STRUCTURAL MODES IN PROTEINS:
A CASE STUDY IN THE PDZ DOMAIN

APPROVED BY SUPERVISORY COMMITTEE

Rama Ranganathan, M.D., Ph.D.

Johann Deisenhofer, Ph.D.

Kevin Gardner, Ph.D.

Hongtao Yu, Ph.D.

DEDICATION

I dedicate this work to my family, my friends, and to all those who have been involved in my education.

ACKNOWLEDGEMENTS

There are so many people to thank for their contributions and support during and in preparation for my doctoral training, that I feel my part in this endeavor was really rather minor.

First, I would like to thank my committee members – Johann Deisenhofer, Kevin Gardner, and Hongtao Yu for their incredibly wise advice and guidance. My only regret is that I did not spend more time seeking their wisdom.

Second, but most importantly, I would like to thank Rama. I am grateful for the original opportunity to join his lab, for the open and equal intellectual environment that he promoted, for the large degree of independence he allowed (including tolerating minor insubordination at times), for the inspiration to approach and tackle difficult yet important problems, and most of all, for the excitement for science that never leaves him. I owe a huge debt to the members of the Ranganathan lab, everyone one of them. I especially thank Dr. Bill Russ who trained me as a rotation student, collaborated with me on several projects, acted as a sounding board for many crazy ideas, and provided a steady friendship during the years that we worked no more than four feet apart. I must also recognize Rohit Sharma, Chris Larson, and Rick McLaughlin who worked before me and with me on projects within the lab.

My main thesis project involved nuclear magnetic resonance spectroscopy – a technique that I neither knew before I began the project nor was routinely practiced in our lab. I am deeply indebted to Pilong Li for his incredibly patient NMR teaching and good-natured friendship throughout the process. I literally would not have been able to do any of my main experiments without the support of the NMR community at UTSW – namely Michael Rosen for allowing me to collaborate with his group, Kevin Gardner for helping me to brainstorm and troubleshoot innovative NMR experiments and for opening his lab and its members as a resource, and to Carlos Amezcua for running the NMR facility and helping me innumerable times.

Outside of the graduate school, I must thank the Medical Scientist Training Program for their initial willingness to invest in my education and for the support and guidance of the directors: Drs. Michael Brown, Andrew Zinn, Dennis McKearin, and Rodney Ulane. I am also

indebted to Robin Downing and Stephanie Robertson who administrated, coordinated, organized, and cared for our whole program like family. I am also incredibly grateful for the members of my MSTP class who stuck together through the many years – both in good times and bad. I miss you already, and I look forward to all the successes you will have in the future.

I am blessed to have many friends both from the medical community and elsewhere, without whom I would not have been able to complete my training, or at least not remain sane while doing so. I am privileged to have a great best friend, Robert, who has been an incredibly faithful companion despite living many miles away. I am especially thankful for the good times we had while living at Normandy, for Carolyn and her family for providing a surrogate family away from my own, for Ashley, David, and Lauren and the good times we had at the Turtle Creek guesthouse. I am extremely grateful to the Billingsleys for generously opening their home to myself and the lineage of previous medical students whom they supported – it is a great contribution to medicine and science that will never be fully appreciated.

Prior to starting medical and graduate school, I was lucky to work in the lab of Dr. Ted Wensel at Baylor Medical Center, and I would like to thank his graduate student at the time, Matthew Sowa, who was the first to teach me molecular biology and how to work with proteins – the basis for my laboratory skills. I also owe and enduring debt to Dr. Daniel Kim-Shapiro at Wake Forest University who sparked my interest in scientific research and provided three years of instruction, guidance, and inspiration.

Finally, I must thank my parents and my family. To say that they really understood what I was working on in the lab for many years would be a stretch. But, perhaps that makes their constant support all the more impressive. I appreciate all that you have done for me, and I have never questioned your love.

STRUCTURAL MODES IN PROTEINS:
A CASE STUDY IN THE PDZ DOMAIN

by

ALAN MATTHEW POOLE

DISSERTATION

Presented to the Faculty of the Graduate School of Biomedical Sciences

The University of Texas Southwestern Medical Center at Dallas

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

The University of Texas Southwestern Medical Center at Dallas

Dallas, Texas

May, 2012

STRUCTURAL MODES IN PROTEINS:
A CASE STUDY IN THE PDZ DOMAIN

Publication No. _____

ALAN MATTHEW POOLE, M.D., Ph.D.

The University of Texas Southwestern Medical Center at Dallas, 2014

Supervising Professor: Rama Ranganathan, M.D.,Ph.D.

Based on studies of protein structures, functional mutagenesis, allosteric control, and evolutionary records of proteins, we propose that proteins are built with an architecture of strong and weak interactions leading to the cooperative behavior of a few residues and the independence of many others. This pattern of heterogeneity determines many characteristics of proteins such as allosteric communication, enzymatic activity, and ligand-binding hot-spots. In addition, this architecture is likely to be a consequence of evolutionary selection and necessary in order to maintain viability while undergoing mutation and adaptation.

Herein, I demonstrate the existence of a heterogeneous architecture in an individual protein (the third PDZ domain from rat PSD95) by measuring physical interactions between all pairs of residues. This global perturbation analysis is performed by making evolutionarily conservative mutations at every position in a protein and observing physical effects by

monitoring a large number of NMR chemical shifts at nuclei distributed throughout the protein. The end result is a matrix of interactions between each mutation and all residues in the protein.

Analysis of this chemical shift perturbation matrix reveals subsets of residues that interact strongly and cooperatively. These residues create structural modes that are present in the both free and peptide-bound PDZ3 and include many residues important for peptide-binding, suggesting that these structural modes are organized for the purpose of protein function. Furthermore, structural modes in PDZ3 are highly correlated with the protein sector identified by Statistical Coupling Analysis (SCA) – a measure of residue coevolution – in the PDZ domain family.

This experiment produced a global map of physical interactions in the PDZ domain. The pattern of interactions is consistent with our model of a heterogeneous architecture composed of cooperative and independent residues. In addition, the correlation between structural modes and the SCA protein sector argues that cooperative physical interactions drive evolution in the PDZ domain family. The connection between physical features of individual proteins and statistical properties of protein families has significant applications for modeling complex physical behavior in proteins, for understanding the robustness and evolvability of natural systems, and for designing novel proteins.

TABLE OF CONTENTS

PRIOR PUBLICATIONS

1. Poole, A.M. and R. Ranganathan, Knowledge-based potentials in protein design. Curr Opin Struct Biol, 2006. 16(4): p. 508-13.

LIST OF FIGURES

LIST OF TABLES

LIST OF DEFINITIONS

Cdc-42	cell division cycle 42 (GTP binding protein)
CRIB	Cdc42/Rac interactive binding domain
CRIPT	cysteine-rich interactor of PDZ three
CT	computed tomography
DHHC	a family of integral membrane proteins containing a DHHC motif
DLG4	Discs-large homolog 4 protein
DNA	deoxyribonucleic acid
E. coli	*Escherichia coli*
GST	glutathione S-transferase
HBLV	hybrid backprojection/lower-value
HSQC	heteronuclear single quantum coherence
L	liter
LV	lower-value
MAGUK	membrane-associated guanylate kinase
MCSD	minimum chemical shift difference
mg	milligram
mM	millimolar
MSA	multiple sequence alignment
NMCAA	next most common amino acid
NMDA	N-methyl-D-aspartic acid
NMR	nuclear magnetic resonance
PC	principal component
PCA	principal component analysis
PDZ	PSD95, Discs-large, Zo-1
PDZ3	third PDZ domain from PSD95
ppm	parts per million
PR	projection reconstruction
PSD95	postsynaptic density-95
RNA	ribonucleic acid
RMS	root mean square
SAP90	synapse associated protein 90, alternative name for PSD95
SCA	Statistical Coupling Analysis
SH3	Src homology 3 domain
uL	microliter
WT	wild-type

CHAPTER 1: Introduction

Proteins are impressive because of their simultaneous simplicity and complexity. From a modest twenty amino acid building blocks, Nature builds molecules with an incredibly wide variety of physical properties and highly complex functions. Combinations of proteins create intra- and inter-cellular signaling networks, maintain and propagate genomes, and perform all functions seen in the theater of life – all while existing in a form that is tolerant to random mutagenesis, yet adaptable to perform new functions on an evolutionary time scale. The central question is how is this possible? What features of proteins enable complex functions in the individual molecule while maintaining robustness and evolvability throughout many generations and across species? Can we make a connection between the structural and chemical features that provide for function and the evolutionary properties seen in the genetic record? A full understanding of this connection will not only help us to better understand how natural proteins work (and break), but also enable us to design novel molecules with the desirable properties found in natural proteins.

Collectively, studies of the biophysical, functional, and evolutionary properties of proteins gradually and iteratively constrain our conception of how proteins work – in essence, providing boundary conditions for any complete description of proteins. This description can be called the general architecture of proteins and is meant to refer to how the constituent amino acid residues of the system interact with each other to determine the properties of the system. This idea encompasses the spatial arrangement of the residues, the distribution and trajectory of accessible conformations, the strength of the interactions between residues, and when it is possible to deconvolve, the mechanisms of such interactions. In addition, the concept of a general architecture also includes non-physical properties such as robustness to mutation and the capability to evolve new functions. These non-physical properties cannot be understood by studying single protein molecules/sequences, but are reflected in the distribution of sequences that comprise an homologous protein family.

One theory for a general description of protein architectures arises from the results of the Statistical Coupling Analysis (SCA). SCA is a conservation-weighted measure of coevolution

between residues in a multiple sequence alignment (MSA) of a protein family [1]. The SCA of many protein families routinely identifies coevolving sets of residues that are *sparse* (typically 15-20% of the total residues in the protein), *distributed* to connect different parts of the protein, and *physically contiguous* in the tertiary structure (but not necessarily the primary structure) [2]. These coevolving sets of residues are termed "sectors" and typically cross secondary structure elements and involve residues both at the surface and in the core of the protein. Sector residues have been shown to interact cooperatively and mediate important protein functions [3-5]. Additionally, more than one sector may be present in a protein family, and the different sectors may be important for different properties of the protein [6]. These results suggest a physical model for proteins in which a central set of residues (the protein sector) interact strongly and cooperatively with each other while the rest of the residues interact more weakly and less cooperatively with other residues. This type of architecture is consistent with known biophysical data about proteins and also provides testable hypotheses to explain the functional and evolutionary properties of proteins.

The assertion that a sparse and distributed set of residues act cooperatively to perform a protein's function has been tested in a few focused studies, but there has never been a global test of residue interactions. Previous tests have generally focused on how individual residues affect protein function and less on how residues interact, and especially less on the physical nature of those interactions. My research focuses on testing the hypothesis that the general architecture of proteins is based on a small subset of strongly and cooperatively interacting residues interspersed among a majority of residues that are weakly coupled. I aim to test this hypothesis by performing a global perturbation analysis to map the pattern of physical interactions in a PDZ domain. This novel experimental approach is the first to make a global map of all pair-wise interactions, and as such, the results are uniquely suited to assess the degree of heterogeneity in residue interactions and to test whether SCA-identified coevolving residues interact physically and cooperatively in individual protein molecules.

Protein structures reveal the spatial topology of amino acids

The obvious starting point for studying the architecture of proteins is their three-dimensional structure. Ever since the first protein structures were solved in 1958, we have classically viewed proteins molecules as adopting a single compact, low-entropy, highly-ordered, folded structure. However, we now know that not all proteins assume well-folded structures. In fact, bioinformatics analysis of genome sequencing collected in the last decade predicts that as much as 1/3 of eukaryotic proteins are fully or partially disordered [7, 8]. Furthermore, test cases indicate that this disorder enhances or is required for some proteins' functions [9, 10]. Thus, a more correct view is that protein structures lie on a continuum somewhere between completely disordered and well-folded with high stability. Although the physical mechanisms determining the amount of entropy in a polypeptide chain are extremely interesting from fundamental and practical perspectives, we must also look at the problem from the viewpoint of evolution. Understanding why proteins evolve to have disorder or stability will help us to understand how they acquire these properties.

For the purposes of this document, however, we will (with noted exceptions) restrict our discussion to the set of proteins that form compact, folded, and well-ordered structures. In this group of proteins, a well-ordered structure is essential for maintaining the stability of the molecule and providing a framework for carrying out the function of the molecule such as organizing active site residues in enzymes. Obtaining a three-dimensional structure of these proteins provides a topological description of the network of amino acids – which residues are close to one another, which residues are in contact, and what are their relevant orientations. Protein structures solved by X-ray crystallography, Nuclear Magnetic Resonance (NMR) spectroscopy, and cryo-electron microscopy reveal huge insights into the mechanisms of how proteins function. These structures expose the organization of active sites, the location of substrate, cofactor, and small molecule binding sites, the topology of protein-protein binding interfaces, and the orientation of multi-protein complexes to name just a few of the useful applications. In addition, structures determined in different functional states (e.g. with and without a ligand) also provide clues as to how proteins perform their functions [11] as well as the range of conformations accessible to the molecule [12]. An oft-cited drawback to

crystallography is that it only determines a single set of atomic coordinates even though we know that proteins access many low energy conformations and even sample folding-unfolding transitions quite frequently [13]. Recently however, there has been a push to use classic structure determination techniques (NMR and X-ray crystallography) to look at dynamics as well. By examining the range of NMR modes consistent with data restraints, B-factors from crystal structures, or sets of crystal structures solved in slightly different conditions, we can now get a sense of the flexibility of the molecule and the ensemble of low energy conformations that it populates [12, 14-16].

On its own however, protein structure analysis cannot answer many important questions. To start, structures reveal the physical arrangement of the residues, but not the energetics of inter-residue interactions. Although it is tempting to assign or infer energetic contributions from features such as hydrogen bonds, salt-bridges, and hydrophobic contacts, the true energies in the native environment are not known [17]. In addition, whatever flexibility and dynamic data can be inferred from crystal structures is approximate; other experiments are needed to determine populations of conformers and the rates of interconversion. Thus, while protein structures cannot provide all the information we would like to know about proteins, they are clearly a necessary foundation and starting point for further experimental and computational studies to deepen our understanding of the general architecture of proteins.

