP INT	0.7166	2.3274	0.4663	2.7326	1.0689	2.6369	0.042
	1AF6	1A0S	2WJR	2QDZ	1UYN	3JTY	3BS0
SASA EXT	7.4312	7.2508	5.4581	7.6328	8.7951	4.4103	6.1195
SASA INT	0.1405	0.2143	0.2535	1.3156	1.2556	1.967	2.8203
VP EXT	7.5241	7.3212	7.4103	5.8185	6.4462	8.5067	7.1437
VP INT	1.7358	1.7399	0.4217	0.1654	0.3513	0.2333	4.8331
	1THQ	2F1V	3DWO	1T16	1TLY	1QJP	3EMN
SASA EXT	4.3432	5.1726	8.5062	8.3151	7.2005	6.0437	5.9726
SASA INT	1.6525	0.7275	0.7932	2.0065	1.6461	0.4147	1.0668
VP EXT	3.8331	7.4538	7.8511	9.772	8.3108	7.017	6.0111
VP INT	1.5692	1.2353	0.9641	3.0637	0.3713	1.0034	1.4676

Figure 14 - Computational mass mutagenesis

Results are from massively swapping amino acids using biophysical (20% solvent accessible surface area) and geometric criteria. (Lipid-facing residues have a projection of Cα-Cβ onto X-Y plane > 0)

First we observe the general trend of high vs low z-score when comparing sensitivity to randomized depth-swapping of lipid-facing residues compared to that of buried residues. One should note the exception of the behavior of 2VQI and 1E54. 2VQI is the large 24-strand usher protein PapC, known to be an oligomer with a large interfacial site. 1E54 is an anion-selective porin called Omp32. The swapping result of 1E54 is a true anomaly amongst all proteins in the PDB dataset, should be treated as an outlier and be re-examined.

Besides the non-porin data, which is the expected high to low z-score for both SASA and VP criteria, there are two classes of interesting data: those with moderately high z-score when swapping interior-facing residues using the geometric criteria (VP INT) compared to low z-score when doing so with interior-facing residues using the biophysical requirement (SASA%). This implies there is a feature of the residues filtered by the geometric criteria that is different from those filtered by the biophysical criteria. Pursuing this direction of thinking revealed that the SASA data I am working with is derived from both monomeric and oligomeric proteins, and for the oligomeric proteins, the protein-protein interfacial residues are no longer considered even though they still pass the geometric criteria. The discrepancy in z-scores may be due to the inclusion of PPI residues into the random swapping in the selection of residues filtered by the geometric criteria. This implies protein-protein interfacial residues themselves may be contributing to the lower sensitivity to depth upon random swapping, similar to Koebnik's result from swapping residues along various strands on OmpA. Section 2.2.7 discusses another way to dissect depth-dependence amongst various groupings of amino acid selections. Chapter 3 will introduce three new features into $E_z\beta$: 1) building homology models from sequences identified in similar cluster(s) as the PDB sequence for which the structure exists, so that (2) we can further probe depth-dependence of amino acids belonging in lipid-facing and protein-protein interfaces while (3) accounting for asymmetry of the outer membrane.

2.2.6 Calculating Orientations of Proteins in Membranes

The $E_z\beta$ training set was initially aligned using the grid-search algorithm; an ensemble of rigid-body rotations along the local x- and y-axis of the experimental structure (θ_x and θ_y in Fig. 15) were assessed for maximal projection of the β-strands on the z-axis. These rigid rotations were coupled with respect to the z-axis. At each rigid rotation step of size 10°, a total energy of insertion was calculated as the sum of individual residue $E_z\beta$ energies as a function of residue type and depth.

We subsequently realigned the same structures using the $E_z\beta$ potential to assess whether orientations remained consistent across the training set. The average change in angle post-$E_z\beta$ alignment between the geometrically determined barrel-axis and the membrane normal was 9.3° ± 16°, consistent with similar measures using other orientation prediction methods (6.5° ± 7.8°) for the same protein dataset (A.L. Lomize, I.D. Pogozheva, et al. 2006). The displacement of barrel center-of-mass from the center of the bilayer was 0.9 ± 1.9Å. Deviations from zero displacement of the barrel center of mass or coincident barrel axis and membrane normal reflect limitations of the training-set alignment based on geometric criteria. Three proteins not in the original training set: α-hemolysin, FpvA, and OprG also optimally aligned at the center of the membrane with the barrel axis nearly coincident with the membrane normal (Fig. 16).

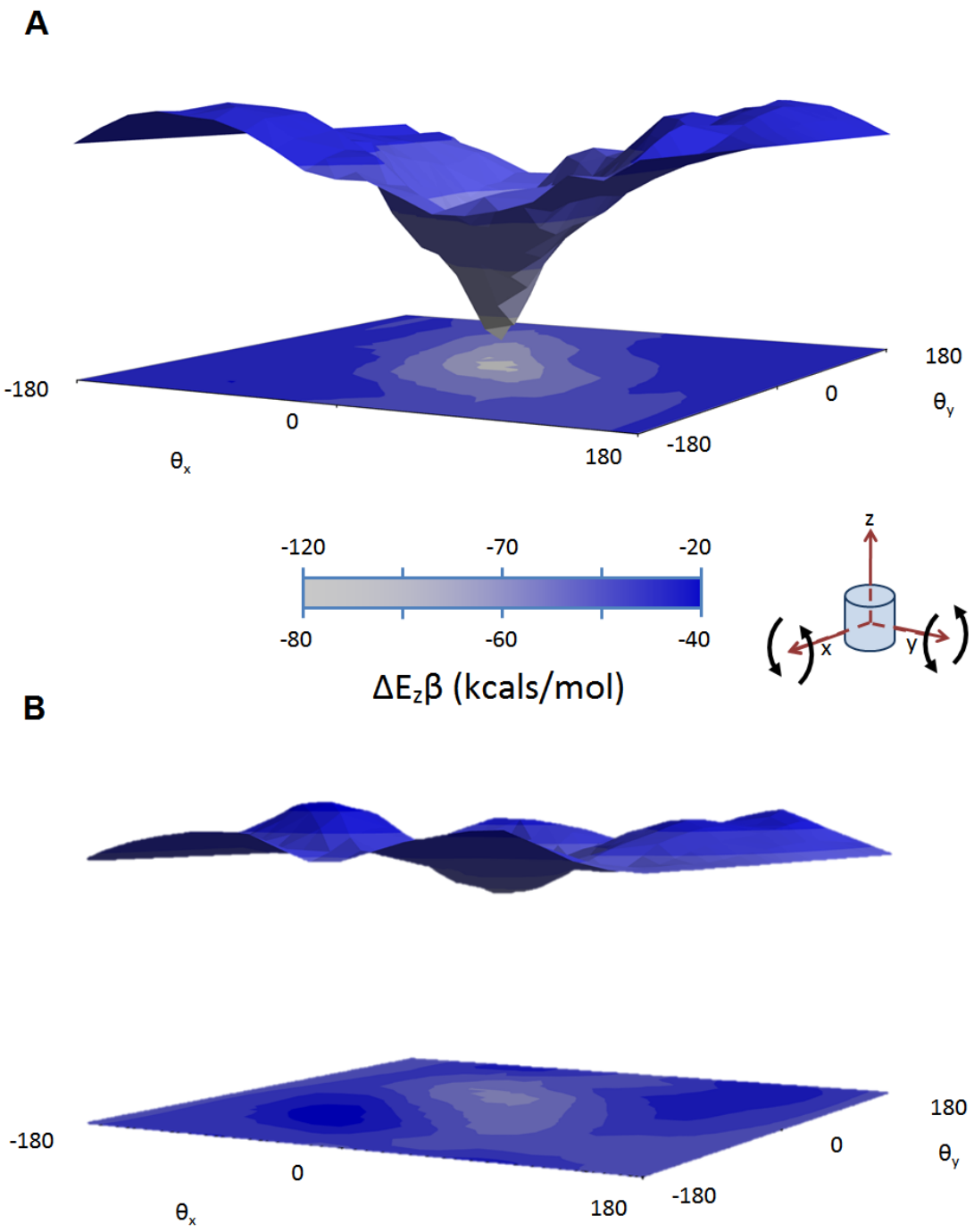
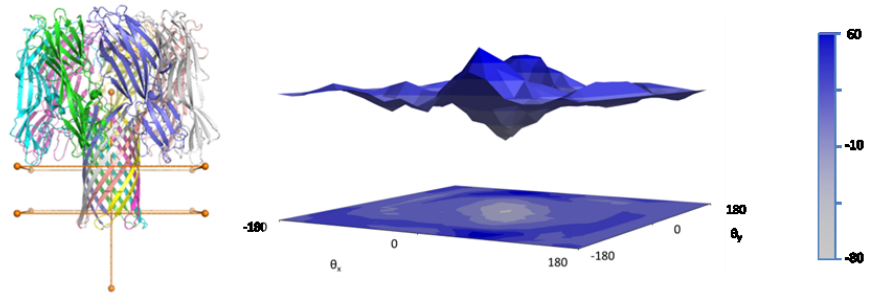


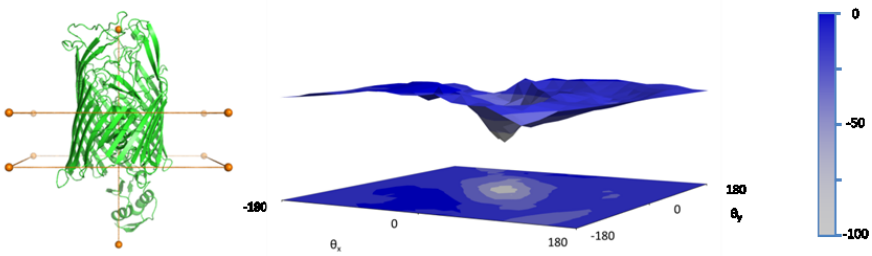
Figure 15 - Energy landscape of barrel orientation

Rigid rotations of (A) FecA, PDB ID 1KMO and (B) TtoA, PDB ID 3DZM about x- and y-axes at z=0 Å.

7AHL (a-hemolysin)
Energy landscape with lowest energy orientation found at z = -4 Å.
OPM angular orientation: 0 ± 1°



2IAH (Pyoverdine Outer Membrane Receptor FpvA)
Energy landscape with lowest energy orientation found at z = 3 Å
OPM angular orientation: 8 ± 0°



2X27 (Outer Membrane Protein OprG)
Energy landscape with lowest energy orientation found at z=1 Å
OPM angular orientation: 7 ± 3°

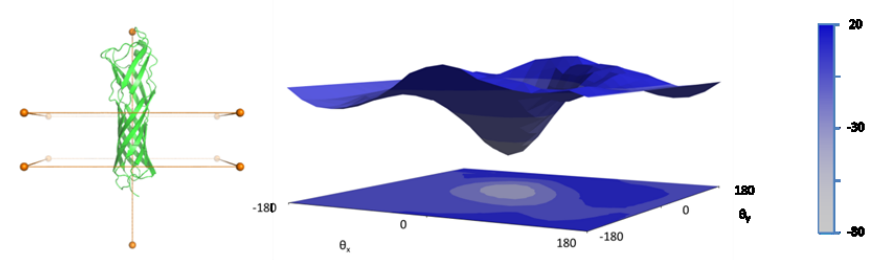


Figure 16 - Alignment of test proteins outside of the training set.

(Left) geometric alignment, (right) Ezβ energy landscape.

Given the discrepancies between $E_z\alpha$ and $E_z\beta$ parameters, one might expect that $E_z\alpha$ would be unsuitable for aligning TMBs. However, alignment of the TMB training set by $E_z\alpha$ was comparable, with center of mass within 1.8 ± 1.2 Å of the membrane center and angular deviation of the membrane normal from the barrel normal of 8.5° ± 11.5°. In terms of OMP placement within the membrane, the similarities between the α- and β-potentials dominate over differences.

Sampling rigid-body rotations of FecA (PDB ID 1KMO) around the minimum orientation resulted in a narrow, funnel-shaped energy landscape (Fig. 15), suggesting that amino acid insertion propensities specify a unique depth and orientation of TMBs within the lipid bilayer. The magnitude of the minimum corresponds with the size of the protein; TtoA (PDB ID 3DZM) which was less than one-third the size of FecA, had significantly shallower minimum.

2.2.7 Lipid facing Residues Dominate Insertion Energetics

Twenty-four TMH structures used in the original $E_z\alpha$ potential were obtained (Senes et al. 2007). For each secondary structural class of protein, the buried and exposed regions were determined by PDB and DSSP criteria. Residues SASA (Kabsch and Sander 1983) of greater than or equal to 20% of the maximal accessible surface area (Glyakina et al. 2007) were considered exposed.

Only lipid-facing residues were included in the calculation of $E_z\beta$, while the original $E_z\alpha$ potential sampled all residues. This was intended to reflect distinct folding pathways employed by TMH and TMB proteins. The two-state folding model of TMHs implies essentially all TM amino acids interact with lipids at some point during folding. In contrast, TMB folding coincides with insertion so that amino acids buried in the native state might never contact lipids during folding. Therefore, it was expected buried amino acids of TMHs would show a depth-dependent bias, supporting the existence of a folding transition-state where all positions interact with lipids to some degree, while buried positions in TMBs would not present a detectable bias.

Distributions of buried amino acids were compared to lipid-facing positions in both TMBs and a set of TMH proteins from the original $E_z\alpha$ potential (Fig. 17). Regardless of secondary structure, lipid-facing residues showed the most pronounced depth dependent bias, consistent with a strong sequence conservation for promoting membrane insertion. A nominal depth dependence was observed for buried, aliphatic amino acids in both TMBs and TMHs. Buried aromatic amino acids did not show any clear propensity to localize in one region of the membrane.

A striking difference was observed in the distribution of buried polar amino acids. TMBs showed a flat distribution across the bilayer, but TMHs had a pronounced bias. Fitting the group of polar amino acids together to parameters describing a sigmoidal distribution showed an approximately tenfold greater ΔE_0 for TMHs over

TMBs (Table 3). This discrepancy supports a two-stage TMH folding model where buried amino acids must interact with the lipid bilayer and thus facilitate insertion.

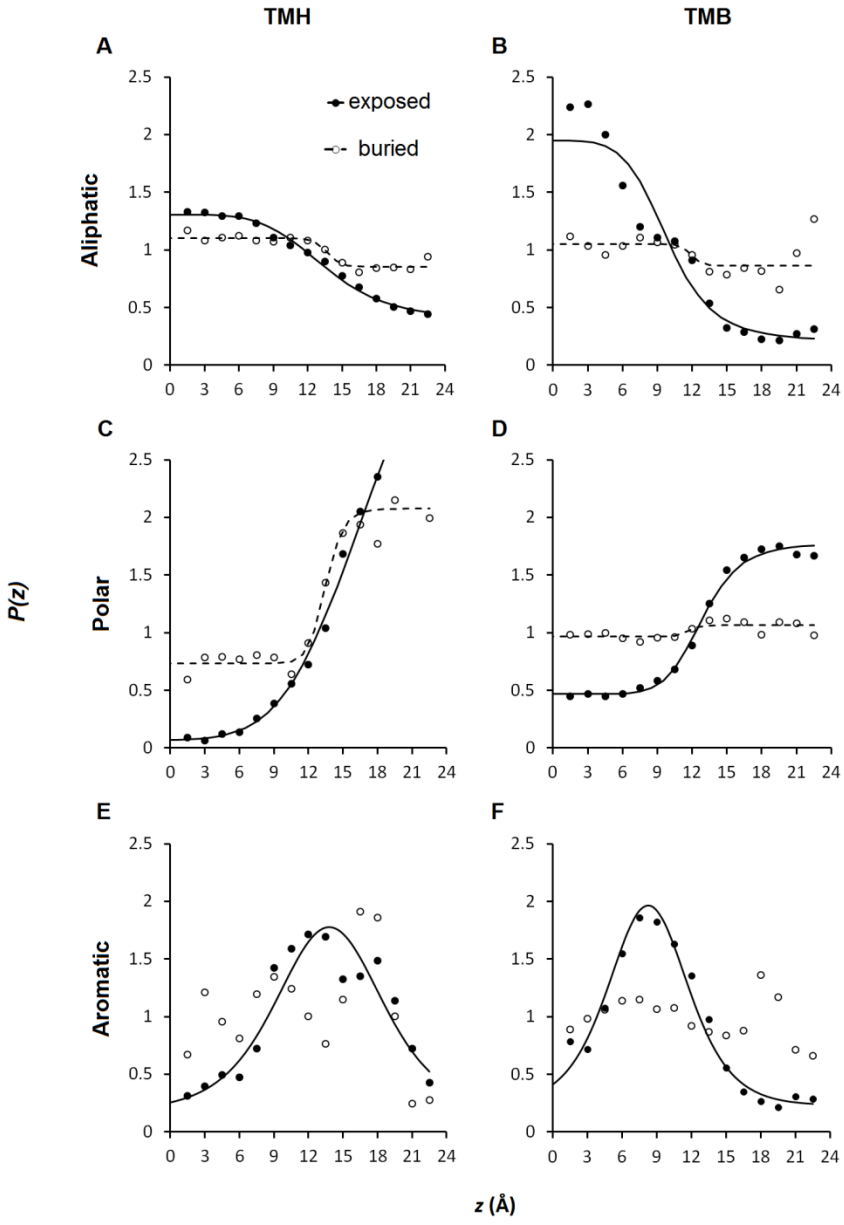


Figure 17 - Depth dependence of lipid facing residues

The distributions of lipid-exposed (filled circles) and buried (open circles) of aliphatic, polar and aromatic residues of TMHs (left) and TMBs (right).

Table 3 - ΔE_0 (kcal/mol) of Buried and Exposed Positions

	TMH		TMB	
Type	Lipid Facing	Buried	Lipid Facing	Buried
L,V,I,A,F[a]	-0.73	-0.15	-1.32	-0.11
N,Q,D,E,R,K	2.64	0.62	0.8	0.06
W,F[a],Y	-1.33	n/a	-1.25	n/a

[a] Phe was considered aliphatic for TMHs and aromatic for TMBs.

2.2.8 Mapping protein-protein interaction sites

The ΔE_z of each residue in the TMB was normalized to the range [0,1] with respect to its ΔE_{min}. Residues were by the normalized insertion energy using a red–white–blue scheme. (Campbell 2004) Amino acids at the oligomerization interface were defined as those that have minimum 40% change in SASA between monomer and oligomer as determined by calculated DSSP values.

2.2.9 $E_z\beta$ moments

For each protein, residues having at least 20% SASA exposed in the monomer are determined by DSSP. The residues should not be interior-facing; this condition was checked manually. To calculate the moment, start with the top view of the protein, looking through the barrel axis. The Cα atoms of these residues are associated with (1) position vectors from a central point (see next section for calculating this point) that are projected onto the membrane then normalized and (2) corresponding $E_z\beta$ energies which are normalized to a range of one's choice. For example, the least favorable energy can be assigned a value of 0 and the best energy 1. The final $E_z\beta$ moment is the inner product between the $E_z\beta$ energies and the normalized position vectors across all of the selected residues. These moment calculations were carried out by Alexander Davis, an undergraduate working for the Nanda Laboratory.

2.2.10 Choosing the appropriate point of central tendency for calculating the $E_z\beta$ moment

To calculate a total $E_z\beta$ moment, a central point must first be picked to assign position vectors to each Cα atom of the included residues. To also guarantee the direction of the moment is invariant given an arbitrary $E_z\beta$ range (min and max energy values), this central point must have the property that if all the energies are equal, the total moment should be the zero vector. This same point has the property of minimizing the sum of distances to these points (Sekino 1999). Such a central point is called the geometric median or Fermat point. Determining the geometric median of these proteins requires an iterative algorithm based on the work of Weiszfeld, (Weiszfeld and Plastria 2009) and such computation is implemented by a Python script written and maintained by Daniel J. Lewis of UCL Geography (2010). See Figure 30 in Appendix 7.10.

From an aligned structure, residues included for moment calculation form a collection of *n* Cα position vectors projected onto the membrane plane $\overrightarrow{p_1} \ldots \overrightarrow{p_n}$, each

paired with associated E_z energies $\overrightarrow{E_1} \ldots \overrightarrow{E_n}$. Each unit position vector can be denoted as: $\overrightarrow{n_1} = \frac{\overrightarrow{p_1} - \vec{g}}{|\overrightarrow{p_1} - \vec{g}|}$, where $\vec{g}$ is the geometric median obtained by Weiszfeld's algorithm using the chosen Cα's (Weiszfeld and Plastria 2009). The total $E_z\beta$ moment is the inner product $\sum E_i \overrightarrow{n_i}$.

By this choice of geometric median $\vec{g}$, we have satisfied the following criteria:

1) $\sum \overrightarrow{n_i} = \vec{0}$.

2)The direction of the moment is invariant to any linear transformation of the ΔE_z range (min and max): $\sum s(E_i + c)\overrightarrow{n_i} = s \sum E_i \overrightarrow{n_i}$.

2.2.11 Discriminating protein-protein interaction sites

If lipid-accessible amino acids are strongly conserved to promote insertion into the bilayer, one might expect surfaces buried upon TMB oligomerization to show a weaker depth-dependent bias. Lipid-facing residues at TMB protein–protein interfaces were noted to have unique amino acid compositions (Koebnik 1999). We tested whether the $E_z\beta$ could predict the binding sites of five oligomers in the data set: ScrY, maltoporin, OMPLA, OprP, and OmpC. An interfacial moment for an aligned β-barrel monomer was defined as the sum of radial moments from the barrel axis to lipid facing amino acids:

$$\mu_{\Delta E_z} = \sum E_i s_i \qquad (8)$$

s_i was the projection of a radial unit vector on the xy-plane from the center of the barrel to the amino acid at position i, and E_i was the $E_z\beta$ insertion energy. This moment matched the protein–protein interface for four out of five oligomers (Fig. 18). With an average of 30% of lipid-facing residues buried in a protein–protein interface, the chance of correctly predicting four out of five protein interfaces by chance was one in two-hundred.

Polar amino acids in the bilayer are hallmarks of membrane protein interaction sites, mediating interprotein networks of hydrogen bonds and ionic interactions (Choma et al. 2000; Zhou et al. 2000; Stanley and Fleming 2007). However, a similarly calculated experimental hydrophobicity-moment (Eisenberg et al. 1984) did not consistently discriminate the binding interface. Only for OMPLA was the most hydrophobic face of the barrel opposite the binding site.

A hydrophobicity-moment does not adequately capture the combined contributions of polar amino acids in the bilayer center and nonpolar amino acids in the headgroup and extra-membrane region. This can be directly visualized on the protein–protein interface by normalizing the $E_z\beta$ insertion energy for each amino acid type from red (most unfavorable) to blue (most favorable) (Fig. 19). Red-colored residues presumably have unfavorable interactions with lipids and/or water, which are relieved upon protein oligomerization. For the sucrose porin ScrY (Fig. 19A), unfavorable positions are found near the center of the bilayer, such as the buried Lys 186, and in the extra-membrane region: Ala 133, Leu 129 and Leu 170 form a hydrophobic surface that binds to Val 428 of an adjacent chain. In the case of OMPLA (Fig. 19B), mainly polar amino acids near the center of the bilayer are detected, resulting in discrimination of the binding interface by both $E_z\beta$ and hydrophobicity-moments. In both cases, these interfacial amino acids are also highly conserved as determined by CONSURF (Armon et al. 2001; Ashkenazy et al. 2010). Not all unfavorable, conserved positions have clear roles in oligomerization, such as the L2-loop in ScrY and Trp 58 in OMPLA. These may play additional structural or functional roles.

The magnitude of the $E_z\beta$ interfacial moment generally correlates with the existence of an oligomeric state in high-resolution structures (Fig. 20), with notable exceptions. Correlation between protein length and $\mu_{\Delta E_z\beta}$ is not straightforward, indicating protein size is not the primary determinant of the magnitude of the

interfacial moment. OMPLA has the lowest $\mu_{\Delta E_z \beta}$ of all the oligomeric proteins in the training set, despite crystallizing as a dimer (PDB ID 1QD6). OMPLA exists in a monomer-dimer equilibrium regulated by calcium and substrate binding; (Snijder et al. 1999; Stanley et al. 2006) the weaker moment is consistent with the need of allosteric modulators to promote dimer formation. A few proteins that crystallized as monomers, 1FEP, 1K24, and 2F1C, had biochemical evidence suggesting the existence of protein–protein interactions (Achtman et al. 1988; Skare et al. 1993; Locher et al. 1998).

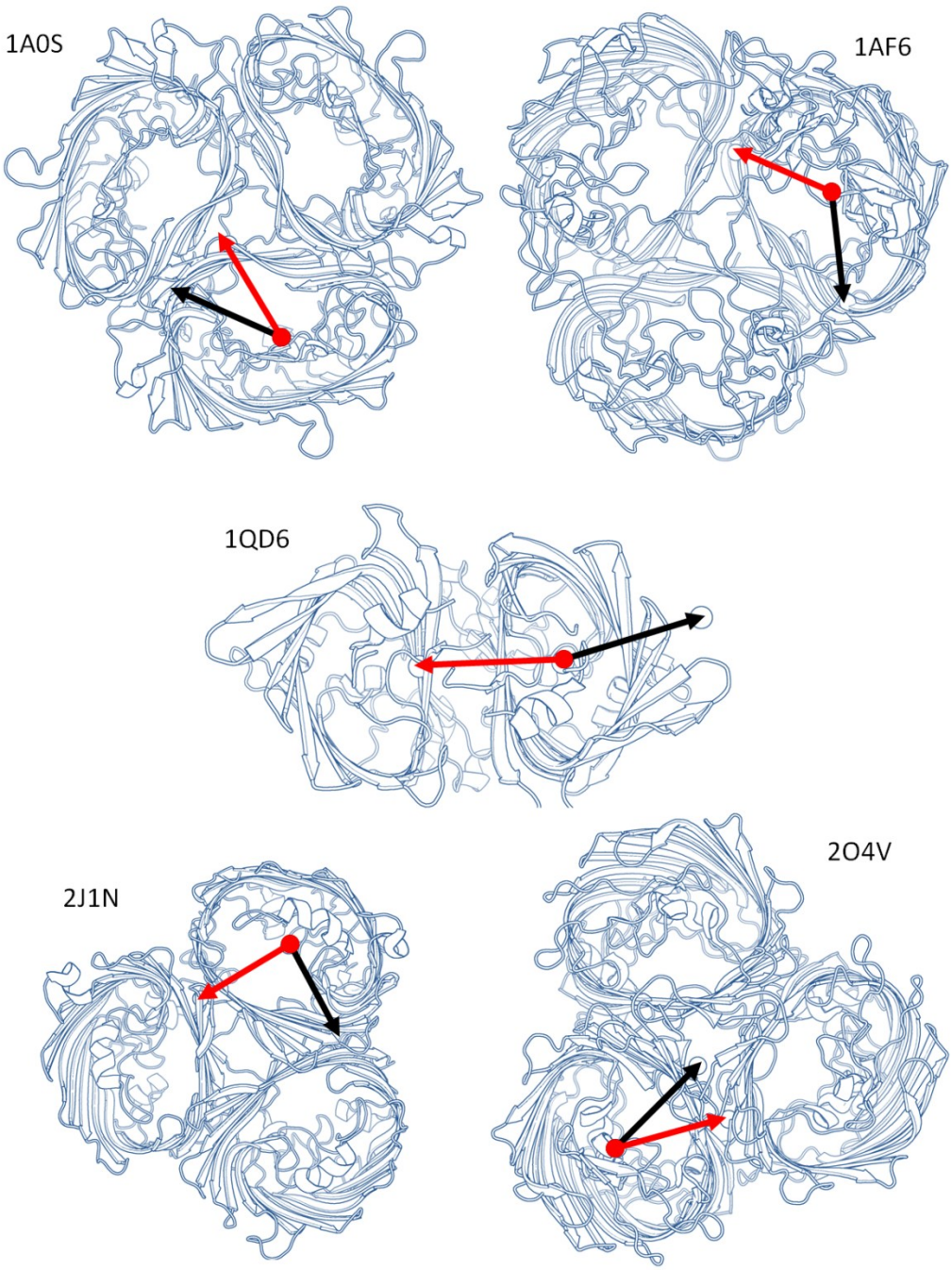


Figure 18 - Interfacial-moment of the barrel exterior

Eq 7 (Red arrow) consistently discriminates the binding interface. The hydrophobicity-moment (black) is computed using amino acid transfer energies from **(Eisenberg et al. 1984)**.

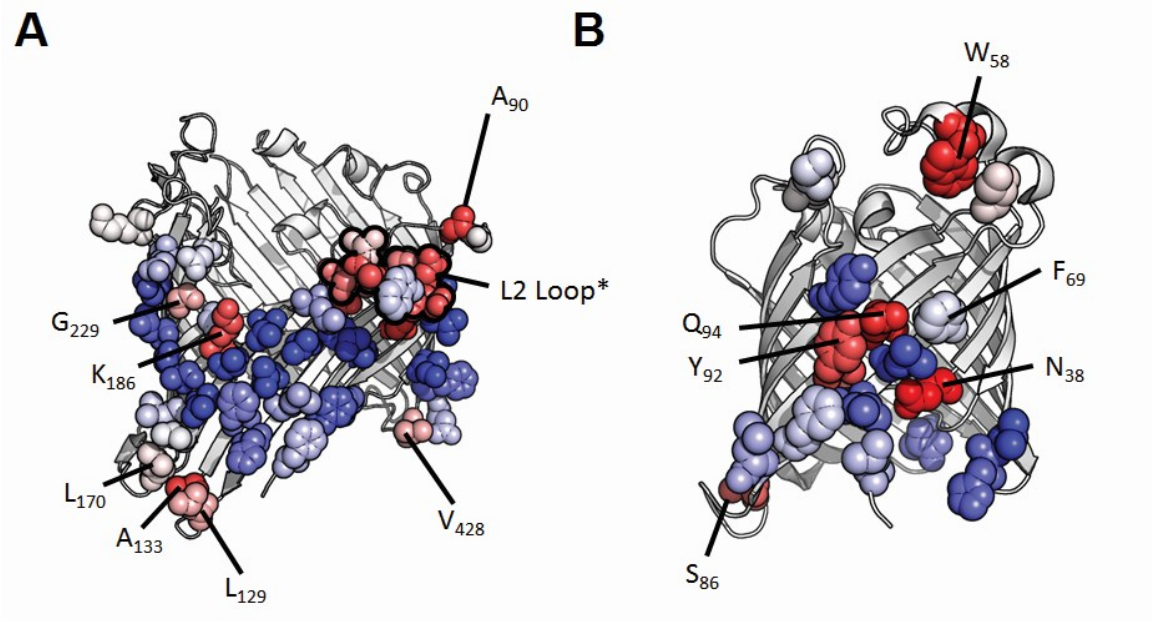


Figure 19 - Interfacial residues colored by $E_z\beta$ potential

(A) Sucrose porin (1A0S) and (B) outer membrane phospholipase A (1QD6) interfacial residues colored by $E_z\beta$ potential.

Amino acids not favorably placed were colored red, while those more favorable for insertion are colored in blue.

*Residues in the L2 loop: Asn 149, Asp 150, Ala 153, and Ser 156.

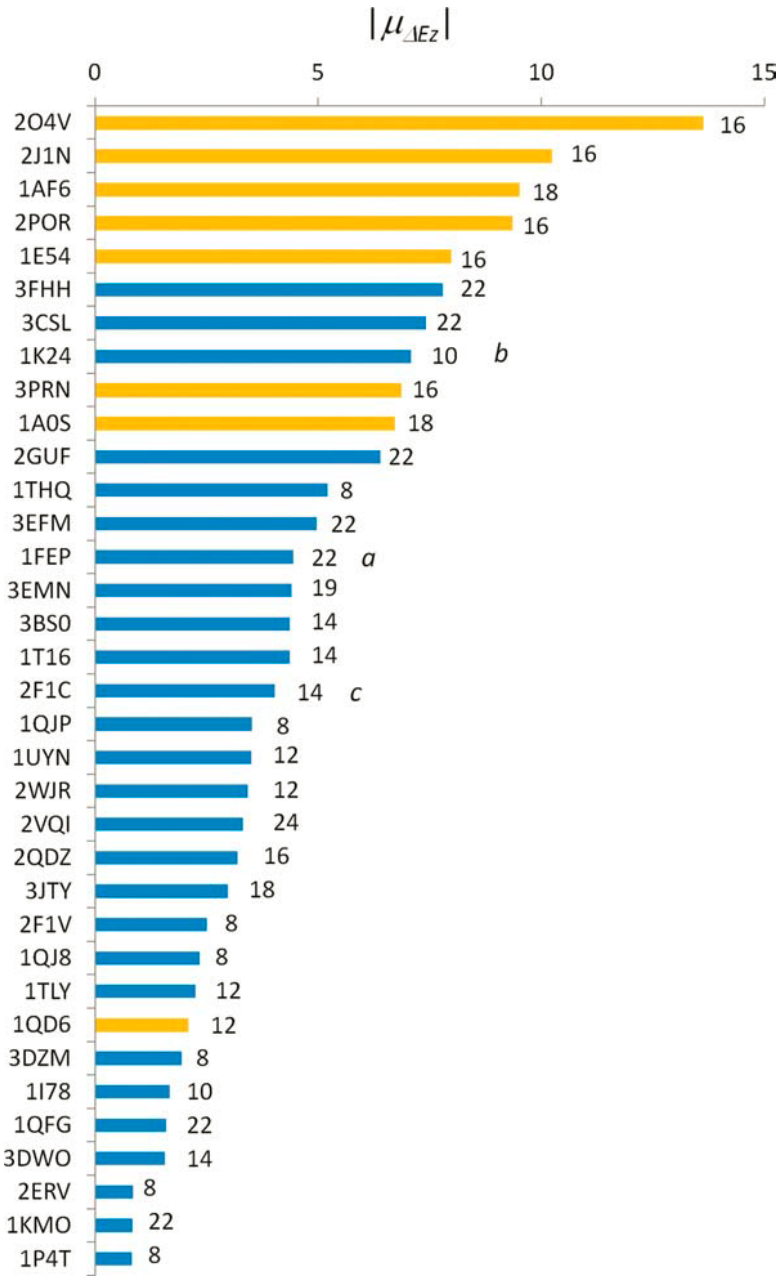


Figure 20 - Magnitudes of the $E_z\beta$ interfacial moment

Large magnitudes of the $E_z\beta$ interfacial moment [Eq. (2)] generally correlate with crystallographic evidence of oligomerization (yellow bars). TMBs without a clear protein-protein interface are shown in blue. Number to the right of the bar is the β-strand count. Biochemical data supports potential presence of oligomers for **(Achtman et al. 1988; Skare et al. 1993; Locher et al. 1998)**.

2.3 Discussion

A statistical analysis of amino-acid depth preferences confirmed TMH and TMB proteins obeyed similar broad physiochemical rules: aliphatic residues preferred the hydrophobic bilayer center while polar amino acids were sparse in the same region, and aromatic belts girdled the protein at the water-lipid interface. A more detailed examination of the $E_z\alpha$ and $E_z\beta$ parameters revealed differences for each class of proteins consistent with unique aspects in structure, folding pathway and lipid environment.

Only recently has a direct experimental thermodynamic scale of transfer energies been attempted for a TMB OMPLA, allowing a direct comparison to $E_z\beta$ (Moon et al 2012). Nineteen substitutions were made at Ala 210, which is within one angstrom of the bilayer center assumed in our calculations. A direct comparison of $\Delta\Delta$Gs of mutation in OMPLA to ΔE_0 from $E_z\beta$ shows a very good correlation (R^2 = 0.77) if proline and phenylalanine are omitted. Proline can be structurally disruptive to both α-helical and β-strand regular secondary structures, but studies of Pro substitutions in bacteriorhodopsin reveal a complex dependence of stability on local structural context. (Yohannan et al. 2004) Often, the effects of proline are on kinetics of folding or post-translational processing. Such effects do not factor in equilibrium thermodynamic measurements, but would certainly affect TMB sequence evolution. The other outlier, phenylalanine, was the most favorable substitution for Ala at position 210 in OMPLA, whereas $E_z\beta$ indicated Phe at z = 0Å was destabilizing. The reason for this discrepancy is not clear, but may result from kinetic constraints imposed by off-pathway interactions between a centrally located Phe and other aromatic groups in a partially folded intermediate (Kleinschmidt and Tamm 1996). This highlights both the challenge of developing a sufficiently general amino acid transfer scale, and the limitations of attributing knowledge-based potential parameters to

purely thermodynamic effects when sequence conservation come from many aspects of protein function.

The $E_z\beta$ potential can be applied to challenges in TMB structural bioinformatics, from structure prediction to identifying protein and lipid binding sites. Although sequence conservation is often used to identify functional sites, it is difficult to determine whether conservation is instead due to constraints of folding and structure (Jaramillo et al. 2002). For example, although the descriptors for TMB geometry have been obtained and compared with those of water-soluble as well as theoretical β-barrels (Pali et al. 2001), more work is needed to show how the geometry of a TMB that is between cylindrical and hyperboloidal affects the distribution of residues according to the size of the residues. In particular, it may be interesting to refine $E_z\beta$ to account for the bulkiness of aromatic residues, which may play a part in their low propensity to partition near the center of the membrane as well as the TMB's oligomerization.