Atomic packing is under evolutionary pressure to be good but not perfect

Upon inspection of well-folded protein structures, it has been observed that the amino acids in the molecule fit together much like pieces in a jigsaw puzzle [18]. This observation has motivated an entire field of inquiry to understand how the amino acids in a protein are packed together and what consequence this has on the function of the molecules. How tightly are proteins packed and how specific are local packing interactions? Since atomic packing is likely to control the physical characteristics of proteins (stability, dynamics, stiffness, etc.) we must try to understand how packing influences the general architecture of protein.

It was noted early on that the atomic packing of proteins approaches that of organic crystals [19-24] with similar densities found in the core and at the surface when one considers solvent packing at the surface [25]. These observations have been very influential and have led to the idea that complimentary packing is essential for protein function [26-30]. In fact, several successful computational protein design programs focus on explicitly optimizing the geometric complimentarity of side-chain packing [27, 31, 32]. However, further research points out that while average packing density is quite high, it is neither maximal nor uniform. When examining free volume distributions, Dill concludes that protein interiors look more like liquids and glasses and that the residues "are more like randomly packed spheres near their percolation threshold [33]." These packing imperfections are likely to be important for the function of the protein as local packing density has been shown to correlate with the local flexibility [34]. Thus, we see that proteins form compact, low-entropy structures, but not so well-packed that they lose all internal degrees of freedom. Another interesting and potentially overlooked caveat to packing studies is that average packing density scales with the resolution of structure determination [35].

In addition, average packing density varies between proteins and appears to be an evolutionarily selectable trait. Most conspicuously, proteins from certain thermophilic organisms have increased compactness and density in order to maximize structural stability [36, 37]. This may arise from an increased number of hydrophobic residues in the cores of these proteins [38], more efficient packing of hydrophobic side-chains in the core, greater involvement of residues in secondary structure elements, and an increased number of hydrogen bonds per residue [37]. However, increased packing density is not the only way to stabilize proteins. Berezovsky and Shakhnovich showed that sequence optimization to enrich for charged residues is also seen in thermophilic organisms [37]. A greater proportion of charged residues on the surface has been noted before [39, 40], but Berezovsky and Shakhnovich explain the disparate mechanisms for thermostability by noting different timescales for evolutionary selection. Proteins from ancient Archaea evolved with high packing densities that provided stability but also high designability (ability to tolerate large variations in sequence while maintaining stability and function). Some thermophilic organisms from

Bacteria, however, colonized extreme environments more recently as descendants from mesophilic organisms and obtained enhanced thermostability by sequence optimization, which is potentially faster (requiring fewer mutations) than re-optimizing the entire protein for high efficiency packing. Thus, we see that there is local variation in packing density inside individual proteins and systematic variation in average packing density for entire organisms. All indications are that packing is under strong evolutionary constraint – but biased toward the individual need of each protein to balance dynamics and stability, rather than perfectly optimal packing.

An example of this balance is illustrated in a study by Fraser et al. in which systematic sampling of electron density around side-chain rotamers reveals populations of cooperative alternate side-chain conformations that are crucial for enzymatic catalysis [41]. These coupled side-chain conformations occur at buried sites in the protein indicating the use of specific structural plasticity to allow for the motions needed to accomplish the specific function of the protein. This type of evidence for functional variation in packing often goes unnoticed because protein structures are driven to a single low energy state when solved at cryogenic temperatures commonly used for crystallography and because the alternate populations can be small and below the traditional threshold used to discriminate signal from noise in electron density maps. Another illustration that natural protein are underpacked comes from the field of computational protein design. In some cases where design algorithms emphasize geometric complementarity, they can produce proteins that are hyper-stable when compared to natural proteins [32]. This result suggests that proteins can be more tightly packed than those found in nature and that extra tight packing and hyperstability may be an evolutionary disadvantage.

One concludes then that compactness, stability, and local packing interactions are indeed important and evolutionarily selected, but these properties are not the sole determinants of protein architecture. It appears that packing density and protein stability are selected for favorable properties – to avoid protease degradation, to inhibit protein aggregation, etc. – but also relaxed to allow functional motions. Many obvious questions arise - are all packing interactions equally important? Is all local packing inhomogeneity functionally important, or is much of it random, and if so, is there a way we can discriminate between the

two? How can we better determine the energetic value of packing and residue interactions? How does the heterogeneity of residue interactions contribute to functional properties of the protein? We thus turn to further experiments to answer these questions in light of the frameworks established by experimental structures.

Protein dynamics provide functionally important conformation transitions

While protein structures provide great insights into the mechanisms of protein function (geometry of active sites, location of allosteric sites, organization of multi-domain complexes, etc.), they are somewhat deceiving in the sense that they generally offer a single snapshot of one low energy configuration of the molecule. We know that proteins populate many low energy microstates around a preferred conformation as well as convert to other energetically-accessible conformations, but it can be difficult to determine the populations of said conformations and microstates. In the case of X-ray crystal structures, there is an experimental limitation due to crystals being kept at cryogenic temperatures to promote stability and protect against radiation damage. Below a characteristic temperature around 200-240K, protein molecules lose flexibility and are frozen into a single conformational substate [42-44]. Another limitation stems from the general approach to refine structures toward a single set of atomic coordinates. NMR structures, on the other hand, often represent the structure as an ensemble of models consistent with the collected structural restraints; however, the structural heterogeneity is a convolution of a true distribution of conformations, but also a lack of sufficient structural restraints. At this time, no single method is sufficient to fully describe the collection of conformations and motions within in a protein. A combination of traditional structural determination, computational simulations, and studies aimed at measuring specific dynamic events within proteins will be needed to provide a more complete picture.

In addition to the random sampling of conformational states accessible to the protein at physiologic temperatures, proteins have evolved to utilize these conformational transitions for functional purposes. In the simplest case, a protein samples two conformations – one active and one inactive – but with the populations strongly biased toward the inactive state. In the presence of an activator, such as phosphorylation, the active state is stabilized and the

populations are inverted [45, 46]. In other cases, the intrinsic dynamics of the molecule correlates with function such as determining the rates of enzymatic catalysis by controlling the access to the active site, controlling allosteric interaction between different parts of the protein, or determining electron transfer kinetics [47-52]. There are several assumptions that go into this conclusion, however. First, when determining atomic motions by NMR, we observe independent kinetics for each nucleus, and we infer that these motions are correlated with each other because they have the same rate constants. In the same way, we also assume that these motions are causative for the functional properties of the molecule because the rate constants of motion match that of function, e.g. catalysis. Although experimental demonstration of concerted atomic motions is extremely difficult, methods are being developed to address this extremely important mechanistic question [52-55]. In fact, the differentiation of functional motions from random thermal motion and non-productive conformational changes is the central challenge the field of protein dynamics.

Given the functional importance of conformational transitions, it is evident that protein dynamics are under evolutionary selection. In the case of systems such as enzymes and motor proteins, the necessity of conformational change is obvious, but there is also evidence for evolutionary selected dynamics in less obvious systems. For instance, conformational plasticity at binding surfaces allows proteins to bind multiple targets [56]. In fact, this form of dynamics may promote evolvability by allowing some degree of functional promiscuity in order to retain an original function while developing a new one [57]. Allosteric communication between subunits or within proteins is also a process that is dependent on dynamics [58]. In fact, the nearly ubiquitous presence of dynamic motions and conformational heterogeneity gives all proteins the potential for allosteric function [59]. Evolution can then select for small perturbations that bias the protein toward functionally useful conformations.

The field of protein dynamics has matured to the point where atomic motions are readily measurable and show strong correlations with protein function. The question remains, however, as to how proteins harness structural heterogeneity into useful and productive motions. What aspect of the protein's architecture governs these motions? Is it that stronger interactions between certain residues in one part of the structure relative to weaker

interactions to other residues allow motion with respect to the rest of the protein? Are there more subtle mechanisms such as rearrangements of hydrogen bonding networks? Is there something special about the architecture of natural proteins that promotes or controls dynamics that might not be present in *de novo* designs [60]? Many of these questions are only beginning to be addressed and will require advances in experimental techniques (such as measuring correlated motions) to provide mechanistic clues along with significant advances in computer simulations to provide atomic detail and breadth of study across many proteins. As high quality dynamic data becomes available, it will be very important that our general architecture of proteins appropriately discriminates functional motions from random thermal fluctuations.

Cooperative interaction of a subset of residues creates allosteric behavior

Any attempt to describe a general architecture of proteins must address the concept of allostery. Allostery allows proteins to respond to signals from other proteins or their environment and enables construction of sophisticated signaling networks through feedback regulation. This phenomenon was originally described as the positive cooperativity of binding a small molecule at distinct sites on separate subunits of a symmetric multimeric protein. Gradually, this definition has expanded to include cooperativity between asymmetric oligomers, negative cooperativity, and communication between different sites in monomers. Even in the past decade, significant evidence (mostly from NMR experiments) has revealed that allosteric communication need not occur through structural reorganization alone, but may involve, in part or solely, changes in the dynamic character of the protein [61-64].

The newfound wealth of experimental knowledge concerning protein dynamics and the ability to measure their entropic contribution has led to a rather extreme view wherein 1) all dynamic proteins are "allosteric;" 2) allosteric effectors can include small molecules, other proteins, DNA/RNA, covalent modifications, changes in the environment, or even mutations; and 3) allosteric communication always occurs by population shifts along pre-existing ensembles of pathways [65]. This all-encompassing view of allostery has only recently been espoused and primarily by a single group [65, 66]. Others, however, actively argue for a more

rigid, function-centric definition of energetic coupling between binding events [67]. In my opinion, the definition wherein any perturbation to the protein affects the entire protein and is thus allosteric loses any descriptive or predictive value of the concept. In addition, the concept that only pre-existing conformations and pathways are accessed by perturbations is difficult to justify and is refuted by experimental evidence [64]. While there are clear cases of pre-existing high-affinity conformations existing in the un-liganded state [46], consistent with classical models [68], it seems unlikely that the range of substates accessible to two different systems will always be completely overlapping. And at what population is a conformation considered to exist in the native state – as long as the energy of a state is less than infinity then it will populated according to the Boltzmann distribution – does that make it a pre-existing conformation? I think that in over-generalizing the concept of allostery, Nussinov & coworkers are really just pointing out that that energy landscape around the native state is less rough or less steep than we may have originally thought.

Even with the realization that allosteric signaling can occur through structural rearrangements or dynamic changes, some sites on the protein appear to be particularly well-suited for propagating an effect of binding to a functional site, and this communication often occurs along a particular pathway of residue interactions. Residue interactions that are important for allosteric communication can be inferred from the protein structures of different functional states [69-71], revealed by NMR chemical shift and dynamic analyses [64, 72], deduced from biochemical experiments [73], measured by double mutant cycle analysis [74], or predicted by statistical analyses of coevolution [1]. In addition, these allosteric residue pathways/networks can also exhibit high order cooperativity [75, 76]. Furthermore the principle of allosterically important residues can also be extended to the model of allostery through conformational selection if a network of cooperative residues is responsible for effecting conformational change following the binding of an allosteric effector. Thus, the existence of selective communication pathways/networks reinforces the idea that evolution has selected for the strengthening of certain residue interactions at the expense of others. This evidence that networks of cooperative interactions between a small subset of residues mediate

allostery makes a strong assertion that a heterogeneous protein architecture is essential for protein function.

The value of identifying potential allosteric sites and understanding residue coupling that propagates perturbations from these sites is immense. Allosteric sites are potential drug targets that may prove to be even more effective than traditional active sites because they often show more inter-species variability among homologs and offer the possibility of increased specificity and fewer side effects [77]. As such, the search for allosteric sites and novel allosteric activators and inhibitors is under active pursuit by both academic researchers and the pharmaceutical industry [77-79]. Two other practical applications of identifying allosteric sites and understanding allosteric communication are the *de novo* design of allosteric proteins and the synthetic combination of proteins to create novel signaling networks or useful protein-based reagents. While the first of these two applications is still in its infancy, the latter is under active development. Successes via "best guesses" or multi-site screening have already occurred [80], but now the rational coupling of allosteric pathways is being tested [81]. The existence of heterogeneous protein architectures with embedded cooperative networks clearly suggests that allosteric drug screening, protein design, and engineering of allosteric coupling between and within proteins can be done more efficiently and effectively than the brute force screening of random combinations. Future experiments will bear out which techniques to identify and control allosteric sites are most effective.

Large-scale mutagenesis reveals that most positions tolerate significant amino acid variation

We have already noted variation in both local and global (average) packing densities, but what do we know about packing specificity? If highly optimized packing were essential for protein structure, function, and dynamics, then mutations that disrupt carefully coordinated side-chain interactions would be expected to have deleterious effects. Several large-scale mutagenesis experiments contain a wealth of data that addresses this very idea of how robust proteins are to mutation. Data comes from diverse in-vivo model systems such as 3-methyl DNA gylocsylase [82], barnase [83], diacylglycerol kinase [84], HIV-1 protease [85], lac repressor [86], RNase A [87], subtilisin [88], T4 lysozyme [89, 90], and TEM1 beta-lactamase [91] as well

as genomic substitution analyses [92]. The collective results indicate that proteins can tolerate substitution at many sites without significantly compromising the structure or function of the molecule. However, the diversity (number of sites) and amount (number of mutations) of tolerance varies between proteins and varies according to the nature and stringency of the selection criteria. There are also plausible reasons to expect that some proteins can tolerate less sequence variation than others. For example, the more functional constraints on a protein, such as the number of binding sites on the surface of the molecule, the less tolerant the molecule should be to mutation [93].

While some of these large-scale mutagenesis experiments focus on determining which sites can accept single mutations, other studies have asked how many mutations a protein can accept before losing structure or function. Theoretical and experimental studies have pointed to the fact that only a limited number of random mutations can occur before compensatory mutations are necessary to restore stability/function [94]. This is most easily shown regarding thermal stability where most mutations seem to be slightly destabilizing [95]. The simplest theories assume that, on average, stability will decline exponentially with the number of random mutations based on the assumption that substitution effects are additive [96]. A slight adjustment to this theory to include the baseline excess stability of the protein seems to account for the experimental data slightly better with the hypothesis being that most mutations are largely tolerated until some stability threshold is reached at which point further mutations are generally deleterious [91]. This hypothesis has significant implications for protein evolution because it implies that proteins have some window of stability within which they must evolve [94].

Many theories that arise from these mutational studies are based on abstracting ideas from the average effect of random mutations, but to understand the architectures of specific proteins, we need to understand the effect of specific mutations. Rather than ask how many sites can accept mutations, we need to ask which sites can accept mutations and are these sites cooperative or independent. Independent site studies are important because they reveal that some positions are highly substitutable while others are much less so – indicating functional or structural constraints on those less substitutable residues. However, these mutation-selection-

sequencing experiments hold the potential to reveal vastly greater amounts of information by also measuring cooperativity between residues. Now that extremely high throughput sequencing is available [97], one should be able to create large mutation libraries and sequence enough variants to observe interaction between all pairs of residues, or create focused libraries to observe higher-order cooperativity. These experiments are extremely exciting and will drive the next generation of hypotheses regarding the energetic coupling of residues and protein evolution[98].

In addition to experimental methods, significant efforts have gone into developing computational methods to predict the structural consequence and thermodynamic stability changes associated with mutation [99] . In general, the results have been disappointing. Algorithms that predict thermal stability changes using various physical and knowledge-based parameters are generally able to match the envelope of effects seen in the entire training set, but the accuracy of individual predictions is low [100, 101]. This low accuracy also precludes the use of such algorithms to identify physical mechanisms mediating the effects.

To summarize, a broad range of mutational experiments reveal that while a protein is still close to its natural sequence, many sites are relatively tolerant to mutations, while a smaller number of sites are very sensitive. However, while many sites are relatively tolerant to accepting mutations and remaining functional, the vast majority of even accepted mutations lower thermal stability. This pattern of mutational affects puts some real constraints on any theory for a general architecture of proteins. Some positions require specific amino acids to maintain proper function while most positions tolerate variation, but are optimized for favorable energetic interactions in their local environment. The latter condition also implies that thermal stability is distributed amongst favorable interactions throughout the protein. Another useful way of stating the same idea would be to say that while there may be few combinations of amino acids at particular positions that can create complex functional behavior in a protein, there are many ways for a protein to establish enough thermal stability to operate effectively.

Proteins have evolved to be evolvable

In using the term "general architecture" of proteins, one naturally gravitates toward the idea of a physical/structural architecture, but the term can also apply to a conceptual framework. I have already reviewed how amino acid interactions contribute to the biophysical properties of proteins – structure, packing density, dynamics, tolerance to mutations, and allostery/cooperativity – properties that apply to single protein sequences. Because of the massive sequence divergence in extant species and the diversity of protein function and specificity, we deem proteins to be evolvable, having the inherent capability to acquire new functions. While at the top level evolution is an organismal property – the species must be able to generate phenotypic diversity to adapt to changes in the environment, threats from predators, competition for energy resources, etc. – it is ultimately a property of protein function and the regulation of protein expression. For proteins, evolvability can only be studied in the context of many related protein sequences, as it is a property we can infer from the ensemble but not the individual. This ensemble property may, however, impose constraints on the general architecture of proteins that are just as significant as the biophysical properties of the individuals.

Protein evolvability can be decomposed into two processes: adaptability and robustness. Adaptability means that proteins must be able to acquire new properties quickly enough by random mutation such that the new, advantageous property can arise before deleterious substitutions render the allele lethal by loss of function or other toxicity. This process is inextricably linked to the concept of robustness; the protein must "tolerate" some amount of sequence divergence, otherwise random mutagenesis would be rapidly lethal. In the literature, these concepts are mainly discussed inside the framework of adaptive evolution (most mutations are selected for a fitness advantage or to compensate for a previous deleterious mutation) versus the neutral theory of evolution wherein most mutations are selectively neutral. While both neutral and adaptive mutations certainly occur, recent reviews of the literature argue that evidence points to adaptive evolution as the predominant mechanism [102, 103]. I will, however, avoid discussing these developed theories in favor of exploring the properties of proteins that allow for adaptability and robustness.

Because new proteins almost always arise from existing proteins, the ability to quickly acquire a new function rests upon a protein either already possessing the new function (although potentially at a low level) or being able to acquire a new function quickly (relatively few mutations). There is significant evidence, especially in enzymes, that proteins often possess promiscuous activities in addition to their native activity and that these promiscuous activities may act as starting points for further positive selection [104]. Similarly, binding proteins may recognize multiple targets (multispecificity) via accessing multiple conformations [104]. These existing low-level activities can then be enhanced by positive selection while maintaining the original activity or gene duplication can occur and allow the separated genes to be optimized for different functions. Thus, promiscuous activities and multispecificity act to speed up the rate of protein evolution by providing a starting point that is already close to the desired activity. In contrast, the fitness density (number of functional constraints) of a protein is expected to be negatively correlated with evolutionary rate; however this relationship appears to be weak or non-existent in several organisms that have been investigated [105].

Robustness to mutation promotes evolvability because it allows the protein to sample sequence variations without losing total function. Although the exact number depends on the protein, the environmental context, and the selection pressure, approximately 1/3 of random mutations are found to be deleterious [94, 102]. This observation indicates that only a subset of residues are essential for function while many other residues can vary significantly in the context of the wild-type protein. This architecture wherein some residues are essential while many are variable appears to be an evolutionarily selected feature. For instance protein families with higher contact densities have higher sequence entropy and are considered to be more designable (more sequences are compatible with that protein's structure and function – i.e. more robust) and display a faster evolutionary rate [106, 107]. Being more designable (and more robust) presumably allows for a greater number of evolutionary paths of sequence variation, making it easier to access regions of sequence space that code for new functions.

Our lab has therefore proposed that a protein architecture with a smaller number of cooperative residues and a larger number of less functionally important residues underlies the adaptability and robustness of natural proteins. Having a small number of cooperative residues

makes the protein more adaptable because a smaller number of mutations can give rise to new complex functions while the larger number of less functional residues promotes robustness. There is one important detail, however. Almost all mutations in a protein tend to be destabilizing, and once a protein accumulates enough destabilizing mutations, it can cease to function or become toxic to the protein via aggregation. Therefore, it appears that nearly all residues in the protein are under weak evolutionary pressure to promote favorable local interactions to provide stability while a subset of residues are under strong evolutionary pressure to provide function for the molecule. Until now, it has been impossible to convincingly test such a hypothesis. With the onset of high-throughput sequencing, however, we now appear to be in a position perform forward evolution experiments with sufficient sequencing statistics to test our proposal. It will also be exciting to test computationally designed proteins in such evolutionary experiments to determine whether different design principles affect the evolutionary potential of proteins.

Conclusions

Thus far, I have reviewed many known properties of proteins – well-ordered tertiary structures with good (but not perfect) atomic packing, functionally important motions, cooperative interactions leading to allosteric behavior, robustness to mutation, and evolvability to acquire new functions. Any general description of proteins must account for all these properties, and any theory to explain how proteins developed or evolved these properties must argue that it is evolutionarily more efficient and robust than competing design principles.

For the past decade, our lab has sought to describe a general architecture of proteins that moves away from the structure-centric view that all atoms and all interactions are important. Largely based on the results of SCA, we have proposed that the functional architecture of proteins is significantly heterogeneous with a small number of strong, cooperative interactions and a large number of weak interactions. In many ways, this is a very successful conceptual construct. It accounts for the heterogeneity of the functional effects of mutations [108]; it identifies functionally important residues [1-6]; and it provides a testable hypothesis to explain the evolvability of proteins. In fact, SCA information was sufficient to

design small proteins (with significant sequence variation from any known natural proteins) that assumed the correct native fold and functioned similar to natural homologs [109, 110].

SCA on its own, however, cannot describe all aspects of the general architecture of proteins. At a high level, it has been difficult to extend SCA-based protein design to larger proteins, and at a low-level, SCA does not predict or explain why most mutations tend to lower thermodynamic stability. In addition, it is not yet clear how statistical coupling relates to protein dynamics. However, there is no reason to expect SCA to reveal all properties of individual proteins; only properties that are conserved in the protein family and that are due to coevolution of similar residues will give rise to statistical coupling. Also, statistical coupling is a function of all evolutionary constraints on proteins, and although cases exist where SCA sectors appear to be devoted to a particular property [6], it is not always possible to define the evolutionary pressure or the mechanism that gives rise to specific couplings.

Up to this point, we have been able to demonstrate the functional importance of SCA sectors, but not the mechanism by which residues are coupled. Because SCA sectors are generally comprised of physically contiguous residues, we make the natural assumption that physical interactions are responsible for the coevolution and cooperative behavior of these residues. Unfortunately, this intuitive hypothesis has been very difficult to test primarily due to the fact that methods to measure the qualitative or quantitative interactions between residues have not existed in a sufficiently high-throughput format. This project aims to address this need by developing a method (termed a global perturbation analysis) to provide a high-throughput and high spatial resolution description of physical interactions between residues in a protein. The resulting data will be first be used to verify the general concept that proteins have a heterogeneous architecture of strong and weak inter-residue interactions. In addition, this global perturbation analysis will be applied to a protein from the PDZ domain family to demonstrate that physical interactions in individual proteins underlie statistical coupling in the protein family.

References

1. Lockless, S.W. and R. Ranganathan, *Evolutionarily conserved pathways of energetic connectivity in protein families.* Science, 1999. **286**(5438): p. 295-9.
2. Suel, G.M., et al., *Evolutionarily conserved networks of residues mediate allosteric communication in proteins.* Nat Struct Biol, 2003. **10**(1): p. 59-69.
3. Ferguson, A.D., et al., *Signal transduction pathway of TonB-dependent transporters.* Proc Natl Acad Sci U S A, 2007. **104**(2): p. 513-8.
4. Shulman, A.I., et al., *Structural determinants of allosteric ligand activation in RXR heterodimers.* Cell, 2004. **116**(3): p. 417-429.
5. Hatley, M.E., et al., *Allosteric determinants in guanine nucleotide-binding proteins.* Proc Natl Acad Sci U S A, 2003. **100**(24): p. 14445-50.
6. Halabi, N., et al., *Protein sectors: evolutionary units of three-dimensional structure.* Cell, 2009. **138**(4): p. 774-86.
7. Fink, A.L., *Natively unfolded proteins.* Curr Opin Struct Biol, 2005. **15**(1): p. 35-41.
8. Dyson, H.J. and P.E. Wright, *Intrinsically unstructured proteins and their functions.* Nat Rev Mol Cell Biol, 2005. **6**(3): p. 197-208.
9. Fuxreiter, M., et al., *Malleable machines take shape in eukaryotic transcriptional regulation.* Nat Chem Biol, 2008. **4**(12): p. 728-37.
10. Mittag, T., L.E. Kay, and J.D. Forman-Kay, *Protein dynamics and conformational disorder in molecular recognition.* J Mol Recognit, 2010. **23**(2): p. 105-16.
11. Wall, M.A., et al., *The structure of the G protein heterotrimer Gi alpha 1 beta 1 gamma 2.* Cell, 1995. **83**(6): p. 1047-58.
12. Yang, L., et al., *Close Correspondence between the Motions from Principal Component Analysis of Multiple HIV-1 Protease Structures and Elastic Network Modes.* Structure, 2008. **16**(2): p. 321-330.
13. Schaeffer, R.D., A. Fersht, and V. Daggett, *Combining experiment and simulation in protein folding: closing the gap for small model systems.* Current Opinion in Structural Biology, 2008. **18**(1): p. 4-9.
14. Best, R.B., et al., *Relation between native ensembles and experimental structures of proteins.* Proceedings of the National Academy of Sciences of the United States of America, 2006. **103**(29): p. 10901-10906.
15. Lange, O.F., et al., *Recognition dynamics up to microseconds revealed from an RDC-derived ubiquitin ensemble in solution.* Science, 2008. **320**(5882): p. 1471-1475.
16. Zoete, V., O. Michielin, and M. Karplus, *Relation between sequence and structure of HIV-1 protease inhibitor complexes: A model system for the analysis of protein Flexibility.* Journal of Molecular Biology, 2002. **315**(1): p. 21-52.
17. DePristo, M.A., P.I.W. De Bakker, and T.L. Blundell, *Heterogeneity and inaccuracy in protein structures solved by X-ray crystallography.* Structure, 2004. **12**(5): p. 831-838.
18. Banerjee, R., et al., *The jigsaw puzzle model: search for conformational specificity in protein interiors.* J Mol Biol, 2003. **333**(1): p. 211-26.
19. Gerstein, M. and C. Chothia, *Packing at the protein-water interface.* Proc Natl Acad Sci U S A, 1996. **93**(19): p. 10167-72.
20. Harpaz, Y., M. Gerstein, and C. Chothia, *Volume changes on protein folding.* Structure, 1994. **2**(7): p. 641-9.
21. Finney, J.L., *Volume occupation, environment and accessibility in proteins. The problem of the protein surface.* J Mol Biol, 1975. **96**(4): p. 721-32.