$E_z\beta$, in concert with existing TMB design potentials, (Jackups and Liang 2005; Jackups et al. 2006; Naveed et al. 2009; Jackups and Liang 2010) can be used to identify amino acids conserved primarily for folding and insertion, allowing a clearer discrimination of positions required for protein or lipid interactions (Adamian et al. 2011). As with the application of $E_z\alpha$ to the design of the CHAMP peptides that bind TMH targets, $E_z\beta$ will be a useful tool in TMB protein engineering. There is gaining interest in modifying TMBs as small molecule biosensors, (Chen et al. 2008; Mohammad et al. 2011) engineered enzymes (Varadarajan et al 2008) and drug delivery agents (Meier et al. 2000; Grosse et al. 2011). In many of these cases, improving TMB stability is an important engineering goal (Chen et al. 2008). To incorporate TMBs into synthetic membranes, $E_z\beta$ can be adjusted to accommodate bilayers of varying thickness to minimize hydrophobic mismatch in the new design (Noor, Marco, and Ulrich). In the future, we plan to incorporate this potential into

software for computational protein design, toward the development of fully de novo TMB proteins.

3. Building Homology Models To Improve $E_z\beta$

$E_z\beta$ is a simple and computationally inexpensive knowledge-based potential. However, the depth-propensities were derived from thirty-five structures of low (<=26%) pairwise homology, and used a SASA >= 20% (Maximum ASA) cutoff criteria for selecting lipid-facing and extracellular residues. By assuming symmetry of the bilayer leaflets in the outer membrane and calculating propensities that symmetrically bin residues may omit important physicochemical features. This chapter describes a preliminary step in building homology models using related sequences of unknown structure in order to improve data count.

3.1 Hidden Markov Models in Structural Prediction of TMBs

A Markov process or chain is a system of states in which the future state depends only on the immediate previous state in some sequence. This type of modeling has many applications in everyday life, ranging from valuation of financial instruments to speech/language/writing recognition to detecting credit card fraud and criminal behavior. Given a set of states (*state space*), a combinatorial set of transition probabilities from states to other states (and themselves) called the *transition probability matrix*, and an initial state probabilities vector π, one can use powers (repeated self-multiplication) of the transition probability matrix on the initial states to determine the behavior of transition in the long run. Hidden Markov models (HMMs) come into play because not all of the state space has been revealed through observation. Hence, HMMs are useful in inferring the hidden intermediate states and their corresponding behavior.

The work of Arne Elofsson and collaborators demonstrates the usefulness of incorporating evolutionary information and HMMs in producing accurate topology predictions for TMHs (Viklund and Elofsson 2004) and recently TMBs (Hayat and Elofsson 2012). Topology in this context could mean a combination of the following

features of the TMB: number of strands, belonging to inner/outer loop/TM segment "compartments", and location of strands. The observation that almost all OMPs are homologous to each other is the underlying assumption behind building HMM profiles using sequence rather than structural alignment data. Remmert et al used it to compile a database of OMPs clustered by sequence homology (Remmert et al. 2009).

3.2 MSA-based method for producing a larger dataset for $E_z\beta$

3.2.1 Preliminary development method for building homology models

Proteins in our dataset are checked for appearing in the same cluster using HHomp. This process eliminates 1AF6, 2F1C, 2POR, 3DWO, 3DZM, 3FHH and 3PRN from our dataset, leaving us with 28 unique TMB clusters (Table 5). Each of the remaining proteins was queried (in FASTA format) against HHomp to retrieve an alignment of homologous sequences (not including the query sequence). The queried FASTA sequence is then realigned with the most homologous cluster(s) in the ClustalX interface (Larkin et al. 2007) using default parameters and a substitution matrix specific to outer membrane proteins (Jimenez-Morales and Liang 2011). Prior to the alignment, all gaps were deleted to avoid producing further gapping (Fig 21). The substitution matrix, $BBTM_{out}$, was submitted as is to the ClustalX MSA parameters for calculating the MSA (Fig 22).

Table 4 - Unique clusters associated with Proteins in our Dataset

number	PDB_ID	Cluster Name	subclusters	# of homol. seqs in MSA
1	1A0S	cluster73	18.1.1 18.1.2	51
2	1AF6		18.1.1	n/a
3	1E54	cluster28	16.2.1-16.2.5 16.2.8	319
4	1FEP	cluster8	22.4.2 22.4.4	81
5	1I78		10.1.1	15
6	1K24		10.2.1	2
7	1KMO		22.2.4	31
8	1P4T	cluster144	8.1.5 8.6.3 8.6.2	76
9	1QD6		12.6.1	85
10	1QFG		22.1.4	224
11	1QJ8		8.3.1	31
12	1QJP	cluster75	8.1.1 nn.31.1	78
13	1T16		14.1.1	162
14	1THQ		8.5.1	14
15	1TLY	cluster108	12.5.1 12.5.2	36
16	1UYN		12.1.6	57
17	2ERV		8.4.1	44
18	2F1C		14.2.1	n/a
19	2F1V		8.2.1	141
20	2GUF		22.4.5	61
21	2J1N	cluster99	16.1.1 16.1.2	77
22	2O4V		16.4.2	75
23	2POR	cluster131	16.2.1 16.2.3	n/a
24	2QDZ	cluster53	nn.5.1 nn.5.2 nn.5.4 cluster43	152
25	2VQI		nn.2.2	206
26	2WJR		nn.36.1	35
27	3BS0	cluster71	14.1.5 14.1.7 14.1.1 cluster62	194

28	3CSL		22.4.6	67
29	3DWO		14.1.1	n/a
30	3DZM	cluster165	8.1.1	n/a
31	3EFM	cluster18	cluster6 22.1.7 22.1.4 22.1.5 22.1.3 22.1.6	531
32	3EMN		nn.54.1	25
33	3FHH		22.4.6	n/a
34	3JTY		nn.9.1	172
35	3PRN	cluster131	16.2.1 16.2.3	n/a

The HHomp web application allows users to download the alignments in many file formats including FASTA and Clustal formats. An Excel add-on developed as a sample of a Bioinformatics toolset called .NET Bio (Microsoft Research) was used to import these aligned sequences into separate sheets with uniform formatting. An Excel script was written to transfer these sequences into a single sheet in order to begin building homology models.

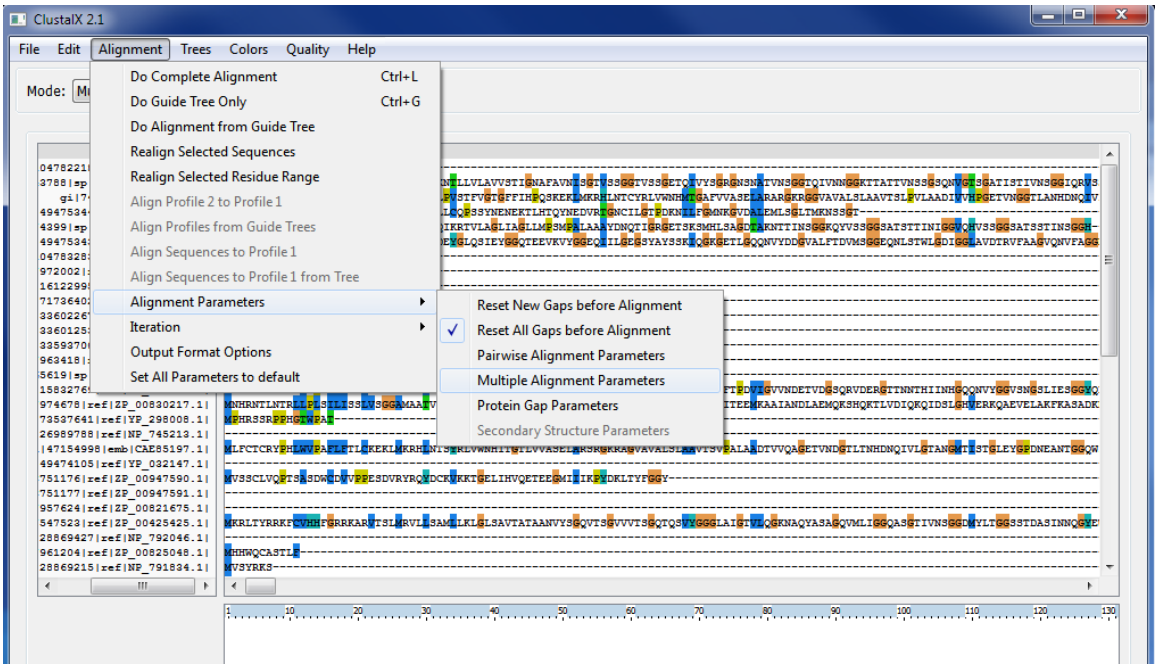


Figure 21 - ClustalX MSA Protocol

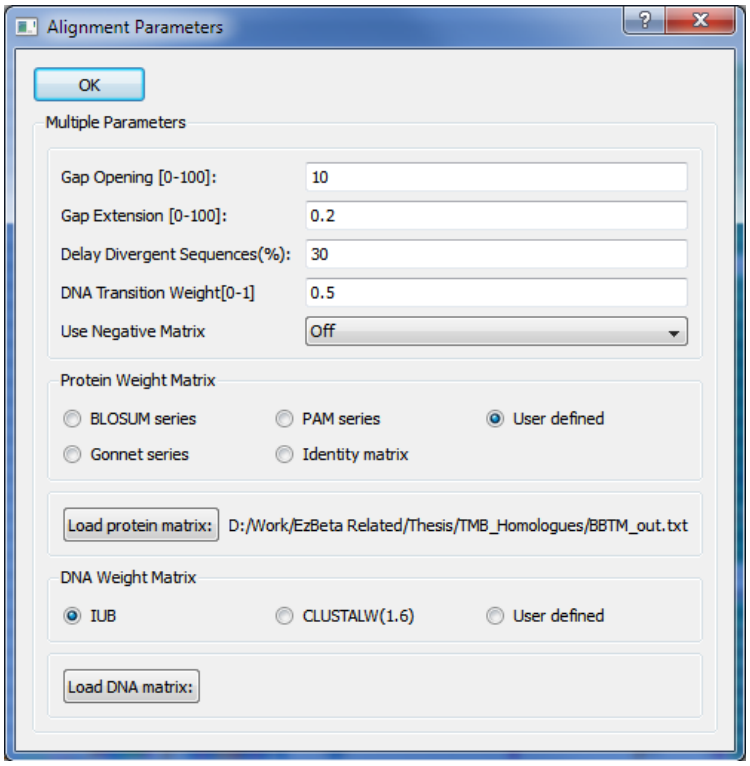


Figure 22 - ClustalX MSA parameters

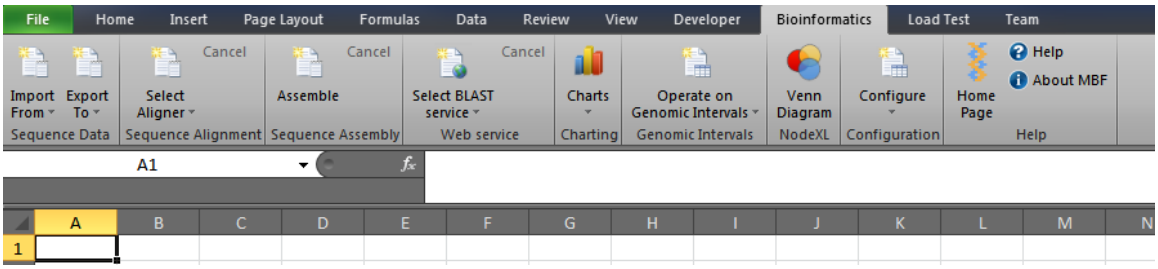


Figure 23 - Microsoft Bioinformatics Add-on for Excel

To build the homology models, the corresponding PDB/DSSP hybrid file was modified such that the "query" sequence remains untouched in the 3-Letter code column, with the 1-Letter code replaced with the "template" sequence (in 1-Letter code format). If the alignment produced a "-", signifying a gap in the alignment, it will be shown as "-" in the template column (Fig. 24).

	A	B	C	D	E	F	G	H	I	J
1	geneID	Chain	ResIndex	3-Code	1-Code	Secondary	%SASA	zCoord of	GLY=1	vectProj
27	gi54309423refYP_13044311	P	96	ALA	N	T	0.018868	8.019	0	-0.95247
28	gi54309423refYP_13044311	P	97	TYR	Q	T	0.308	12.548	0	-1.53511
29	gi54309423refYP_13044311	P	98	ILE	V		0.115789	15.092	0	1.533398
30	gi54309423refYP_13044311	P	99	THR	R	S	0.010753	15.828	0	1.342894
31	gi54309423refYP_13044311	P	100	PRO	Y	G	0	20.408	0	1.118723
32	gi54309423refYP_13044311	P	101	ALA	Q	G	0	15.745	0	1.236546
33	gi54309423refYP_13044311	P	102	GLY	S	G	0.086538	14.865	1	1.292407
34	gi54309423refYP_13044311	P	103	GLU	N	G	0.4	18.763	0	0.075591
35	gi54309423refYP_13044311	P	104	THR	K	G	0.021505	16.624	0	1.34804
36	gi54309423refYP_13044311	P	105	GLY	V	T	0.067308	12.224	1	1.529894
37	gi54309423refYP_13044311	P	106	GLY	G		0	12.237	1	0.667442
38	gi54309423refYP_13044311	P	107	ALA	-		0	8.444	0	-0.92401
39	gi54309423refYP_13044311	P	108	ILE	-		0	9.52	0	1.134806
40	gi54309423refYP_13044311	P	109	GLY	-		0	5.578	1	1.425751
41	gi54309423refYP_13044311	P	110	ARG	R		0.02518	4.258	0	-1.35638
42	gi54309423refYP_13044311	P	111	LEU	L	T	0.009434	2.01	0	1.204454
43	gi54309423refYP_13044311	P	112	GLY	G	T	0.009615	4.772	1	1.517722
44	gi54309423refYP_13044311	P	113	ASN	N		0.005319	7.731	0	-1.51515
45	gi54309423refYP_13044311	P	114	GLN	E		0.037975	6.06	0	-1.06722
46	gi54309423refYP_13044311	P	115	ALA	D		0.106918	9.524	0	0.891095
47	gi54309423refYP_13044311	P	116	ASP	D	S	0.201058	8.509	0	-1.44415
48	gi54309423refYP_13044311	P	117	THR	L	E	0.016129	5.291	0	1.542637
49	gi54309423refYP_13044311	P	118	TYR	Y	E	0.096	4.563	0	-1.50719
50	gi54309423refYP_13044311	P	119	VAL	S	E	0.006098	1.253	0	1.549569

aligned_1A0S | 1A0S_PreHybrid | 1A0S_Hybrid | 1A0S Hybrid Values | 1A0S_ThreadSeq

Figure 24 - Creating homology models using the PDB/DSSP Hybrid File

Only the first column (originally PDBID modified into GeneID) and 1-Letter Code column were modified into a template sequence column. The 3-Letter column remained as the query sequence column. All other information refers to the query sequence.

3.2.2 Future Work Needed

We are currently developing a semi-automated method in building homology models using the computational tools built using the PDB/DSSP Hybrid file type (Excel spreadsheet). The $BBTM_{out}$ matrix used in the MSA requires further understanding as the first generation of this method used the matrix as is. Perhaps, like the PAM 250 matrix (Dayhoff 1978), the $BBTM_{out}$, which is based on an "evolutionary time unit of 40", may require several iterations of self-multiplication to produce better alignments.

#	bbTMout																							
#	Lowest score = -16,						Highest score = 25																	
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	6	-12	-11	-13	-12	-11	-10	1	-13	-1	-1	-13	-5	-3	-8	-3	0	-9	-8	0	-12	-4	-6	-16
R	-12	14	-7	-9	-12	-5	-8	-10	-6	-12	-12	-4	-13	-13	-14	-11	-8	-12	-8	-12	-8	-9	-8	-16
N	-11	-7	13	-1	-12	-3	-8	-9	-7	-13	-12	-7	-8	-12	-14	-3	-3	-14	-10	-12	6	-8	-7	-16
D	-13	-9	-1	14	-12	-7	-5	-8	-12	-13	-11	-6	-12	-13	-14	-8	-11	-14	-11	-13	7	-6	-8	-16
C	-14	-13	-11	-12	25	-11	-11	-14	-14	-14	-15	-14	-14	-14	-13	-14	-12	-14	-11	-13	-11	-12	-11	-16
Q	-11	-5	-3	-7	-10	12	-4	-12	-2	-12	-12	-6	-9	-12	-10	-8	-4	-10	-8	-11	-5	-8	-7	-16
E	-10	-8	-8	-5	-12	-4	17	-11	-12	-12	-12	-13	-14	-12	-16	-11	-8	-13	-13	-11	-6	3	-9	-16
G	1	-10	-9	-8	-12	-12	-11	8	-14	-6	-6	-13	-9	-7	-9	-4	-3	-11	-10	-5	-8	-1	-7	-16
H	-13	-6	-7	-12	-12	-2	-12	-14	13	-12	-12	-10	-13	-11	-14	-9	-12	-9	-5	-13	-9	-13	-9	-16
I	-1	-12	-13	-13	-13	-12	-12	-6	-12	5	2	-14	-3	-1	-13	-9	-5	-6	-8	3	-13	-9	-7	-16
L	-1	-12	-12	-11	-15	-12	-12	-6	-12	2	4	-13	-2	1	-11	-9	-5	-5	-6	2	-11	-9	-6	-16
K	-13	-4	-7	-6	-12	-6	-13	-13	-10	-14	-13	14	-12	-12	-12	-10	-9	-13	-12	-13	-6	-13	-9	-16
M	-5	-13	-8	-12	-12	-9	-14	-9	-13	-3	-2	-12	12	-3	-13	-10	-4	-9	-9	-3	-10	-11	-7	-16
F	-3	-13	-12	-13	-12	-12	-12	-7	-11	-1	1	-12	-3	6	-10	-10	-8	-3	-2	0	-12	-9	-6	-16
P	-8	-14	-14	-14	-12	-10	-16	-9	-14	-13	-11	-12	-14	-10	12	-8	-9	-13	-15	-12	-14	-12	-10	-16
S	-3	-11	-3	-8	-12	-8	-11	-4	-9	-9	-9	-10	-10	-10	-8	11	-1	-12	-11	-8	-5	-7	-7	-16
T	0	-8	-3	-11	-12	-4	-8	-3	-12	-5	-5	-9	-4	-8	-9	-1	9	-11	-8	-3	-7	-5	-5	-16
W	-9	-12	-14	-14	-12	-10	-13	-11	-9	-6	-5	-13	-9	-3	-13	-12	-11	9	-1	-5	-14	-12	-8	-16
Y	-8	-8	-10	-11	-12	-8	-13	-10	-5	-8	-6	-12	-9	-2	-15	-11	-8	-1	6	-7	-10	-11	-7	-16
V	0	-12	-12	-13	-13	-11	-11	-5	-13	3	2	-13	-3	0	-12	-8	-3	-5	-7	4	-12	-8	-6	-16
B	-12	-8	6	7	-11	-5	-6	-8	-9	-13	-11	-6	-10	-12	-14	-5	-7	-14	-10	-12	14	-5	-8	-16
Z	-4	-9	-8	-6	-12	-8	3	-1	-13	-9	-9	-13	-11	-9	-12	-7	-5	-12	-11	-8	-5	15	-8	-16
X	-6	-8	-7	-8	-11	-7	-9	-7	-9	-7	-6	-9	-7	-6	-10	-7	-5	-8	-7	-6	-8	-8	1	-16
*	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	1

Figure 25 - An OMP specific substitution matrix called $BBTM_{out}$

(Jimenez-Morales and Liang 2011).

3.3. Incorporating OM Leaflet Composition Asymmetry into $E_z\beta$

Due to the low count of residues we faced in our first dataset of thirty-five TMBs, our previous design of $E_z\beta$ assumed symmetry of the lipid bilayer. That is, the outer and inner leaflets were assumed to be of similar composition. With a semi-automated homology model generating method as described in section 3.2, we have produced a total of 3042 homology models (Table 6), and can now gather propensities such that the bin (z in [1.5 Å, 4.5 Å]) is separate from (z in [-4.5 Å, -1.5 Å]). Once the effects of incorporating substitution matrices is better understood, the appropriately modified matrix will be the new input matrix in deriving the MSA for the homology model building process. Propensities were solved for similarly to those in $E_z\beta$, but this time depended on which leaflet the residue was counted in. Furthermore, to normalize against over-counting amino acids within a large cluster compared to those from smaller clusters, each count of a residue is inversely weighted by its associated cluster's size. Let cluster *j* have respective size j_{size}. The new formula for the propensity $P_{res,bin}$ is:

$$P_{res,bin} = \frac{n_{res,bin}}{n_{tot} f_{res} f_{bin}} = \frac{n_{res,bin} n_{tot}}{n_{res} n_{bin}} = \frac{\sum \frac{1}{j_{size}} n_{res,bin,j} \sum \frac{1}{j_{size}} n_{tot,j}}{\sum \frac{1}{j_{size}} n_{res,j} \sum \frac{1}{j_{size}} n_{bin,j}} \quad (9)$$

These four component sum functions can be computed. See Section 7.9 for an Excel VBA solution to computing $n_{res,bin}$ when each PDB's cluster is equally weighted.

Also with a larger dataset of amino acids, we can better quantify the behavior of amino acids in the context of quaternary structures. In particular, we can now consider the different depth-dependence profiles of amino acids in lipid-facing residues in both monomeric and oligomeric TMBs compared to those of amino acids at protein-protein interfaces of oligomeric TMBs only. The criteria as reported for

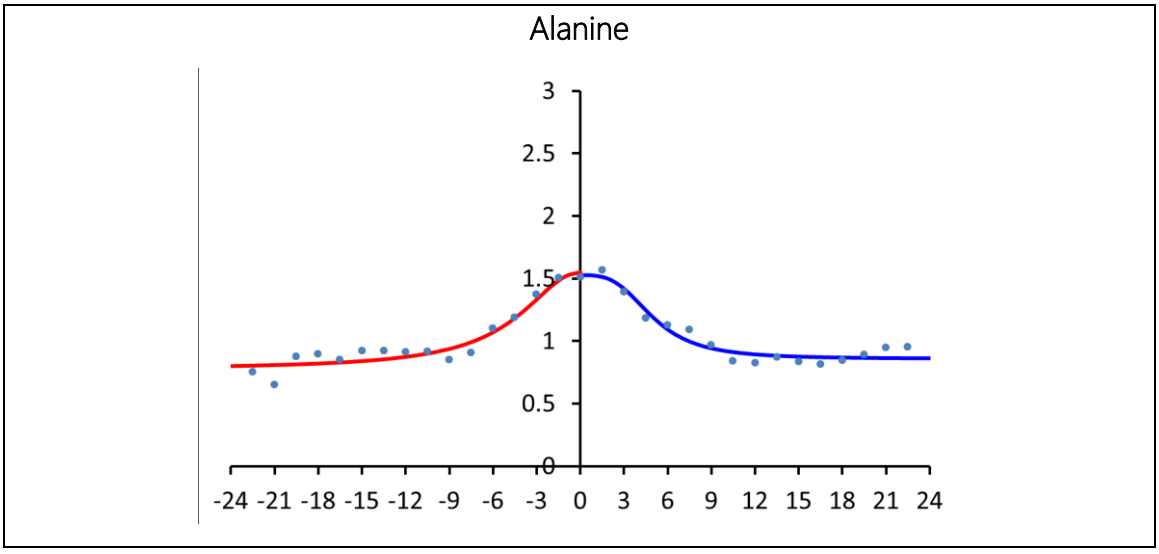
discerning PPI residues (40% change in SASA% using DSSP of monomer vs DSSP of oligomer in addition to the 20% SASA criteria) was used to produce the datasets for $P_z\beta_{PPI}$ and $P_z\beta_{non\text{-}PPI}$.

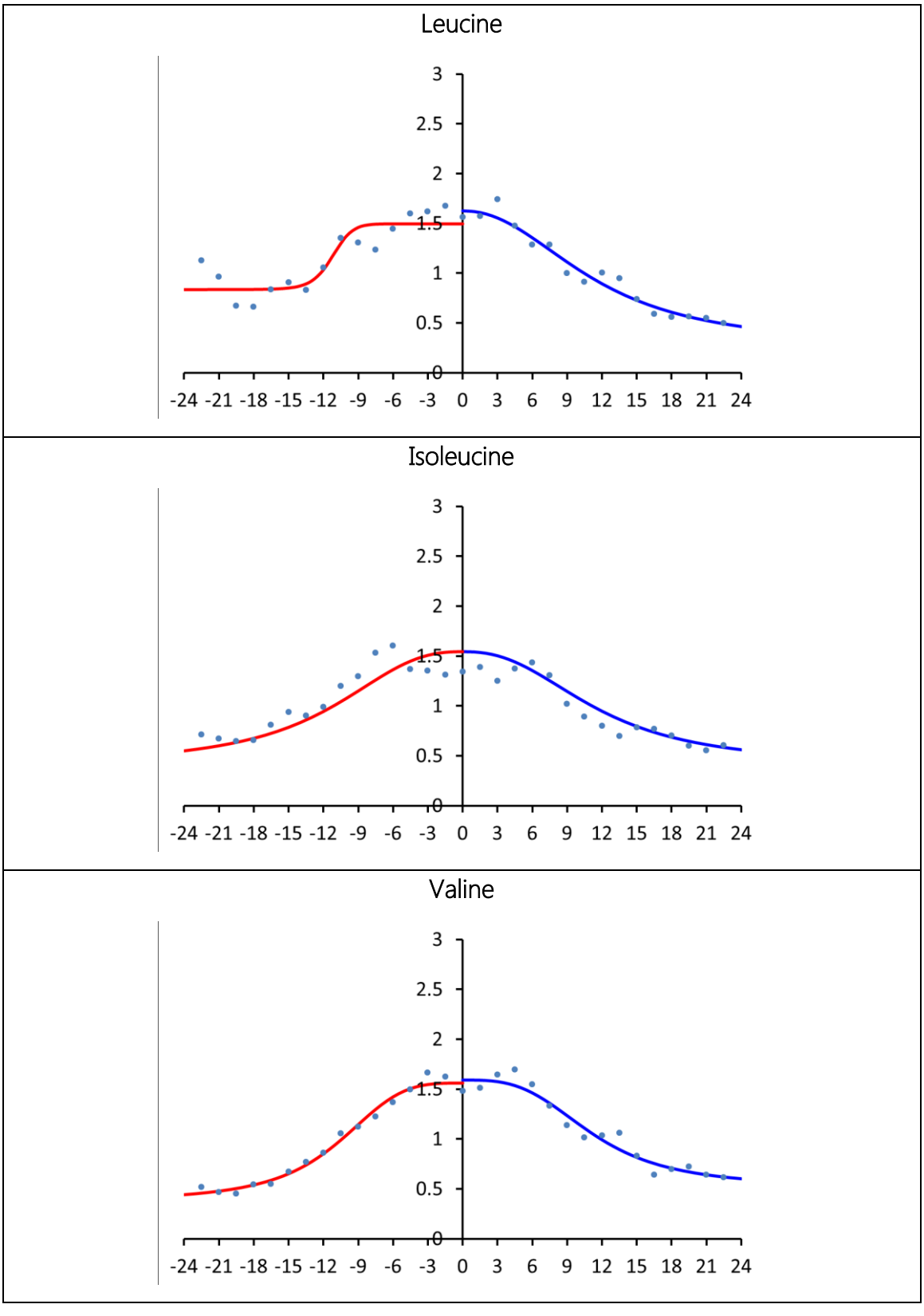
3.3.1 Preliminary data and analysis: $P_z\beta_{non\text{-}PPI}$ and $P_z\beta_{PPI}$

The following two tables are as follows: because we incorporated asymmetry of composition in the OM leaflets, amino acid propensities may exhibit a leaflet-dependence as well. This requires two further classes of $E_z\beta$ parameters to obtain: $E_z\beta_{\text{inner leaflet}}$ and $E_z\beta_{\text{outer leaflet}}$ for both PPI and non-PPI datasets. For each propensity chart in the table, blue lines represent non-linear fits for outer leaflet residue bin propensities. Those in red represent fits for inner leaflet residue bin propensities. All datapoints are midpoints of z bins. For example the midpoint of z bin [0,3] is 1.5 Å.

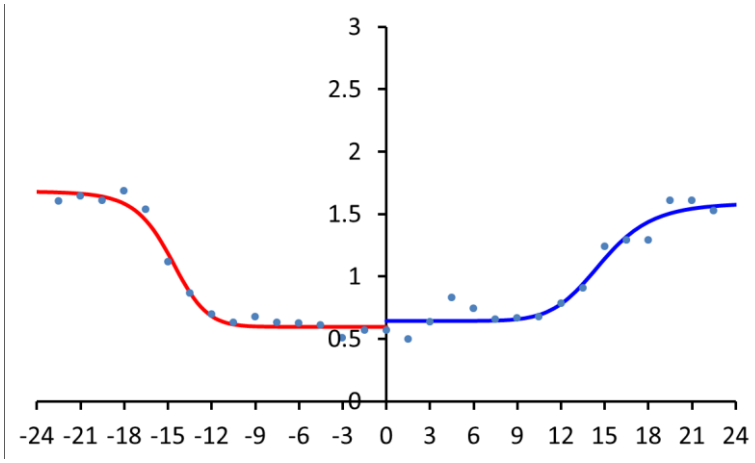
Table 5 - Depth-dependent propensities for non-PPI dataset $P_z\beta_{non\text{-}PPI}$

Propensity P_z as a function of depth z from center of the bilayer

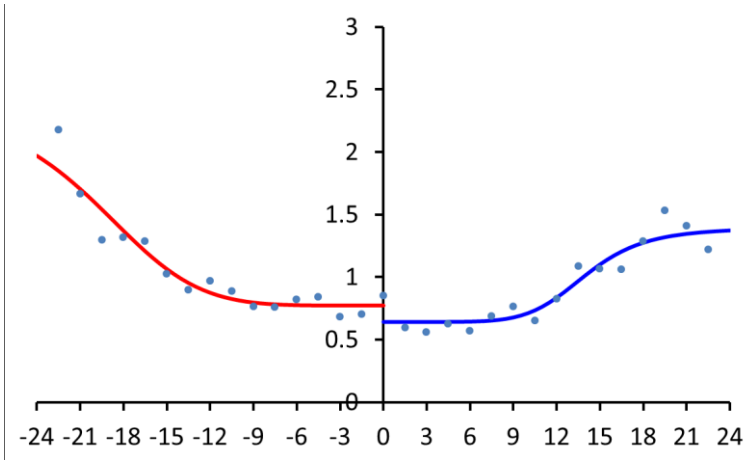



Leucine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Isoleucine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Valine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24

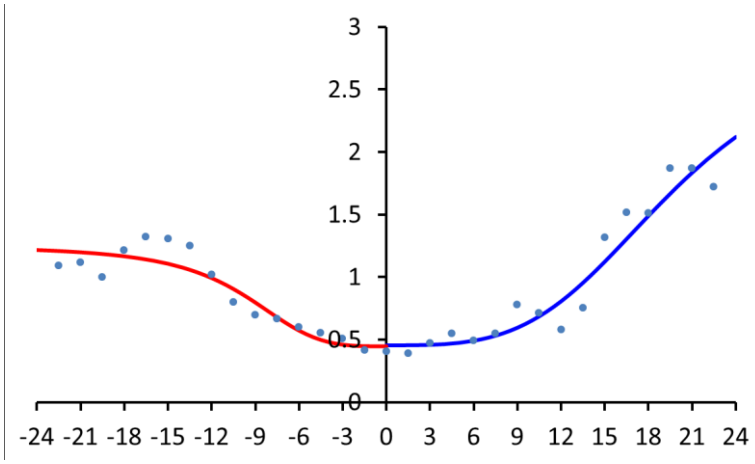
Aspartic Acid

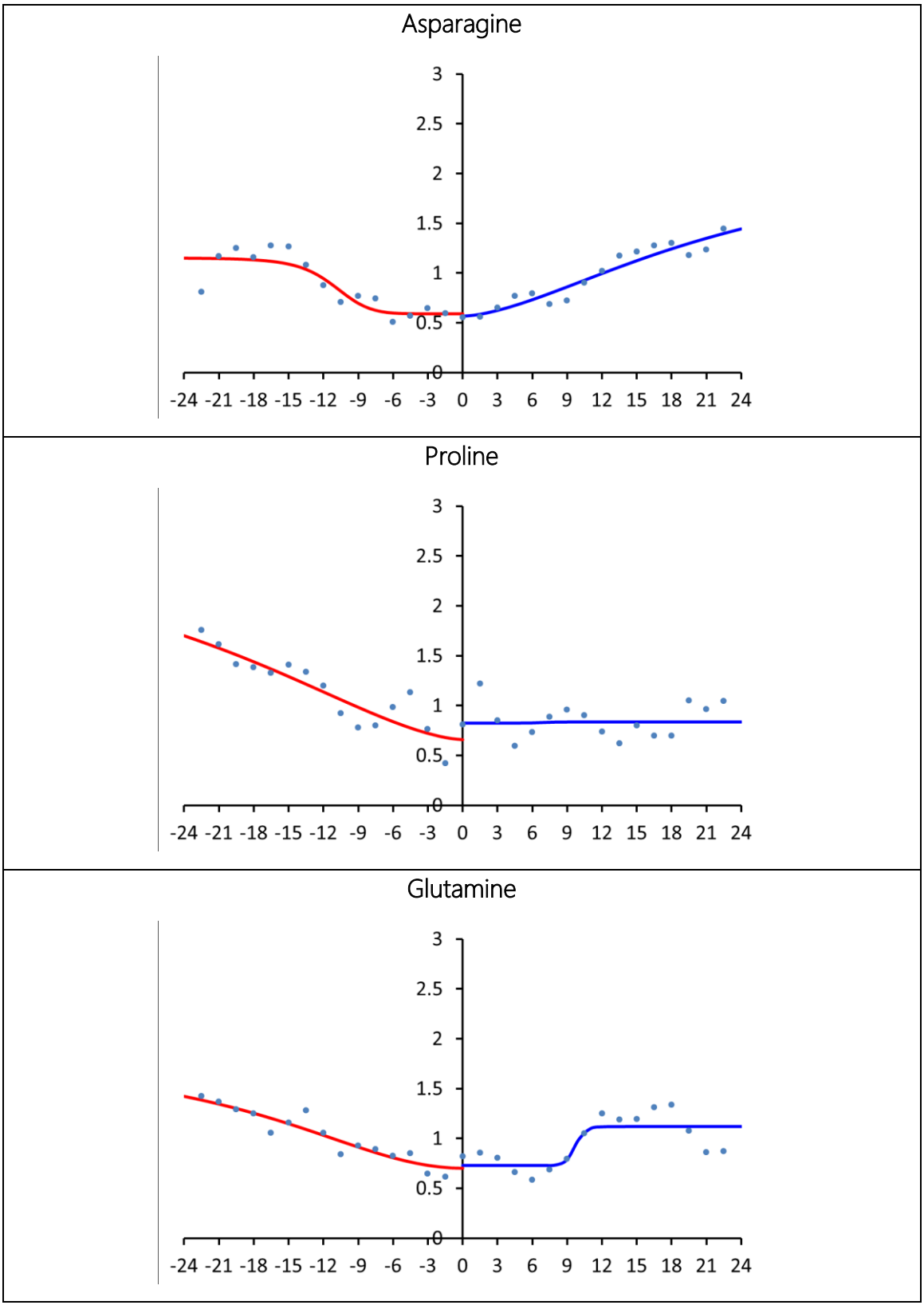


Glutamic Acid

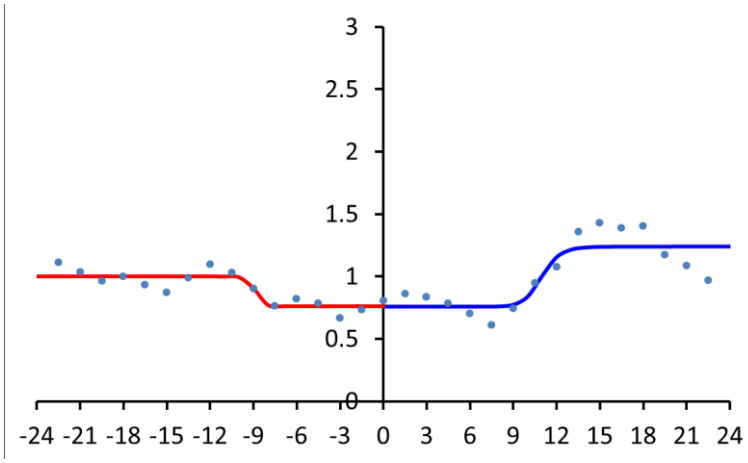


Lysine

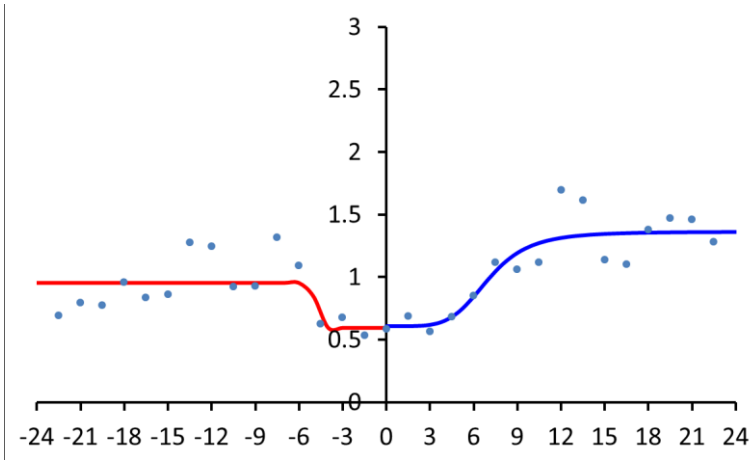



Asparagine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Proline
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Glutamine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24