22. Chothia, C., *Structural invariants in protein folding.* Nature, 1975. **254**(5498): p. 304-8.
23. Richards, F.M., *The interpretation of protein structures: total volume, group volume distributions and packing density.* J Mol Biol, 1974. **82**(1): p. 1-14.
24. Richards, F.M., *Areas, volumes, packing and protein structure.* Annu Rev Biophys Bioeng, 1977. **6**: p. 151-76.
25. Tsai, J., et al., *The packing density in proteins: standard radii and volumes.* J Mol Biol, 1999. **290**(1): p. 253-66.
26. Chen, J. and W.E. Stites, *Packing is a key selection factor in the evolution of protein hydrophobic cores.* Biochemistry, 2001. **40**(50): p. 15280-9.
27. Dahiyat, B.I. and S.L. Mayo, *Probing the role of packing specificity in protein design.* Proc Natl Acad Sci U S A, 1997. **94**(19): p. 10172-7.
28. Godoy-Ruiz, R., et al., *A stability pattern of protein hydrophobic mutations that reflects evolutionary structural optimization.* Biophys J, 2005. **89**(5): p. 3320-31.
29. Kellis, J.T., Jr., K. Nyberg, and A.R. Fersht, *Energetics of complementary side-chain packing in a protein hydrophobic core.* Biochemistry, 1989. **28**(11): p. 4914-22.
30. Desjarlais, J.R. and T.M. Handel, *De novo design of the hydrophobic cores of proteins.* Protein Sci, 1995. **4**(10): p. 2006-18.
31. Sheffler, W. and D. Baker, *RosettaHoles: rapid assessment of protein core packing for structure prediction, refinement, design, and validation.* Protein Sci, 2009. **18**(1): p. 229-39.
32. Kuhlman, B., et al., *Design of a novel globular protein fold with atomic-level accuracy.* Science, 2003. **302**(5649): p. 1364-8.
33. Liang, J. and K.A. Dill, *Are proteins well-packed?* Biophys J, 2001. **81**(2): p. 751-66.
34. Halle, B., *Flexibility and packing in proteins.* Proceedings of the National Academy of Sciences of the United States of America, 2002. **99**(3): p. 1274-1279.
35. Seeliger, D. and B.L. de Groot, *Atomic contacts in protein structures. A detailed analysis of atomic radii, packing, and overlaps.* Proteins, 2007. **68**(3): p. 595-601.
36. Banerji, A. and I. Ghosh, *A new computational model to study mass inhomogeneity and hydrophobicity inhomogeneity in proteins.* European Biophysics Journal, 2009. **38**(5): p. 577-587.
37. Berezovsky, I.N. and E.I. Shakhnovich, *Physics and evolution of thermophilic adaptation.* Proc Natl Acad Sci U S A, 2005. **102**(36): p. 12742-7.
38. Zeldovich, K.B., I.N. Berezovsky, and E.I. Shakhnovich, *Protein and DNA sequence determinants of thermophilic adaptation.* PLoS Comput Biol, 2007. **3**(1): p. e5.
39. Suhre, K. and J.M. Claverie, *Genomic correlates of hyperthermostability, an update.* J Biol Chem, 2003. **278**(19): p. 17198-202.
40. Cambillau, C. and J.M. Claverie, *Structural and genomic correlates of hyperthermostability.* J Biol Chem, 2000. **275**(42): p. 32383-6.
41. Fraser, J.S., et al., *Hidden alternative structures of proline isomerase essential for catalysis.* Nature, 2009. **462**(7273): p. 669-73.
42. Parak, F.G., *Proteins in action: the physics of structural fluctuations and conformational changes.* Curr Opin Struct Biol, 2003. **13**(5): p. 552-7.
43. Jardetzky, O., *Protein dynamics and conformational transitions in allosteric proteins.* Prog Biophys Mol Biol, 1996. **65**(3): p. 171-219.
44. Rasmussen, B.F., et al., *Crystalline ribonuclease A loses function below the dynamical transition at 220 K.* Nature, 1992. **357**(6377): p. 423-4.
45. Mulder, F.A., et al., *Studying excited states of proteins by NMR spectroscopy.* Nat Struct Biol, 2001. **8**(11): p. 932-5.
46. Volkman, B.F., et al., *Two-state allosteric behavior in a single-domain signaling protein.* Science, 2001. **291**(5512): p. 2429-33.

47. Allemann, R.K., R.M. Evans, and E.J. Loveridge, *Probing coupled motions in enzymatic hydrogen tunnelling reactions.* Biochem Soc Trans, 2009. **37**(Pt 2): p. 349-53.
48. Eisenmesser, E.Z., et al., *Enzyme dynamics during catalysis.* Science, 2002. **295**(5559): p. 1520-3.
49. Eisenmesser, E.Z., et al., *Intrinsic dynamics of an enzyme underlies catalysis.* Nature, 2005. **438**(7064): p. 117-121.
50. Wang, C., et al., *Dynamics of ATP-binding cassette contribute to allosteric control, nucleotide binding and energy transduction in ABC transporters.* J Mol Biol, 2004. **342**(2): p. 525-37.
51. Wolf-Watz, M., et al., *Linkage between dynamics and catalysis in a thermophilic-mesophilic enzyme pair.* Nat Struct Mol Biol, 2004. **11**(10): p. 945-9.
52. Watt, E.D., et al., *The mechanism of rate-limiting motions in enzyme function.* Proc Natl Acad Sci U S A, 2007. **104**(29): p. 11981-6.
53. Vogeli, B. and R. Riek, *Side chain: backbone projections in aromatic and ASX residues from NMR cross-correlated relaxation.* J Biomol NMR, 2010. **46**(2): p. 135-47.
54. Vogeli, B. and L. Yao, *Correlated dynamics between protein HN and HC bonds observed by NMR cross relaxation.* J Am Chem Soc, 2009. **131**(10): p. 3668-78.
55. Bouvignies, G., et al., *Identification of slow correlated motions in proteins using residual dipolar and hydrogen-bond scalar couplings.* Proc Natl Acad Sci U S A, 2005. **102**(39): p. 13885-90.
56. Ascenzi, P. and M. Fasano, *Allostery in a monomeric protein: the case of human serum albumin.* Biophys Chem, 2010. **148**(1-3): p. 16-22.
57. Tokuriki, N. and D.S. Tawfik, *Protein dynamism and evolvability.* Science, 2009. **324**(5924): p. 203-7.
58. Kern, D. and E.R. Zuiderweg, *The role of dynamics in allosteric regulation.* Curr Opin Struct Biol, 2003. **13**(6): p. 748-57.
59. Gunasekaran, K., B. Ma, and R. Nussinov, *Is allostery an intrinsic property of all dynamic proteins?* Proteins: Structure, Function and Genetics, 2004. **57**(3): p. 433-443.
60. Walsh, S.T., et al., *Dynamics of a de novo designed three-helix bundle protein studied by 15N, 13C, and 2H NMR relaxation methods.* Biochemistry, 2001. **40**(32): p. 9560-9.
61. Tsai, C.J., A. del Sol, and R. Nussinov, *Allostery: absence of a change in shape does not imply that allostery is not at play.* J Mol Biol, 2008. **378**(1): p. 1-11.
62. Cooper, A. and D.T. Dryden, *Allostery without conformational change. A plausible model.* Eur Biophys J, 1984. **11**(2): p. 103-9.
63. Popovych, N., et al., *Dynamically driven protein allostery.* Nat Struct Mol Biol, 2006. **13**(9): p. 831-8.
64. Bruschweiler, S., et al., *Direct observation of the dynamic process underlying allosteric signal transmission.* J Am Chem Soc, 2009. **131**(8): p. 3063-8.
65. del Sol, A., et al., *The origin of allosteric functional modulation: multiple pre-existing pathways.* Structure, 2009. **17**(8): p. 1042-50.
66. Tsai, C.J., A. Del Sol, and R. Nussinov, *Protein allostery, signal transmission and dynamics: a classification scheme of allosteric mechanisms.* Mol Biosyst, 2009. **5**(3): p. 207-16.
67. Fenton, A.W., *Allostery: an illustrated definition for the 'second secret of life'.* Trends Biochem Sci, 2008. **33**(9): p. 420-5.
68. Monod, J., J. Wyman, and J.P. Changeux, *On the Nature of Allosteric Transitions: A Plausible Model.* J Mol Biol, 1965. **12**: p. 88-118.
69. Gandhi, P.S., et al., *Structural identification of the pathway of long-range communication in an allosteric enzyme.* Proc Natl Acad Sci U S A, 2008. **105**(6): p. 1832-7.
70. Lee, M., et al., *Dihydroorotase from Escherichia coli: loop movement and cooperativity between subunits.* J Mol Biol, 2005. **348**(3): p. 523-33.
71. Datta, D., et al., *An allosteric circuit in caspase-1.* J Mol Biol, 2008. **381**(5): p. 1157-67.

72. Masterson, L.R., et al., *Allosteric cooperativity in protein kinase A.* Proc Natl Acad Sci U S A, 2008. **105**(2): p. 506-11.
73. Rakauskaite, R. and J.D. Dinman, *rRNA mutants in the yeast peptidyltransferase center reveal allosteric information networks and mechanisms of drug resistance.* Nucleic Acids Res, 2008. **36**(5): p. 1497-507.
74. Piper, D.R., et al., *Cooperative interactions between R531 and acidic residues in the voltage sensing module of hERG1 channels.* Cell Physiol Biochem, 2008. **21**(1-3): p. 37-46.
75. Ohtaka, H., A. Schon, and E. Freire, *Multidrug resistance to HIV-1 protease inhibition requires cooperative coupling between distal mutations.* Biochemistry, 2003. **42**(46): p. 13659-66.
76. Sadovsky, E. and O. Yifrach, *Principles underlying energetic coupling along an allosteric communication trajectory of a voltage-activated K+ channel.* Proceedings of the National Academy of Sciences of the United States of America, 2007. **104**(50): p. 19813-19818.
77. Shen, A., *Allosteric regulation of protease activity by small molecules.* Mol Biosyst, 2010. **6**(8): p. 1431-43.
78. Hardy, J.A., et al., *Discovery of an allosteric site in the caspases.* Proceedings of the National Academy of Sciences of the United States of America, 2004. **101**(34): p. 12461-12466.
79. Hardy, J.A. and J.A. Wells, *Searching for new allosteric sites in enzymes.* Current Opinion in Structural Biology, 2004. **14**(6): p. 706-715.
80. Ostermeier, M., *Engineering allosteric protein switches by domain insertion.* Protein Eng Des Sel, 2005. **18**(8): p. 359-64.
81. Lee, J., et al., *Surface sites for engineering allosteric control in proteins.* Science, 2008. **322**(5900): p. 438-42.
82. Guo, H.H., J. Choe, and L.A. Loeb, *Protein tolerance to random amino acid change.* Proc Natl Acad Sci U S A, 2004. **101**(25): p. 9205-10.
83. Axe, D.D., N.W. Foster, and A.R. Fersht, *A search for single substitutions that eliminate enzymatic function in a bacterial ribonuclease.* Biochemistry, 1998. **37**(20): p. 7157-66.
84. Wen, J., X. Chen, and J.U. Bowie, *Exploring the allowed sequence space of a membrane protein.* Nat Struct Biol, 1996. **3**(2): p. 141-8.
85. Parera, M., et al., *HIV-1 protease catalytic efficiency effects caused by random single amino acid substitutions.* Mol Biol Evol, 2007. **24**(2): p. 382-7.
86. Markiewicz, P., et al., *GENETIC-STUDIES OF THE LAC REPRESSOR .14. ANALYSIS OF 4000 ALTERED ESCHERICHIA-COLI LAC REPRESSORS REVEALS ESSENTIAL AND NONESSENTIAL RESIDUES, AS WELL AS SPACERS WHICH DO NOT REQUIRE A SPECIFIC SEQUENCE.* Journal of Molecular Biology, 1994. **240**(5): p. 421-433.
87. Smith, B.D. and R.T. Raines, *Genetic Selection for Critical Residues in Ribonucleases.* Journal of Molecular Biology, 2006. **362**(3): p. 459-478.
88. Shafikhani, S., et al., *Generation of large libraries of random mutants in Bacillus subtilis by PCR-based plasmid multimerization.* Biotechniques, 1997. **23**(2): p. 304-10.
89. Poteete, A.R., D. Rennell, and S.E. Bouvier, *Functional significance of conserved amino acid residues.* Proteins, 1992. **13**(1): p. 38-40.
90. Rennell, D., et al., *Systematic mutation of bacteriophage T4 lysozyme.* J Mol Biol, 1991. **222**(1): p. 67-88.
91. Bershtein, S., et al., *Robustness-epistasis link shapes the fitness landscape of a randomly drifting protein.* Nature, 2006. **444**(7121): p. 929-32.
92. Eyre-Walker, A. and P.D. Keightley, *High genomic deleterious mutation rates in hominids.* Nature, 1999. **397**(6717): p. 344-7.
93. Zuckerkandl, E., *Evolutionary processes and evolutionary noise at the molecular level. I. Functional density in proteins.* J Mol Evol, 1976. **7**(3): p. 167-83.

94. Tokuriki, N. and D.S. Tawfik, *Stability effects of mutations and protein evolvability.* Curr Opin Struct Biol, 2009. **19**(5): p. 596-604.
95. Sanchez, I.E., et al., *Point Mutations in Protein Globular Domains: Contributions from Function, Stability and Misfolding.* Journal of Molecular Biology, 2006. **363**(2): p. 422-432.
96. Bloom, J.D., et al., *Thermodynamic prediction of protein neutrality.* Proc Natl Acad Sci U S A, 2005. **102**(3): p. 606-11.
97. Metzker, M.L., *Sequencing technologies - the next generation.* Nat Rev Genet, 2010. **11**(1): p. 31-46.
98. McLaughlin, R.N., Jr., et al., *The spatial architecture of protein function and adaptation.* Nature, 2012. **491**(7422): p. 138-42.
99. Guerois, R., J.E. Nielsen, and L. Serrano, *Predicting changes in the stability of proteins and protein complexes: A study of more than 1000 mutations.* Journal of Molecular Biology, 2002. **320**(2): p. 369-387.
100. Khan, S. and M. Vihinen, *Performance of Protein Stability Predictors.* Human Mutation, 2010. **31**(6): p. 675-684.
101. Potapov, V., M. Cohen, and G. Schreiber, *Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details.* Protein Eng Des Sel, 2009. **22**(9): p. 553-60.
102. Camps, M., et al., *Genetic constraints on protein evolution.* Crit Rev Biochem Mol Biol, 2007. **42**(5): p. 313-26.
103. DePristo, M.A., D.M. Weinreich, and D.L. Hartl, *Missense meanderings in sequence space: a biophysical view of protein evolution.* Nat Rev Genet, 2005. **6**(9): p. 678-87.
104. Peisajovich, S.G. and D.S. Tawfik, *Protein engineers turned evolutionists.* Nat Methods, 2007. **4**(12): p. 991-4.
105. Pal, C., B. Papp, and M.J. Lercher, *An integrated view of protein evolution.* Nat Rev Genet, 2006. **7**(5): p. 337-48.
106. Zeldovich, K.B. and E.I. Shakhnovich, *Understanding protein evolution: from protein physics to Darwinian selection.* Annu Rev Phys Chem, 2008. **59**: p. 105-27.
107. Zhou, T., D.A. Drummond, and C.O. Wilke, *Contact density affects protein evolutionary rate from bacteria to animals.* J Mol Evol, 2008. **66**(4): p. 395-404.
108. Clackson, T. and J.A. Wells, *A hot spot of binding energy in a hormone-receptor interface.* Science, 1995. **267**(5196): p. 383-6.
109. Socolich, M., et al., *Evolutionary information for specifying a protein fold.* Nature, 2005. **437**(7058): p. 512-518.
110. Russ, W.P., et al., *Natural-like function in artificial WW domains.* Nature, 2005. **437**(7058): p. 579-83.

CHAPTER 2: A global chemical shift perturbation assay

Since the fundamental unit of a protein is the amino acid residue, understanding the architecture of a protein rests on understanding the interactions of these residues. Unfortunately, we cannot directly measure the physical forces between residues, so we must use surrogate quantities as readouts of these interactions. This is usually accomplished by making some sort of perturbation (such as a mutation) at one residue and then observing the resulting change in a biophysical or functional property of the protein. As described below, there are currently many methods for measuring residue interactions, each having their own strengths and weaknesses. However, none of the existing methods are suitable for measuring the physical interactions between all pairs of residues in a protein. In this chapter, I describe the development of an NMR-based assay that maps physical interactions between all pairs of residues in a protein. Although still labor-intensive, it is sufficiently high-throughput to provide a global assessment of the physical interactions between all residues in a protein.

Alternative Methods

Currently, there are several techniques to measure interactions between residues in a protein and many rely on mutagenesis to change the amino acid at a particular site (a perturbation to the system) and then measure some property of this variant protein. The main factors to consider are combinatorial complexity, speed of the assay, and the readout or information gathered from the assay. To illustrate combinatorial complexity, consider a 100 amino acid protein. Creating a single mutant at every position (such as an alanine scan) would require 100 variants, while a library of every possible single mutation would necessitate 19*100 = 1,900 variants. If one wanted to measure pairwise interactions of mutations using double mutant cycle analysis, then about 5,000 variants are needed to create all double mutants (when mutating to a single amino acid, such as an alanine scan) while almost two million variants are required to obtain all combinations of double mutants mutated to all possible amino acids. Higher order (third, fourth, etc.) mutation schemes require even greater numbers of mutants.

	Single Amino Acid	All amino acids
Single Mutants	100	1900
Double Mutants	4950	~1.8 million
Triple Mutants	161,700	~1.1 billion

Table 2-1 Combinatorial Complexity of Mutations in a 100 Residue Protein

Thus, combinatorial complexity quickly climbs out of the realm of conventional biophysical experiments involving protein expression and purification which generally have the capacity to handle tens to hundreds of variants. Functional selection systems, such as bacterial growth and selection, can handle libraries in the thousands to millions, but once one goes above second order combinations, the combinatorial complexity quickly outgrows reasonable library sizes. I will now review several existing methods to interrogate residue interactions and discuss their advantages, drawbacks, and limitations on the complexity they can handle due to restrictions inherent to each assay as well as time and monetary considerations.

Thermodynamic mutant cycle analysis can measure the interaction energies between a pair of amino acids by comparing the wild-type protein, two single mutants, and the corresponding double mutant. This formalism is very powerful because it measures actual energetic quantities, however, the energy is only related to the specific assay – thermodynamic stability, ligand affinity, catalytic power, etc. One drawback to this approach is that you do not gain any information about the mechanism of interaction. Additionally, mutant cycle analysis is usually restricted to focused studies because the assay generally requires purified protein which is labor intensive and time consuming to obtain. Another method to observe residue interactions is to solve the structure of single mutants and observe how the effects of the mutation are propagated to other residues [1]. This can be a powerful method because it can reveal the detailed structural changes, but it is slow, is subject to crystal contact artifacts, is potentially not as sensitive as other methods, and does not provide direct information about the energetic value of structural changes. Computer simulations can also be performed to calculate interaction energies between residues. For instance, equilibrium molecular dynamics simulations can calculate average interaction energies over the course of the simulation

trajectory, and molecular mechanics force fields can be used to computationally predict the effect of mutations [2]. More recently, several non-equilibrium dynamics techniques have emerged explicitly for the purpose of identifying allosteric signaling pathways and long range residue coupling [3-8]. These computational methods (especially the newer non-equilibrium methods) are very appealing because they generate information with atomic resolution, have the ability to identify mechanisms of residue coupling, and can be run in parallel for many mutations or proteins. While molecular simulations probably represent the future of mechanistic studies of protein function, the accuracy of these simulations is currently difficult to determine and there are no means to validate the findings of such a study without performing experiments.

Finally, recent developments in high throughput sequencing offer the ability to create libraries of single and double mutants, screen by some functional or biophysical property, and then sequence the input and selected libraries to determine the effects of single, double, and higher order mutants (see recent studies using high throughput sequencing [9, 10]). In effect, this is a high throughput implementation of the thermodynamic mutant cycles discussed above. While this is clearly the most exciting type of current and future experiment to interrogate residue interactions, assays compatible with high throughput screening and selection must be developed, and it is still an energetic measurement that foregoes any information about the mechanism of interaction.

Assaying residue interactions with chemical shifts

Since none of the above experiments met our requirements of measuring all pair-wise residue interactions in a reasonable amount of time while also providing some information about the mechanisms of interaction, we turned to NMR-based methods. NMR chemical shifts are very sensitive indicators of the chemical environment of NMR-active nuclei and are extremely useful for reporting small structural or chemical changes in proteins. By recording the chemical shifts of a wild-type protein and comparing those to the chemical shifts of a mutant protein, one can determine which nuclei in the protein experience a change in chemical environment due to a mutation. Thus, in a single experiment, one can monitor all residues in a

protein and determine which ones are coupled (using chemical shifts as a reporter) to a mutation. By repeating this experiment for a single mutation at all sites in a protein, one can create a matrix of all pair-wise interactions between the residues in a protein.

This approach has distinct advantages and limitations. The advantages are that chemical shifts are sensitive to many mechanisms of perturbation and to very small quantities of change. Changes to secondary structure, local structure conformation, electric fields from polar groups or formal charges, hydrogen bond strength, and aromatic ring orientation all influence the chemical shift of a nucleus [11, 12]. In addition, chemical shifts can be obtained in a reasonable amount of time and are extremely precise and repeatable measurements. Chemical shifts do not, however, provide any information about the energetic value of a perturbation (mutation) as a whole or the energetic change at a particular residue or nucleus. Also, since they are a convolution of many structural or chemical features, it is difficult to determine the relative contributions of the factors that determine a chemical shift change. Given the advantages, drawbacks, and alternative experiments, this method of measuring chemical shift perturbation due to mutations offered the best option to obtain a sensitive, precise, and high-resolution mapping of all pair-wise interactions between residues in a protein.

PSD95 PDZ3 as a model system

Performing a global mutation-based perturbation analysis using NMR methods places many constraints on the choice of a protein to study. Firstly, I needed to choose a system that would conceptually test the idea that proteins have a heterogeneous architecture of some strong and many weak interactions with cooperative strong interactions mediating protein function. Second, our experimental design of collecting high quality NMR spectra on many protein variants requires that the model protein express well in *E. coli,* be easy to isolate at high purity, and be soluble and monomeric at high concentrations. Additionally, the protein must be relatively small to facilitate NMR spectrum acquisition (high signal/noise ratio and disperse resonances in the spectrum) and must have a function that can be assayed by NMR experiments.

The third PDZ domain of PSD95 of the Norway rat (hereafter referred to as PDZ3) is a capable model system to test a global perturbation analysis. PSD95 (also known as SAP90 or DLG4) is a member of the membrane-associated guanylate kinase (MAGUK) protein family and is the most abundant scaffold protein in the postsynaptic density of neurons. As shown in Figure 2-1, this protein contains 3 PDZ domains along with an SH3 and a guanylate kinase domain and has been found to bind to NMDA receptors, K+ channels, neuronal nitric oxide synthase, and the cysteine-rich PDZ-binding protein, CRIPT among others [13]. The third PDZ domain has been found to specifically interact with several proteins including citron [14], neuroligin [15], DHHC5 [16], and CRIPT [17], and is well characterized both structurally and functionally. The PDZ domain family is deep, diverse, and amenable to SCA. Experiments using small numbers of mutations have been consistent with ascribing functional importance to coevolving networks of residues in this domain, making this PDZ domain an ideal system to globally test whether evolutionary properties present in the PDZ domain family are reflected in the physical properties an individual PDZ domain. In addition, PDZ3 meets all of our experimental criteria by being highly expressed in *E. coli* (50-100 mg/L), effectively purified to 95% purity by one-step affinity tag purification, monomeric and soluble at concentrations up to several millimolar, and offering high quality NMR spectra with sharp and well-dispersed resonance peaks.

To simulate the function of PDZ3, a 9-mer peptide (TKNYKQTSV) corresponding to the C-terminus of CRIPT is introduced in solution. The peptide is acetylated on the N-terminus and has a free carboxylic acid group on the C-terminus. PDZ3 binds this CRIPT peptide with a K_D of 0.93 uM, and the X-ray structure of the complex is shown in panel B of Figure 2-1.

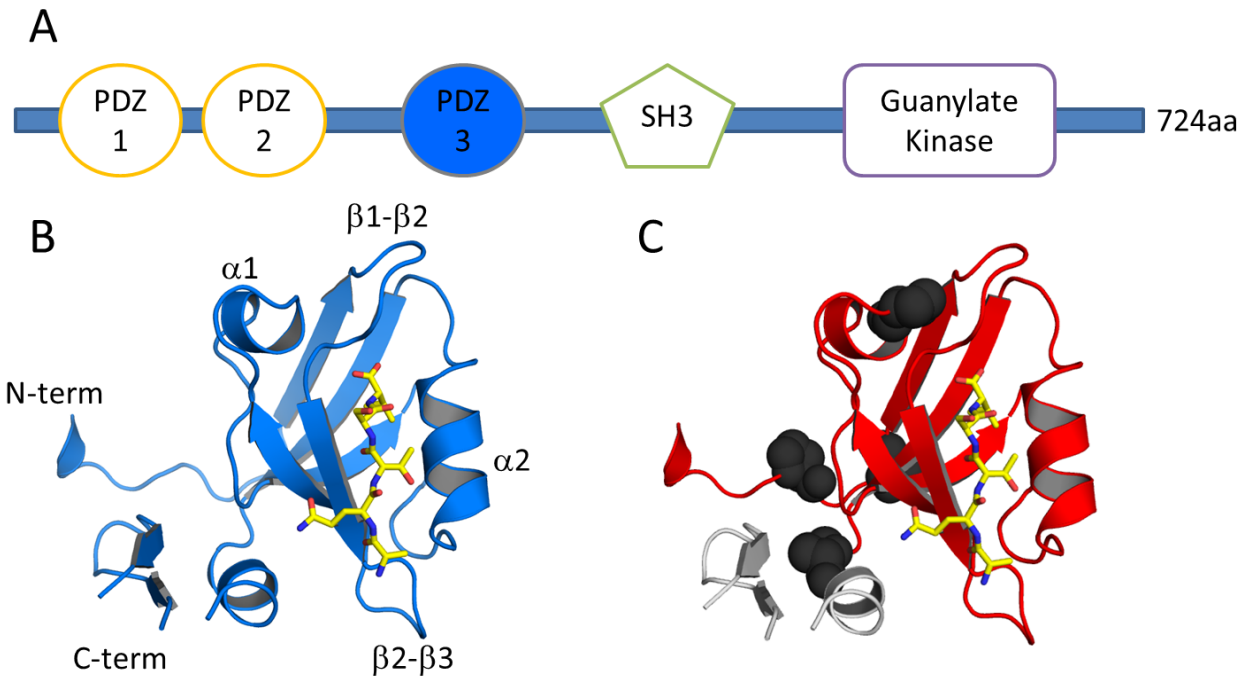


Figure 2-1: PSD95 from *Rattus norvegicus*.
A) PSD95 contains 3 N-terminal PDZ domains (circles), an SH3 domain (pentagon), and a C-terminal guanylate kinase domain (rectangle). PDZ3 is colored blue. B) The crystal structure of the third PDZ domain from PSD95 is shown (cartoon) in complex with a peptide (yellow sticks) derived from the C-terminus of CRIPT. C) Structure of PDZ3 with prolines shown in black spheres (prolines produce no resonance in an HNCO experiment) and residues 300-393 (chemical shifts used for analysis) colored red in cartoon. Images created from PDB 1BE9 [18].