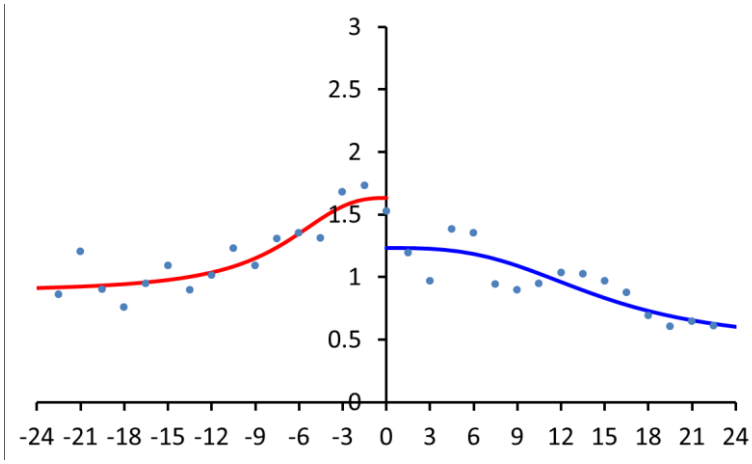
Arginine



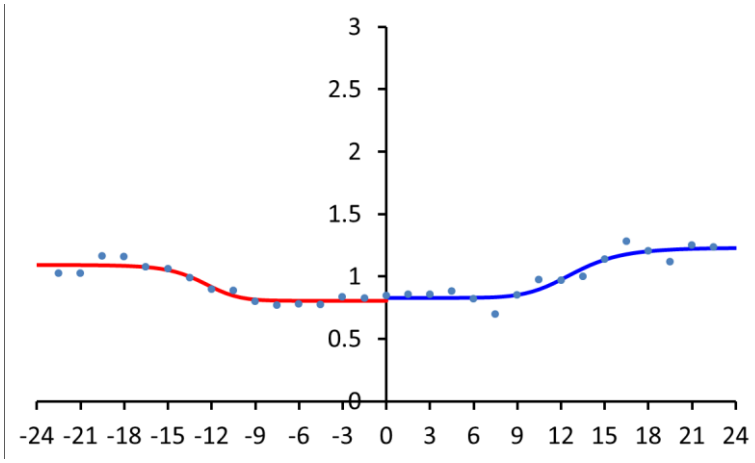
Histidine



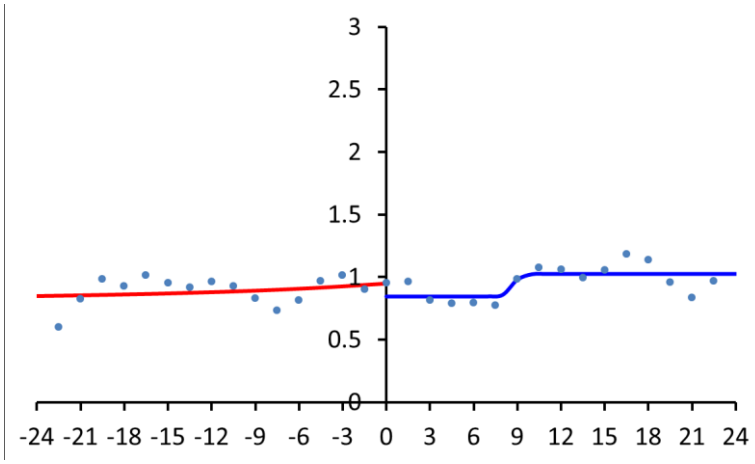
Methionine



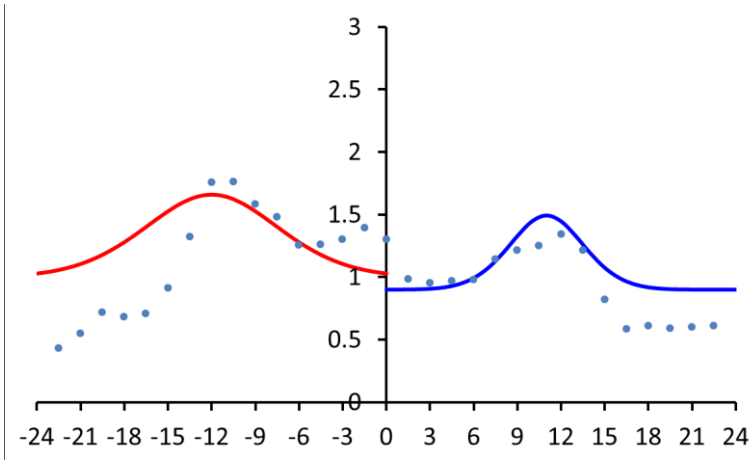
Serine



Threonine



Phenylalanine



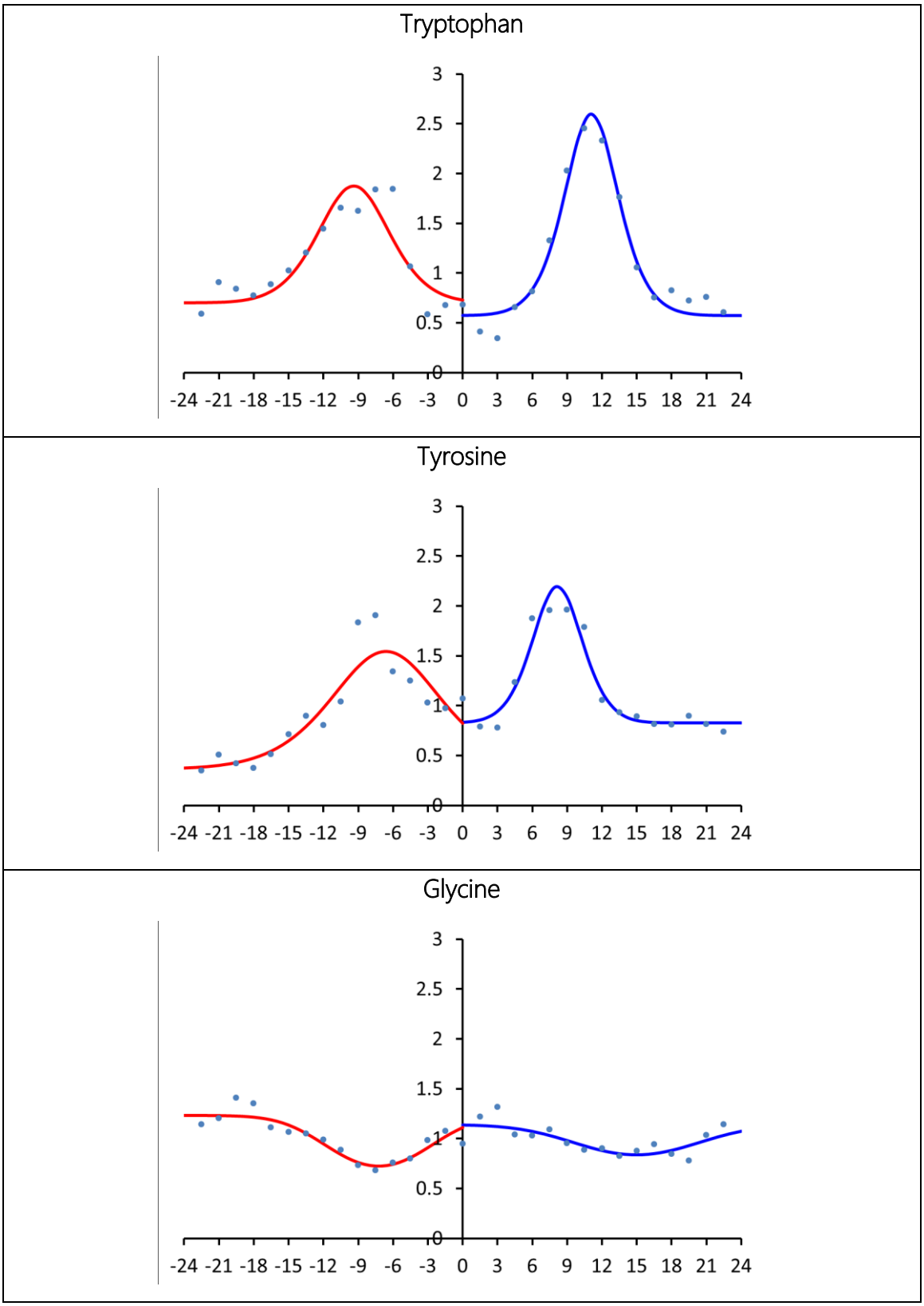
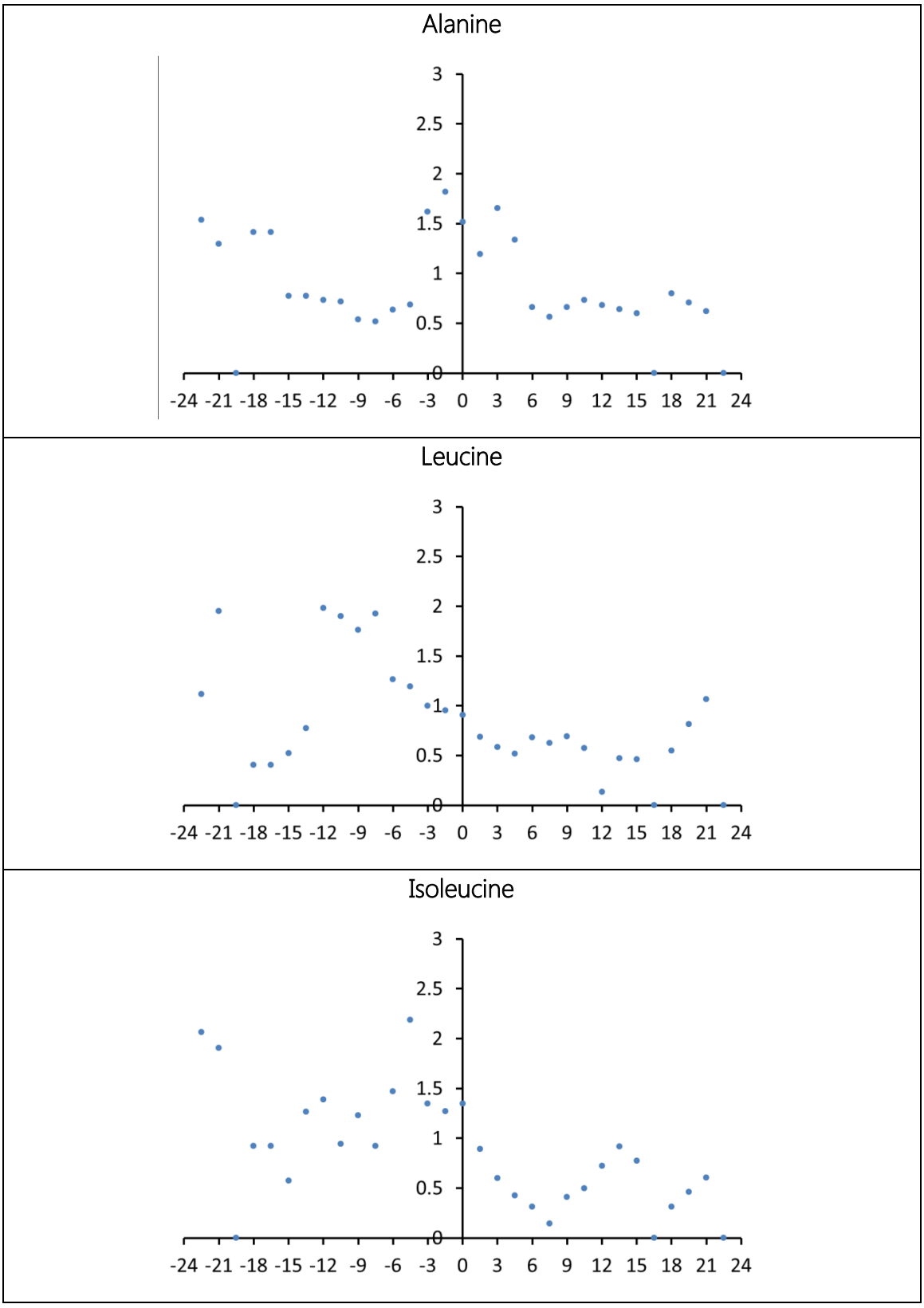
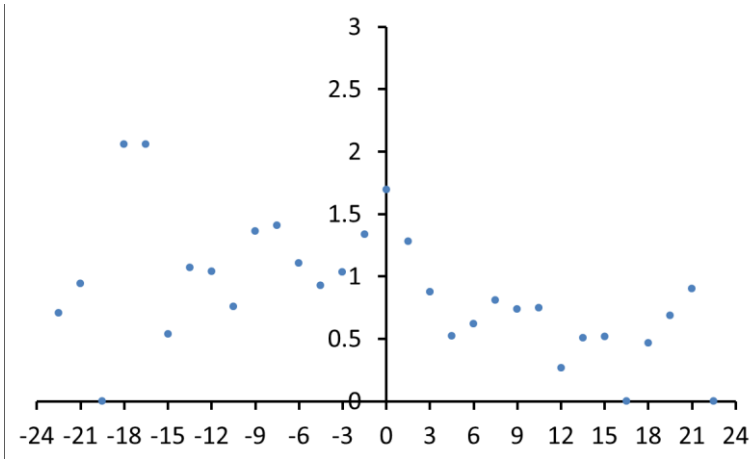

Tryptophan
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Tyrosine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Glycine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24

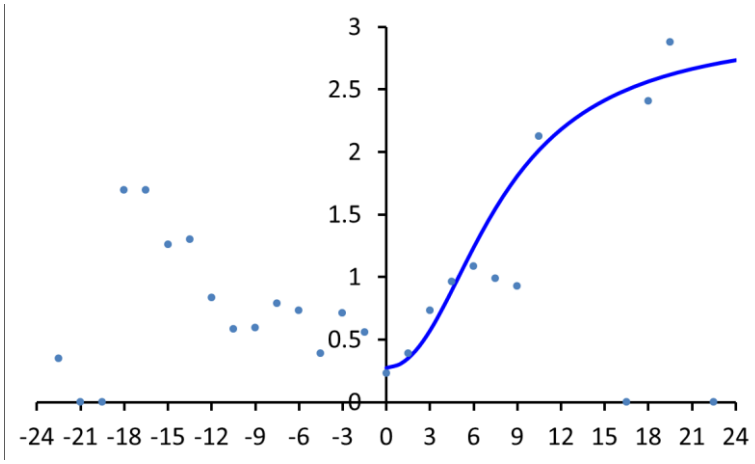
Table 6 - Depth-dependent propensities for PPI dataset $P_z\beta_{PPI}$



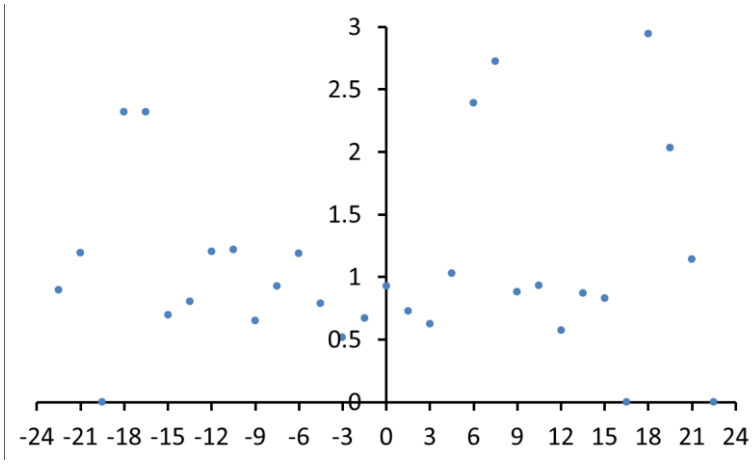
Valine

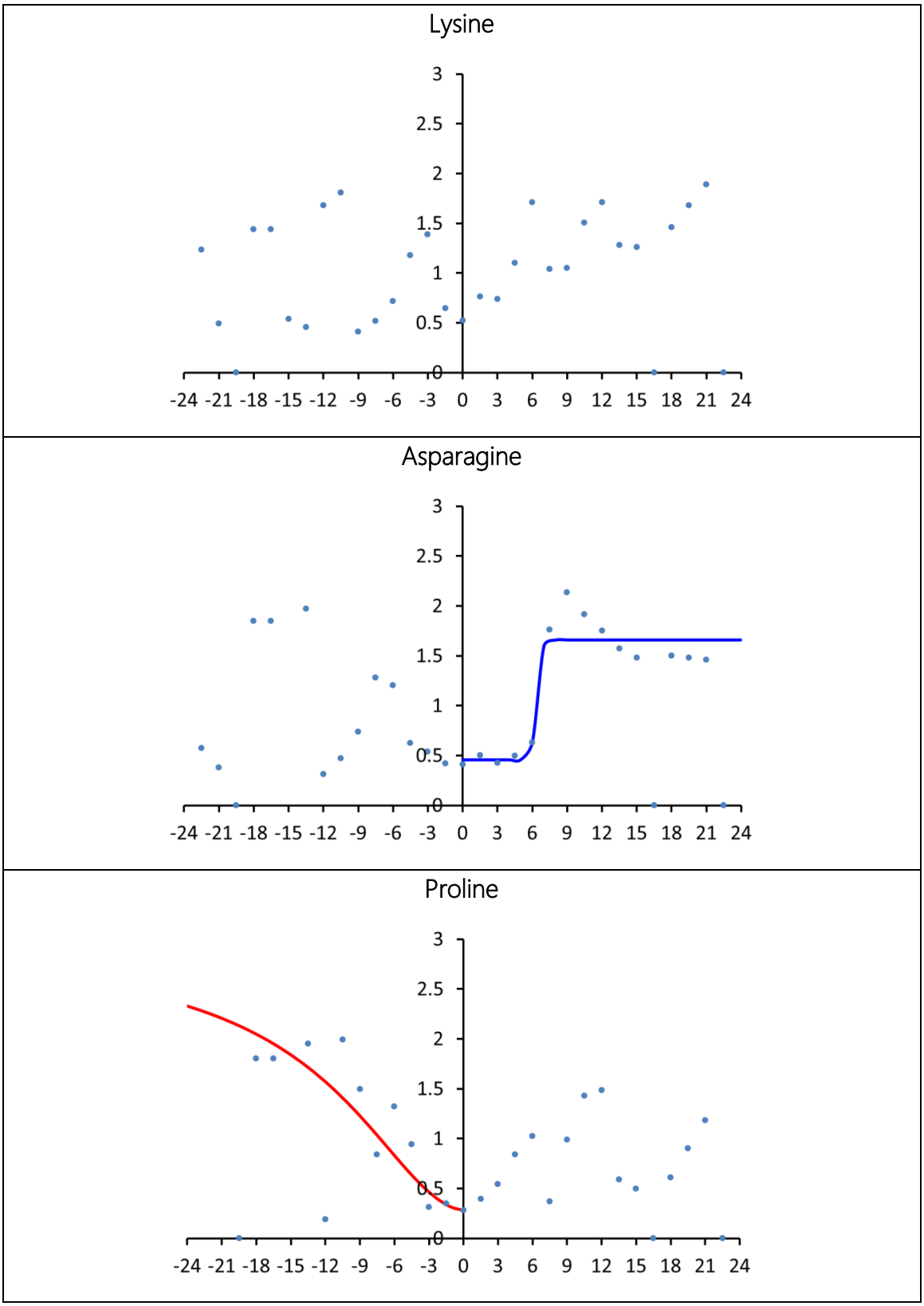


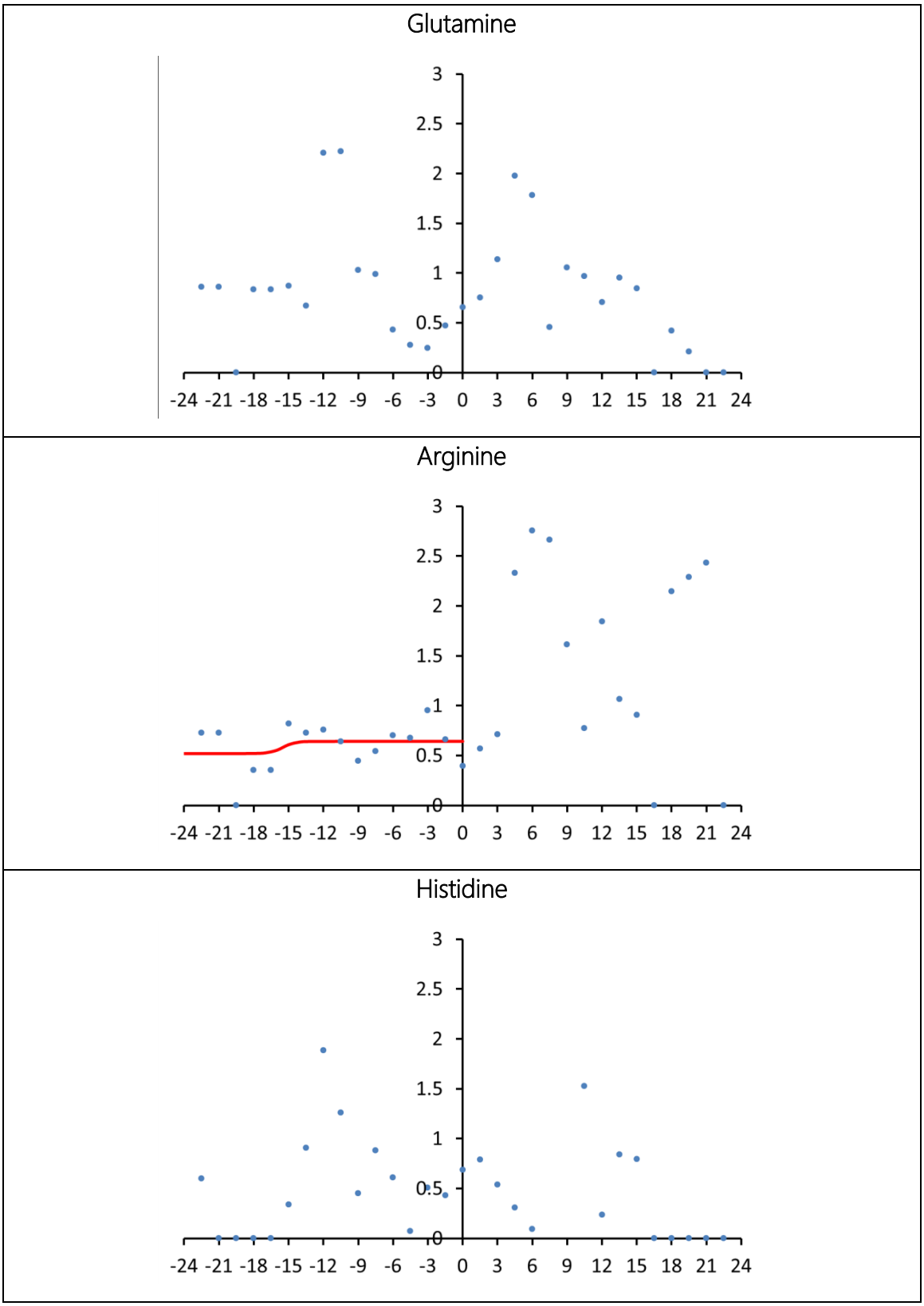
Aspartic Acid



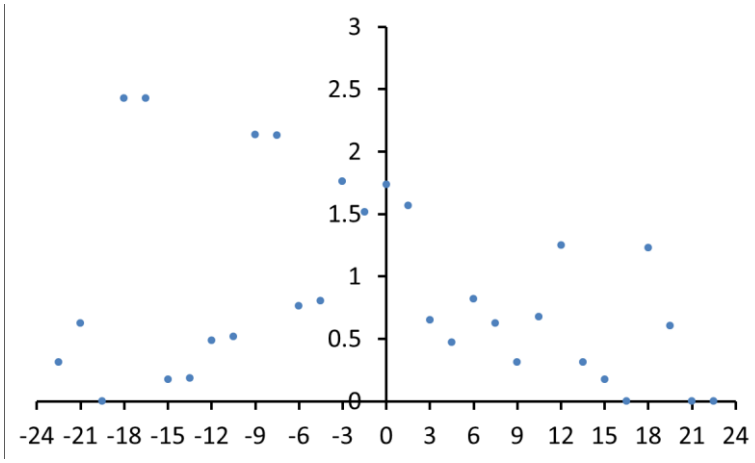
Glutamic Acid



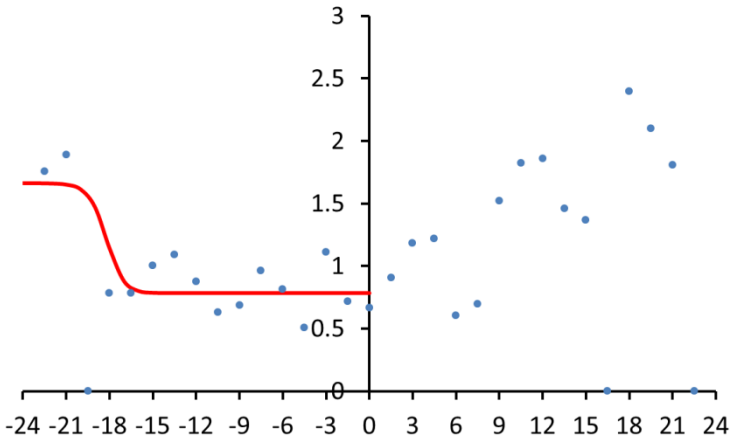

Lysine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Asparagine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Proline
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24


Glutamine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Arginine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24
Histidine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24

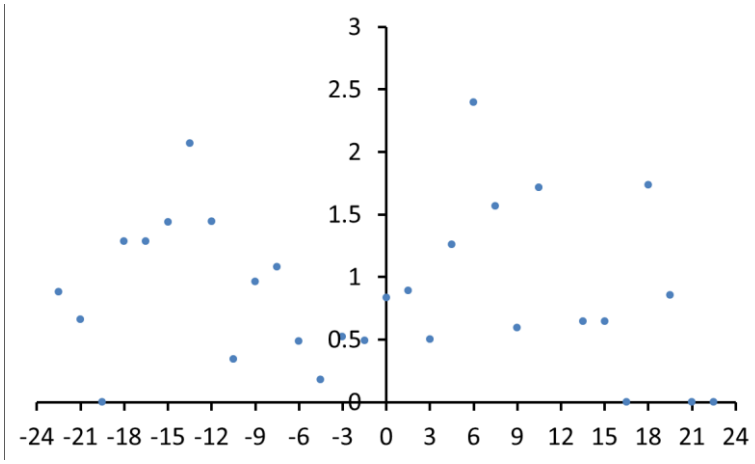
Methionine



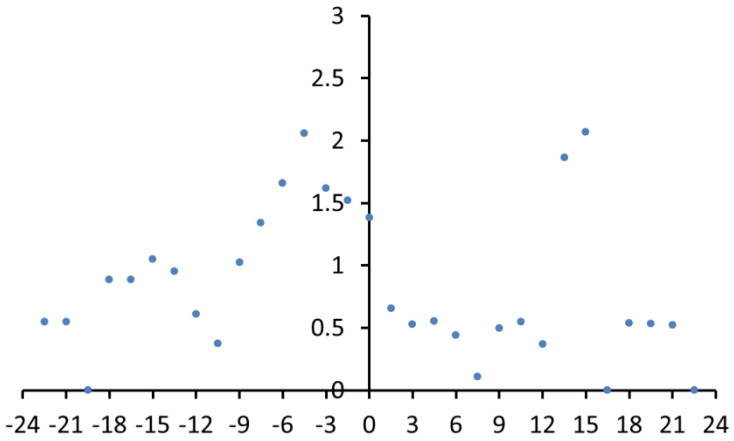
Serine



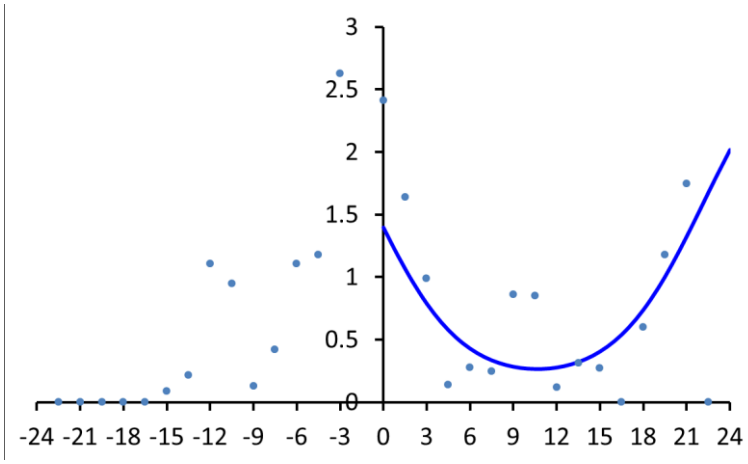
Threonine



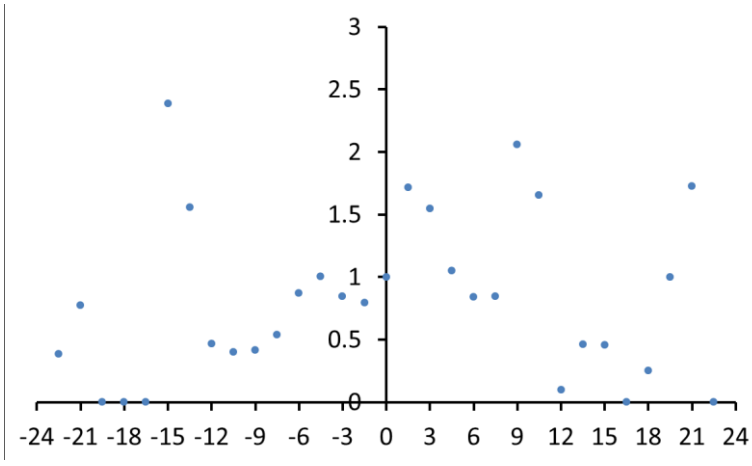
Phenylalanine

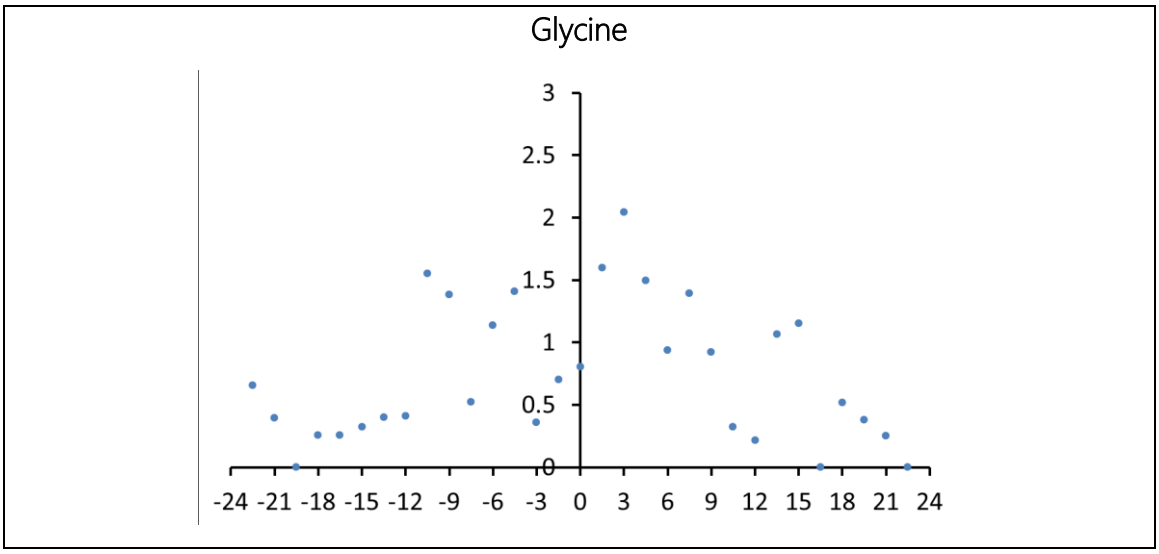


Tryptophan



Tyrosine




Glycine
3
2.5
2
1.5
1
0.5
0
-24 -21 -18 -15 -12 -9 -6 -3 0 3 6 9 12 15 18 21 24

To give a perspective of the size differences between the two datasets, non-PPI and PPI, the structurally weighted n_{tot} values given by

$$n_{tot} = \sum \frac{1}{j_{size}} n_{tot,j} \quad (10)$$

are 4425 and 138, respectively. For PPI residues, the next iteration of our method should aim to carefully incorporate more clusters. For non-PPI residues, there are clear signals consistent with the asymmetric composition of the outer membrane. In particular, the positively charged residue Lys is heavily favored at the outer leaflet of the bilayer. This high propensity makes sense due to an observed phenomenon called positive-outside rule due to the negative charges found in the LPS region.

The aromatics Trp and Tyr regain their prominence as exhibiting aromatic behavior. Our criteria of 20% SASA does not exclude residues facing inwards the barrel, which may contribute to a higher propensity at the outer leaflets. It is clear that Phe participates in aromatic behavior moreso than it does for nonpolar behavior, and is better fit according to a weighted combination of Gaussian and sigmoidal fits.

The aliphatics show symmetric behavior across both leaflets, implying that aliphatic placement is not dependent on leaflet environment, and that phospholipids in the inner leaflet and LPS in the outer leaflet show similar interactions with aliphatic residues near the center of the bilayer. Leucine, which was the highest propensity residue across the outer membrane according to the previous symmetric $E_z\beta$, now has a more dampened signal after inverse weights with respect to cluster size were introduced. In terms of protein design, aromatics are more likely to be selected at the headgroup region.

The distribution of glycines across the two leaflets is also interesting because the anti-Gaussian behavior is not as apparent in the outer leaflet as they are in the inner leaflet. Thus glycines at the extracellular cap region are still there to relieve structural tension, and rescued by the aromatics (Jackups and Liang 2005).

3.3.2 Future work

We have recently developed an asymmetric version of $E_z\beta$. and attempted to characterize amino acid depth-dependence at protein-protein interaction sites compared to lipid-facing residues. However due to low counts of PPI residues, depth-dependent propensity fits are far from finalized and require more ways to acquire a larger dataset without compromising its quality.

Because this positive-outside rule might be more applicable to larger proteins that reach further into LPS such as iron transporters and porins compared to smaller TMBs such as OMPLA, and because there can exist proteins of low sequence homology with similar fold, it may be important to investigate depth-dependent propensity profiles for clusters of unique folds. Perhaps quantifying similarities and differences between amino acid propensity profiles between any two homologous-fold clusters can lead us to building phylogenetic trees based on structure and function of OMPs, not just their sequences.

4. Excel as a tool in structural bioinformatics

Structural bioinformatics involves sorting, organizing, computing, analyzing and graphing data, and Excel is a suitable tool for all these needs. Although Excel is a spreadsheet application well-associated with financial analysis, it has potential use in the fields of physical, chemical and biological sciences. This section of the thesis describes tools that were built using Excel's programming language called Visual Basic for Applications (VBA).

There are two ways of programming Excel: The first is to automate and then manipulate a sequence of steps, usually starting from recording that sequence of steps into generated code called macros. If the user is unfamiliar with the names of functions and properties Excel has to offer, recording these macros can give insight on which relevant objects, functions, properties and events to call on, and more importantly, the order in which these components are called. These components can be combined to become powerful subunits of a larger code structure called procedures, which are collectively called modules. Hence, the Excel programming language, Visual Basic for Applications (VBA), is considered mainly a procedural programming language.