PDZ domains are ~90 amino acid protein-protein interaction modules that typically recognize the C-termini of binding targets [19], but have also been shown, in a small number of cases, to bind internal sequence motifs [20, 21] and phospholipids [22]. In eukaryotes, PDZ domains are often found in multidomain scaffolding proteins where they are important for assembling specialized subcellular signaling complexes. In prokaryotes, PDZ domains adopt similar tertiary structure, but are circularly permuted with respect to their eukaryotic counterparts and are often partnered with various protease domains. While typically considered to be passive peptide binding domains, examples have been discovered where PDZ domains play a dynamic role in signal transduction [23] or are allosterically regulated by interactions away from the conserved peptide-binding interface [24].

Due to the abundance and importance of PDZ domains, much effort has been invested in studying the binding specificity of these proteins. While original studies of small numbers of

PDZ-peptide interactions suggested that a small number of specificity classes exist, more recent large scale ligand screens reveal that specificity profiles for individual domains are less distinct than previously thought, but are well-dispersed in selectivity space to minimize cross-reactivity within a single organism [25]. Another study using unbiased screening of random peptide libraries (via phage display) against many human and *Caenorhabditis elegans* PDZ domains revealed that specificity classes do exist and are evolutionarily conserved [26]. These large scale studies have been very useful for defining PDZ domain specificity, for understanding how organisms optimize large numbers of specific protein-protein interactions, and for identifying targets and strategies for therapeutic interventions [27, 28].

Implementation

This NMR-based global perturbation analysis is conducted by studying 81 evolutionarily conserved single mutation variants of the wild-type PDZ3. An HNCO spectrum is acquired for each mutant; peaks corresponding to all bonded amide-carbonyl groups are identified; and the chemical shift difference from wild-type is calculated for each HNC(O) group. Spectra are also acquired for the same mutants in the presence of saturating quantities of a target peptide ligand for a total of 162 unique spectra. The resulting chemical shift perturbation data is then examined for patterns of residue interactions and analyzed by principal component analysis (PCA) to identify mutations that show cooperative effects, manifested by similar patterns of perturbation.

Evolutionarily conserved mutations

One way to test a system is to make a small perturbation to its ground state and observe the response of the system. The ideal perturbation is one that is large enough to elicit a response, but small enough to keep the system near its native state. In a biological system, it is also preferable for the perturbation to be related to a natural perturbation of the system. In order to closely approximate an ideal perturbation, I chose to make single mutations to either the consensus amino acid from the multiple sequence alignment (MSA) of the PDZ domain family or, when an amino acid identity for PSD95 PDZ3 matched the consensus, the next most

common amino acid (NMCAA). In doing so, I attempted to make subtle perturbations to the native state of the PDZ domain that are likely to be observed during evolution and to be well-tolerated by the structure. The positional sequence conservation for a 240 sequence alignment of PDZ domains (alignment courtesy of Steve Lockless, see Appendix 1:) is shown in Figure 2-2 along with the single mutation constructs that were created. The PDZ3 construct used in this study contains 119 amino acids which includes an additional 14 N-terminal and 24 C-terminal residues that flank the conserved PDZ domain fold and are not part of the multiple sequence alignment. Since the purpose of this experiment was to compare the physical properties of a single protein to the statistical properties of the protein family, I chose to only mutate the 81 residues corresponding to the conserved structure and alignable sequence of the PDZ domain.

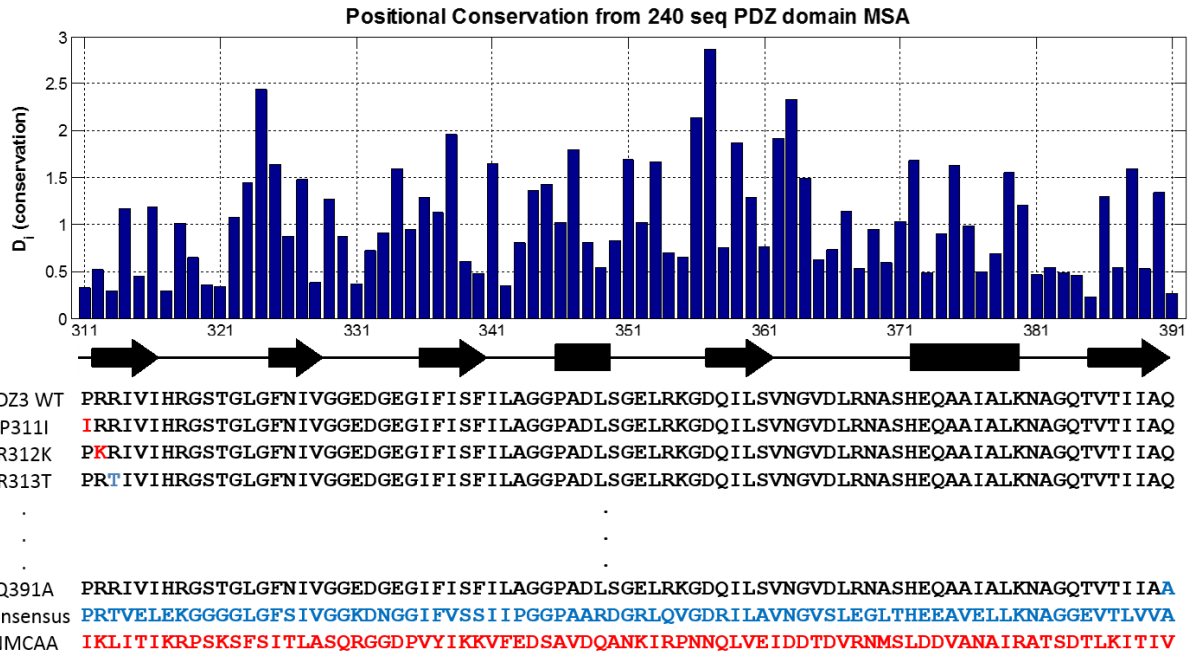


Figure 2-2: PDZ family sequence conservation and PDZ3 single mutants.
The sequence entropy (conservation) at each position of a diverse 240 sequence MSA of the PDZ family is shown. Positions have been truncated to those present in PSD95 PDZ3 and the sequence numbering corresponds to PDB 1BE9. Secondary structure is provided for reference. The wild-type PDZ3 sequence is shown for residues 311-391, and the most common (consensus), blue, and next most common amino acid (NMCAA), red, residues are shown for each position. The sequences of several single mutant constructs are provided for illustration.

An advantage of using mutagenesis is that the perturbation is localized to a particular site in the protein. The disadvantage, however, is that mutations at different positions in the protein are not likely to be energetically equivalent. In addition, my decision to use evolutionarily conservative mutations means that the mutations are not chemically equivalent. The more common approach of mutating every position to alanine does not, however, make the mutations any more consistent and may be more disruptive to the system than using evolutionarily conservative mutations.

Although data was collected for 81 mutants, two of these variants were excluded from analysis. D357N was excluded because it appeared to have a perturbation pattern by chemical shift perturbation that was more consistent with a global perturbation than a subtle perturbation near the native state. As it turns out, this mutant was 23^0C destabilized relative to WT when thermodynamic stability was measured by differential scanning calorimetry – the most of any mutant in our library. Therefore, we excluded it because it did not meet our criterion of a subtle perturbation. Another mutant, N381A was excluded because the protein formed a dimer in the presence of CRIPT peptide. This property was confirmed by NMR relaxation experiments showing the T2 values for this protein in the presence of peptide were significantly shorter (50ms versus 85ms) when compared to other ligand-bound PDZ domains. The remaining 79 mutations were all well-folded, stable, and gave excellent NMR spectra. In addition, all mutants retained good affinity for CRIPT peptide with K_D < 20 uM which allowed for saturated binding at reasonable protein and peptide concentrations. Thus, our mutation strategy appears to be effective in making subtle perturbations to PDZ3. A full list of all mutations included in the analysis is provided in Appendix 2:.

NMR Spectra

To obtain chemical shift measurements on ~160 proteins, I needed a very quick NMR experiment. Initially, I acquired ^{1}H-^{15}N HSQC spectra (~20-40 minutes per spectrum) on a test set of ~25 proteins. Although the spectral quality was excellent, these two-dimensional spectra contained significant spectral overlap, making it difficult or impossible to identify or distinguish some of the peaks. It also became apparent that given the degree of peak dispersion in a 2-D

spectrum, it would not be possible to accurately estimate the quantity of chemical shift changes without obtaining residue assignments for the peaks in every spectrum. Alternately stated, due to the density of the peaks, it was difficult to accurately guess which mutant peak corresponded to which wild-type peak. Obtaining residue assignments is a time consuming process that requires three-dimensional spectra with long acquisition times and significant hands-on processing. At the time this experiment began in 2006, obtaining chemical shift assignments on this many proteins was not a practical option. However, there have been recent developments in time-compressed data acquisition schemes, high-volume data handling, automated phasing, automated peak detection, and automated chemical shift assignment (mainly thanks to NMR-based structural genomics efforts) that would make high-volume chemical shift assignments easier today.

In order to gain better peak dispersion, I chose to use an HNCO NMR experiment. This experiment correlates chemical shifts from all bonded backbone amide nitrogen, amide proton, and carbonyl carbon nuclei. Using conventional, direct sampling in both indirect (nitrogen and carbon) dimensions, this experiment requires at least 12 hours to obtain sufficiently high resolution in the indirect dimensions. Due to the small size and good spectral behavior (sharp peaks) of PDZ3 and the sensitivity of the HNCO experiment, the signal/noise ratio was actually in vast excess of what was required. Therefore, I decided to use a new reduced-dimensionality data acquisition scheme for multi-dimensional NMR. Although several methods (shown to be mathematically analagous) had been described for expediting NMR data acquisition by reduced or joint sampling of the indirect dimensions [29], I chose the projection-reconstruction (PR) method [30] because it is conceptually intuitive, it requires only small modifications to the data acquisition practices, PR pulse sequences were already being incorporated into our default suite of experiments in the Varian VNMR and Biopack software suites, and processing algorithms were readily available and easy to use. Projection-reconstruction works best when there is a high signal-to-noise ratio and a low density of peaks. Since an HNCO spectrum only contains ($N_{residues} - N_{prolines} - 1$) resonance peaks, this was an ideal case. I was able to acquire a very high resolution PR-HNCO spectrum in less than 3 hours.

Basic spectral acquisition involved acquiring a small number (6-10) of equally spaced two dimensional (2D) projections that were processed using standard methods in the nmrPipe software package (see Appendix 2: for a sample processing script). Due to the time-consuming nature of the manual phasing of every single 2D projection (thousands were collected), I implemented an automatic phasing program in Matlab (see Appendix 4:) based on the principle of maximizing "white space" described in a paper by Balacco and Cobas [31]. Once properly Fourier-transformed and phased 2D spectra were obtained, 3D HNCO spectra were calculated using the projection-reconstruction technique as implemented by the PR-Calc program developed by Coggins and Zhou [32]. PR of NMR spectra is analogous to the computed tomography (CT) method of medical imaging, commonly called a "CAT scan," wherein a 3D image of internal anatomy is calculated from a series of 2D X-ray images taken from many angles. There are several options for projection-reconstruction algorithms, but I settled on the lower-value (LV) implementation because it is simple and has the least projection artifacts compared to other algorithms. The LV algorithm computes the intensity value of each voxel in 3D space by comparing the corresponding values in each 2D spectrum and keeping the lowest value. A hybrid backprojection/lower-value (HBLV) algorithm which keeps the lowest *k* (where $1 < k <$ number of projections) values for each voxel was also explored. The HBLV algorithm gave higher signal using $k = 2$ or 3, but with more shadowing artifacts in the center of the spectrum. In practice, the LV and HBLV (with $k = 2$ or 3) algorithms produced similar spectral quality, but the LV algorithm was chosen for simplicity. An example PR-Calc control file is given in Appendix 5:.

Peak detection in NMR Spectra

The first step in analyzing NMR spectra is to assign resonance peaks to their corresponding residues in the protein. Depending on the size and complexity of the protein, tis can be a time-consuming process. Peak assignments for WT PDZ3 in the free and peptide-bound states were made using standard techniques. In the WT spectrum, resonances are missing for residues 299, 308, 311, 346, and 394 because these residues are proline and therefore have no amide proton and are not detected by the HNCO experiment (locations

indicated in panel C of Figure 2-1). In addition, the WT free spectrum is missing resonances for residues 320, 321, and 323 in the $\beta1-\beta2$ loop, most likely due to chemical exchange associated with loop mobility, leaving 109 visible peaks. In the WT spectrum with peptide, 111 peaks are visible – residues 321 and 323 are now present, likely due to reduced mobility of the $\beta1-\beta2$ loop as corroborated by lower B-factors in the peptide-bound crystal structure.

From the mutant spectra, I developed a strategy to measure the chemical shift differences compared to WT without embarking on the lengthy process of making explicit peak assignments for each spectrum. The process starts with locating peaks using a built in function of NMRView followed by manual adjustment according to the following procedure. After automated peak-picking in NMRView, I first looked to see if there were the correct number of peaks – either an equal number to the WT spectrum or one less (or one more) if a residue was mutated to (or from) a proline. I then overlaid the WT and mutant spectra and inspected the weakest peaks to see if they appeared to be true peaks rather than noise, PR artifacts, or Fourier transform artifacts. Finally, I checked a few spots in the spectrum that are likely to have peaks very close together to see if there were any instances of multiple peaks picked as single peaks due to overlap in the spectrum. At any point, I would manually add or subtract peaks as necessary. If I still did not find the expected number of peaks after these manual adjustments, then I obtained an explicit peak to residue assignment for that mutant as described below.