When the user records a macro, the user will notice that there are certain repeated elements used for the compiler to interpret such as Application, Workbook, WorkSheet, Range, Chart, etc. This is because VBA is also an object-oriented programming language, with the previous items as examples of basic objects the user can work with. Attached with the basic objects are many functions and properties, each thoroughly documented in the Microsoft Developer's Network Library as well as the Object Library. However, because VBA does not allow the use of inheritance, polymorphism, overloading of functions, encapsulation and abstraction, it is not a pure object-oriented programming language. Nevertheless, it is one of the most necessary languages used in the financial analysis industry, because of its ability to

create complex spreadsheets and forms, extending functionality by calling Windows-native and third-party software functions, deploy web services and more (Kimmel et al 2004)

Microsoft announced no plan to continue supporting VBA, but has recently offered an additional exciting avenue for Excel automation. Microsoft has developed tools for one to more easily develop and sell applications/software beyond those third-party software specialized in extending Office automation. This foresight in the early 2000's brings forth two important products to those interested in getting their feet wet in programming/software engineering: the .NET framework and Visual Studio. The .NET framework is a software development framework that facilitates programming as well as development of software across multiple operating systems and web applications across multiple browsers. Visual Studio is a free integrated development environment – it is a software that helps the developer organize and test run the layout (form) and the source code behind (function) his/her developing product. Visual Studio 2008/2010 Professional, along with other professional-edition software, is also freely available to students with a valid school email address from Microsoft's Dreamspark website (Microsoft). Both of these tools are thoroughly documented (Microsoft).

With the introduction of these two main products, Microsoft offers VBA developers a .NET-based alternative for automating Office called Visual Studio Tools for Office (VSTO) (Meister, 2008). VSTO seamlessly combines Office and desktop development, thus expanding the boundaries of software development. However, because VBA has been integrated deeply into all kinds of business and scientific operations, Microsoft plans to keep VBA in future shipments of Office (VSTOTeam, 2008).

This chapter will demonstrate how Excel's semi-object-oriented components coupled with procedural programming can be an effective approach for solving problems in structural bioinformatics of membrane proteins. First, we shall review the two most commonly used text file types used in encapsulating information on structural properties of proteins. The information in both file types can then be merged into one "hybrid" file type, which can be a powerful input file for performing bioinformatics calculations on the data.

Another feature of Excel that is highly useful for scientific visualization is the Charts object, which is automatable. In particular, this thesis will highlight the use of surface charts in visualizing dynamic energy landscapes.

4.1 File types used in structural bioinformatics

4.1.1 The Protein Data Bank file

The PDB is database of structural coordinates obtained by X-ray crystallography and nuclear magnetic resonance imaging methods. It can be accessed on the web at: http://www.pdb.org. There are currently almost 82,000 structures available for water-soluble, membrane, fibrillar, and other searchable protein/macromolecular structures (Bernstein et al. 1977). Knowing the 3-d structural coordinates, and thus the possible spatial configurations of a protein, is important to understanding the function of the protein. For example, it may elucidate the location of an active site of an enzyme or the constriction of a pathway of an ion channel.

The PDB is a fixed width delimited text file, meaning all data fields have a max character limit. As we near the ability to obtain structures of larger proteins, this may become problematic (boscoh.com, 2007) and the PDB format will have to support dynamic fields in the near future. Upon opening the PDB as a text file in Excel, the user is immediately prompted by the Text Import Wizard dialogue menu to manually format the PDB by assigning delimiters, which can be found here:

http://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html. Facing the same task for many PDBs becomes tedious and may require an automated solution. Create a list of PDB IDs (which are always four-characters and start with a number). The following formatting snippet can be used within a loop construct that iterates through the PDB ID list:

```
Workbooks.OpenText _
        filename:=PDB_FileAddress, _
        StartRow:=1, _
        DataType:=xlFixedWidth, _
        FieldInfo:= Array( _
                Array(0, 1), _
                Array(6, 1), _
                Array(11, 1), _
                Array(17, 1), _
                Array(20, 1), _
                Array(22, 1), _
                Array(28, 1), _
                Array(38, 1), _
                Array(46, 1))
```

Where "PDB_FileAddress" is a String variable for the full address of the file.

4.1.2 The Dictionary of Secondary Structure Prediction

The geometric patterning of hydrogen bonds can reveal the state of the secondary structure of an amino acid is participating in. Kabsch and Sanders' database is derived directly from the 3-D coordinates of the PDB (Kabsch and Sander 1983). In addition to the secondary structure label assigned to a residue, we are also interested in the "buriedness" of the amino acid, known as the solvent accessible surface area (SASA). By considering the protein as a surface collection of spheres, their algorithm takes a theoretical sphere that rolls along the surface and integrates the amount of surface area covered (Lee and Richards 1971). When an amino acid is fully buried, its SASA is 0.

Glyakina recently calculated the maximal accessible surface area of all amino acids from the PDB database (Glyakina et al. 2007). The metric we are interested in using is the percentage burial compared to maximal burial, which we call SASA% for short. In our derivation of the first generation of $E_z\beta$ parameters, we used residues that were considered 20% exposed.

4.1.3 The PDB/DSSP Hybrid file

The PDB/DSSP hybrid file serves as an input file towards Excel-programmed computational experiments. It is useful because it consists of both three-dimensional coordinates and secondary structure information. The protein's 3-D structural coordinates (PDB file) as well as the corresponding secondary structural information (DSSP file) are needed to create this specially formatted spreadsheet (Fig. 26).

The PDB/DSSP Hybrid spreadsheet contains a number of sheets: a table of $E_z\beta$ parameters as well as one for maximal accessible surface areas, the parsed and formatted PDBs and DSSPs, an aligned version of the PDB called "aligned_<PDB_ID>", and an intermediate file called the "<PDB_ID>_PreHybrid" that reports Cαs and Cβs of all amino acids (except glycine) for determining whether an amino acid is facing lipid or is buried inside the barrel structure. The final two sheets are the hybrid files. One contains calculations based on the aligned PDB and prehybrid information and the other is duplicated as values only, called "<PDB_ID>_Hybrid_Values". The Hybrid Values sheet is henceforth fed in as inputs to future computational experiments and analysis.

The data fields in the PDB/DSSP hybrid (for this work) are:

- PDB ID (4 characters)
- chain (1 character)
- the residue's index number
- 3-letter amino acid code
- 1-letter amino acid code
- secondary structure (1 character)
- solvent accessible surface area (SASA) as a percentage of the maximum accessible surface area (Glyakina et al. 2007)
- z-coordinate of the Cβ for all residues except glycine
- a binary number flagging when residue is glycine
- the magnitude of vector $\overrightarrow{C\alpha C\beta}$ after projection onto the XY-plane

For purposes of anticipating other bioinformatics studies with this tool, a PDB/DSSP hybrid file can be customized to have more or less columns of data depending on the problem at hand.

L8

	A	B	C	D	E	F	G	H	I	J	K
1	pdbID	Chain	ResIndex	3-Code	1-Code	Secondary	%SASA	zCoord of	GLY=1	vectProj	
2	1A0S	P	71	SER	S		0.590909	19.495	0	-1.53415	
3	1A0S	P	72	GLY	G		0.278846	16.781	1	1.451312	
4	1A0S	P	73	PHE	F	E	0.21519	13.432	0	1.461179	
5	1A0S	P	74	GLU	E	E	0.265116	12.994	0	-1.48804	
6	1A0S	P	75	PHE	F	E	0.050633	10.029	0	1.5295	
7	1A0S	P	76	HIS	H	E	0.23445	8.744	0	-1.50945	
8	1A0S	P	77	GLY	G	E	0.038462	5.345	1	1.48997	
9	1A0S	P	78	TYR	Y	E	0.1	2.526	0	-1.34211	
10	1A0S	P	79	ALA	A	E	0	1.886	0	1.406916	
11	1A0S	P	80	ARG	R	E	0.057554	-1.48	0	-1.32067	
12	1A0S	P	81	SER	S	E	0	-1.99	0	1.450794	
13	1A0S	P	82	GLY	G	E	0	-3.7	1	-1.52261	
14	1A0S	P	83	VAL	V	E	0	-6.436	0	1.547869	
15	1A0S	P	84	ILE	I	E	0.021053	-9.304	0	-1.48022	
16	1A0S	P	85	MET	M	E	0.230392	-11.002	0	1.477811	
17	1A0S	P	86	ASN	N	E	0.12234	-14.939	0	-0.40075	
18	1A0S	P	87	ASP	D	T	0.560847	-13.635	0	1.507475	
19	1A0S	P	88	SER	S	T	0.278409	-17.154	0	-0.38646	
20	1A0S	P	89	GLY	G	S	0.067308	-13.463	1	1.325202	
21	1A0S	P	90	ALA	A	S	0.132075	-17.266	0	0.379301	
22	1A0S	P	91	SER	S		0.318182	-16.323	0	-1.42372	
23	1A0S	P	92	THR	T		0.155914	-15.167	0	1.288035	
24	1A0S	P	93	LYS	K		0.307359	-16.885	0	-1.09075	
25	1A0S	P	94	SER	S		0.193182	-11.357	0	1.078616	

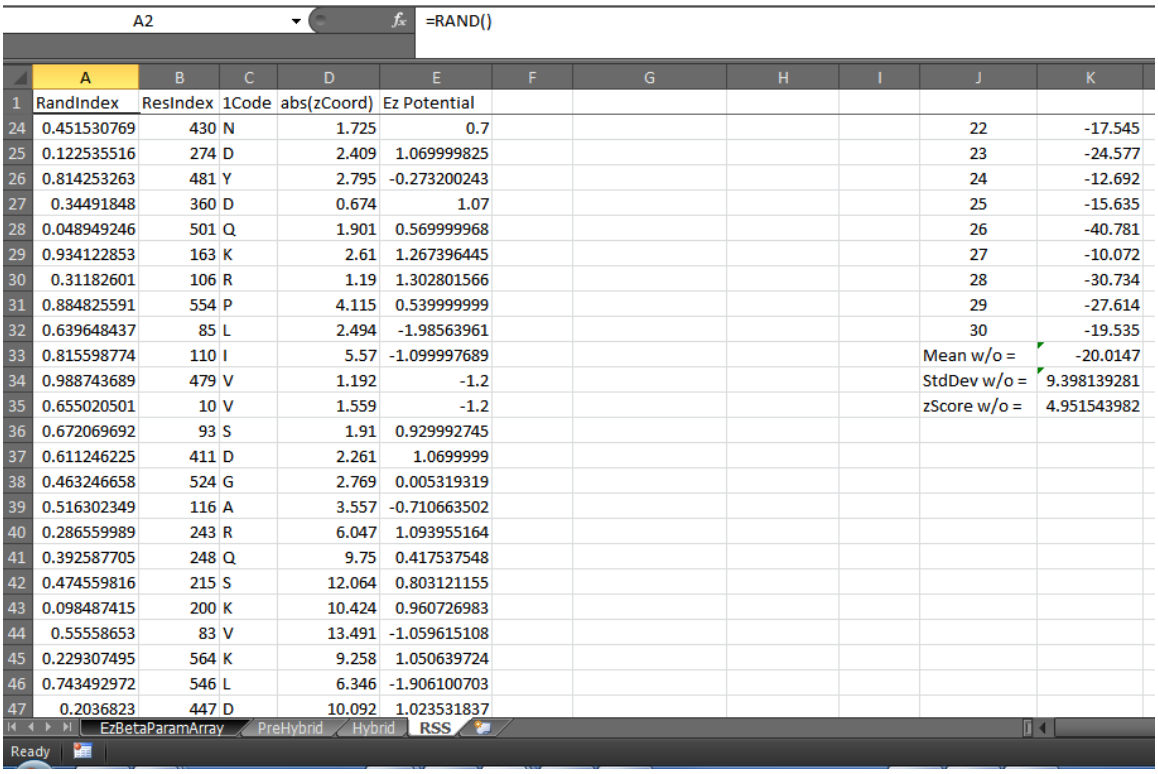
1A0S

Figure 26 - The PDB/DSSP Hybrid File

A hybrid file can be constructed however the user likes. For the purposes of assessing insertion energetics and structural prediction, this PDB/DSSP hybrid suffices as the key input for computational studies.

4.2 Random Mutagenesis of Protein Sequences using Excel

In order to simulate a randomization experiment in which all of the amino acids are swapped with each other, a nice Excel trick is to sort the residues using a random index. To achieve this, create a column of random numbers using the rand() function and a column of residues. This gives a decimal value between 0 and 1. Choose the sort range to include only the columns of residue index numbers and the original random indices. Link the amino acid single-letter codes with the corresponding index numbers using a VLookup function, which looks up an array of user's interest to find a corresponding value given the position of the search value. Rand() has a special property that allows it to recalculate all cell formulas upon the user hitting "refresh calculations". Thus, all cell formulas with Rand() will have a new random number between 0 and 1. Since we can sort only the residue indices according to these random number indices, we will essentially have randomly swapped residues. (Fig. 27) The thirty trials of randomization were automated, each time outputting the new total $E_z\beta$ score, where a derived $E_z\beta$ parameters file along with the aligned PDB hybrid file are fed as input. (Fig. 28) At the end of the thirty trials, a standard deviation and z-score is determined. High z-score indicates a higher overall sensitivity to changes in z-depth.



A2 =RAND()

	A	B	C	D	E	J	K
1	RandIndex	ResIndex	1Code	abs(zCoord)	Ez Potential		
24	0.451530769	430	N	1.725	0.7	22	-17.545
25	0.122535516	274	D	2.409	1.069999825	23	-24.577
26	0.814253263	481	Y	2.795	-0.273200243	24	-12.692
27	0.34491848	360	D	0.674	1.07	25	-15.635
28	0.048949246	501	Q	1.901	0.569999968	26	-40.781
29	0.934122853	163	K	2.61	1.267396445	27	-10.072
30	0.31182601	106	R	1.19	1.302801566	28	-30.734
31	0.884825591	554	P	4.115	0.539999999	29	-27.614
32	0.639648437	85	L	2.494	-1.98563961	30	-19.535
33	0.815598774	110	I	5.57	-1.099997689	Mean w/o =	-20.0147
34	0.988743689	479	V	1.192	-1.2	StdDev w/o =	9.398139281
35	0.655020501	10	V	1.559	-1.2	zScore w/o =	4.951543982
36	0.672069692	93	S	1.91	0.929992745		
37	0.611246225	411	D	2.261	1.0699999		
38	0.463246658	524	G	2.769	0.005319319		
39	0.516302349	116	A	3.557	-0.710663502		
40	0.286559989	243	R	6.047	1.093955164		
41	0.392587705	248	Q	9.75	0.417537548		
42	0.474559816	215	S	12.064	0.803121155		
43	0.098487415	200	K	10.424	0.960726983		
44	0.55558653	83	V	13.491	-1.059615108		
45	0.229307495	564	K	9.258	1.050639724		
46	0.743492972	546	L	6.346	-1.906100703		
47	0.2036823	447	D	10.092	1.023531837		

EzBetaParamArray | PreHybrid | Hybrid | RSS

Ready

Figure 27 - Performing random mass mutagenesis in Excel

The data in column "RandIndex" consist only of random numbers between 0 and 1 as defined by the cell formula "=RAND()". Everytime the spreadsheet is refreshed (intentionally or not), all of these random numbers change value simultaneously. This feature could be used to perform mass mutagenesis for an entire protein or a customized selection of amino acids.

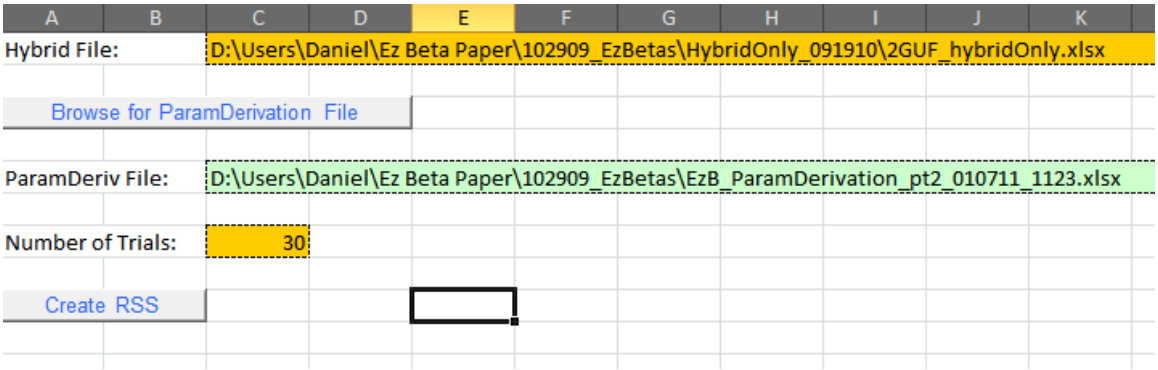


Figure 28 - The Random Sequence Swapping (RSS) User Interface

Parameters for creating the RSS file are highlighted in color.

4.3 Visualization of a 4-D Surface Chart in Excel

Visualizations that are low-cost in memory are desirable. We present a method for stitching three dimensional scattered data from multiple worksheets into a dynamic "animation-like" surface chart in Excel. This method is useful when (1) the user hard-codes the data points to conserve memory; employing such strategy scales better than soft-coding data values, (2) the data values are hard-coded by an unknown source, or (3) the function is complex and requires a user-defined function to output values into cells. In particular, we demonstrate an application in biology where rigid motion (rotation and translation are the only transformations applied to an object in 3-D space) is used to model the free energy gain/loss by surveying various placements and orientations of membrane proteins with respect to their environment. Our strategy involves a simple concept of scrolling through an order of worksheets, and can be extended to even more dimensions (i.e. scrolling through workbooks if necessary)

4.3.1 Visualizing Multidimensional Data such as Dynamic Energy Landscapes in Excel

Excel is an excellent choice for developing solutions to problems in the sciences requiring organization and visualization of data. We say Excel is an object model, because almost all of its tools and components can be automated through an extensive assortment of functions and properties arranged in hierarchical fashion and available for the user to manipulate and call.

Excel is a container of sheets of data and has many functions useful for financial analysis. Since the data could be hypothetically organized in any of the higher dimensions n>=3 (for example, carefully constructed discrete arrays of three-dimensional data per worksheet), it is possible to represent data in n+1 dimensions when considering the data across the entire workbook. Such organized data could be "stitched" together to form a more visual form of media, perhaps a more continuous animation, and provide insight into dynamics of the subject being studied.

One highly manipulable Excel object is the Charting object, consisting of a plethora of chart types and all of their properties. When a Chart is paired with a control tool such as a scrollbar, the user has now created (either manually or automatically) a dynamic chart. In terms of energy landscapes, individual landscapes can fit each slice of a collection of energy landscapes in the form of surface charts. The motion of "flipping through" the scrollbar values, which is indirectly linked with the dynamic chart, can offer the sense of animation.

We were interested in validating that the TMB's energy landscape "funnels" most (Onuchic, Luthey-Schulten, and Wolynes 1997) where the $E_z\beta$ is at its lowest, representing its natural depth and angular conformation in the membrane. Every energy landscape "snapshot" was calculated using a fixed z-depth from -40 to 40 Å. Rotations about x- and y-axis were performed at increments of 10° from -90° to 90°. The following subsections describe the code behind generating such animation tools in Excel, which are meant only for visualizing the $E_z\beta$ energies and that could be extended to all kinds of multidimensional visualization purposes.

4.3.2 Preparing the Dynamic Range: The "Setup_MainSheet" Subroutine

Viewing the data using our dynamic surface chart to view individual 3-D slices of 4-D animation is like using a microfilm machine to view individual slides of archived newspaper articles. The machine is composed of a magnifying lens and a slide sorter consisting of multiple slides. The "Setup_Mainsheet" subroutine is responsible for loading a particular slide's data to be magnified and viewed. The worksheet responsible for preparing this view at the particular 3-D data is called "DataForPlot". As with the microfilm machine, the user manually scrolls to a certain slide, which is properly indexed. Similarly, our data is indexed in numerical order from -40 to 40 Å (for the biology example). "DataForPlot" also reports the index number of the slide for viewing. Since the scrollbar can only take numbers 1, 2, 3 and so on, the index number of the 3-D data needs to be translated into the scrollbar value via a simple mathematical formula. Depending on the scrollbar value, and therefore the index number of the 3-D data to "view", the dynamic range reports the corresponding data values of the particular 3-D slice of the 4-D animation.

4.3.3 Displaying the Surface Chart Animation: The "Setup_PlotSheet" Subroutine

While the "DataForPlot" sheet is like the prepared slide for microfilm viewing, the surface chart animation is the viewer itself. The "Setup_PlotSheet" subtroutine prepares both the surface chart visualization tool and the scrollbar. The visualization tool is the interface between the user and the data. To control the index number of the 3-D slice of the 4-D animation, the subroutine generates a scrollbar, whose value dictates which particular 3-D data to view. When the scrollbar value is changed, the value of index reported in "DataForPlot" updates accordingly, and vice versa.

4.3.4 Giving the Surface Chart Analytical Meaning: the "ColorGrad" Function

When a surface chart is created, Excel does not create meaningful legend colors. They start off as random colors. There exist useful websites demonstrating how

to implement gradient-based color schemes (Pope 2007) for heat maps (contour plots) (Blumenstock 2003, The "How" Blog) and therefore surface charts by extension. "ColorGrad" paints the surface chart with a meaningful color gradient, so that visualization and analysis will be easier for those reading the surface chart.

4.3.5 Putting It All Together with the "test_S3DP" Subroutine

The test subroutine, here called "test_S3DP" (short for testing surface 3-D plot), is a subroutine without arguments used to call the two main subroutines: one that generates a sheet "Setup_MainSheet" with virtual data, and another "Setup_PlotSheet" that generates the surface plot. It is necessary to write at least one subroutine without arguments because subroutines with arguments cannot be directly run unless the arguments are indicated in a call. Therefore this type of subroutine call needs to be performed within a testing subroutine. The next page shows a schematic of the interplay between all the sheets created through the VBA code we have discussed (Fig. 29).

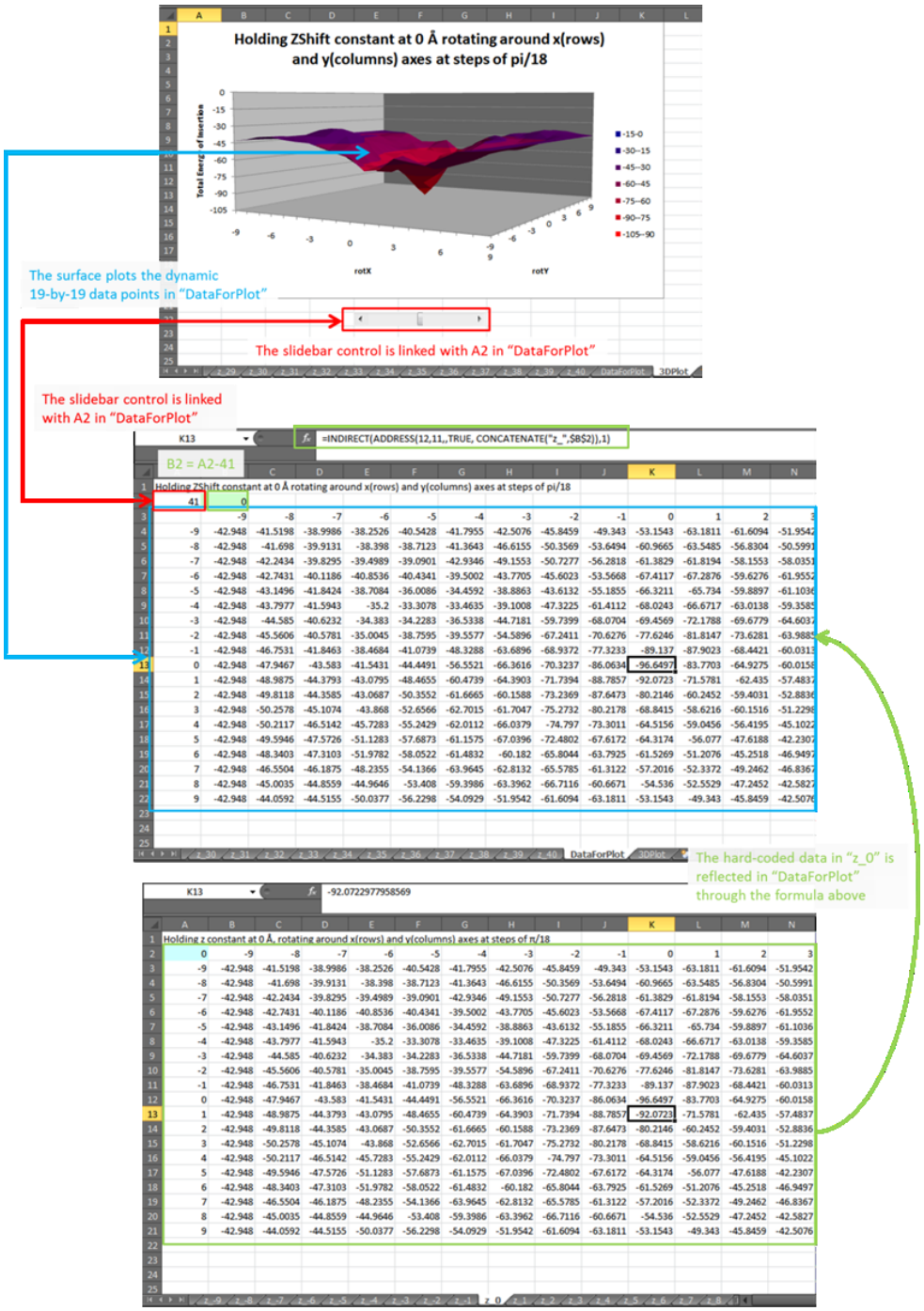


Figure 29 - A schematic of sheets "z_-40" to "z_40", "DataForPlot" and "3DPlot" interacting

4.4 Creating Animations of The Insertion Pathway using Excel

The concept of multiple models in NMR-based PDB structures could be extended and applied to creating animations as well. Given that the energy landscape files (see 4.3) are prepared such that they contain all energy values at all combinations of rotations about x- and y-axis coupled, arranged in primary order of z-depth where each sheet represents a unique z-depth from z= -40 to 40 Å.

An Excel VBA script was written to shell into our Linux terminal using PuTTY commands) and submit the z-aligned PDB to ProtCAD, an in-house protein design software library we continue to develop and maintain (Summa 2002) in order to apply rigid transformations according to angular conformations (θ_x, θ_y) corresponding to the best energy value reported for that z-depth slice.

Once the VBA script has accessed the Linux terminal, it is no longer in the realm of Windows. However, there is a helpful function called "SendKeys" which simulates keystrokes given to any machine, including the user's own. "SendKeys" is useful in automating scriptable software. For example, PyMOL, a molecular visualization software, is scriptable but only takes in Python-based scripts. One way around this, although tedious, is to automate a VBA script that automates customized PyMOL scripts. One application for taking this approach is visualizing customized selections of residues as shown in the PDB/DSSP hybrid file, which Excel can perform sorting and filtering upon for certain criteria. For example, using the hybrid file, one can quickly apply the Autofilter tool to retrieve the residue indices of all residues that have greater than or equal to 20% SASA in a TMB. Then a customized button correctly wired with a script can call PyMOL to open, call the PDB file, and create a selection based on the visible residue indices from the Autofilter selection in the hybrid file. Such extreme automation has been attempted with success in our lab.

5. Future Studies and Conclusion

There has been recent uptrend in studying OMPs. We have made major progress in structural and topology prediction and understanding their assembly pathways. However, the insertion of OMPs is still elusive and difficult to study using experimental setups. We have recently developed and are continuing to improve a knowledge-based potential specifically for understanding patterns in amino acid depth-dependence, called $E_z\beta$. These propensities of amino acids, which reflect their efficiency to partition into a range of depths from the center of the lipid bilayer, can be converted into an insertion energy term. An advantage to using this statistical potential is its cost-efficiency in *de novo* designing or redesigning OMPs.

We showed even with the assumption of symmetric bilayers, that $E_z\beta$ was able to predict orientation and higher-order structural features such as protein-protein interaction sites. One issue we have not yet addressed is the determination of the center of the bilayer in our starting assumptions. For symmetric $E_z\beta$, the center of the bilayer was determined as the z-coordinate of the barycenter of TM strands as calculated previously (Lomize et al. 2006). It may be necessary to conduct a study validating general statistical potentials by using the given rules to iteratively calculate optimal orientations of theoretical TMBs.

6. References

Achtman M, Neibert M, Crowe B, Strittmatter W, Kusecek B, Weyse E, Walsh M, Slawig B, Morelli G, Moll A. 1988. Purification and characterization of eight class 5 outer membrane protein variants from a clone of Neisseria meningitidis serogroup A. J. Experimental Med. 168(2):507-525.

Adamian L, Naveed H, Liang J. 2011. Lipid-binding surfaces of membrane proteins: evidence from evolutionary and structural analysis. Biochimica et Biophysica Acta (BBA)-Biomembranes 1808(4):1092-1102.

Andersen OS, Koeppe RE. 2007. Bilayer thickness and membrane protein function: an energetic perspective. Annu. Rev. Biophys. Biomol. Struct. 36:107-130.

Armon A, Graur D, Ben-Tal N. 2001. ConSurf: an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. J. Mol. Biol. 307(1):447-463.

Ashkenazy H, Erez E, Martz E, Pupko T, Ben-Tal N. 2010. ConSurf 2010: calculating evolutionary conservation in sequence and structure of proteins and nucleic acids. Nucleic Acids Res. 38:W529-W533.

Bennun SV, Hoopes MI, Xing C, Faller R. 2009. Coarse-grained modeling of lipids. Chemistry and physics of lipids 159(2):59-66.

Bernstein FC, Koetzle TF, Williams GJ, Meyer Jr. EE, Brice MD, Rodgers JR, Kennard O, Shimanouchi T, Tasumi M. 1978. The Protein Data Bank: A Computer-based

Archival File for Macromolecular Structures. Arch. of Biochem. And Biophys. 185(2):584-591.

Bitto E, McKay DB. 2003. The periplasmic molecular chaperone protein SurA binds a peptide motif that is characteristic of integral outer membrane proteins. J. Biol. Chem. 278:49316-49322.

Blaber M, Zhang XJ, Matthews BW. 1993. Structural basis of amino acid alpha helix propensity. Science 260(5114):1637-1640.

Blumenstock J. 2003. Heatmap Tool (VBA). Retrieved from: http://senorjosh.jblumenstock.com/archives/2003/04/heatmap_tool_vba.shtml Accessed: November 15, 2011.

Booth PJ, Clarke J. 2010. Membrane protein folding makes the transition. Proc. Nat. Acad. Sci. 107(9):3947-3948.

Botelho AV, Huber T, Sakmar TP, Brown MF. 2006. Curvature and hydrophobic forces drive oligomerization and modulate activity of rhodopsin in membranes. Biophys. J. 91(12):4464-4477.

Bowie JU. 2004. Membrane proteins: a new method enters the fold. Proc. Nat. Acad. Sci. USA 101(12):3995-3996.

Brandl M, Weiss MS, Jabs A, Suhnel J, Hilgenfeld R. 2001. C-H ⋯ π-interactions in proteins. J. Mol. Biol. 307(1):357-377.

Braun V, Rehn K. 1969. Chemical Characterization, Spatial Distribution and Function of a Lipoprotein (Murein-Lipoprotein) of the *E. coli* Cell Wall. Eur. J. Biochem. 10(3):426-438.

Burgess NK, Dao TP, Stanley AM, Fleming KG. 2008. β-Barrel proteins that reside in the *Escherichia coli* outer membrane *in vivo* demonstrate varied folding behavior *in vitro*. J. Biol. Chem. 283:26748-26758.

Campbell RL (2004) My PyMOL script repository. Available at http://pldserver1.biochem.queensu.ca/~rlc/work/pymol.

Caputo GA, London E. 2003. Cumulative effects of amino acid substitutions and hydrophobic mismatch upon the transmembrane stability and conformation of α-helices. Biochemistry 42(11):3275-3285.

Carter DM, Gagnon JN, Damlaj M, Mandava S, Makowski L, Rodi DJ, Pawelek PD, Coulton JW. 2006. Phage Display Reveals Multiple Contact Sites between FhuA, an Outer Membrane Receptor of *Escherichia coli*, and TonB. J. Mol. Biol. 357(1):236-251.

Cepeda MS, Boston R, Farrar JT, Strom BL. 2003. Comparison of logistic regression versus propensity score when the number of events is low and there are multiple confounders. Am. J. Epidemiol. 158(3):280-287.

Chamberlain AK, Lee Y, Kim S, Bowie JU. 2004. Snorkeling preferences foster an amino acid composition bias in transmembrane helices. J. Mol. Biol. 339(2):471-479.

Chen M, Khalid S, Sansom MSP, Bayley H. 2008. Outer membrane protein G: Engineering a quiet pore for biosensing. Proc. Nat. Acad. Sci. 105(17):6272-6277.

Choma C, Gratkowski H, Lear JD, DeGrado WF. 2000. Asparagine-mediated self-association of a model transmembrane helix. Nat. Struct. Mol. Biol. 7:161-166.

CodePlex. (n.d.). ".NET Bio". Retrieved from: http://bio.codeplex.com . Accessed: April 23, 2012.

Cordes FS, Bright JN, Sansom MSP. 2002. Proline-induced distortions of transmembrane helices. J. Mol. Biol. 323:951-960.

Davis JH, Clare DM, Hodges RS, Bloom M. 1983. Interaction of a synthetic amphiphilic polypeptide and lipids in a bilayer structure. Biochemistry 22:5298-5305.

Dawson JP, Melnyk RA, Deber CM, Engelman DM. 2003. Sequence context strongly modulates association of polar residues in transmembrane helices. J. Mol. Biol. 331(1):255-262.

Dayhoff MO, Schwartz RM, Orcutt BC. 1978. A model for evolutionary change in proteins. In: Dayhoff MO, editor. Altas of Protein Sequence and Structure. Vol 5. Washington DC: Nat. Biochem. Res. Found. p. 345-352.

Debnath D, Nielsen KL, Otzen D. 2010. *In vitro* association of fragments of a β-sheet membrane protein. Biophys. Chem. 148:112-120.

Dong C, Beis K, Nesper J, Brunkan-LaMontagne AL, Clarke BR, Whitfield C, Naismith JH. 2006. Wza the translocon for *E. coli* capsular polysaccharides defines a new class of membrane protein. Nature 444:226-229.

Eisenberg D, Weiss RM, Terwilliger TC. 1984. The hydrophobic moment detects periodicity in protein hydrophobicity. Proc. Nat. Acad. Sci. 81(1):140-144.

Ellena JF, Lackowicz P, Mongomery H, Cafiso DS. 2011. Membrane Thickness Varies Around the Circumference of the Transmembrane Protein BtuB. Biophys. J. 100(5):1280-1287.

Emberly EG, Mukhopadhyay R, Tang C, Wingreen NS. 2004. Flexibility of β-sheets: Principal component analysis of database protein structures. Proteins: Structure, Function, and Bioinformatics 55(1):91-98.

Engelman D, Steitz T, Goldman A. 1986. Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. Ann. Rev. Biophys. Biophys. Chem. 15:321-353.

Engelman DM. 2005. Membranes are more mosaic than fluid. Nature 438:578-580.

Ferguson AD, Chakraborty R, Smith BS, Esser L, van der Helm D, Deisenhofer J. 2002. Structural basis of gating by the outer membrane transporter FecA. Science 295:1715-1719.

Geula S, Naveed H, Liang J, Shoshan-Barmatz V. 2012. Structure-based Analysis of VDAC1 Protein. J. Biol. Chem. 287:2179-2190.

Glyakina AV, Garbuzynskiy SO, Lobanov MY, Galzitskaya OV. 2007. Different packing of external residues can explain differences in the thermostability of proteins from thermophilic and mesophilic organisms. Bioinformatics 23(17):2231-2238.

Gouaux E, Hobaugh M, Song L. 1997. α-Hemolysin, γ-hemolysin, and leukocidin from Staphylococcus aureus: Distant in sequence but similar in structure. Protein science 6(12):2631-2635.

Grosse W, Essen LO, Koert U. 2011. Strategies and Perspectives in Ion-Channel Engineering. ChemBioChem 12(6):830-839.

Gu Y, Li D. 1999. The van der Waals Interaction between a Spherical Particle and a Cylinder. Journal of Colloid and Interface Science 217(1):60-69.

Hagan CL, Kim S, Kahne D. 2010. Reconstitution of outer membrane protein assembly from purified components. Science 328(5980):890-892.

Hagan CL, Silhavy TJ, Kahne D. 2011. β-Barrel membrane protein assembly by the Bam complex. Ann. Rev. Biochem. 80:189-210.

Hayat S, Elofsson A. 2012. BOCTOPUS: improved topology prediction of transmembrane β-barrel proteins. Bioinformatics 28(4):516-522.

Hayat S, Park Y, Helms V. 2011. Statistical analysis and exposure status classification of transmembrane β-barrel residues. Comp. Biol. Chem. 35(2):96-107.

Hessa T, Kim H, Bihlmaier K, Lundin C, Boekel J, Andersson H, Nilsson IM, White SH, Von Heijne G. 2005. Recognition of transmembrane helices by the endoplasmic reticulum translocon. Nature 433:377-381.

Hessa T, Meindl-Beinker NM, Bernsel A, Kim H, Sato Y, Lerch-Bader M, Nilsson IM, White SH, Von Heijne G. 2007. Molecular code for transmembrane-helix recognition by the Sec61 translocon. Nature 450:1026-1030.

Heuzenroeder MW and Reeves P. 1981. The tsx protein of Escherichia coli can act as a pore for amino acids. J. Bacteriol. 147(3):1113-1116.

Hiller S, Wagner G. 2009. The role of solution NMR in the structure determinations of VDAC-1 and other membrane proteins. Curr. Op. Struct. Biol. 19(4):396-401.

Ho BK. Parsing PDB Files: Sometimes You Really Should Reinvent the Wheel. Available from: http://boscoh.com/protein/parsing-pdb-files-sometimes-you-really-should-reinvent-the-wheel

Last accessed: March 12, 2012.

Ho BK, Curmi PMG. 2002. Twist and shear in β-sheets and β-ribbons. J. Mol. Biol. 317(2):291-308.

Hong H, Park S, Jiménez RHF, Rinehart D, Tamm LK. 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, Tamm LK. 2004. Elastic coupling of integral membrane protein stability to lipid bilayer forces. Proc. Nat. Acad. Sci. of the United States of America 101(12):4065-4070.

Hsieh D, Davis A, Nanda V. 2012. A knowledge-based potential highlights unique features of membrane α-helical and β-barrel protein insertion and folding. Protein Science. 21(1):50-62.

Huschilt JC, Millman BM, Davis JH. 1989. Orientation of a-helical peptides in a lipid bilayer. Biochim. Biophys. Acta – Biomembranes 979(1):139-141.

Huysmans GHM, Radford SE, Brockwell DJ, Baldwin SA. 2007. The N-terminal helix is a post-assembly clamp in the bacterial outer membrane protein PagP. J. Mol. Biol. 373(3):529-540.

Inokuchi K, Mutoh N, Matsuyama S, Mizushima S. 1982. Primary structure of the ompF gene that codes for a major outer membrane protein of *Escherichia coli* K-12. Nucleic Acids Res. 10(21):6957-6968.

Introduction to Protein Data Bank Format. Last modified August 2011. Available from: http://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html

Irbäck A, Mitternacht S. 2008. Spontaneous β-barrel formation: An all-atom Monte Carlo study of Aβ16-22 oligomerization. Proteins: Structure, Function, and Bioinformatics 71(1):207-214.

Ishi JN, Okajima Y, Nakae T. 1981. Characterization of lamB Protein from the Outer Membrane of Escherichia coli that Forms Diffusion Pores Selective For Maltose-Maltodextrins. FEBS Letters. 134(2):217-220.

Jackson SE, Craggs TD, Huang J. 2006. Understanding the folding of GFP using biophysical techniques. Exp. Rev. Proteom. 3(5):545-559.

Jackups R, Cheng S, Liang J. 2006. Sequence motifs and antimotifs in β-barrel membrane proteins from a genome-wide analysis: the Ala-Tyr dichotomy and chaperone binding motifs. J. Mol. Biol. 363(2):611-623.

Jackups R, Liang J. 2005. Interstrand pairing patterns in β-barrel membrane proteins: The positive-outside rule, aromatic rescue, and strand registration prediction. J. Mol. Biol. 354(4):979-993.

Jackups R, Liang J. 2010. Combinatorial analysis for sequence and spatial motif discovery in short sequence fragments. Comp. Biol. Bioinformatics, IEEE/ACM Transactions on 7(3):524-536.

Jaramillo A, Wernisch L, Héry S, Wodak SJ. 2002. Folding free energy function selects native-like protein sequences in the core but not on the surface. Proc. Nat. Acad. Sci. 99(21):13554.

Jaud S, Fernández-Vidal M, Nilsson IM, Meindl-Beinker NM, Hübner NC, Tobias DJ, Von Heijne G, White SH. 2009. Insertion of short transmembrane helices by the Sec61 translocon. Proc. Nat. Acad. Sci. 106(28):11588-11593.

Jiménez-Morales D, Adamian L, Liang J. 2008. Detecting remote homologues using scoring matrices calculated from the estimation of amino acid substitution rates of beta-barrel membrane proteins. Eng. in Med. and Biol. Soc., 2008. 30th Annual International Conference of the IEEE; 2008 Aug 20-25; Vancouver, BC.

Jiménez-Morales D, Liang J. 2011. Pattern of Amino Acid Substitutions in Transmembrane Domains of β-Barrel Membrane Proteins for Detecting Remote Homologs in Bacteria and Mitochondria. PLoS One 6(11):e26400.

Kabsch W, Sander C. 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22(12):2577-2637.

Kamio Y, Nikaido H. 1976. Outer membrane of Salmonella typhimurium: accessibility of phospholipid head groups to phospholipase c and cyanogen bromide activated dextran in the external medium. Biochemistry 15(12):2561-2570.

Killian JA, von Heijne G. 2000. How proteins adapt to a membrane-water interface. Trends in Biochemical Sciences 25(9):429-433.

Kleffel B, Garavito RM, Baumeister W, Rosenbusch JP. 1985. Secondary structure of a channel-forming protein: porin from *E. coli* outer membranes. EMBO J. 4(6):1589-1592.

Kleinschmidt JH, Bulieris PV, Qu J, Dogterom M, den Blaauwen T. 2011. Association of neighboring β-strands of outer membrane protein A in lipid bilayers revealed by site directed fluorescence quenching. J. of Mol. Biol. 407(2):316-332.

Kleinschmidt JH, Tamm LK. 1996. Folding intermediates of a β-barrel membrane protein. Kinetic evidence for a multi-step membrane insertion mechanism. Biochemistry 35(40):12993-13000.

Kleinschmidt JH, Tamm LK. 2002. Secondary and tertiary structure formation of the [beta]-barrel membrane protein OmpA is synchronized and depends on membrane thickness. J. Mol. Biol. 324(2):319-330.

Kleinschmidt JH. 2006. Folding kinetics of the outer membrane proteins OmpA and FomA into phospholipid bilayers. Chemistry and physics of lipids 141(1-2):30-47.

Koebnik R. 1999. Membrane assembly of the *Escherichia coli* outer membrane protein OmpA: exploring sequence constraints on transmembrane β-strands. J. Mol. Biol. 285(4):1801-1810.

Koh E, Kim T. 2005. Minimal surface as a model of β-sheets. Proteins: Structure, Function, and Bioinformatics 61(3):559-569.

Labischinski H, Barnickel G, Bradaczek H, Naumann D, Rietschel ET, Giesbrecht P. 1985. High state of order of isolated bacterial lipopolysaccharide and its

possible contribution to the permeation barrier property of the outer membrane. J. Bacteriol. 162(1):9-20.

Larkin M, Blackshields G, Brown N, Chenna R, McGettigan P, McWilliam H, Valentin F, Wallace I, Wilm A, Lopez R, et al. 2007. Clustal W and Clustal X version 2.0. Bioinformatics 23(21):2947-2948.

Lasdon LS, Fox RL, Ratner MW. 1973. Nonlinear optimization using the generalized reduced gradient method. Cleveland (OH): Case Western Reserve Univ. Cleveland OH Dept of Operations Research. Accession No.: AD0774723.

Lasters I, Wodak SJ, Alard P, Van Cutsem E. 1988. Structural principles of parallel beta-barrels in proteins. Proc. Nat. Acad. Sci. 85(10):3338-3342.

Lazaridis T. 2003. Effective energy function for proteins in lipid membranes. Proteins: Structure, Function, and Bioinformatics 52(2):176-192.

Lee B, Richards FM. 1971. The interpretation of protein structures: estimation of static accessibility. J. Mol. Biol. 55(3):379-IN4.

Lenaz G. 1987. Lipid fluidity and membrane protein dynamics. Bioscience Reports 7(11):823-837.

Lew S, Ren J, London E. 2000. The effects of polar and/or ionizable residues in the core and flanking regions of hydrophobic helices on transmembrane conformation and oligomerization. Biochemistry 39(32):9632-9640.

Lewis DJ (2010) Volunteered geographic information - a geography/GIS blog. Available at http://danieljlewis.org/2010/07/09/computing-the-geometric-median-in-python/

Locher KP, Rees B, Koebnik R, Mitschler A, Moulinier L, Rosenbusch JP, Moras D. 1998. Transmembrane signaling across the ligand-gated FhuA receptor: crystal structures of free and ferrichrome-bound states reveal allosteric changes. Cell 95(6):771-778.

Lomize AL, Pogozheva ID, Lomize MA, Mosberg HI. 2006. Positioning of proteins in membranes: a computational approach. Protein science 15(6):1318-1333.

Lomize MA, Lomize AL, Pogozheva ID, Mosberg HI. 2006. OPM: orientations of proteins in membranes database. Bioinformatics 22(5):623-625.

McLachlan AD. 1979. Gene duplications in the structural evolution of chymotrypsin. J. Mol. Biol. 128(1):49-79.

Meier W, Nardin C, Winterhalter M. 2000. Reconstitution of channel proteins in (polymerized) ABA triblock copolymer membranes. Angewandte Chemie International Edition 39(24):4599-4602.

Meister, C. (2008, May 25). "Please Read First – What is VSTO and Non-VSTO Resources".

Retrieved from:

http://social.msdn.microsoft.com/Forums/en-US/vsto/thread/063a23a6-1595-4c83-a25f-6c94658c4649/

Accessed: October 31, 2011.

Meng G, Fronzes R, Chandran V, Remaut H, Waksman G. 2009. Protein oligomerization in the bacterial outer membrane (Review). Molecular membrane biology 26(3):136-145.

Merkel JS, Regan L. 1998. Aromatic rescue of glycine in [beta] sheets. Folding and Design 3(6):449-456.

Microsoft Dreamspark. (n.d.) Retrieved from: https://www.dreamspark.com/ Accessed: October 3, 2011.

Microsoft Developer Network. (n.d.) ".NET Development". Retrieved from: http://msdn.microsoft.com/library/aa139615

Accessed: September 20, 2011.

Minor Jr DL, Kim PS. 1994. Measurement of the β-sheet-forming propensities of amino acids. Nature 367(6464):660-663.

Mitra K, Ubarretxena-Belandia I, Taguchi T, Warren G, Engelman DM. 2004. Modulation of the bilayer thickness of exocytic pathway membranes by membrane proteins rather than cholesterol. Proc. Nat. Acad. Sci. USA 101(12):4083-4088.

Mohammad MM, Howard KR, Movileanu L. 2011. Redesign of a Plugged β-Barrel Membrane Protein. J. Biol. Chem. 286:8000-8013.

Morona R, Klose M, Henning U. 1984. *Escherichia coli* K-12 outer membrane protein (OmpA) as a bacteriophage receptor: analysis of mutant genes expressing altered proteins. J. Bacteriol. 159(2):570-578.

Mouritsen O, Bloom M. 1984. Mattress model of lipid-protein interactions in membranes. Biophys J. 46:141-153.

Murzin AG, Lesk AM, Chothia C. 1994. Principles determining the structure of β-sheet barrels in proteins Ⅱ. The observed structures. J. Mol. Biol. 236(5):1382-1400.

Nava J, Kreinovich V. 2012. Towards Symmetry-Based Explanation of (Approximate) Shapes of Alpha-Helices and Beta-Sheets (and Beta-Barrels) in Protein Structure. Symmetry 4(1):15-25.

Naveed H, Jackups R, Liang J. 2009. Predicting weakly stable regions, oligomerization state, and protein-protein interfaces in transmembrane domains of outer membrane proteins. Proc. Nat. Acad. Sci. 106(31):12735-12740.

Naveed H, Xu Y, Jackups R, Liang J. 2012. Predicting three-dimensional structures of transmembrane domains of β-barrel membrane proteins. J. Am. Chem. Soc. 134(3):1775-1781.

Novotný J, Bruccoleri RE, Newell J. 1984. Twisted hyperboloid strophoid as a model of the β-barrels in proteins. J. Mol. Biol. 177(3):567-573.

Mizuno T, Chou MY, Inouye M. 1983. A comparative study on the genes for three porins of the *Escherichia coli* outer membrane. DNA sequence of the osmoregulated ompC gene. J. Biol. Chem. 258:6932-6940.

Muhammad N, Dworeck T, Fioroni M, Schwaneberg. Engineering of the *E. coli* outer Membrane Protein FhuA to overcome the Hydrophobic Mismatch in Thick Polymeric Membranes. J. Nanobiotechnology 9:8.

Nanda V, Xu F, Hsieh D. 2009. Chapter 16: Modulation of Intrinsic Properties by Computational Design. Protein Engineering and Design. Boca Raton (FL). 75:327-341.

Onuchic JN, Luthey-Schulten Z, Wolynes PG. 1997. Theory of protein folding: the energy landscape perspective. Ann. Rev. Phys. Chem. 48:545-600.

Oomen CJ, Van Ulsen P, Van Gelder P, Feijen M, Tommassen J, Gros P. 2004. Structure of the translocator domain of a bacterial autotransporter. The EMBO Journal 23:1257-1266.

Overbeeke N, Bergmans H, Van Mansfield F, Lugtenberg B. 1983. Complete nucleotide sequence of phoE, the structural gene for the phosphate limitation inducible outer membrane pore protein of *Escherichia coli* K12. J. Mol. Biol. 163(4):513-532.

Park Y, Hayat S, Helms V. 2007. Prediction of the burial status of transmembrane residues of helical membrane proteins. BMC Bioinformatics 8:302.

Persikov AV, Ramshaw JAM, Kirkpatrick A, Brodsky B. 2000. Amino acid propensities for the collagen triple-helix. Biochemistry 39:14960-14967.

Phale PS, Philippsen A, Kiefhaber T, Koebnik R, Phale VP, Schirmer T, Rosenbusch JP. 1998. Stability of trimeric OmpF porin: the contributions of the latching loop L2. Biochemistry 37(45):15663-15670.

Pieretti G, Carillo S, Lindner B, Kim KK, Lee KC, Lee JS, Lanzetta R, Parrilli M, Corsaro MM. 2012. Characterization of the Core Oligosaccharide and the O-Antigen Biological Repeating Unit from Halomonas stevensii Lipopolysaccharide: The First Case of O-Antigen Linked to the Inner Core. Chemistry-A European Journal. 18(12):3729-3735.

Pope A. (2007, April 28). "XY Scatter Colouration Plot". Retrieved from: http://www.andypope.info/charts/spectrum.htm Accessed: November 19, 2011.

Popot JL, Engelman DM. 1990. Membrane protein folding and oligomerization: the two-stage model. Biochemistry 29(17):4031-4037.

Pugsley AP, Schnaitman CA. 1978. Identification of three genes controlling production of new outer membrane pore proteins in *Escherichia coli* K-12. J. Bacteriol. 135(3):1118-1129.

Raetz C, Dowhan W. 1990. Biosynthesis and function of phospholipids in *Escherichia coli*. J. Biol. Chem. 265:1235-1238.

Reboul CF, Mahmood K, Whisstock JC, Dunstone MA. 2012. Predicting giant transmembrane β-barrel architecture. Bioinformatics 28(10):1299-1302.

Rees D, DeAntonio L, Eisenberg D. 1989. Hydrophobic organization of membrane proteins. Science 245(4917):510-513.

Regan L. 1994. Protein Structure: Born to be beta. Curr. Biol. 4(7):656-658.

Reitz S, Cebi M, Reiß P, Studnik G, Linne U, Koert U, Essen LO. 2009. On the function and structure of synthetically modified porins. Angewandte Chemie International Edition 48(26):4853-4857.

Remmert M, Biegert A, Linke D, Lupas A, Söding J. 2010. Evolution of outer membrane β-barrels from an ancestral ββ-hairpin. Mol. Biol. and Evol. 27(6):1348-1358.

Remmert M, Linke D, Lupas AN, Söding J. 2009. HHomp—prediction and classification of outer membrane proteins. Nucleic acids res. 37(suppl 2):W446-W451.

Rice P, Longden I, Bleasby A. 2000. EMBOSS: the European molecular biology open software suite. Trends in Genetics 16(6):276-277.

Ried G, Henning U. 1987. A unique amino acid substitution in the outer membrane protein OmpA causes conjugation deficiency in *Escherichia coli* K-12. FEBS Letters 223(2):387-390.

Rigel NW and Silhavy TJ. 2011. Making a beta-barrel: assembly of outer membrane proteins in Gram-negative bacteria. Curr. Op. Microbiol. 15(2):189-193.

Rizzitello AE, Harper JR, Silhavy TJ. 2001. Genetic evidence for parallel pathways of chaperone activity in the periplasm of *Escherichia coli*. J. Bacteriol. 183(23):6794-6800.

Rosenbusch JP. 1974. Characterization of the major envelope protein from *Escherichia coli*. J. Biol. Chem. 249:8019-8029.

Russell SJ, Cochran AG. 2000. Designing stable beta-hairpins: Energetic contributions from cross-strand residues. J. Am. Chem. Soc. 122(50):12600-12601.

Schein SJ, Colombini M, Finkelstein A. 1976. Reconstitution in planar lipid bilayers of a voltage-dependent anion-selective channel obtained from paramecium mitochondria. J. Memb. Biol. 30(1):99-120.

Schleiff E, Soll J, Küchler M, Kühlbrandt W, Harrer R. 2003. Characterization of the translocon of the outer envelope of chloroplasts. J. Cell Biol. 160(4):541-551.

Schulz GE. 2002. The structure of bacterial outer membrane proteins. Biochimica et Biophysica Acta (BBA)-Biomembranes 1565(2):308-317.

Schweizer M, Henning U. 1977. Action of a major outer cell envelope membrane protein in conjugation of *Escherichia coli* K-12. J. Bacteriol. 129(3):1651.

Sekino J. 1999. n-Ellipses and the minimum distance sum problem. Amer. Math. Monthly 106(3):193-202.

Senes A, Chadi DC, Law PB, Walters RFS, Nanda V, DeGrado WF. 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(2):436-448.

Senes A, Engel DE, DeGrado WF. 2004. Folding of helical membrane proteins: the role of polar, GxxxG-like and proline motifs. Curr. Op. Struct. Biol. 14(4):465-479.

Senes A, Ubarretxena-Belandia I, Engelman DM. 2001. The Cα—H··· O hydrogen bond: A determinant of stability and specificity in transmembrane helix interactions. Proc. Nat. Acad. Sci. 98(16):9056-9061.

Seshadri K, Garemyr R, Wallin E, Heijne GV, Elofsson A. 1998. Architecture of β-barrel membrane proteins: Analysis of trimeric porins. Protein Science 7(9):2026-2032.

Skare J, Ahmer B, Seachord C, Darveau R, Postle K. 1993. Energy transduction between membranes. TonB, a cytoplasmic membrane protein, can be chemically cross-linked in vivo to the outer membrane receptor FepA. J. Biol. Chem. 268:16302-16308.

Sklar JG, Wu T, Kahne D, Silhavy TJ. 2007. Defining the roles of the periplasmic chaperones SurA, Skp, and DegP in *Escherichia coli*. Genes & Development 21:2473-2484.

Smith CK, Withka JM, Regan L. 1994. A Thermodynamic Scale for the β-Sheet Forming Tendencies of the Amino Acids. Biochemistry 33(18):5510-5517.

Snijder H, Ubarretxena-Belandia I, Blaauw M, Kalk K, Verheij H, Egmond M, Dekker N, Dijkstra B. 1999. Structural evidence for dimerization-regulated activation of an integral membrane phospholipase. Nature 401:717-721.

Sonntag I, Schwarz H, Hirota Y, Henning U. 1978. Cell envelope and shape of *Escherichia coli*: multiple mutants missing the outer membrane lipoprotein and other major outer membrane proteins. J. Bacteriol. 136(1):280-285.

Stanley AM, Chuawong P, Hendrickson TL, Fleming KG. 2006. Energetics of outer membrane phospholipase A (OMPLA) dimerization. J. Mol. Biol. 358(1):120-131.

Stanley AM, Fleming KG. 2007. The role of a hydrogen bonding network in the transmembrane β-barrel OMPLA. J. Mol. Biol. 370(5):912-924.

Stec B, Kreinovich V. 2005. Geometry of Protein Structures. I. Why Hyperbolic Surfaces are a Good Approximation for Beta-Sheets. Geombinatorics. 15(1):18-27.

Summa CM. 2002. Computational methods and their applications for de novo functional rotein design and membrane protein solubilization [Ph.D. thesis]. [Philadelphia (PA)]: University of Pennsylvania School of Medicine.

Surrey T, Jähnig F. 1992. Refolding and oriented insertion of a membrane protein into a lipid bilayer. Proc. Nat. Acad. Sci. 89(16):7457-7461.

Surrey T, Jähnig F. 1995. Kinetics of folding and membrane insertion of a β-barrel membrane protein. J. Biol. Chem. 270:28199-28203.

Tamm LK, Hong H, Liang B. 2004. Folding and assembly of β-barrel membrane proteins. Biochimica et Biophysica Acta (BBA)-Biomembranes 1666(1-2):250-263.

The "How" Blog. 2009. How to Create A Heat Map in Excel. Retrieved from: http://how.best-free-information.com/2009/04/how-to-create-a-heat-map-in-excel/ Accessed: November 15, 2011.

Thompson JD, Higgins DG, Gibson TJ. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 22(22):4673-4680.

Tommassen J. 2010. Assembly of outer-membrane proteins in bacteria and mitochondria. Microbiology 156(9):2587-2596.

Tokunaga M , Tokunaga H, Okajima Y, Nakae T. Characterization of Porins from the Outer Membrane of Salmonella typhimurium 2. Physical Properties of the Functional Oligomeric Aggregates. Eur. J. Biochem. 95(3):441-448.

Tusnády GE, Dosztányi Z, Simon I. 2004. Transmembrane proteins in the Protein Data Bank: identification and classification. Bioinformatics 20(17):2964-2972.

Viklund H, Elofsson A. 2004. Best α-helical transmembrane protein topology predictions are achieved using hidden Markov models and evolutionary information. Protein Science 13(7):1908-1917.

Vogt J, Schulz GE. 1999. The structure of the outer membrane protein OmpX from *Escherichia coli* reveals possible mechanisms of virulence. Structure 7(10):1301-1309.

VSTOTeam. (2008, January 16). "The Reports of VBA's Demise Have Been Greatly Exaggerated".

Retrieved from: http://blogs.msdn.com/b/vsto/archive/2008/01/16/the-reports-of-vba-s-demise-have-been-greatly-exaggerated.aspx

Accessed: September 21, 2011.

Wallin E, Tsukihara T, Yoshikawa S, Heijne GV, Elofsson A. 1997. Architecture of helix bundle membrane proteins: an analysis of cytochrome c oxidase from bovine mitochondria. Protein Science 6(4):808-815.

Walters R, DeGrado W. 2006. Helix-packing motifs in membrane proteins. Proc. Nat. Acad. Sci. 103(37):13658-13663.

Walther DM, Papic D, Bos MP, Tommassen J, Rapaport D. 2009. Signals in bacterial β-barrel proteins are functional in eukaryotic cells for targeting to and assembly in mitochondria. Proc. Nat. Acad. Sci. 106(8):2531-2536.

Walther DM, Rapaport D, Tommassen J. 2009. Biogenesis of β-barrel membrane proteins in bacteria and eukaryotes: evolutionary conservation and divergence. Cell. Mol. Life Sci. 66(17):2789-2804.

Wang Q, Cheung MS. 2012. A Physics-Based Approach of Coarse-Graining the Cytoplasm of *Escherichia coli* (CGCYTO). Biophysical Journal 102(10):2353-2361.

Weiszfeld E, Plastria F. 2009. On the point for which the sum of the distances to n given points is minimum. Annals of Op. Res. 167(1):7-41.

Wiener MC, White SH. 1991. Fluid bilayer structure determination by the combined use of X-ray and neutron diffraction. I. Fluid bilayer models and the limits of resolution. Biophys. J. 59(1):162-173.

Wolfe PB, Wickner W, Goodman JM. 1983. Sequence of the leader peptidase gene of *Escherichia coli* and the orientation of leader peptidase in the bacterial envelope. J. Biol. Chem. 258(19):12073-12080.

Yin H, Slusky JS, Berger BW, Walters RS, Vilaire G, Litvinov RI, Lear JD, Caputo GA, Bennett JS, DeGrado WF. 2007. Computational design of peptides that target transmembrane helices. Science's STKE 315(5820):1817-1822.

Yohannan S, Yang D, Faham S, Boulting G, Whitelegge J, Bowie JU. 2004. Proline substitutions are not easily accommodated in a membrane protein. J. Mol. Biol. 341(1):1-6.

Zalk R, Israelson A, Garty ES, Azoulay-Zohar H, Shoshan-Barmatz V. 2005. Oligomeric states of the voltage-dependent anion channel and cytochrome c release from mitochondria. Biochem. J. 386(1):73-83.

Zhou FX, Cocco MJ, Russ WP, Brunger AT, Engelman DM. 2000. Interhelical hydrogen bonding drives strong interactions in membrane proteins. Nat. Struct. Biol. 7:154-160.

Zwizinski C, Wickner W. 1980. Purification and characterization of leader (signal) peptidase from *Escherichia coli*. J. Biol. Chem. 255:7973-7977.

7. Supplementary Information

7.1 Perl script for Theoretical β-Barrel modeling (Vikas Nanda)

```
#!/usr/bin/perl

use Math::Trig;

format PDB =
ATOM   @>>>  CA  ALA  @>>>    @###.###@###.###@###.###
$res,                 $res,  $x,     $y,     $z
.

open (PDB, ">$ARGV[3]");


$numStrands = $ARGV[0];
$shear = $ARGV[1];
$length = $ARGV[2];

$a = 3.3;
$b = 4.4;

$alpha = pi / 2.0 - atan (($shear*$a)/($numStrands*$b));
$radius = sqrt( ($shear*$a)*($shear*$a) + ($numStrands*$b)*($numStrands*$b) ) /
(2.0*$numStrands*sin(pi/$numStrands));

print rad2deg($alpha) . " " . $radius . "\n";

for $N (0 .. $numStrands - 1)
{
    for $n (0 .. $length - 1)
    {
        $theta = ($n*$a + $N*$b) * cos($alpha) / $radius;
        $x = $radius * cos ($theta);
        $y = $radius * sin ($theta);
        $z = ($n*$a - $N*$b) * sin ($alpha);

        $res = $N * $length + $n + 1;

        write (PDB);
    }
}
```

7.2 Matlab script for Ruled Surface model of β-Barrel

```
%function [basePolygon, nextBase, SectionCutPolygon, Strands] =
getBetaBarrelPlain(noStrands, noAApStrand, bTwist, bHeight, rotation, translation)
%characteristics of Beta Barrel-----------------------------------------------
%b = 4.4; %perpendicular distance between neighboring strands
%a = 3.3; %distance between neighboring C-alphas
N = noStrands; % number of strands
n = noAApStrand; % number of AAs per strand
%rotation and translation are 1 by 3 vectors
%remember, Twist is in radians, not degrees
phi = rotation(1);
theta = rotation(2);
psi = rotation(3);
deltaX = translation(1);
deltaY = translation(2);
deltaZ = translation(3);

%Set up Tensors to Acquire Data of each Polygonal Cross Section or Strand--
SectionCutPolygon = zeros(3,N,n);
Strands = zeros(3,n,N);

%Draw Base Polygons and Store their Info as a Matrix-----------------------
t = 0:2*pi/N:2*pi;   %parameter t
startX = cos(t);
startY = sin(t);
startZ = zeros(1,(N+1));

s = sqrt(a^2+b^2);   %polygonal side length
R = .5*s*csc(pi/N);  %circumradius
%r = R*cos(pi/N);     %inradius

basePolygon = [startX; startY; startZ].*R;
% plot3(basePolygon(1,:),basePolygon(2,:),basePolygon(3,:));
% hold on
offset1 = zeros(3,(noStrands+1));
offset1(3,:) = bHeight*ones(1,(noStrands+1));
nextBase = offset1 + basePolygon;

%apply Twist to Top Base-------------------------------------------------
nextBase = rotateZ(bTwist)*nextBase;
%apply rigid motion to the top and bottom bases----------------------------
basePolygon = rotateX(phi)*rotateY(theta)*rotateZ(psi)*basePolygon +
translation'*ones(1,N+1);
nextBase = rotateX(phi)*rotateY(theta)*rotateZ(psi)*nextBase + translation'*ones(1,N+1);
%draw the two bases-------------------------------------------------------
plot3(basePolygon(1,:),basePolygon(2,:),basePolygon(3,:));
hold on
lastX = nextBase(1,:);
lastY = nextBase(2,:);
lastZ = nextBase(3,:);
plot3(lastX,lastY,lastZ);
hold on
%Draw straight lines from First Base's Vertices to that of the Top Base----
    L = zeros(3, 2, N);
    for k = 1:N
        L(:,:,k) = [basePolygon(:,k),nextBase(:,k)];
        line(L(1,:,k),L(2,:,k),L(3,:,k));
    end
    hold on
    axis equal
%OK, now that we have the structure, let's set up AA points on each strand-
% Lstart = L(:,1,:);
% Lend = L(:,2,:);
for m = 1:N
    Strands(:,:,m) = linspaceVects(L(:,1,m)',L(:,2,m)',n);
end

for p=1:n
    SectionCutPolygon(:,:,p) = linspaceVects(Strands(:,p,1)',Strands(:,p,N),N);
end
```

7.3 C++ ProtCAD Script Alignment of TMBs

```
#include <iostream>
#include <string>
#include <cstring>
#include "ensemble.h"
#include "PDBInterface.h"
#include <math.h>
#include <vector>

#define PI 3.14159
using namespace std;
/*betaBarrelAligner_v2 - given a PDB, number of chains, number of strands per chain
(assuming protein is symmetric in this respect), and start and end of each strand,this
script extracts the CA index numbers that correspond to the start and end of each beta
strand, then converts each CA-to-next-CA into a vector.

The way we determine the angle of rotation is through a grid-search. A vector is
considered "optimal" if the sum of the projections of each CA-to-next-CA vector against
this independent vector is maximized with respect to the grid size, which will be finer
each of the 5 cycles. Each cycle after the 1st will have its grid set up with the
previous optimal vector as its starting point.

The resulting pdb is of format aligned_PDB_ID(.pdb)*/

//usage:
//argument 0 = betaBarrelAligner_v2
//argument 1 = <the pdb you are using to get optimal rotation angle (hint: needs to be
one chain only)>
//argument 2 = <the pdb you want aligned (can be multichained)>
//argument 3 = <the pdb that you output after alignment>

//note: to optimize alignment, consider trimming your TMB to minimal strands and use that
for argv[1]

int main(int argc, char* argv[])
{
//user types something like "betaBarrelAligner_v2 1GFL_A.pdb 1GFL.pdb aligned_1GFL.pdb"
string PDBinfile    = argv[1];
string PDBusefile   = argv[2];
string alignedPDB   = argv[3];

//read in proteins
PDBInterface* thePDB   = new PDBInterface(PDBinfile);
ensemble* theEnsemble  = thePDB->getEnsemblePointer();
molecule* pMol         = theEnsemble->getMoleculePointer(0);
protein* betaBarrel    = static_cast<protein*>(pMol);

PDBInterface* thePDB2  = new PDBInterface(PDBusefile);
ensemble* theEnsemble2 = thePDB2->getEnsemblePointer();
molecule* pMol2        = theEnsemble2->getMoleculePointer(0);
protein* betaBarrel2   = static_cast<protein*>(pMol2);

//program asks user for number of chains and strands (per chain). User is then prompted
for a series of start and end pairs for //each labeled strand.

int numOfChains, numOfStrands;

cout << "how many chains are in your TMB? : ";
cin  >> numOfChains;

cout << "how many strands? : ";
cin  >> numOfStrands;

vector< vector< vector< int> > > chainArray;

for(int i = 0; i < numOfChains; i++)
{
    vector< vector< int > > strandArray;
    for(int j = 0; j < numOfStrands; j++)
    {
```

```
        vector< int > indexArray;

        int tempNum1, tempNum2;
        cout << "input strand start index for chain "
            << static_cast<char> (i+65) << ", strand " << j+1 << ": ";
        cin  >> tempNum1;

        cout << "input strand end index for chain "
            << static_cast<char> (i+65) << ", strand " << j+1 << ": ";
        cin  >> tempNum2;

        for (int k = tempNum1; k<= tempNum2; k++)
        {
            indexArray.push_back(k);
        }
        tempNum1 = 0;
        tempNum2 = 0;

        strandArray.push_back(indexArray);
    }
    chainArray.push_back(strandArray);
}

UInt resInChainCount = 0;

dblVec barrelCentroid(3);
barrelCentroid[0] = 0.0; barrelCentroid[1] = 0.0; barrelCentroid[2] = 0.0;

for (vector< vector< vector<int> > > ::size_type u = 0; u < chainArray.size(); u++)
{
    for (vector< vector <int> >::size_type v = 0; v < chainArray[u].size(); v++)
    {
        for (vector< int >::size_type w = 0; w < chainArray[u][v].size(); w++)
        {
        barrelCentroid =
        barrelCentroid + betaBarrel -> getCoords(u,betaBarrel->getIndexFromResNum(u,
chainArray[u][v][w]), "CA");
        resInChainCount++;
        }
    }
}

double resInChain = (double)resInChainCount;
barrelCentroid    = (barrelCentroid/resInChain) * -1.0;

cout << "the centroid of this beta barrel protein is: "
    << barrelCentroid[0] << " " << barrelCentroid[1] << " " << barrelCentroid[2] << endl;

betaBarrel->translate(barrelCentroid);      //center the barrel

//setting up simulation optimizing parameters
double bestProjection = 0;
double bestPhi, bestTheta, bestPsi;

double phiMin   = -2.0*PI; double phiMax   = 2.0*PI;
double thetaMin = -2.0*PI; double thetaMax = 2.0*PI;
double psiMin   = -2.0*PI; double psiMax   = 2.0*PI;
double step     = (20.0/180.0)*2.0*PI;
int countTimesOfSim = 0;

//initializing zAxis vector
dblVec zAxisVector(3);
zAxisVector[0] = 0.0;
zAxisVector[1] = 0.0;
zAxisVector[2] = 1.0;

//start grid search
for (UInt h = 0; h<5; h++)
{
    dblVec zAxisVector(3);
    for(double phi = phiMin; phi <= phiMax; phi += step)
```

```
        {
        for(double theta = thetaMin; theta <= thetaMax; theta += step)
        {
            for(double psi = psiMin; psi <= psiMax; psi += step)
            {
                betaBarrel->eulerRotate(phi,theta,psi);
                double currProjection = 0.0;

                for (vector< vector< vector< int > > >::size_type u = 0; u <
chainArray.size(); u++)
                {
                    //cout << "chain " << static_cast<char> (u+65) << ": ";
                    for (vector< vector< int > >::size_type v = 0; v <
chainArray[u].size(); v++)
                    {
                        double dotProd = 0.0;
                        vector< UInt > tempArray;
                        vector< dblVec > resInTempStrand;
                        //cout << "strand " << v+1 << ": " << endl;

                        for(vector< int >::size_type w = 0; w < chainArray[u][v].size();
w++)//for each residue in strand...
                        {
                            tempArray.push_back(chainArray[u][v][w]);
                            //initializing temp 3d coord variables
                            dblVec betaStrCrds(3); dblVec tempFirst(3); dblVec
tempSec(3); dblVec tempDiff(3);

                            betaStrCrds = betaBarrel->
                                getCoords(u,betaBarrel-
>getIndexFromResNum((int)u,chainArray[u][v][w]),"CA");
                            resInTempStrand.push_back(betaStrCrds);

                            //start getting vector differences
                            if(w>=1)
                            {
                                tempSec   = betaStrCrds;
                                tempFirst = betaBarrel->
                                    getCoords(u,betaBarrel->
                                        getIndexFromResNum((int)u,chainArray[u][v][w-
1]),"CA");

                                tempDiff  = tempSec - tempFirst;


//tempDiffNormalized is introduced to normalize each strand length to help debias
//the axis
                                dblVec tempDiffNormalized(3);
                                tempDiffNormalized[0] =
tempDiff[0]/(double)chainArray[u][v].size();
                                tempDiffNormalized[1] =
tempDiff[1]/(double)chainArray[u][v].size();
                                tempDiffNormalized[2] =
tempDiff[2]/(double)chainArray[u][v].size();
//dotProd = CMath::dotProduct(tempDiffNormalized,zAxisVector);
                                dotProd = tempDiffNormalized[2];
                                if(dotProd < 0)
                                {
                                dotProd = -1*dotProd;
                                    }
                                    currProjection += dotProd;
                                }
                            }
                        }
                    }

                    //evaluate projection
                    if (currProjection > bestProjection)
                    {
                        bestProjection = currProjection;
                        bestPhi   = phi;
```

```
                        bestTheta = theta;
                        bestPsi   = psi;
                    }

                    countTimesOfSim++;
                    betaBarrel->undoEulerRotate(phi,theta,psi);
                }
            }
        }

        phiMin   = bestPhi - step;
        phiMax   = bestPhi + step;
        thetaMin = bestTheta - step;
        thetaMax = bestTheta + step;
        psiMin   = bestPsi - step;
        psiMax   = bestPsi + step;
        step     = step/5;
    }

    betaBarrel2->translate(barrelCentroid);
    betaBarrel2->eulerRotate(bestPhi,bestTheta + PI/3.0 ,bestPsi);
    pdbWriter(betaBarrel2, alignedPDB);

    return 0;
}
```