Assigning a limited number of datasets

For mutant spectra that did not clearly have the expected number of peaks, I was concerned that measuring chemical shift differences between WT and mutant spectra would be greatly complicated by missing or extra peaks arising from peak overlap, spectral artifacts, or chemical exchange processes. Therefore, I chose to pursue conventional peak assignments for this subset of mutant spectra. Assigning an HNCO spectrum simply requires a complementary HN(CA)CO spectrum which correlates the $C_{i-1}(O)H_iN_i$ and $C_i(O)$ nuclei. Since the HN(CA)CO pulse sequence has significantly less sensitivity and the spectrum has twice as many peaks as the HNCO, I did not try to obtain a PR-HN(CA)CO spectrum. A conventional HN(CA)CO spectrum required ~12 hours of spectrometer time to collect the data. To assign the HNCO spectrum

quickly, I first used a Matlab script to make a preliminary residue assignment based on chemical shift similarity to the WT spectrum (see Appendix 6:). With the preliminary assignments, I then manually inspected the HNCO/HN(CA)CO strips and made correction to the assignments as necessary according to standard NMR peak to residue assignment methods. For each assigned mutant, I was able to either find the missing peak(s), identify the extra peak(s), or confirm that a peak was indeed missing and not simply overlapped with another peak. In total, I assigned 18 free and 21 of the peptide-bound spectra. These explicit peak assignments were also very useful for estimating errors in measuring chemical shift changes without explicit peak assignments as described below.

Quantitating chemical shift change

Quantitating the chemical shift change upon mutation requires matching each mutant peak to a WT peak to measure the difference. This is a straightforward process for the datasets with assigned peak lists, but for the unassigned datasets, I had to choose an algorithm to decide which peaks should be paired together. This process starts with obtaining an optimal superposition of each mutant spectrum to the WT spectrum to correct for small differences in chemical shift referencing between spectrometers that lead to small chemical shift offsets.

Since the mutations were designed to be subtle, the mutant spectra should not be wholly different from the WT spectrum – there should be some nuclei that have the same chemical shifts and some nuclei that now have different chemical shifts. This is the case for every single mutant, although there were a few mutants that did show a large number of chemical shift changes. An example of the spectral comparison between WT and a mutant is shown in Figure 2-3 for the H^1 and N^{15} dimensions. Using the principle stated above that many chemical shifts should be the same and the assumption that direction of chemical shift change for the affected nuclei should (on average) be random, the mutant spectrum is aligned to the WT spectrum by performing a nonlinear least squares minimization on the distance between WT and mutant peaks for only the closest 50 peaks. The overlay was done in each of the three chemical shift dimensions (H^1, N^{15}, C^{13}(O)) independently, and this procedure was found to be extremely quick, robust, and insensitive to the actual number of peak distances minimized.

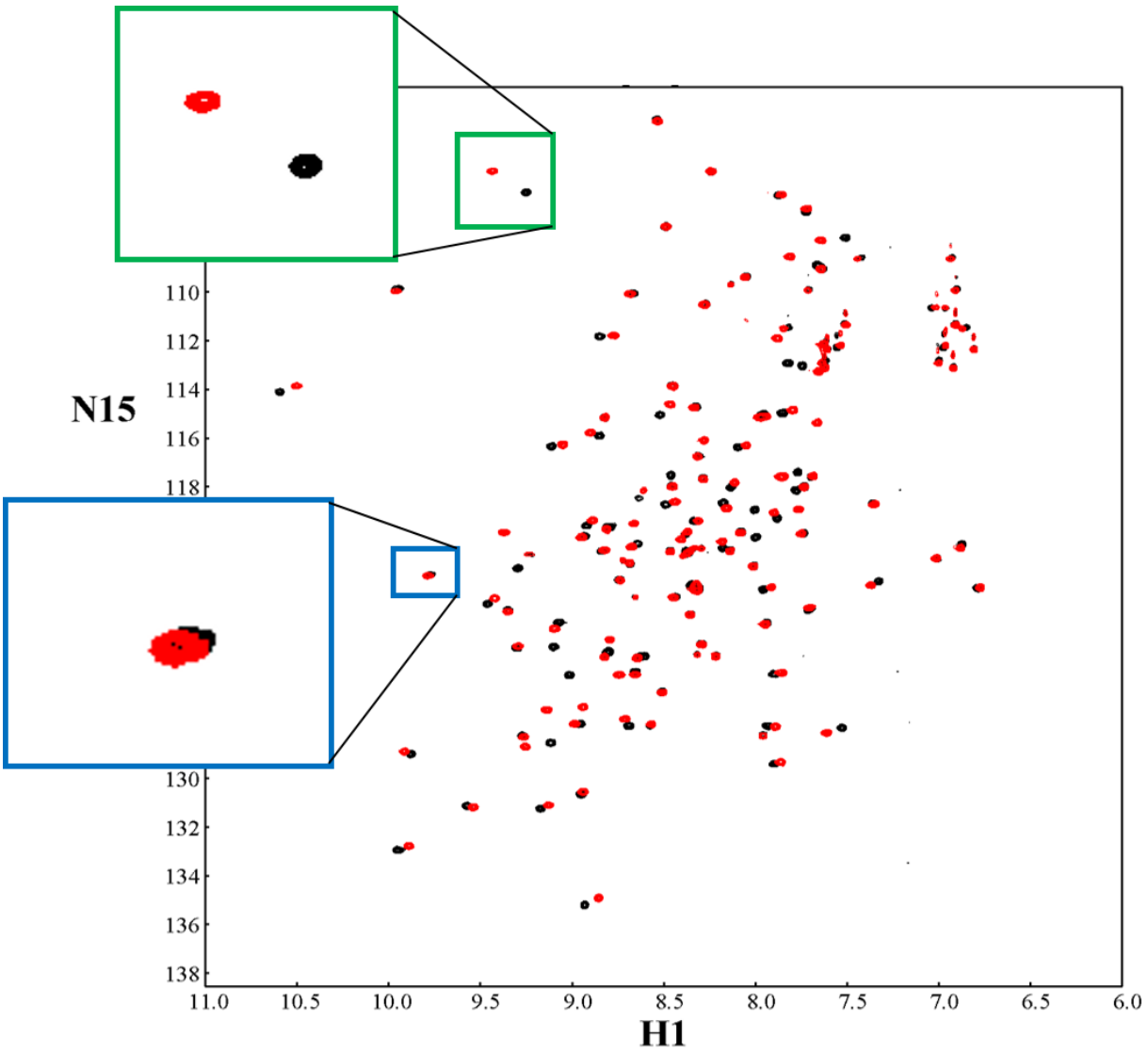


Figure 2-3: Overlay of H1-N15 spectra of WT-pep and A347V-pep HNCO spectra.
The WT spectrum (black) is show overlaid with a mutant spectrum, A347V (red), both in the presence of CRIPT peptide. An example of a small (nearly zero) chemical shift change is shown in the blue magnified region while an example of a significant chemical shift change is highlighted in the green region.

Once the mutant peaks were properly aligned to the WT peaks, one must choose how to calculate chemical shift changes without having explicit peak assignments. In the literature, the most common method for calculating the chemical shift changes is the "minimal chemical shift difference" (MCSD) method [33] which simply finds the closest mutant peak to each WT peak

and measures that chemical shift difference. The advantage of the minimum chemical shift difference method is that it will never overestimate the magnitude of chemical shift change. The disadvantage is that it will make many incorrect assignments when there is significant chemical shift change or when regions of the spectrum are crowded with peaks. An alternative method is to match each WT peak to a mutant peak one-to-one such that each mutant peak only corresponds to a single WT peak. I implemented a variant of this concept which I will call the "iterative matching" method using the following procedure:

1) calculate the distance from each WT peak to its closest mutant peak
2) select the WT-mutant peak match with the shortest distance and give the WT residue assignment to that mutant peak
3) remove that WT-mutant peak pair from further consideration
4) iteratively repeat steps 1-3 until all WT peaks are matched to mutant peaks.

This iterative matching method has the advantage that it can often find more correct peak assignments, but also has drawbacks including that it requires mutant peak lists with the same number of peaks as WT and that it opens the possibility of over-estimating chemical shift changes if WT and mutant peaks are incorrectly matched. See Figure 2-4 for an illustration of how the MCSD and iterative matching methods compare.

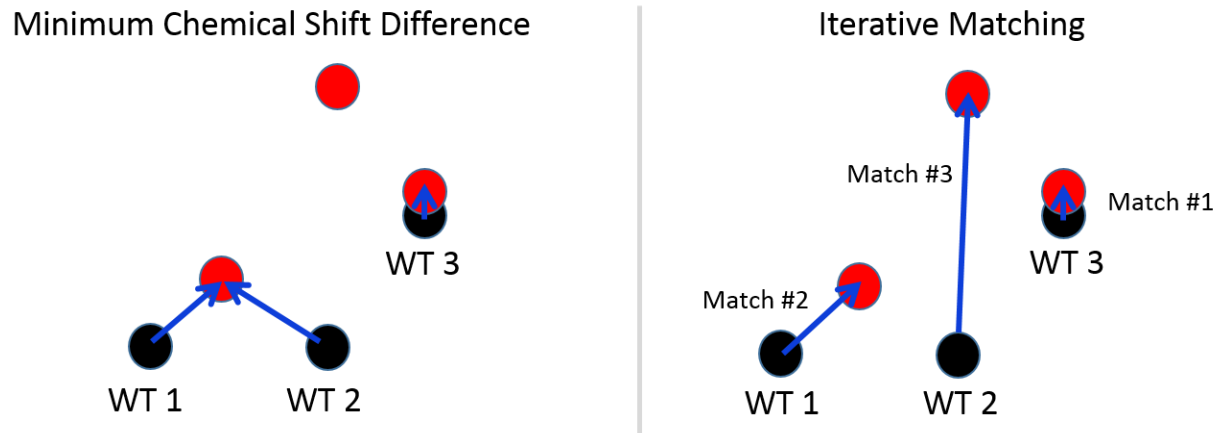


Figure 2-4: Peak matching algorithms to measure chemical shift change.
Hypothetical illustration to compare the MCSD and iterative matching methods. WT resonance peaks are shown in black and mutant residue peaks are shown in red. The iterative matching method attempts to better pair WT and mutant peaks by iteratively matching the closest pairs of peaks.

Error Analysis

In the process of trying to correctly identify all the peaks in the mutant spectra, I acquired residue assignments for 18 free and 21 peptide-bound spectra. Using these sets of assignments as a testing ground, I was able to estimate the errors made by the MCSD method and iterative matching algorithms. Figure 2-5 shows that the iterative matching algorithm makes fewer errors than the MCSD method: 42 vs. 112 for the 18 assigned free spectra and 65 vs. 135 for the 21 assigned peptide-bound spectra. It is important to note, however, that the iterative matching algorithm can produce chemical shift changes that are both too small and too large while the MCSD method is limited to either calculating the correct chemical shift change or underestimating the chemical shift change. Based on its superior accuracy, I chose to use the iterative matching algorithm for all datasets that were not explicitly assigned using the HNCO/HN(CA)CO strategy.

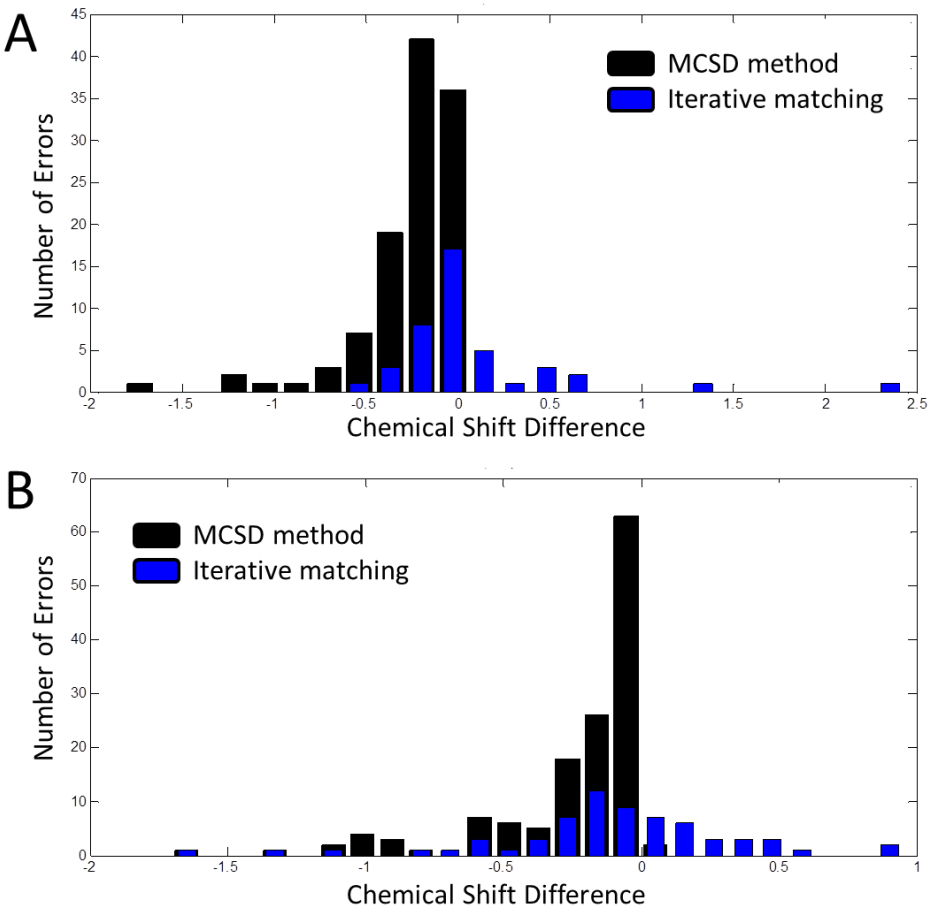


Figure 2-5: Accuracy comparison of the MCSD method and iterative matching algorithm. A) The distribution of chemical shift measurement errors made by the MCSD method and the iterative matching algorithm for the 18 assigned datasets without peptide. 112 total errors are made by the MCSD method while 42 errors are made by the iterative matching algorithm. B) The same as (A) for the 21 assigned datasets with peptide. 139 total errors are made by the MCSD method while 65 errors are made by the iterative matching algorithm. Chemical shift difference ($\Delta\delta = \text{norm}(\Delta\delta H^{1}, \Delta\delta N^{15}, \Delta\delta C^{13})$)

In addition to evaluating the best peak-matching algorithm, assigning ~25% of the spectra allowed an estimation of the error in the chemical shift change measurements for the datasets that were not assigned. In Figure 2-6, the number of incorrectly matched peaks (using the iterative matching algorithm) is plotted versus the number of perturbations with a chemical shift change greater than 0.1 ppm. This graph clearly shows that spectra with fewer chemical shift changes (more similar to WT) have fewer errors in the peak matching process. In fact, spectra with less than 20 perturbations greater than 0.1 ppm tend to have one or zero mismatched peaks with the single error usually occurring at the site of mutation. Above 20

perturbations, the number of errors in each dataset increases with the number of perturbations greater than 0.1 ppm.