7.4 Excel VBA Script for Calculating $E_z\beta$ for PDB/DSSP Hybrid Files

```
Function calcEZ_Beta(deltaE As Double, zMid As Double, sTransition As Double, _
    z As Double, distroType As String) As Double
'-------------------------------------------------------------------------------------
    Select Case distroType
        Case "A", "M", "V", "L", "I", "N", "D", "Q", "E", "K", "R", "H", "P", "S", "T"
            calcEZ_Beta = deltaE / (1 + ((z / zMid) ^ sTransition))
        Case "F", "Y", "W", "G"
            calcEZ_Beta = _
                deltaE * Exp((-1 * (z - zMid) ^ 2) / (2 * (sTransition ^ 2)))
        Case Else
            calcEZ_Beta = 0
    End Select
End Function

Function TotalEzBeta(startOfLetters As Range, startOfZs As Range) As Double

'StartOfX means one cell as a starting point of the sequence
Dim numOfRes As Integer, EzBPA_WkSht As Worksheet
Set EzBPA_WkSht = Worksheets("EzBetaParamArray")

TotalEzBeta = 0
numOfRes = Range(startOfLetters, startOfLetters.End(xlDown)).Rows.Count
For i = 0 To numOfRes - 1
    If startOfLetters.Offset(i, 0).Value = "C" Then
        TotalEzBeta = TotalEzBeta + 0
    Else
        TotalEzBeta = _
            TotalEzBeta + calcEZ_Beta( _
                Application.WorksheetFunction.VLookup( _
                    startOfLetters.Offset(i, 0).Value, _
                    EzBPA_WkSht.Range("EzBetaParamArray"), 2, 0), _
                Application.WorksheetFunction.VLookup( _
                    startOfLetters.Offset(i, 0).Value, _
                    EzBPA_WkSht.Range("EzBetaParamArray"), 3, 0), _
                Application.WorksheetFunction.VLookup( _
                    startOfLetters.Offset(i, 0).Value, _
                    EzBPA_WkSht.Range("EzBetaParamArray"), 4, 0), _
                startOfZs.Offset(i, 0).Value, _
                startOfLetters.Offset(i, 0).Value _
                )
    End If
Next
End Function
```

7.5 Excel VBA Script for Randomly Swapping Sequences within TMBs

```
Sub RSS()
'given
' - HybridFile name, ParamDerivFile name, number of trials
' - HybridFile name -> Hybrid/Prehybrid values
' - ParamDerivFile  -> EzBetaParamArray
' - number of trials

'output
' - function that calculates energy of any sequence given z-depth and aa-type
' --> means I need EzBeta for individual AAs
' - randomly swap these rows by a random number index, then re-sort

'Sheets setup

Dim RSS SetupSheet As Worksheet
Set RSS_SetupSheet = ThisWorkbook.Sheets("RSS Interface")

Dim targetHybrid As Workbook, paramDerivFile As Workbook

Dim targetHybridName As String
targetHybridName = RSS_SetupSheet.Range("C1").Value
Dim paramDerivFileName As String
paramDerivFileName = RSS_SetupSheet.Range("C5").Value

Dim numOfTrials As Integer
numOfTrials = RSS_SetupSheet.Range("C7").Value

Dim RSS_Workbook As Workbook
Set RSS_Workbook = Workbooks.Add

'Assuming both files (paramDeriv,newHybrid) exist
If FileExists(paramDerivFileName) Then
    If FileIsOpen(paramDerivFileName) Then
        Set paramDerivFile =
Application.Workbooks(ExtractFileNameRaw(paramDerivFileName))
    Else
        Workbooks.Open (paramDerivFileName)
        Set paramDerivFile =
Application.Workbooks(ExtractFileNameRaw(paramDerivFileName))
    End If
Else
    Exit Sub
End If

If FileExists(targetHybridName) Then
    If FileIsOpen(targetHybridName) Then
        Set targetHybrid = Application.Workbooks(ExtractFileNameRaw(targetHybridName))
    Else
        Workbooks.Open (targetHybridName)
        Set targetHybrid = Application.Workbooks(ExtractFileNameRaw(targetHybridName))
    End If
Else
    Exit Sub
End If

Dim EzBetaParamArray() As Variant, index As Integer, ref_Params As Range
index = 0
ReDim EzBetaParamArray(1 To 19, 1 To 5)

Dim RSS_EzBetaParamArray As Worksheet
Set RSS_EzBetaParamArray = RSS_Workbook.Sheets.Add(Before:=Sheets(1))

RSS_EzBetaParamArray.Name = "EzBetaParamArray"
RSS_EzBetaParamArray.Tab.ColorIndex = 1

Set ref_Params = paramDerivFile.Sheets("displayOfParams").Range("A4:I25")

With RSS_EzBetaParamArray.Range("A1")
    .Value = "Residue"
```

```
    .Offset(0, 1).Value = "Paq"
    .Offset(0, 2).Value = "DeltaE0"
    .Offset(0, 3).Value = "Zmid"
    .Offset(0, 4).Value = "n"
End With

For Each AA In Split("A,D,E,H,I,K,L,M,N,P,Q,R,S,T,V,F,W,Y,G", ",")
    index = index + 1
    EzBetaParamArray(index, 1) = AA
    RSS_EzBetaParamArray.Cells(index + 1, 1).Value = EzBetaParamArray(index, 1)
    EzBetaParamArray(index, 2) = WorksheetFunction.VLookup(Convert_AA_1to3(CStr(AA)),
ref_Params, 3, 0)
    RSS_EzBetaParamArray.Cells(index + 1, 2).Value = EzBetaParamArray(index, 2)
    EzBetaParamArray(index, 3) = WorksheetFunction.VLookup(Convert_AA_1to3(CStr(AA)),
ref_Params, 5, 0)
    RSS_EzBetaParamArray.Cells(index + 1, 3).Value = EzBetaParamArray(index, 3)
    EzBetaParamArray(index, 4) = WorksheetFunction.VLookup(Convert_AA_1to3(CStr(AA)),
ref_Params, 7, 0)
    RSS_EzBetaParamArray.Cells(index + 1, 4) = EzBetaParamArray(index, 4)
    EzBetaParamArray(index, 5) = WorksheetFunction.VLookup(Convert_AA_1to3(CStr(AA)),
ref_Params, 9, 0)
    RSS_EzBetaParamArray.Cells(index + 1, 5) = EzBetaParamArray(index, 5)
Next

Dim RSS TargetPreHybrid As Worksheet
Set RSS_TargetPreHybrid = RSS_Workbook.Worksheets.Add(after:=RSS_EzBetaParamArray)
RSS_TargetPreHybrid.Name = "PreHybrid"
'The Prehybrid should be already trimmed by user if necessary.
targetHybrid.Sheets(Left(ExtractFileNameRaw(targetHybridName), 4) &
"_PreHybrid").Activate
Cells.Select
Selection.Copy
RSS_TargetPreHybrid.Range("A1").PasteSpecial Paste:=xlPasteValues
Application.CutCopyMode = False

Dim RSS_TargetHybrid As Worksheet
Set RSS_TargetHybrid = RSS_Workbook.Worksheets.Add(after:=RSS_TargetPreHybrid)
RSS_TargetHybrid.Name = "Hybrid"
targetHybrid.Sheets(Left(ExtractFileNameRaw(targetHybridName), 4) & "_Hybrid").Activate
Cells.Select
Selection.Copy
RSS_TargetHybrid.Range("A1").PasteSpecial Paste:=xlPasteValues
Application.CutCopyMode = False

Dim RSS ExptSht As Worksheet
Set RSS_ExptSht = RSS_Workbook.Worksheets.Add(after:=RSS_TargetHybrid)
RSS_ExptSht.Name = "RSS"

Dim numOfRows As Integer
numOfRows = Range(RSS_TargetHybrid.Range("A2"),
RSS_TargetHybrid.Range("A1").End(xlDown)).Rows.Count

With RSS_ExptSht
    .Range("A1").Value = "RandIndex"
    .Range("A2").Formula = "=rand()"
    .Range("B1").Value = "TrueResIndex"
    .Range("C1").Value = "1Code"
    .Range("D1").Value = "abs(zCoord)"
    .Range("E1").Value = "Ez Potential"
End With

targetHybrid.Close savechanges:=False
paramDerivFile.Close savechanges:=False

Range(RSS_TargetHybrid.Range("C1"), RSS_TargetHybrid.Range("C1").End(xlDown)).Copy
RSS_ExptSht.Range("B1").PasteSpecial Paste:=xlPasteValues

Dim i As Integer
For i = 1 To numOfRows
    Range("C1").Offset(i, 0).Formula = _
        "=Vlookup(B" & i + 1 & ",'" & RSS_TargetHybrid.Name & "'!C:E, 3, 0)"
```

```
    Range("C1").Offset(i, 1).Value = _
        Math.Abs(Application.WorksheetFunction.VLookup( _
            Range("B" & i + 1).Value, RSS_TargetHybrid.Range("C:H"), 6, 0))
    'now that we have Paqs, we offset the vlookup index by 1 (used to be 2, 3, 4)

    If Range("C" & i + 1).Value = "C" Then
        Range("C1").Offset(i, 2).Value = 0
    Else
        Range("C1").Offset(i, 2).Value = _
            calcEZ_Beta( _
                Application.WorksheetFunction.VLookup( _
                    Range("C" & i + 1), _
                        RSS_EzBetaParamArray.Range("A1:E20"), 3, 0), _
                Application.WorksheetFunction.VLookup( _
                    Range("C" & i + 1), _
                        RSS_EzBetaParamArray.Range("A1:E20"), 4, 0), _
                Application.WorksheetFunction.VLookup( _
                    Range("C" & i + 1), _
                        RSS_EzBetaParamArray.Range("A1:E20"), 5, 0), _
                Range("D" & i + 1).Value, _
                Range("C" & i + 1).Value _
                )
    End If
Next

RSS_ExptSht.Range("A2").AutoFill Destination:=Range(Range("A2"),
Range("A2").Offset(numOfRows - 1, 0))

Range("G2").Value = "Total EzB Energy = "
Range("H2").Value = WorksheetFunction.Sum(Range(Range("E2"), Range("E2").End(xlDown)))

Range("K2").Value = WorksheetFunction.Round(Range("H2").Value, 3)
Range("A1:K1").EntireColumn.AutoFit

'so far this gets us original sequence's EzB energy
'the following performs the 30 (or x=numOfTrials) trials of mass mutagenesis

Dim j As Integer
For j = 1 To numOfTrials
    With Range("J2").Offset(j, 0)
        .Value = j
        .HorizontalAlignment = xlCenter
    End With
    Columns("A:B").Select

    Selection.Sort Key1:=Range("A2"), Order1:=xlAscending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, _
        DataOption1:=xlSortNormal
    'Replace old calculated EzBeta values with updated ones
    Range("C1").Activate

    Dim k As Integer
    For k = 1 To numOfRows
        If Range("C" & k + 1).Value = "C" Then
            ActiveCell.Offset(k, 2).Value = 0
        Else
            ActiveCell.Offset(k, 2).Value = _
                calcEZ_Beta( _
                    Application.WorksheetFunction.VLookup( _
                    Range("C" & k + 1), _
                        RSS_EzBetaParamArray.Range("A1:E20"), 3, 0), _
                    Application.WorksheetFunction.VLookup( _
                    Range("C" & k + 1), _
                        RSS_EzBetaParamArray.Range("A1:E20"), 4, 0), _
                    Application.WorksheetFunction.VLookup( _
                    Range("C" & k + 1), _
                        RSS_EzBetaParamArray.Range("A1:E20"), 5, 0), _
                    Range("D" & k + 1).Value, _
                    Range("C" & k + 1).Value _
                    )
```

```
        End If
    Next
    Range("H2").Value = WorksheetFunction.Sum(Range(Range("E2"),
Range("E2").End(xlDown)))
    Range("J2").Offset(j, 1).Value = WorksheetFunction.Round(Range("H2").Value, 3)
Next

'Report Stats
With Range("J2")
    .Value = "OrigSeq(0)"
    .Font.Bold = True
    .Font.ColorIndex = 48
    .Offset(numOfTrials + 1, 0).Value = "Mean w/o = "
    .Offset(numOfTrials + 1, 1).Formula = "=average(K" & 3 & ":K" & 2 + numOfTrials & ")"
    .Offset(numOfTrials + 2, 0).Value = "StdDev w/o = "
    .Offset(numOfTrials + 2, 1).Formula = "=stdev(K" & 3 & ":K" & 2 + numOfTrials & ")"
    .Offset(numOfTrials + 3, 0).Value = "zScore w/o = "
    .Offset(numOfTrials + 3, 1).Formula = "=abs(K" & 2 & "-K" & 3 + numOfTrials & ")/K" &
4 + numOfTrials
End With

Range("A1:K1").EntireColumn.AutoFit

'delete remnant sheets (1,2,3)
Application.DisplayAlerts = False
RSS_Workbook.Sheets(Array("Sheet1", "Sheet2", "Sheet3")).Delete
Application.DisplayAlerts = True

'rename file
RSS_Workbook.SaveAs filename:=ThisWorkbook.Path & "\" & _
    "RSS_" & RSS_TargetHybrid.Range("A2").Value & "_" & timeStamp

End Sub
```

7.6 Excel VBA Script for Generating Dynamic Energy Landscapes

```
Option Explicit

Dim cellWidth As Double
Dim cellHeight As Double
Dim pi As Double

Sub test_S3DP()
'how to use if first testing this code:
'1)delete these last two sheets: "DataForPlot" and "3DPlot"
'2)click once to place cursor within this subroutine
'3)press F5 to run this subroutine

'note:
'If user wants to import his/her own data,
'the user will need to edit the transformation in Setup MainSheet
'as suited to the user's needs

cellWidth = Range("A1").Width
cellHeight = Range("A1").Height
pi = WorksheetFunction.pi()

Dim xLen As Integer, yLen As Integer
xLen = countSteps(-1 * pi / 2, pi / 2, pi / 18)
yLen = xLen

Setup_MainSheet CInt(xLen), CInt(yLen)
'Code can run if sheet "DataForPlot" does not exist
Setup_PlotSheet 255, 0, 0, 0, 0, 150, -105, 0, 15
'Code can run if sheet "3DPlot" does not exist
End Sub
'-------------------------------------------------------------

Sub Setup_MainSheet(xLen As Integer, yLen As Integer)
Dim mainSheet As Worksheet
Dim i As Integer, j As Integer
Set mainSheet = Sheets.Add(after:=Sheets(Worksheets.Count))
mainSheet.Name = "DataForPlot"

With mainSheet
    .Range("A1:I1").Merge
    .Range("A2").Value = Int(81 * Rnd)
    .Range("B2").Formula = "=$A$2-41"
    .Range("B2").Interior.ColorIndex = 35

    For j = -1 * (yLen / 2) To (yLen / 2)
        .Range("A3").Offset(0, j + (yLen / 2) + 1).Value = j        'Column labels
        For i = -1 * (xLen / 2) To (xLen / 2)
            .Range("A3").Offset(i + (xLen / 2) + 1, 0).Value = i    'Row labels
            .Cells(i + (xLen / 2) + 4, j + (yLen / 2) + 2).Formula = _
                "=indirect(address(" & _
                    i + (xLen / 2) + 3 & "," & _
                    j + (yLen / 2) + 2 & ",,TRUE, Concatenate(" & _
                    Chr(34) & "z_" & Chr(34) & ",$B$2)),1)"         'Virtual data
        Next
    Next

    'dynamic title
    .Range("A1").Formula = "=concatenate(" _
        & Chr(34) & "Holding ZShift constant at " & Chr(34) & ",$B$2," & _
         Chr(34) & " " & ChrW$(197) & " rotating around x(rows) and y(columns) axes at
steps of pi/18"
        & Chr(34) & ")"
    .Range(Cells(3, 1), Cells(3 + yLen + 1, 1 + xLen + 1)).Select

End With
ThisWorkbook.Names.Add Name:="rngToPlot", RefersTo:=Selection
End Sub

'-------------------------------------------------------------
```

```
Sub Setup_PlotSheet(R_start As Integer, G_start As Integer, B_start As Integer, _
    R_end As Integer, G_end As Integer, B_end As Integer, _
    minScale As Integer, maxScale As Integer, majorStep As Integer)

Dim plotSheet As Worksheet
Set plotSheet = Sheets.Add(after:=Sheets(Worksheets.Count))
plotSheet.Name = "3DPlot"

Dim surfChartObj As ChartObject

Set surfChartObj = plotSheet.ChartObjects.Add( _
    Left:=0, Top:=0, Width:=11 * cellWidth, Height:=20 * cellHeight)

With surfChartObj.Chart
    .SetSourceData Source:=Sheets("DataForPlot").Range("rngToPlot"), PlotBy:=xlRows
    .ChartType = xlSurface
    .Location where:=xlLocationAsObject, Name:=plotSheet.Name
    .HasTitle = True
    .ChartTitle.Text = "='DataForPlot'!R1C1"
'Link to the "dynamic title" in "DataForPlot"

    With .Axes(xlValue)     'z-axis
        .HasTitle = True
        .AxisTitle.Characters.Text = "Total Energy of Insertion"
        .MinimumScale = minScale
        .MaximumScale = maxScale
        .MajorUnit = majorStep
        .Crosses = xlMinimum
    End With

    With .Axes(xlCategory)  'x-axis
        .HasTitle = True
        .AxisTitle.Characters.Text = "rotX"
        .TickLabelPosition = xlTickLabelPositionLow
        .TickLabelSpacing = 3
        .TickLabels.Orientation = 0
        .TickLabels.Offset = 450
    End With

    With .Axes(xlSeries)    'y-axis
        .HasTitle = True
        .AxisTitle.Characters.Text = "rotY"
        .TickLabelPosition = xlTickLabelPositionLow
        .TickLabelSpacing = 3
        .TickLabels.Orientation = 0
        .TickLabels.Offset = 45
    End With

    'Rotation, Elevation and Perspective of the Surface Chart
    .Elevation = 5
    .Perspective = 25
    .Rotation = 30
    .HeightPercent = 60

    .ChartGroups(1).Has3DShading = True
    .Walls.Interior.ColorIndex = 15
    .Floor.Interior.ColorIndex = 15

End With
```

```
'Select and edit features and binding cell for scrollbar object
plotSheet.ScrollBars.Add(cellWidth * 4, cellHeight * 21, cellWidth * 3,
cellHeight).Select
With Selection
    .min = 1
    .max = 81
    .SmallChange = 1
    .LargeChange = 5
    .LinkedCell = Sheets(Sheets.Count - 1).Name & "!$A$2"
    .Display3DShading = True
End With

Dim x() As Variant, numOfColors As Integer
'x is an N-by-3 matrix
numOfColors = surfChartObj.Chart.Legend.LegendEntries.Count
ReDim x(1 To numOfColors, 1 To 3)
x = colorGrad(R_start, G_start, B_start, R_end, G_end, B_end, numOfColors)
Dim i As Integer

'After computing color gradient, set legend colors in order
With surfChartObj.Chart.Legend
    For i = 1 To numOfColors
        .LegendEntries(i).LegendKey.Interior.Color = RGB(x(i, 1), x(i, 2), x(i, 3))
    Next
End With

End Sub
'-------------------------------------------------------------

Function colorGrad(R_start As Integer, G_start As Integer, B_start As Integer, _
    R_end As Integer, G_end As Integer, B_end As Integer, numOfColors) As Variant
'this is written to help autogenerate colors for the legend entries. Given a start and
'end color (both in RGB format), colorGrad determines a linear interpolation (gradient)
and returns an array of RGB values.

'inspired by "How to Create a Heat Map in Excel"
'URL: http:// how.best-free-information.com/2009/04/how-to-create-a-heat-map-in-excel/

Dim colorMatrix() As Variant
ReDim colorMatrix(1 To numOfColors, 1 To 3)
Dim R As Double, g As Double, b As Double, i As Integer, j As Integer, k As Integer

'Determining red components of gradient
For i = 1 To numOfColors
    Dim currColorValueR As Integer
    currColorValueR = Int(R_start + ((R_end - R_start) / (numOfColors - 1)) * (i - 1))
    If (currColorValueR <= 255 And currColorValueR >= 0) Then
        colorMatrix(i, 1) = currColorValueR
    Else
        MsgBox ("Incorrect Value (0-255)")
        Exit Function
    End If
Next

'Determining green components
For j = 1 To numOfColors
    Dim currColorValueG As Integer
    currColorValueG = Int(G_start + ((G_end - G_start) / (numOfColors - 1)) * (j - 1))
    If (currColorValueG <= 255 And currColorValueG >= 0) Then
        colorMatrix(j, 2) = currColorValueG
    Else
        MsgBox ("Incorrect Value (0-255)")
        Exit Function
    End If
Next
```

```
'Determining blue components
For k = 1 To numOfColors
    Dim currColorValueB As Integer
    currColorValueB = Int(B_start + ((B_end - B_start) / (numOfColors - 1)) * (k - 1))
    If (currColorValueB <= 255 And currColorValueB >= 0) Then
        colorMatrix(k, 3) = currColorValueB
    Else
        MsgBox ("Incorrect Value (0-255)")
        Exit Function
    End If
Next

colorGrad = colorMatrix
End Function

'-------------------------------------------------------------

Function countSteps(vStart As Double, vEnd As Double, vStep As Double) As Integer
'use this simple function to determine how many gridpoints (columns) one needs
Dim v As Double, vCt As Integer

For v = vStart To vEnd Step vStep
    vCt = vCt + 1
Next

countSteps = vCt
End Function
```

7.7 Excel VBA Script for Stitching Multiple PDBs into an Animation

```
'
Declare PtrSafe Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As LongPtr)
'-------------------------------------------------------------
Function inQuotes(strToQuote As String) As String
inQuotes = Chr(34) & strToQuote & Chr(34)
End Function
'-------------------------------------------------------------
Function logonToPuTTY2(PuTTY_EXE As String, IP_Session As String, login As String, pwd As
String, sendFeed As String) As Long
Dim taskID As Long
taskID = Shell(inQuotes(PuTTY_EXE) & " " & _
    inQuotes("-load") & " " & inQuotes(IP_Session) & " " & _
    inQuotes("-l") & " " & login & " " & _
    inQuotes("-pw") & " " & pwd, _
    vbNormalFocus)
AppActivate taskID, True
Sleep (2000)
AppActivate taskID, True
SendKeys sendFeed
 AppActivate taskID, True
logonToPuTTY2 = taskID
End Function

Sub getAnimationParams_test()
Dim targetSheet As Worksheet
Dim targetRange As Range
Dim reportedMin As Double
Dim reportedMinAddress As Range
Dim shtIndex As Integer
Dim commandStr1 As String, commandStr2 As String, commandStr3 As String
Dim PDB_ID As String
PDB_ID = Mid(ThisWorkbook.Name, 5, 4)
Debug.Print PDB_ID

Dim testArr() As Integer
ReDim testArr(1 To 81, 1 To 2)
Dim trajectory() As String
ReDim trajectory(1 To 81)

commandStr1 = "cd protcad/src/test/Correct_Alignment_PDBs_091310~"
commandStr2 = commandStr1 & "mkdir " & PDB_ID & "_animation~exit~"
commandStr3 = commandStr1

Dim taskID As Long
'toggle off when doing z = 1 to 40, toggle on when doing z = -40 to 0 (or however you
partition)

taskID = logonToPuTTY2( _
        "D:\Users\username\Downloads\PuTTY.exe", "192.76.178.25", "username", "password",
commandStr2)

For shtIndex = -40 To 0
    Set targetSheet = ThisWorkbook.Sheets("z_" & shtIndex)
    targetSheet.Activate
    Set targetRange = targetSheet.Range("B3:T21")
    reportedMin = WorksheetFunction.Min(targetRange)
    targetSheet.Range("B3:T21").Find(What:=Left(CStr(reportedMin), 5),
LookAt:=xlPart).Activate
    Debug.Print shtIndex, Round(reportedMin, 5), ActiveCell.End(xlToLeft).Value,
ActiveCell.End(xlUp).End(xlUp).Offset(1, 0).Value

    'solution: create 2d array!
    Dim tempArrLeft As Integer, tempArrRight As Integer
    testArr(shtIndex + 41, 1) = ActiveCell.End(xlToLeft).Value
    tempArrLeft = testArr(shtIndex + 41, 1)
    testArr(shtIndex + 41, 2) = ActiveCell.End(xlUp).End(xlUp).Offset(1, 0).Value
    tempArrRight = testArr(shtIndex + 41, 2)
    targetSheet.Range("A1").Activate
```

```
    'rotateProteinEuler: tag to commandStr
    trajectory(shtIndex + 41) = PDB_ID & "_animation/aligned_" & PDB_ID & "_" & shtIndex
& "_" & tempArrLeft & "_" & tempArrRight & ".pdb"
    commandStr3 = commandStr3 & "rotateProtein_061611 " & _
        "aligned_" & PDB_ID & ".pdb" & " " & _
        trajectory(shtIndex + 41) & _
        " " & tempArrLeft & " " & tempArrRight & " " & shtIndex & "~sleep 3~"
    'Debug.Print commandStr3

    'Sleep 2000
Next
taskID = logonToPuTTY2( _
        "D:\Users\Daniel\Downloads\PuTTY.exe", "192.76.178.25", "hsiehd", "nandalab2",
commandStr3)
Debug.Print commandStr3

End Sub
'-------------------------------------------------------------
Sub createNMRModel()
'create animation file
'first create the sorted collection or else the destination file will be included,
resulting in some fatal loop
Dim theCol As Collection
Set theCol = GetFilesInDateOrder(ThisWorkbook.Path & _
    "\" & Mid(ThisWorkbook.Name, 5, 4) & "_animation")

Dim fileCounter As Integer
fileCounter = 0

Dim myAnimationFile As String
myAnimationFile = ThisWorkbook.Path & _
    "\" & Mid(ThisWorkbook.Name, 5, 4) & "_animation" & _
    "\" & Mid(ThisWorkbook.Name, 5, 4) & "_animation.pdb"
destNum = FreeFile()
Open myAnimationFile For Append As destNum

For Each theItem In theCol
    fileCounter = fileCounter + 1
    Dim sourceNum As Integer
    sourceNum = FreeFile()
    Open theItem For Input As sourceNum
    Print #destNum, "MODEL       " & createAppropriateNumbering(fileCounter)
    Do While Not EOF(sourceNum)
        Line Input #sourceNum, Temp
        Print #destNum, Temp
    Loop

    Dim FoR_Num As Integer
    FoR_Num = FreeFile()
    Open "D:\Work\EzBeta Related\FoR.txt" For Input As FoR Num

    Do While Not EOF(FoR_Num)
        Line Input #FoR_Num, Temp
        Print #destNum, Temp
    Loop

    Print #destNum, "ENDMDL"

    Close #FoR_Num
    Close #sourceNum
Next

Close #destNum
End Sub
```

```
Function createAppropriateNumbering(targetNum As Integer) As String
If targetNum < 10 Then
    createAppropriateNumbering = " " & CStr(targetNum)
ElseIf targetNum < 100 And targetNum > 9 Then
    createAppropriateNumbering = CStr(targetNum)
End If

End Function

Function GetFilesInDateOrder(strFolderPath As String) As Collection

    Dim colFiles As Collection
    Dim fso As Object, fdr As Object, filTemp As Object
    Dim lngIndex As Long, lngInsert As Long
    Set colFiles = New Collection
    Set fso = CreateObject("Scripting.Filesystemobject")
    Set fdr = fso.GetFolder(strFolderPath)
    For Each filTemp In fdr.Files
        ' If it's the first entry just add it
        If colFiles.Count = 0 Then
            colFiles.Add filTemp, filTemp.Name
        Else
            ' otherwise check modified date to see where to add it
            lngInsert = 0
            For lngIndex = 1 To colFiles.Count
                If filTemp.DateLastModified >= colFiles(lngIndex).DateLastModified Then
                    lngInsert = lngIndex
                    Exit For
                End If
            Next lngIndex
            If lngInsert = 0 Then
                ' it's the latest one, so add it to the end
                colFiles.Add filTemp, filTemp.Name
            Else
                ' add it in the position specified by lngInsert
                colFiles.Add filTemp, filTemp.Name, lngInsert
            End If
        End If
    Next filTemp
    Set GetFilesInDateOrder = colFiles
'    This is test code just to check the order
'    For lngIndex = 1 To colFiles.Count
'        Debug.Print colFiles(lngIndex).Name
'    Next lngIndex
End Function
```

7.8 Excel VBA Script for Building Homology Models

```
'
Function getActualSeqLength(startRng As Range) As Integer
'finds length (minus gaps) of PDB (preferred) or homolog sequence (not the intended use)
Dim count As Integer, i As Integer
Dim summarySht As Worksheet
Set summarySht = Sheets("SummarySheet")

'go down list of IDs to find cell with PDB_ID in first four places

count = 0
For i = 1 To Range(startRng.Offset(0, 1), startRng.End(xlToRight)).count
    If Not (startRng.Offset(0, i).Value = "-") Then
        count = count + 1
    End If
Next

getActualSeqLength = count
End Function

Function getPDBPositions(startRng As Range) As Variant
Dim count As Integer, i As Integer
'Dim startRng As Range
Dim summarySht As Worksheet
Set summarySht = Sheets("SummarySheet")

Dim positionsList() As Variant
ReDim positionsList(1 To getActualSeqLength(startRng), 1 To 2)

count = 0
'startRng.Activate
For i = 1 To Range(startRng.Offset(0, 1), startRng.End(xlToRight)).count
    'ActiveCell.Offset(0, 1).Activate
    If Not (CStr(startRng.Offset(0, i).Value) = "-") Then
        count = count + 1
        positionsList(count, 1) = CStr(startRng.Offset(0, i).Value)
        positionsList(count, 2) = startRng.Offset(0, i).Column
    End If
Next

Debug.Print count
For i = 1 To UBound(positionsList)
    Debug.Print positionsList(i, 1), positionsList(i, 2)
Next
getPDBPositions = positionsList
End Function

Sub testGPP()
Dim targetRng As Range
Dim summarySht As Worksheet
Set summarySht = Sheets("SummarySheet")
Set targetRng = getPDBSeqInMSA("1A0S")
Dim positionsList As Variant
positionsList = getPDBPositions(targetRng)
End Sub

Sub threadProt(templateRng As Range, targetRng As Range)
'modifies the PDB Hybrid Workbook
'saves as the target's name

Dim count As Integer, i As Integer
Dim summarySht As Worksheet
Set summarySht = Sheets("SummarySheet")
Dim thdSht As Worksheet
Dim thdWbk As Workbook

Dim templateProtein As Variant
templateProtein = getPDBPositions(templateRng)
Dim PDB_ID As String
PDB_ID = CStr(Left(templateRng.Value, 4))
```

```
Dim targetPDBHybrid As Workbook
Set targetPDBHybrid = Workbooks.Open("D:\Users\Daniel\Ez Beta
Paper\102909_EzBetas\HybridOnly_091910\" & PDB_ID & "_hybridOnly.xlsx")
'copy the sheet
targetPDBHybrid.Sheets(PDB_ID & "_Hybrid").Copy
after:=targetPDBHybrid.Sheets(Sheets.count)
Set thdSht = Sheets(Sheets.count)
thdSht.Name = PDB_ID & "_ThreadSeq"

'Debug.Print UBound(templateProtein)

thdSht.Range("A1").Value = "geneID"
For i = 1 To UBound(templateProtein, 1)
    thdSht.Range("A1").Offset(i, 0).Value = targetRng.Value
    'Debug.Print targetRng.Row, templateProtein(i, 2)
    thdSht.Range("E1").Offset(i, 0).Value = summarySht.Cells(targetRng.Row,
templateProtein(i, 2)).Value
Next

thdSht.Range("A1").EntireColumn.AutoFit

'make the changes to column A (PDBID => ID), 3-code, 1-code

'save the workbook as
targetPDBHybrid.SaveAs Filename:=ThisWorkbook.Path & "\" & PDB_ID & "_thd_" &
targetRng.Value & ".xlsx", FileFormat:=51
targetPDBHybrid.Close savechanges:=False
End Sub

Sub testTP()
Dim summarySht As Worksheet
Set summarySht = Sheets("SummarySheet")
Dim i As Integer, numOfRows As Integer, j As Integer
numOfRows = Range(summarySht.Range("A1"), summarySht.Range("A1").End(xlDown)).count
j = 9  'change this value
For i = 1 To numOfRows
    If Not (i = j) Then
        threadProt summarySht.Range("A" & j), summarySht.Range("A" & i)
    End If
Next
End Sub
```

7.9 Excel VBA Script for Deriving $n_{res,bin}$ When Each PDB Structure (Identified in Different Clusters) is Equally Weighted

```
'Calc n res bin, Calc n bin, Calc n res, and Calc n total are calculated very similarly
and involves proper filtering parameters.
For purposes of maintaining brevity, only Calc n res bin is shown.

Function Calc_n_res_bin(Residue As String, zRng_LBound As Double, zRng_UBound As Double)
As Double

Dim i As Integer, n_resBin As Double
Dim currAutoFilter As AutoFilter
n_resBin = 0#
Dim clusterInfo As range
ThisWorkbook.Activate
Set clusterInfo = range(Sheets(1).range("A2"), Sheets(1).range("A2").Offset(0,
1).End(xlDown))

For i = 3 To Sheets.Count
    Dim sizeOfCluster_i As Variant, n_resBin_i As Integer

    Sheets(i).Activate
    Sheets(i).AutoFilterMode = False
    Sheets(i).range("A1:J1").AutoFilter

    sizeOfCluster_i = Application.WorksheetFunction.VLookup(Sheets(i).Name, clusterInfo,
2, 0)

    n_resBin_i = CDbl(filterFor_P_ResBin(Sheets(i), Sheets(i).range("A1:J1"), 5, Residue,
, 8, zRng_LBound, zRng_UBound))
    n_resBin = n_resBin + (1 / sizeOfCluster_i) * n_resBin_i

Next

Calc_n_res_bin = n_resBin
End Function

'-------------------------------------------------------------

Function filterFor_P_ResBin( _
    targetSheet As Worksheet, targetRange As range, myField As Integer, myCriteria As
String, Optional myCriteria_2 As Variant, _
    Optional zField As Variant, Optional zF_LBound As Variant, Optional zF_Ubound) As
Integer
'For the purpose of counting residues belonging to single and multiple residue indices,
identity and bin types, this function filters for these specific criteria

targetSheet.Activate
targetSheet.AutoFilterMode = False
Dim numOfRows As Long
numOfRows = range(targetSheet.range("A1"), targetSheet.range("A1").End(xlDown)).Count

targetRange.AutoFilter
Dim myCriteriaList() As String
myCriteriaList = Split(myCriteria, ",")


If IsMissing(myCriteria_2) And IsMissing(zField) And IsMissing(zF_LBound) And
IsMissing(zF_Ubound) Then
    If UBound(myCriteriaList) > 1 Then
        targetRange.AutoFilter Field:=myField, Criteria1:=myCriteriaList,
Operator:=xlFilterValues
    Else
        If IsNumeric(myCriteriaList) Then
            targetRange.AutoFilter Field:=myField, Criteria1:="=" & CStr(myCriteria),
Operator:=xlFilterValues
        Else
            targetRange.AutoFilter Field:=myField, Criteria1:=CStr(myCriteria),
Operator:=xlFilterValues
```

```
        End If
    End If
ElseIf IsMissing(myCriteria_2) And Not (IsMissing(zField)) And Not (IsMissing(zF_LBound))
And Not (IsMissing(zF_Ubound)) Then
    If UBound(myCriteriaList) > 1 Then
        targetRange.AutoFilter Field:=myField, Criteria1:=myCriteriaList,
Operator:=xlFilterValues
        targetRange.AutoFilter Field:=zField, Criteria1:=">=" & zF_LBound,
Operator:=xlAnd, Criteria2:="<=" & zF_Ubound
    Else
        If IsNumeric(myCriteriaList) Then
            targetRange.AutoFilter Field:=myField, Criteria1:="=" & CStr(myCriteria),
Operator:=xlFilterValues
            targetRange.AutoFilter Field:=zField, Criteria1:=">=" & zF_LBound,
Operator:=xlAnd, Criteria2:="<=" & zF_Ubound
        Else
            targetRange.AutoFilter Field:=myField, Criteria1:=CStr(myCriteria),
Operator:=xlFilterValues
            targetRange.AutoFilter Field:=zField, Criteria1:=">=" & zF_LBound,
Operator:=xlAnd, Criteria2:="<=" & zF_Ubound
        End If
    End If
ElseIf Not (IsMissing(myCriteria_2)) And (IsMissing(zField)) And (IsMissing(zF_LBound))
And (IsMissing(zF_Ubound)) Then
    'assuming range between two numbers (i.e. between two z values, criteria 1 should be
<criteria 2, neither is splittable)
    targetRange.AutoFilter Field:=myField, Criteria1:=">=" & myCriteria, Operator:=xlAnd,
Criteria2:="<=" & myCriteria_2
End If


filterFor_P_ResBin = numOfRowsAutofiltered(targetSheet, numOfRows)

End Function

'-------------------------------------------------------------

Function numOfRowsAutofiltered(targetSheet As Worksheet, maxOfRows As Long) As Integer
'Counts number of visible rows remaining after filtering for your criteria

        targetSheet.Activate
        range(range("A2"), range("A2").End(xlDown)).Select
        Selection.SpecialCells(xlCellTypeVisible).Select

        If (Selection.Count > maxOfRows) Then
            numOfRowsAutofiltered = 0
        Else
            numOfRowsAutofiltered = Selection.Count
        End If
        range("A1").Select
End Function
```

7.10 Calculating the Geometric Median using the Weiszfeld Algorithm

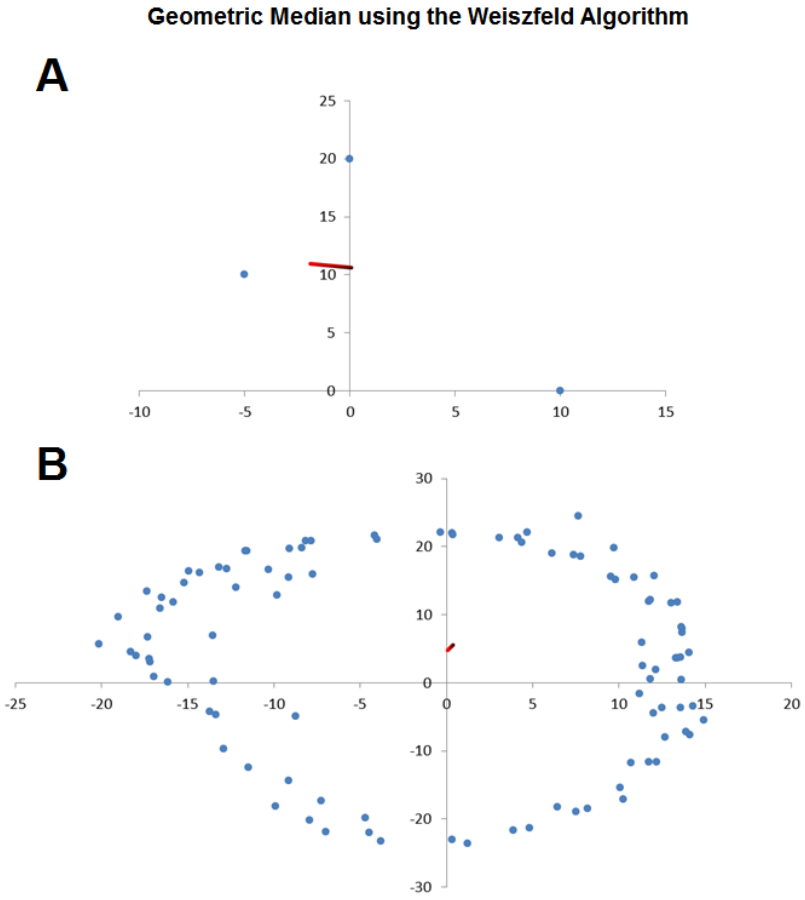


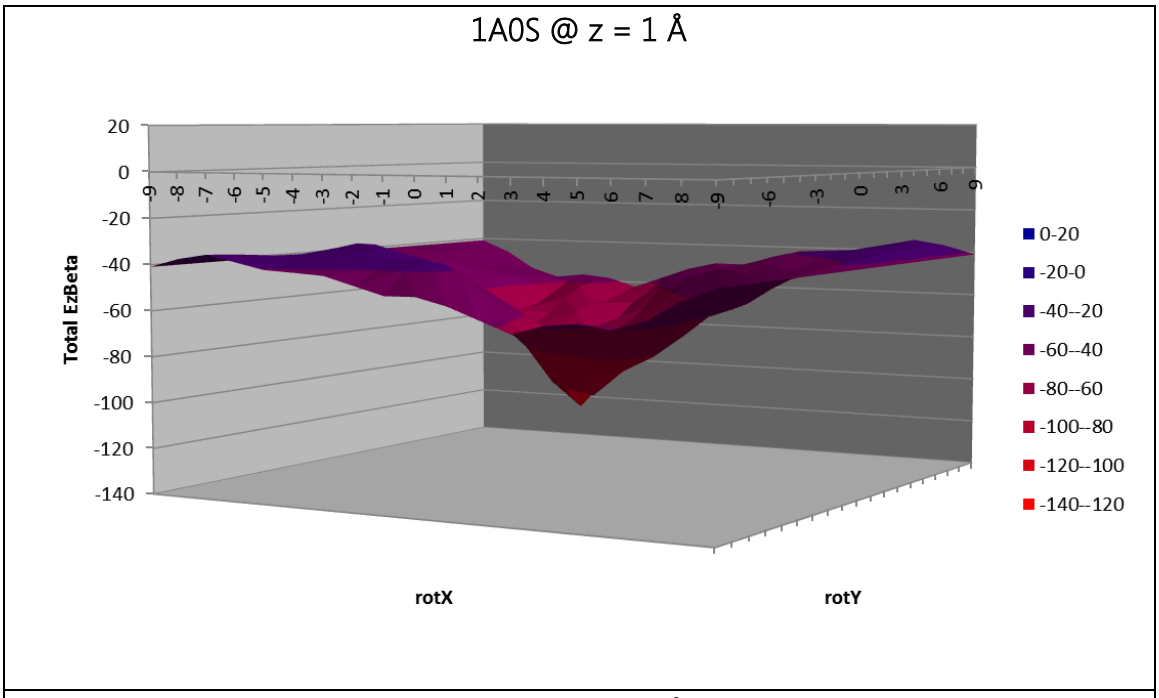
Figure 30 - Calculating the geometric median using the Weiszfeld algorithm

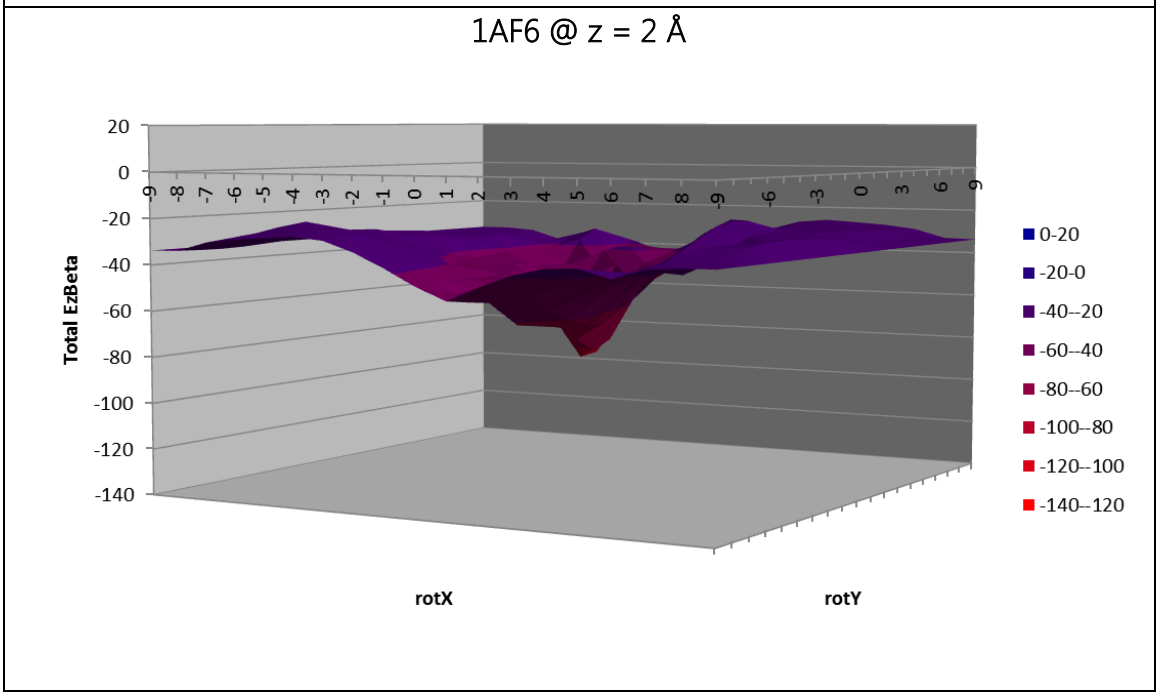
Given data points (shown in blue), the geometric median is arrived at using the iterative Weiszfeld algorithm. The process of sequence convergence is colored in red-to-black gradient, with the initial point colored red. This convergence process is not linear. The initial guess of the Weiszfeld algorithm is the center of mass. The inverse weights of adjusting the geometric center are iteratively calculated by the sum of individual deviations with respect to the guess of the geometric center. Geometric median of (Fig. A) a triangle and (Fig. B) the 2-dimensional Cα coordinates of the sucrose porin (PDB 1A0S).

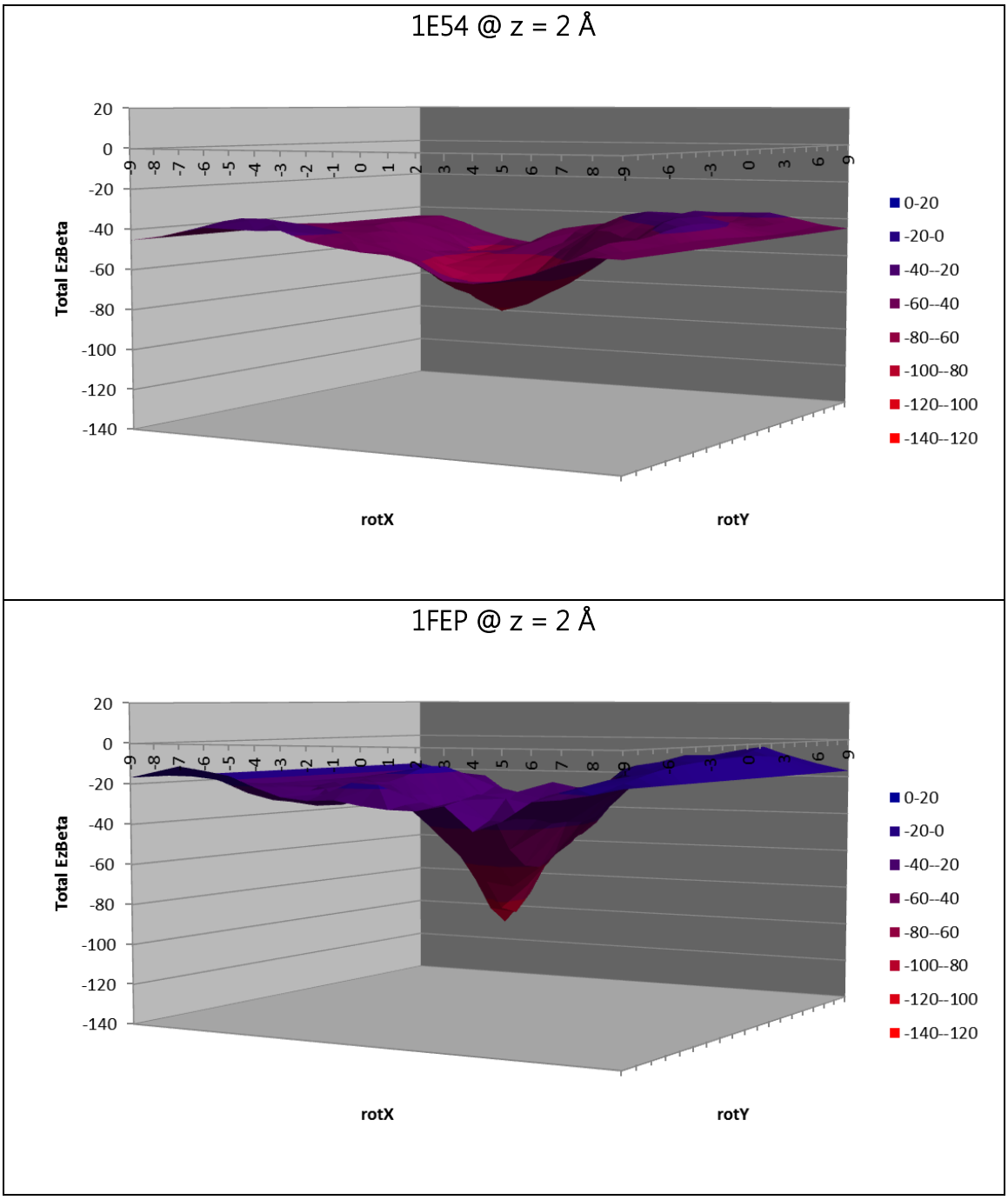
7.11 Energy Landscapes of Proteins in Dataset

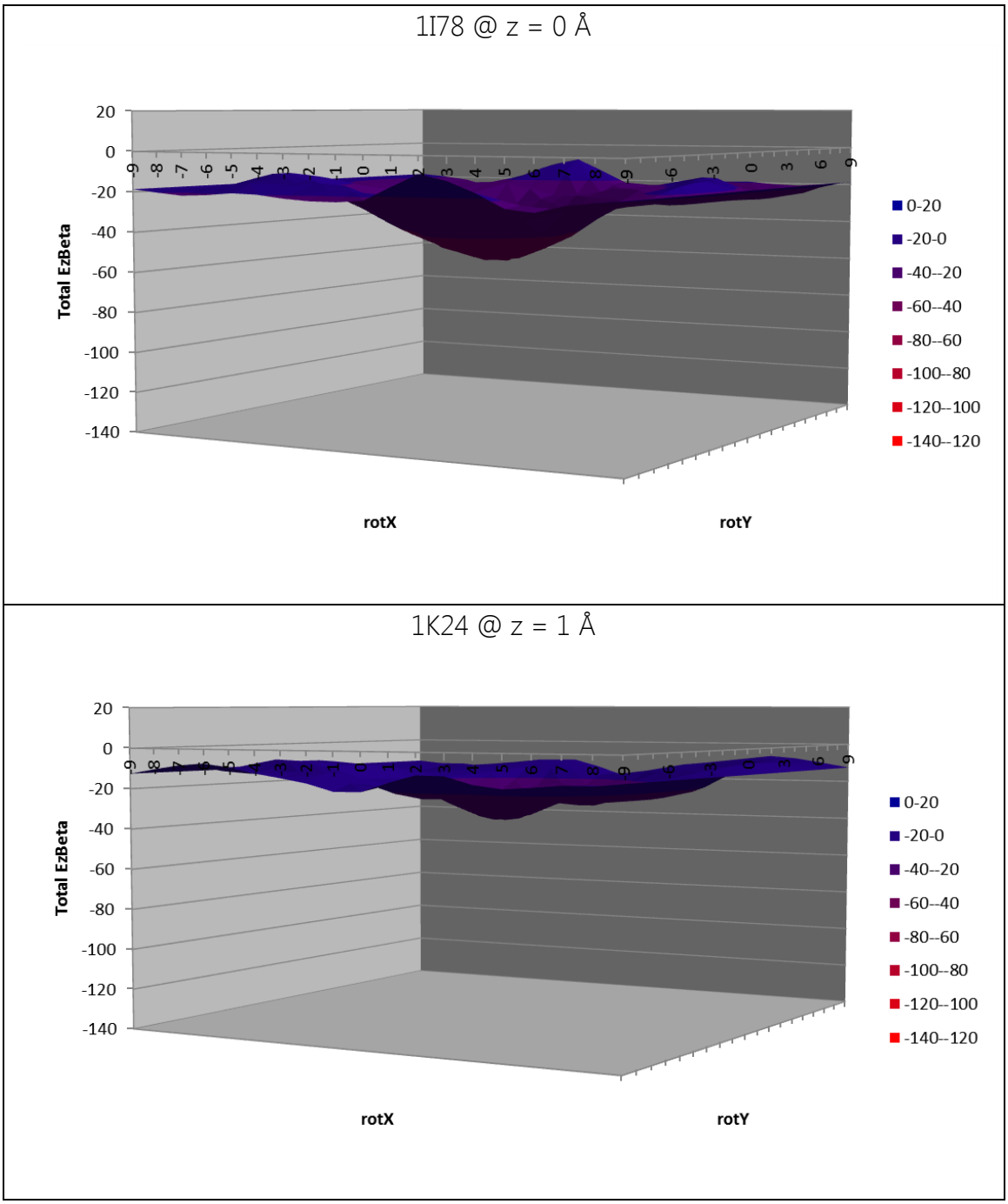
For each protein in our dataset, a snapshot is taken where the $E_z\beta$ energy landscape at a fixed z-depth with respect to the bilayer contains the lowest energy funnel point. The energy surface is colored using a blue-to-red gradient scheme, with red as the lowest energy value.

Table 7 - Energy Landscapes of Proteins in Dataset at the Energy Minimum Depth

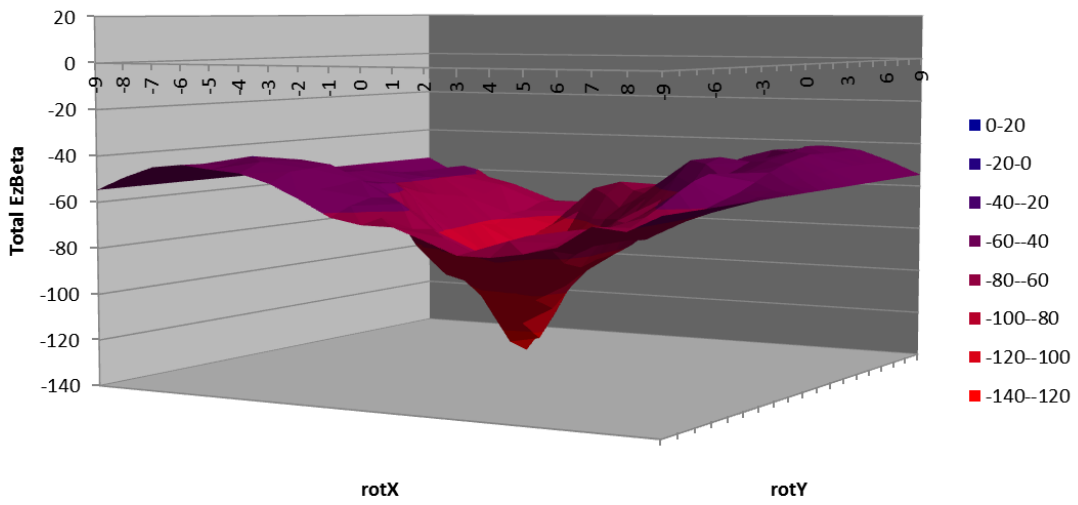




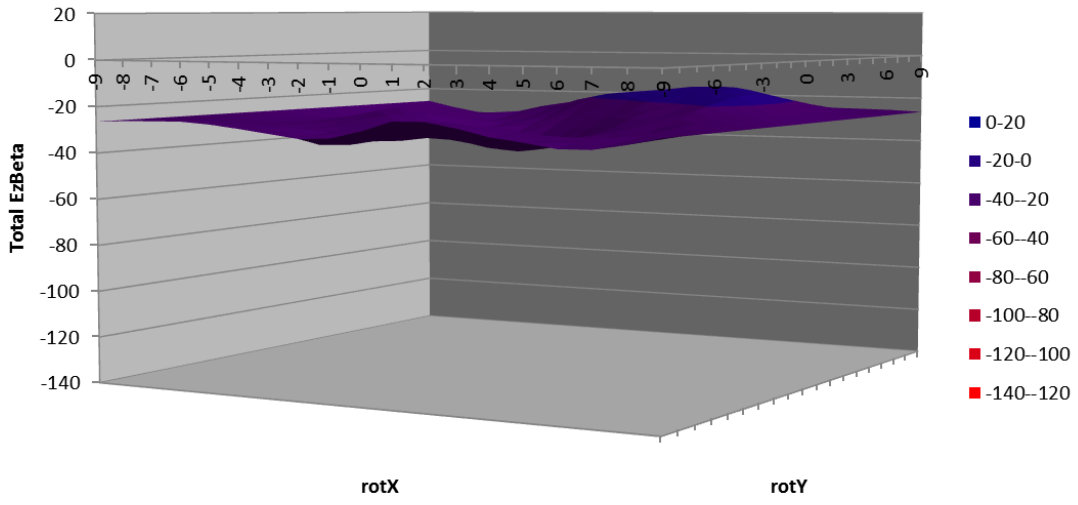

1E54 @ z = 2 Å
Total EzBeta
20
0
-20
-40
-60
-80
-100
-120
-140
-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9
-9 -6 -3 0 3 6 9
rotX
rotY
0-20
-20-0
-40--20
-60--40
-80--60
-100--80
-120--100
-140--120
1FEP @ z = 2 Å
Total EzBeta
20
0
-20
-40
-60
-80
-100
-120
-140
-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9
-9 -6 -3 0 3 6 9
rotX
rotY
0-20
-20-0
-40--20
-60--40
-80--60
-100--80
-120--100
-140--120


1I78 @ z = 0 Å
Total EzBeta
20
0
-20
-40
-60
-80
-100
-120
-140
rotX
rotY
0-20
-20-0
-40--20
-60--40
-80--60
-100--80
-120--100
-140--120
1K24 @ z = 1 Å
Total EzBeta
rotX
rotY
0-20
-20-0
-40--20
-60--40
-80--60
-100--80
-120--100
-140--120

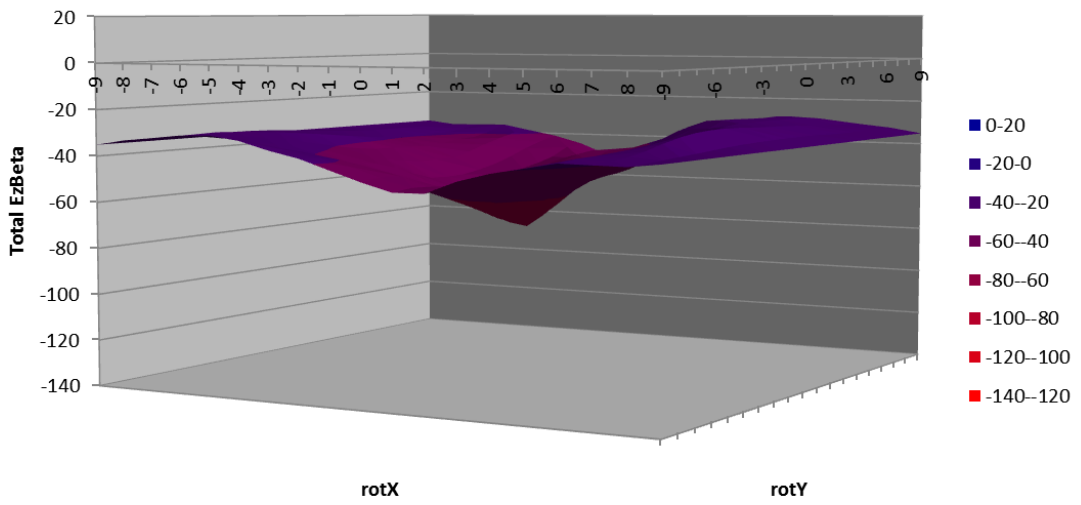
1KMO @ z = 0 Å



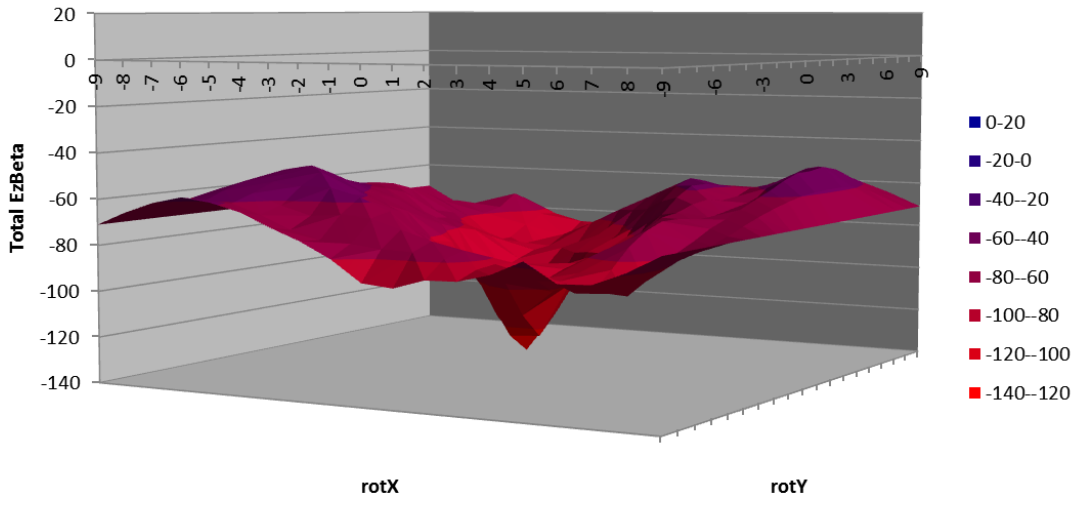
1P4T @ z = 1 Å



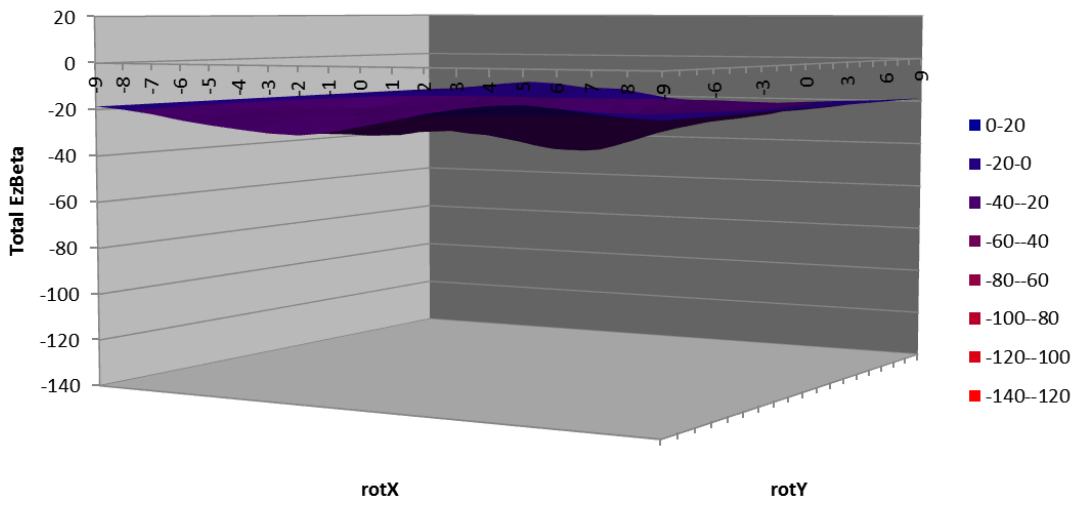
1QD6 @ z = 1 Å



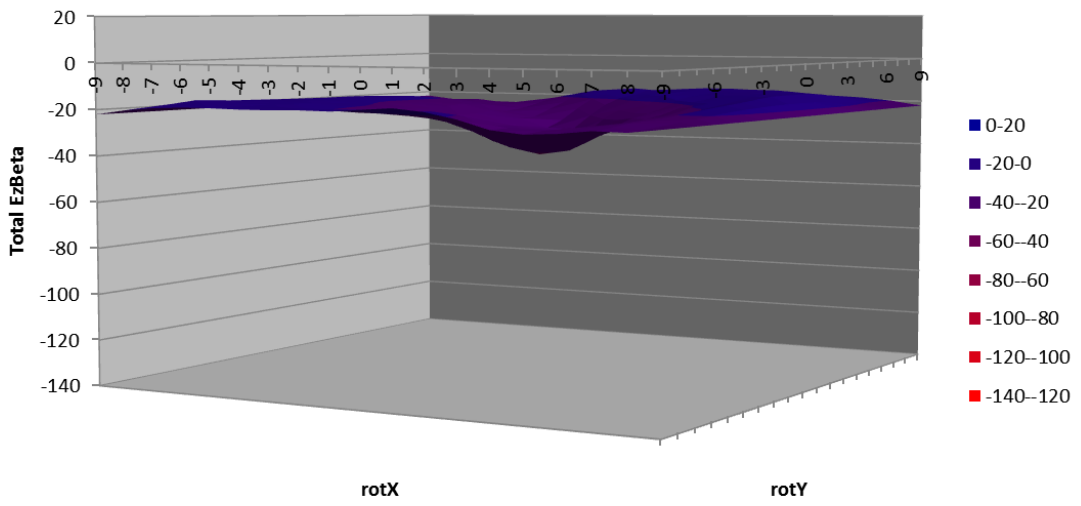
1QFG @ z = 0 Å

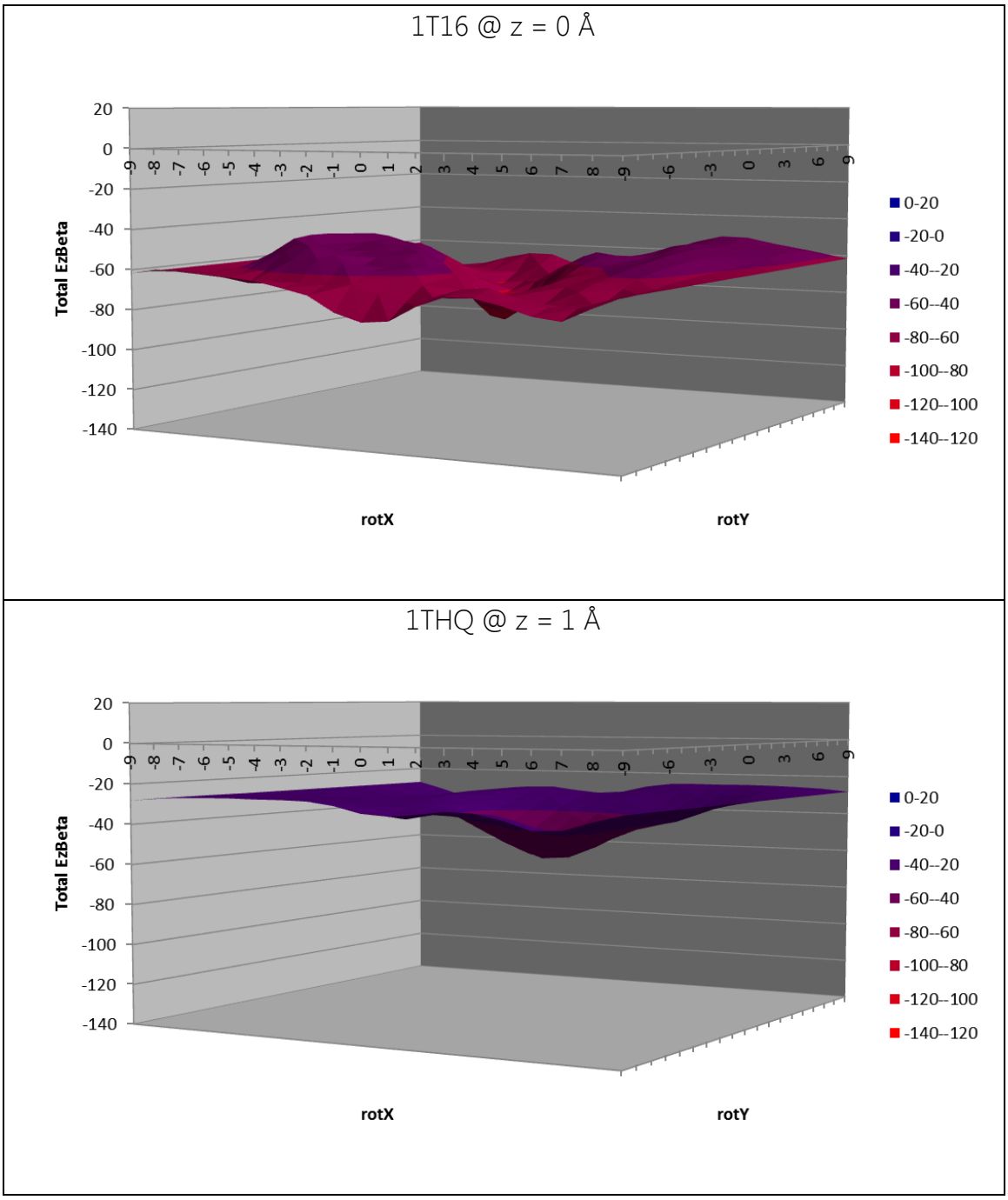


1QJ8 @ z = 0 Å

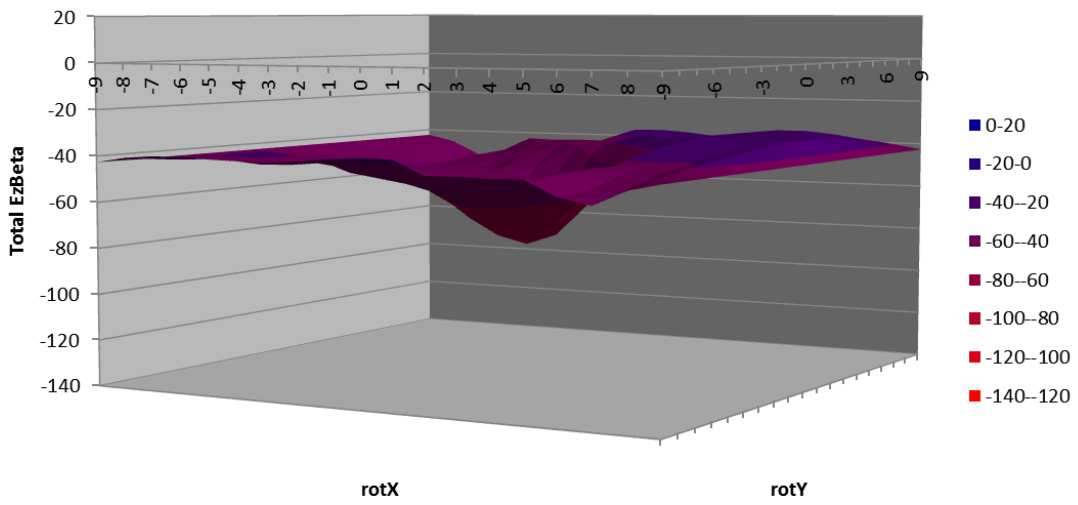


1QJP @ z = 1 Å

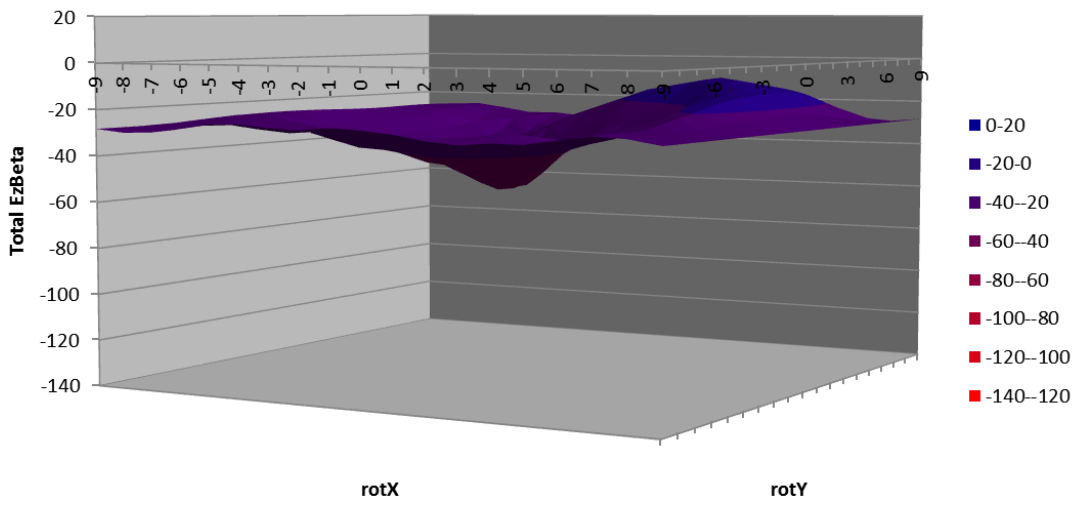



1T16 @ z = 0 Å
Total EzBeta
20
0
-20
-40
-60
-80
-100
-120
-140
-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2
3 4 5 6 7 8 9
-9 -6 -3 0 3 6 9
rotX
rotY
0-20
-20-0
-40--20
-60--40
-80--60
-100--80
-120--100
-140--120
1THQ @ z = 1 Å
Total EzBeta
20
0
-20
-40
-60
-80
-100
-120
-140
-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2
3 4 5 6 7 8 9
-9 -6 -3 0 3 6 9
rotX
rotY
0-20
-20-0
-40--20
-60--40
-80--60
-100--80
-120--100
-140--120

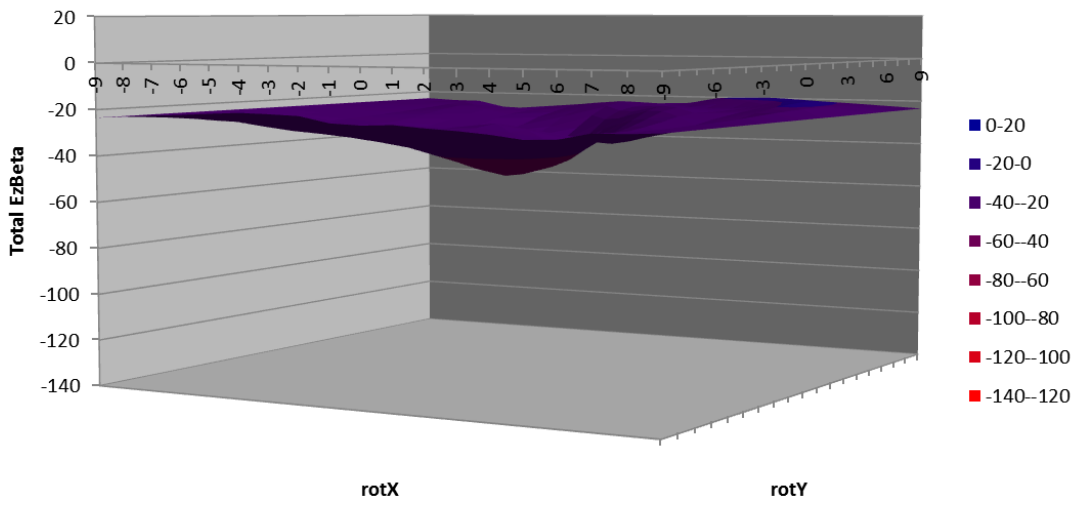
1TLY @ z = 2 Å



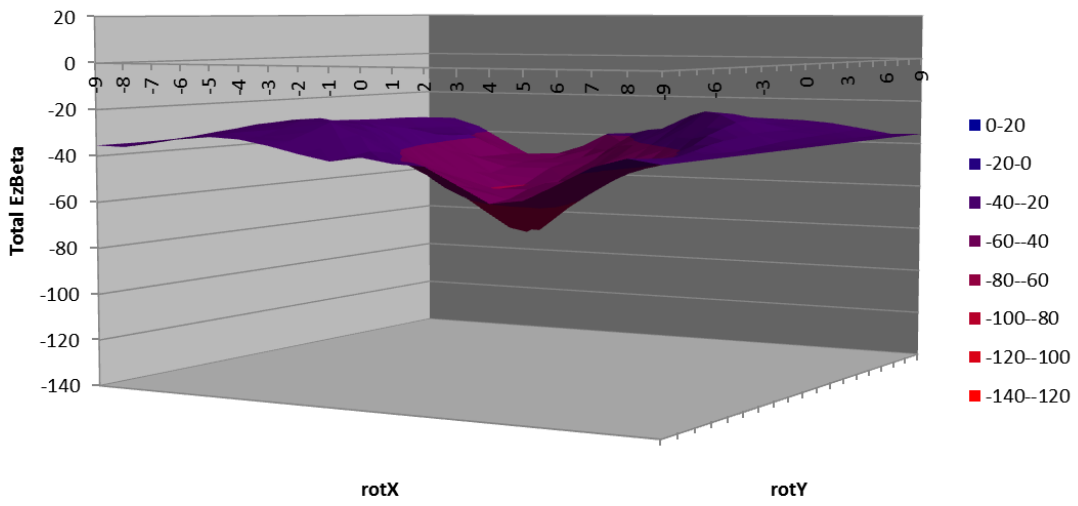
1UYN @ z = 1 Å



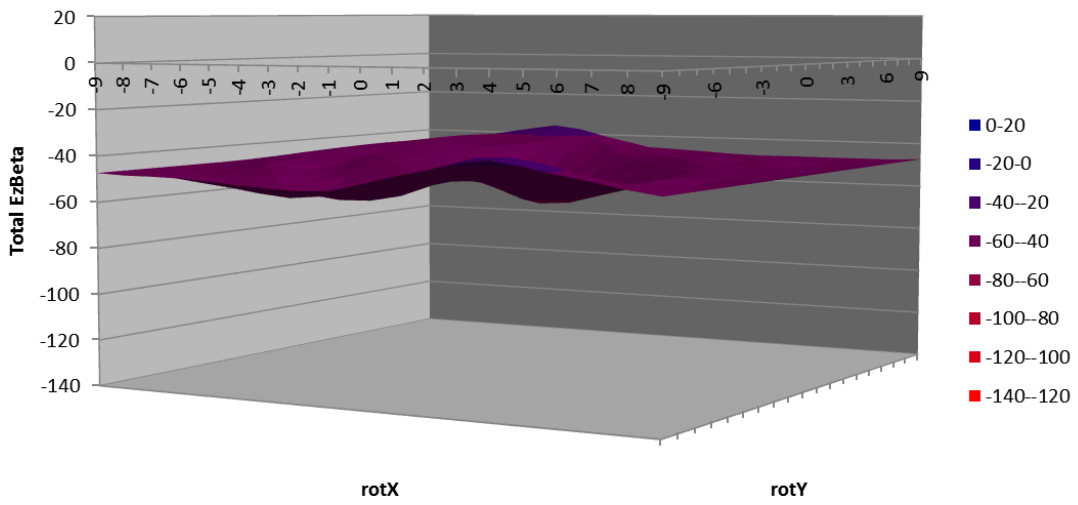
2ERV @ z = 1 Å



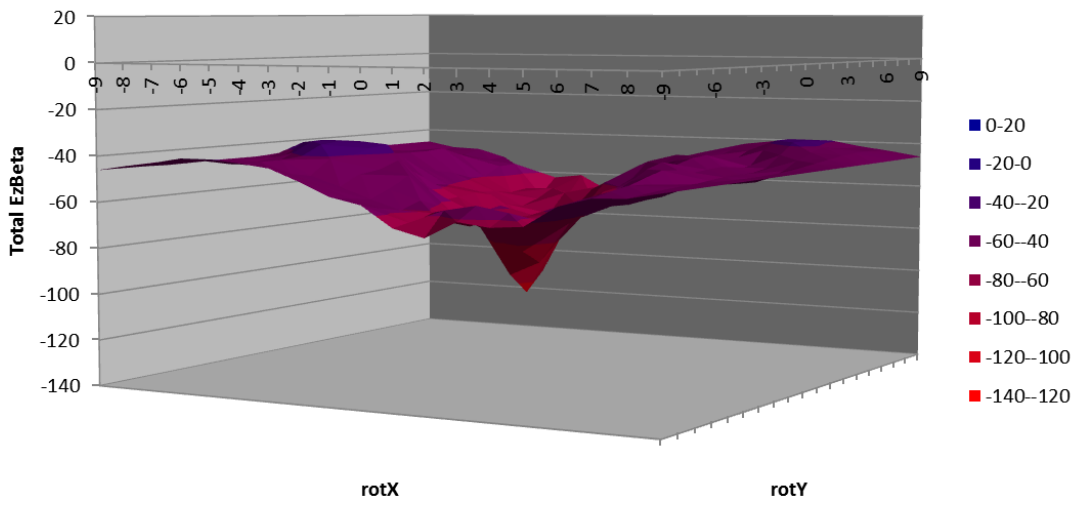
2F1C @ z = 2 Å



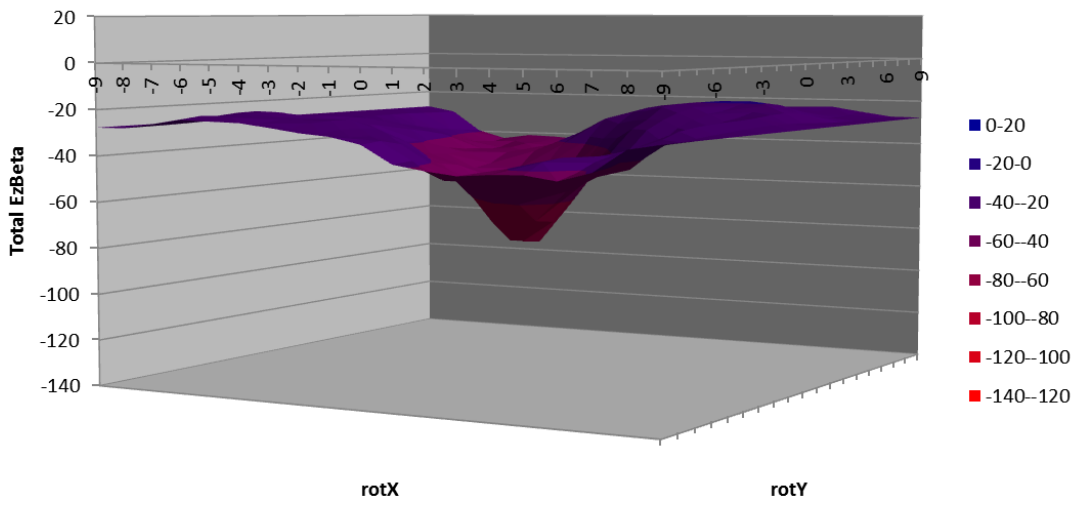
2F1V @ z = 0 Å



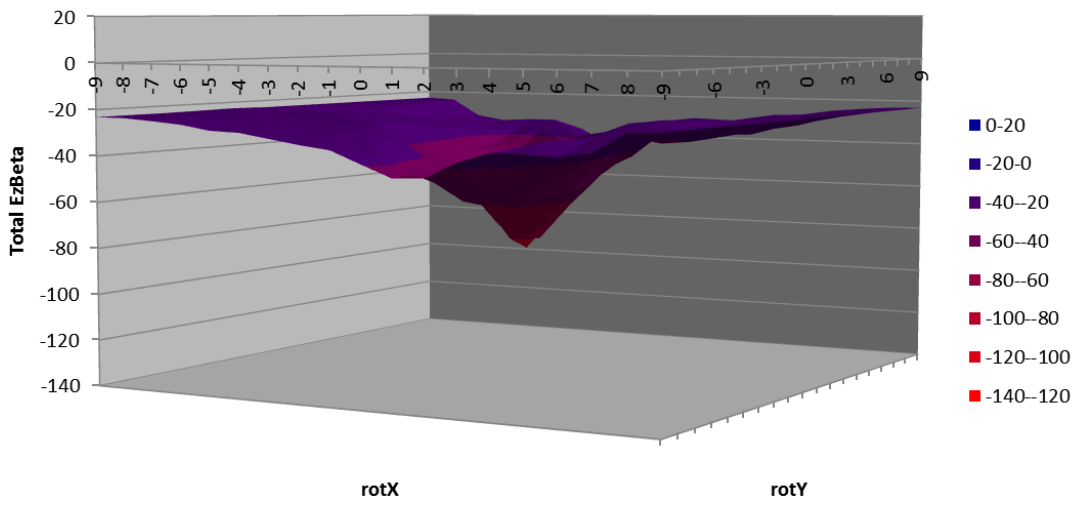
2GUF @ z = 2 Å



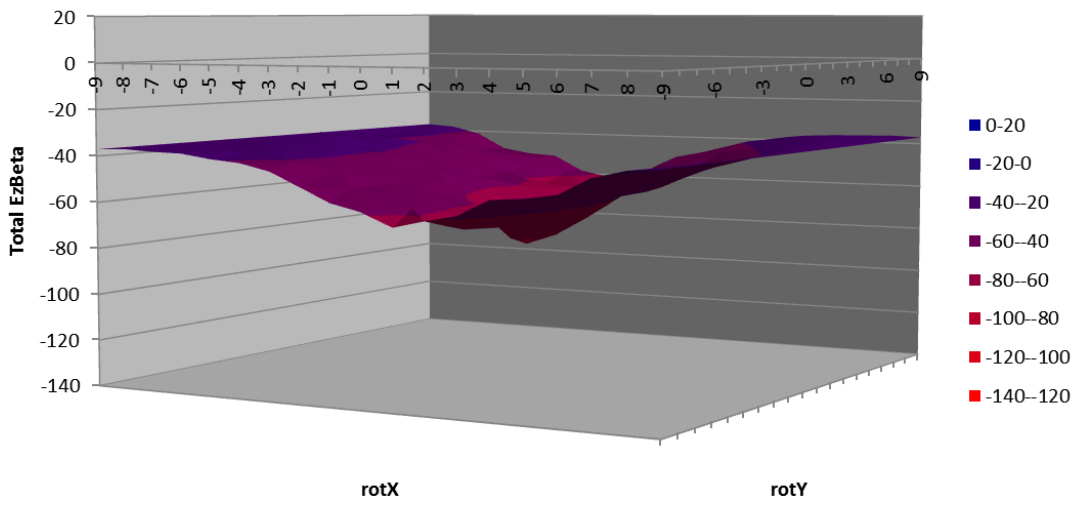
2J1N @ z = 2 Å



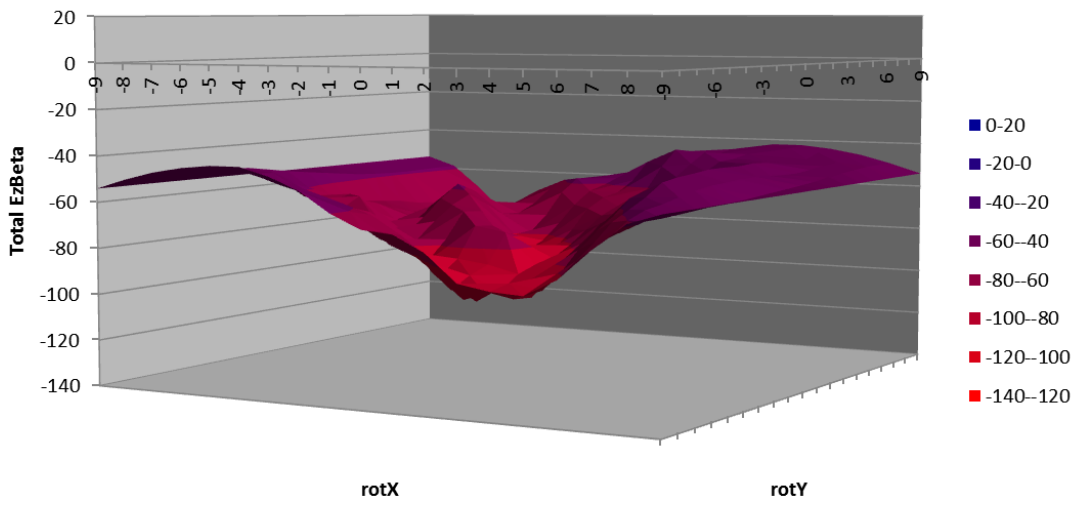
2O4V @ z = 2 Å



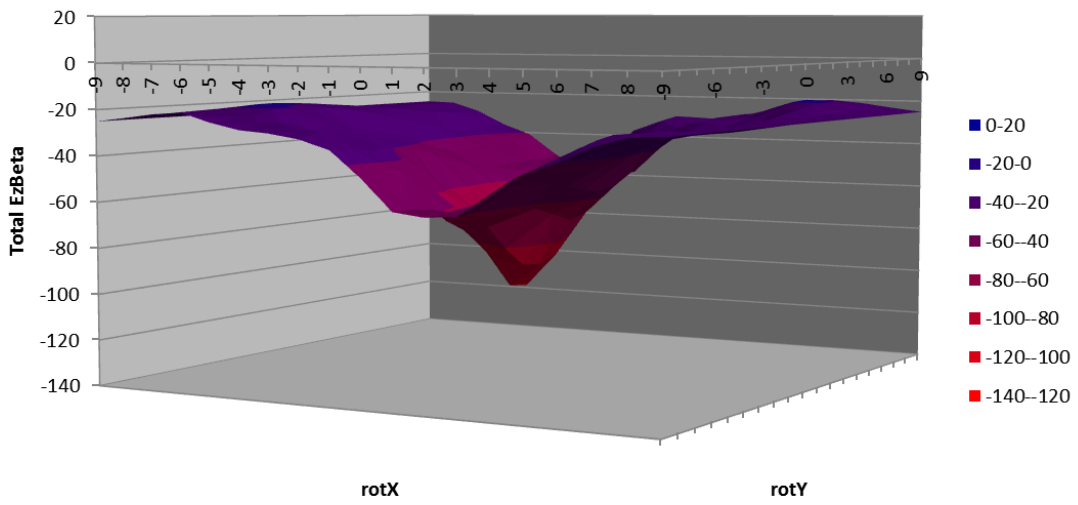
2POR @ z = 2 Å



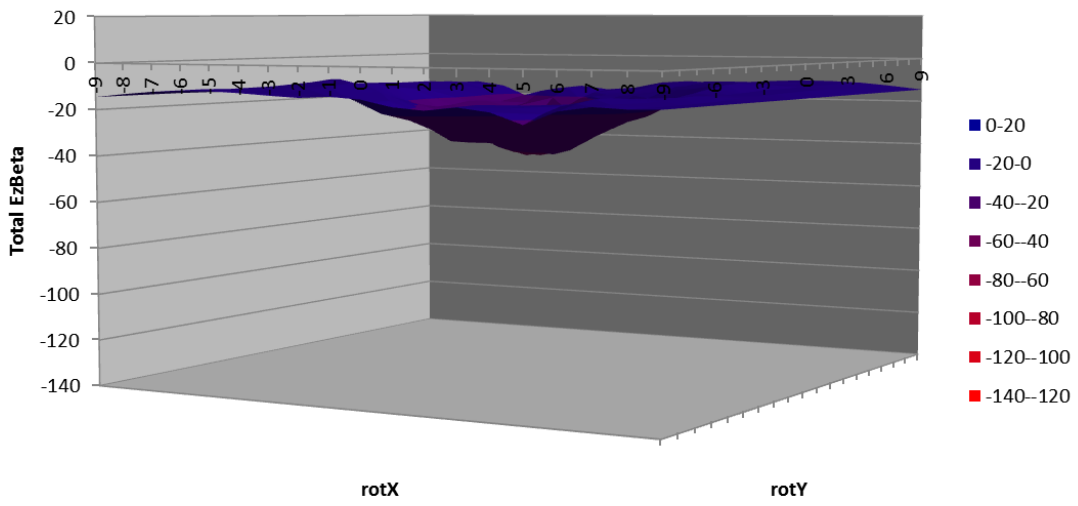
2QDZ @ z = 1 Å



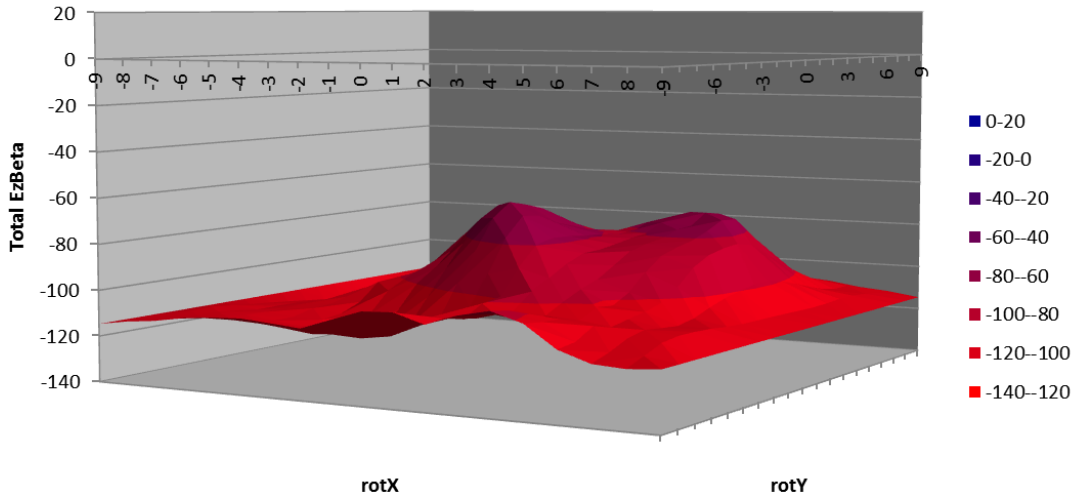
2VQI @ z = 2 Å



2WJR @ z = 3 Å

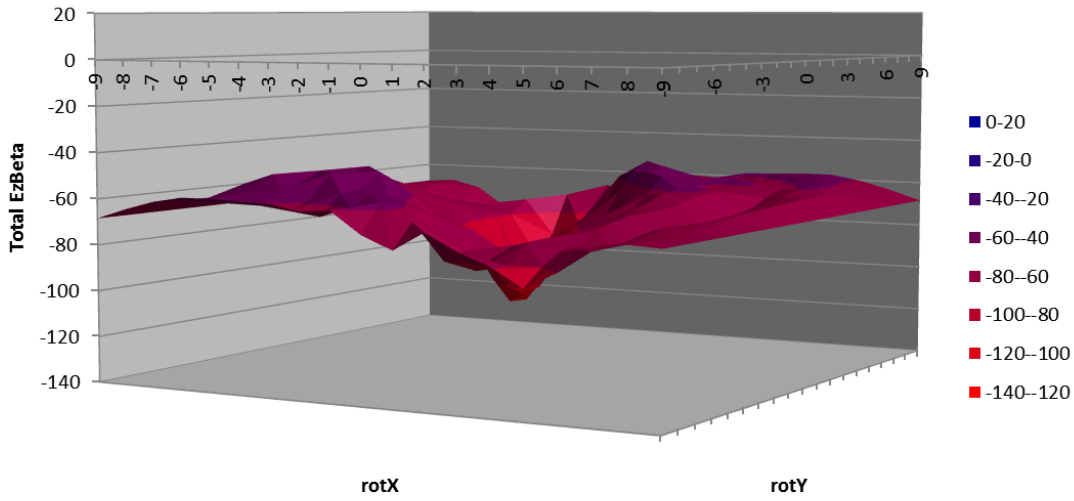


3BS0 @ z = -9 Å

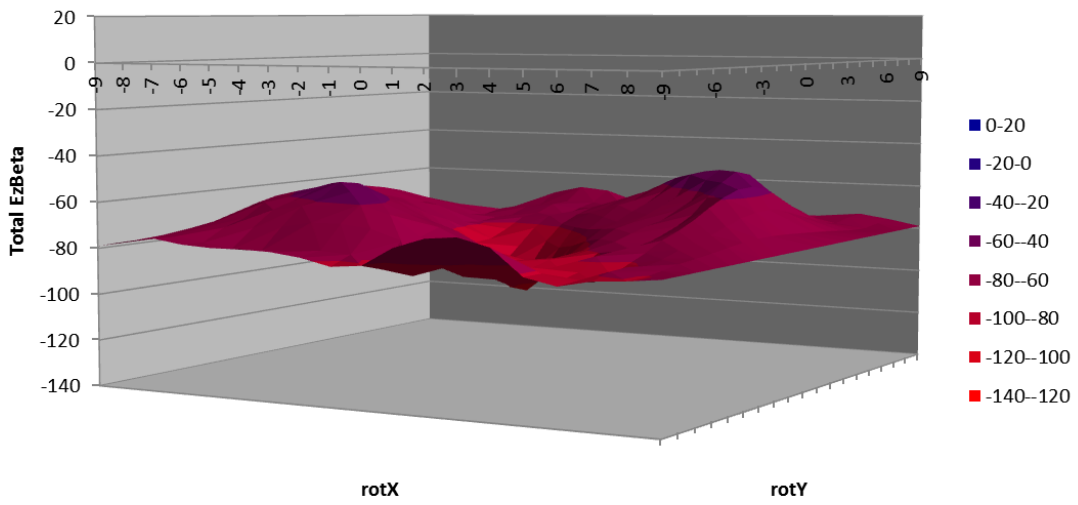


*This TMB resembles one complete barrel with a half-formed barrel pinching off from the extracellular loops. The resulting "best orientation" is therefore a barrel tilted on its side.

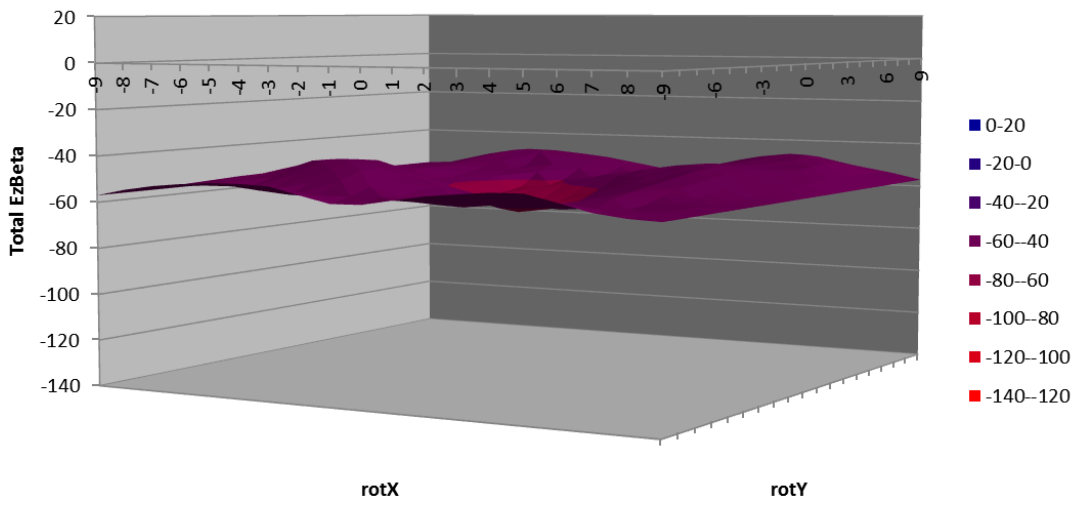
3CSL @ z = 1 Å



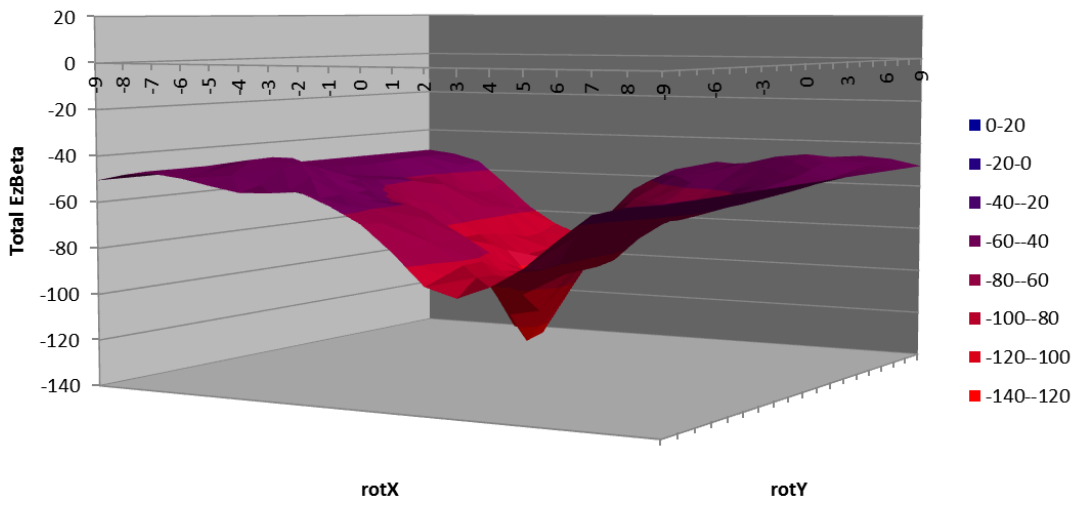
3DWO @ z = 0 Å



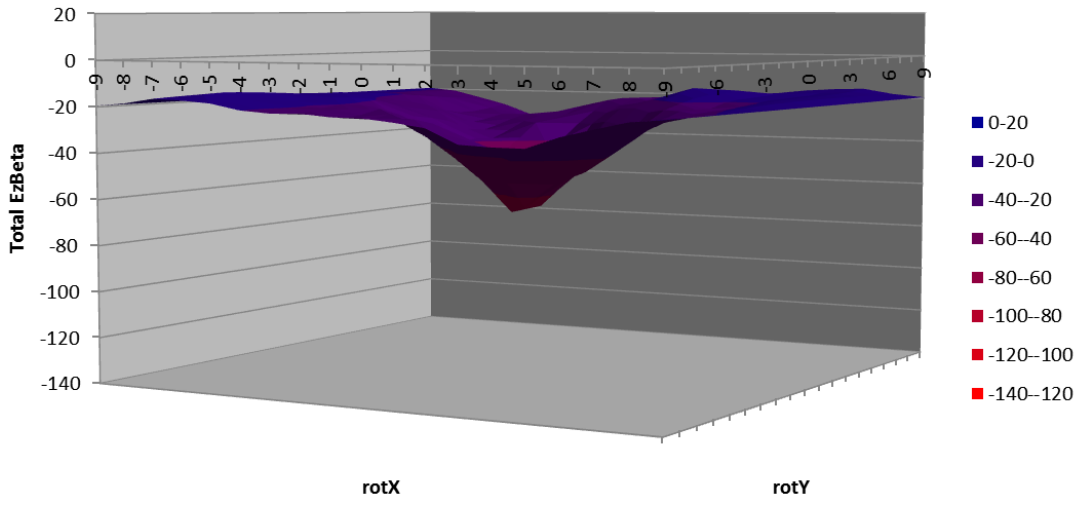
3DZM @ z = 0 Å



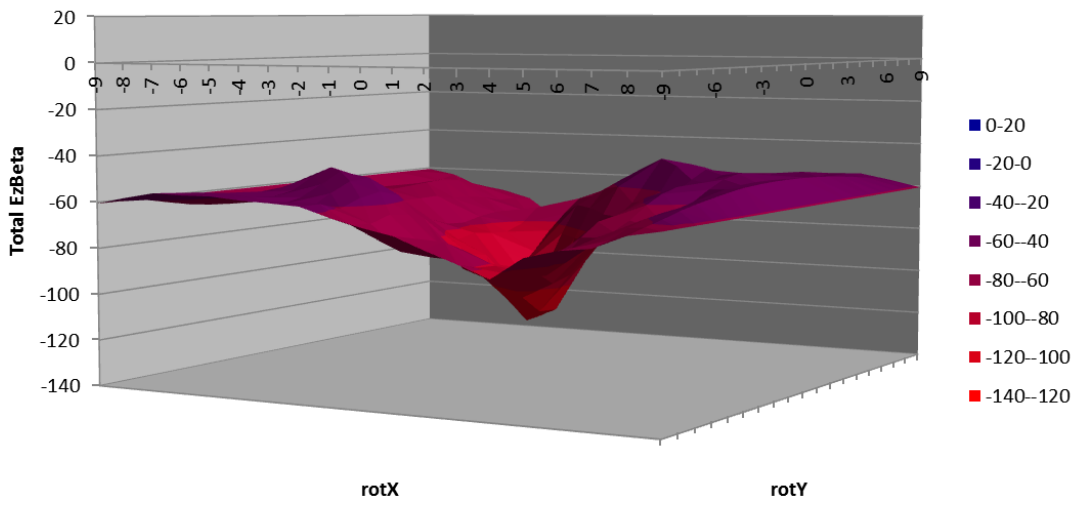
3EFM @ z = 0 Å



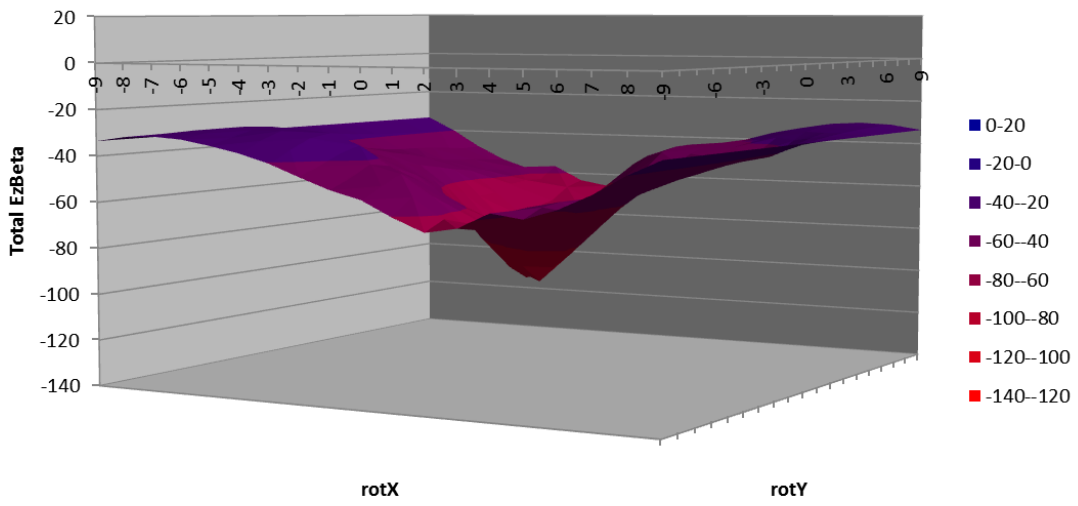
3EMN @ z = 3 Å

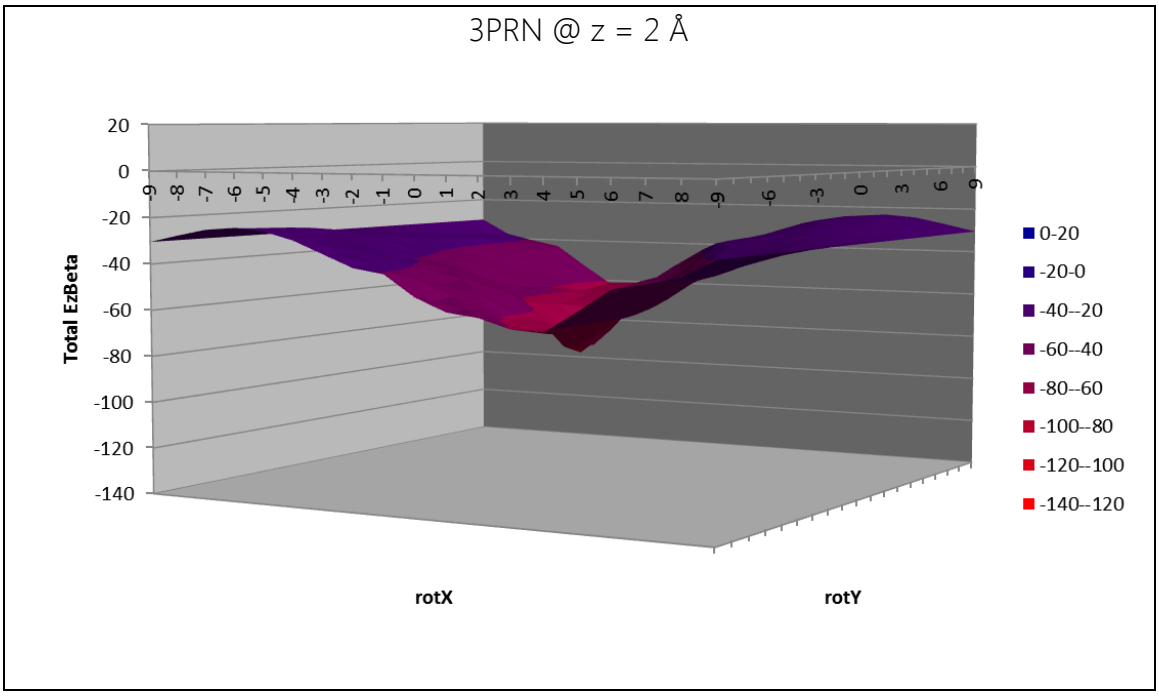


3FHH @ z = 0 Å



3JTY @ z = 1 Å




3PRN @ z = 2 Å
Total EzBeta
20
0
-20
-40
-60
-80
-100
-120
-140
-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9
-9 -6 -3 0 3 6 9
rotX
rotY
0-20
-20-0
-40--20
-60--40
-80--60
-100--80
-120--100
-140--120

Curriculum Vitae

Daniel Hsieh

Education

Ph.D. Computational Biology and Molecular Biophysics, October 2012.
Rutgers, The State University of New Jersey, Piscataway, NJ 08854.

B.A. Mathematics and Economics, May 2006.
New York University, New York, NY 10003.

Experience

Graduate Research	Center for Advanced Biotechnology and Medicine Rutgers, The State University of New Jersey	2006 - 2012
Teaching	Rutgers, The State University of New Jersey	2009 - 2012
Summer Analyst	J.P. Morgan and Chase Securities, New York City, NY 10017 North American Credit High Grade Research	2004 - 2005

Publications

Nanda V, Hsieh D, Davis A. 2012. Prediction and design of outer membrane protein protein-protein interaction sites. In: Ghirlanda G and Senes A, Eds. Membrane protein design. In press.

Hsieh D, Nanda V. (under review). Dynamic surface charts for scattered 4-D data in Excel spreadsheets. Electronic Journal of Spreadsheets in Education.

Hsieh D, Davis A, Nanda V. 2012. A knowledge-based potential highlights unique features of membrane alpha-helical and beta-barrel protein insertion and folding. Protein Science, 21: 50-62.

Nanda V, Xu F, Hsieh D. 2009. Modulation of intrinsic properties by computational design – from stability to catalysts. In: Cochran J and Park SJ, Eds. Protein engineering and design. 2009, CRC Press.