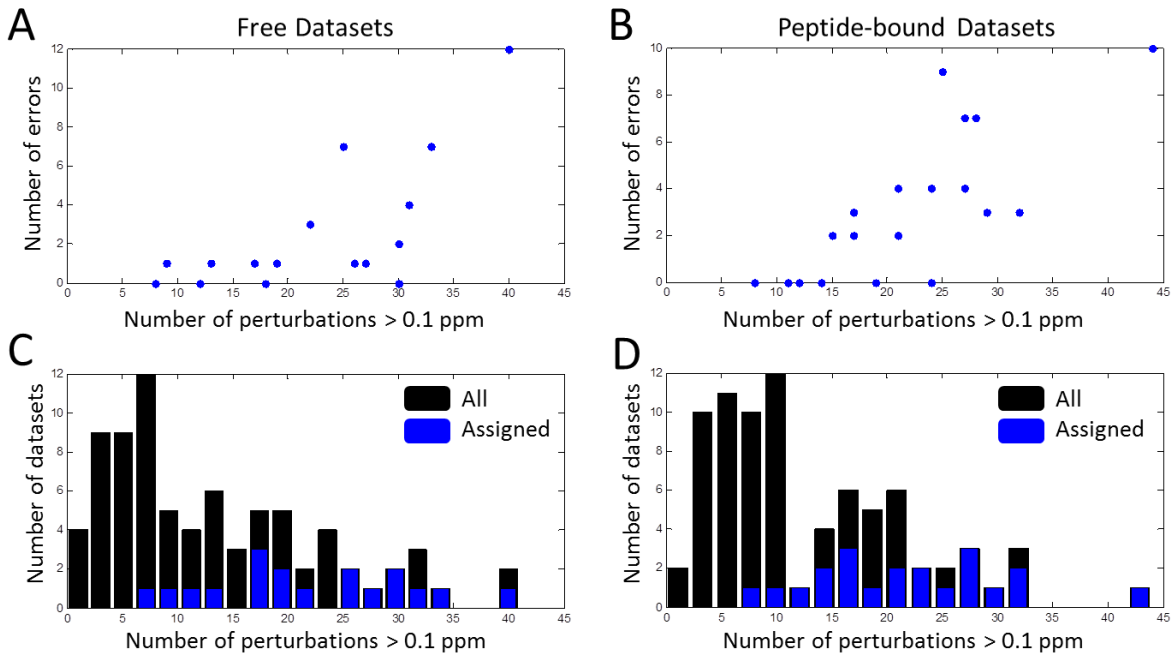


Figure 2-6: Mutants with more perturbed residues are more difficult to correctly match peaks.
A) The number of errors made by the iterative matching algorithm for each of the 18 spectra (without peptide) is plotted against the number of perturbations greater than 0.1 ppm for each dataset. Above 20 perturbations > 0.1 ppm, there is a strong positive correlation with the number of errors. B) Same as (A), but for peptide-bound spectra. C & D) Histograms showing the number of free and peptide-bound spectra with the indicated number of perturbations > 0.1 ppm are shown in black while assigned datasets are shown in blue. The assigned spectra preferentially cover the datasets with more perturbations, and hence, more predicted errors.

In panels C and D of Figure 2-6, histograms of the number of perturbations greater than 0.1 ppm are shown for both the free and peptide-bound datasets along with an indication of which datasets have been assigned. The assigned datasets span the range of 10 to 40 perturbations greater than 0.1 ppm, but are concentrated in datasets that with greater numbers of significant perturbations. This concentration is advantageous because many of the datasets expected to have a significant number of errors have been assigned and the vast majority of the unassigned datasets are expected to have few errors. The error analysis

presented here argues that the iterative matching peak assignment method is highly accurate and produces relatively few errors. Avoiding the need to collect HNCACO spectra to make conventional peak assignments for each mutant spectrum saved weeks of valuable spectrometer time and contributed significantly to making data collection feasible.

Conclusions

In this chapter, I described the creation of a global chemical shift perturbation assay for the purpose of mapping the interactions between all pairs of residues in a protein. This assay is fast compared to other biophysical assays of residue interactions (such as solving protein structures) and offers atomic resolution that is not available in other biochemical experiments. The global chemical shift perturbation assay was implemented in PSD95 PDZ3, which represents an ideal test case.

Several technical considerations were made in order to make this experiment possible. First, the use of the projection-reconstruction method for acquiring high resolution HNCO spectra significantly lowered the amount of spectrometer time required to obtain the necessary data and was essential to enabling the investigation of mutations at all positions in the protein. Alongside fast spectral acquisition, I implemented several automated processes including automatic phase correction, chemical shift referencing, and peak analysis. These processes allowed the collection and processing of HNCO spectra for ~160 mutant proteins.

After obtaining residue assignments for a significant number of the HNCO spectra, I was also able to evaluate the best strategy for calculating chemical shift change for unassigned mutants. An iterative matching method chosen because it was found to be more accurate than the more common MCSD method. I assigned ~25% of the mutant spectra which allowed me to estimate the incidence of errors which was found to be quite small. In addition, these residue assignments will be used to calculate chemical shift change for the mutants for which they are available, further lowering the number of possible errors in the dataset.

By combining conventional NMR experiments with new time-saving data acquisition schemes and high-throughput automated data processing, I have created a global chemical shift perturbation analysis that is extremely powerful and reveals information about residue

interactions that is not available by any other means. The results of this experiment in PDZ3 are presented and discussed in the following chapter.

Methods

Mutant library creation

An expression construct for rat PSD95 PDZ3 was obtained from Rod Mackinnon's laboratory which consisted of a pGEX-4T1 expression plasmid with the PDZ3 sequence cloned C-terminal to a glutathione S-transferase (GST) domain with a linker region containing a thrombin protease recognition sequence. The sequence of the expressed polypeptide after thrombin cleavage is:
GSPEFLGEEDIPREPRRIVIHRGSTGLGFNIVGGEDGEGIFISFILAGGPADLSGELRKGDQILSVNGVDLRNASHE
QAAIALKNAGQTVTIIAQYKPEEYSRFEANSRVDSSGRIVTD.
(PDZ domain boundaries indicated by underline)

Of note, the construct contains the correct genomic sequence for rat PSD95 PDZ3, but has a single mutation (328 I→V) when compared to the original published structures 1BFE and 1BE9 from the Mackinnon laboratory which are used for illustration purposes in this document. Also, the construct contains two N→D mutations near the C-terminus when compared to 1BFE and 1BE9 structures, but both of these residues are outside of the PDZ domain and are not well-resolved in the structures.

Using a multiple sequence alignment of 240 diverse PDZ domains (Appendix 1:) assembled previously in the lab by Steve Lockless, I identified the most common (consensus) and next most common amino acid (NMCAA) at each position. Appropriate primers were designed to mutate residues 311-391 of rat PSD95 PDZ3 (PDB 1BE9 numbering) to the consensus amino acid (or the NMCAA whenever the wild-type PDZ3 residue matched the consensus amino acid), and single mutant constructs were obtained using the QuikChange® (Stratagene) site-directed mutagenesis kit. Positive mutant clones were confirmed by DNA sequencing. Each single mutant clone was then transformed into *E. coli* BL21-DE3 cells, and cell stocks were made using 800uL of BL21 cells (OD = 1.0, obtained during log phase growth) + 200uL of 80% glycerol and frozen in liquid nitrogen and stored at -80^{O}C. A full list of mutations considered is provided in Appendix 2:.

Protein expression

Scrapings from BL21-DE3 cell frozen cell stocks were streaked on LB + ampicillin (AMP) plates or plated from fresh plasmid transformation. From a plate, several colonies were used to inoculate 2mL of MDG+AMP and grown overnight at 30C. 50mL of pre-warmed M9+AMP+trace metals medium was inoculated with 250uL of saturated MDG culture and grown at 37C until the culture reached on OD of 0.5. The 50mL culture was spun down, supernatant removed, and pellet resuspended in pre-warmed 250mL M9+AMP+trace metals medium. Culture was allowed to grow at 37C until an OD of 0.5 was reached at which point the culture was cooled to 25C and expression induced with 500uM final concentration of IPTG. Expression was allowed to continue for 12-14 hours until maximum OD around 2.5 was reached. Cells were centrifuged and resuspended in Buffer A for immediate purification or freezing for later purification.

Protein Purification

Cells were suspended in Buffer A and the protease inhibitors PMSF, Leupeptin, and Pepstatin were added. Cells were sonicated in a vial submersed in an ice water bath to prevent excessive heating. Lysed

cell contents were centrifuged at high speed (20,000 rpm in a Sorvall SS34 rotor for 30 minutes. Supernatant was added to 1.5-2.0mL bed volume of GST resin and incubated at 4C for 45 minutes. GST resin was then washed 3x with Buffer A (140mM NaCl, 2.7mM KCl, 10mM Na_2HPO_4, 1.8mM KH_2PO_4, pH 7.3) and 3x with NMR buffer (50mM NaCl, 25mM KPO_4, 1mM EDTA, 0.02% NaN_3, pH 7.0). Most supernatant was removed and 20-100 units of thrombin was added and sample continuously mixed at room temperature. Thrombin cleavage was monitored by SDS-PAGE and digest was discontinued once cleavage appeared to reach ~75% completion. Supernatant was then eluted from the GST resin using a small disposable column, and resin was washed with 4x200uL of NMR buffer of which the flow-through was added to the original elution. 30uL of benzamidine sepharose resin was added to the eluted sample and incubated at 4C for 20 minutes to remove residual thrombin. Benzamidine resin was then removed by filtration through a small disposal column and washed with 200uL NMR which was also collected. Final sample yields of wild-type and mutant PDZ domains were typically 1-1.5 mL of 1-1.5 mM protein.

NMR sample preparation

Purified protein samples were concentrated using an Amicon Ultra-4 3kDa MWCO filter unit to a final volume of 270uL. 30uL of D_2O was added for a final volume of 300uL. Samples were added to 5mm D_2O-matched Shigemi NMR tubes and sealed.

The peptide TKNYKQTSV was synthesized using Fmoc chemistry with an N-terminal acetyl group and a C-terminal carboxylate group and purified by reversed-phase HPLC. Stock peptide solution was made by dissolving lyophilized peptide in NMR buffer and adjusting the pH to 7.0 with NaOH. Stock peptide solution was added to protein NMR samples to a final concentration of 4mM.

NMR spectral acquisition

All data was acquired on Varian Unity Inova spectrometers operating at 600 MHz with room temperature or cold probes at 25^0C sample temperature. HNCO NMR spectra were acquired using the ghn_co pulse sequence provided in the Varian Biopack software package. Data for projection reconstruction spectra were typically acquired with 8 projections: 0, +/- 22.5^0, +/- 45^0, +/- 67.5^0, 90^0, with 128 complex points in the tilted dimensions and 4 transients per point to accommodate phase cycling.

NMR spectrum processing

Intermodulated projections (ex: +/- 22.5^0) were first split using the PRSP software. Individual projections were then processed using the NMRPipe software software package using the following protocol to produce frequency domain projection data: (A) direct dimension: 1) baseline correction using polynomial fitting, 2) signal apodization using the sine-bell function, 3) zero-filling to double the number of data points, 4) Fourier transform, 5) phase correction. (B) indirect dimension: 1) forward-backward linear prediction to double the number of data points, 2) sine-bell apodization, 3) zero-filling, 4) Fourier transform, 5) phase correction. A sample processing script is provided in Appendix 2:.

A custom automatic phasing routine was written implemented in Matlab to determine optimal phases for the Fourier-transformed 2-D spectra using the maximum white space principle [31]. Matlab m-files are provided in Appendix 4:.

Properly phased and Fourier-transformed projection spectra were used to reconstruct a 3-D HNCO spectrum using the PR-Calc software. Input spectra were scaled using baseline noise values and processed using the lower-value algorithm. A sample PR-Calc control file is included in Appendix 5:.

Peak picking and adjustment

Processed PR-HNCO spectra were loaded into the NMRViewJ software [34, 35] and peak detection was performed at an appropriate noise floor. Peaks were then manually edited for overlaps, noise peaks, artifact peaks, etc. Peaklists were adjusted until a correct number of peaks was detected. If too many or

too few peaks existed after manual editing, and HNCACO spectrum was obtained for explicit residue assignment.

HNCO/HN(CA)CO residue assignments

For WT PDZ3 and each mutant that was explicitly assigned, HNCACO spectra were acquired with conventional linear sampling of the indirect dimensions including 50-70 complex points per indirect dimension. For WT PDZ3, residue assignments were made using conventional HNCO/HN(CA)CO peak assignment procedures in NMRViewJ. For each mutant, a preliminary assignment was made using the iterative matching algorithm using the wild-type assignments as a reference. The mutant HNCO & HN(CA)CO spectra were then examined in NMRViewJ and corrections were made to the assigned residue-peak pairs as necessary.

References

1. Jain, R.K. and R. Ranganathan, *Local complexity of amino acid interactions in a protein core.* Proc Natl Acad Sci U S A, 2004. **101**(1): p. 111-6.
2. Asada, T., et al., *Simulation study of interactions and reactivities between NADH cytochrome b5 reductase and cytochrome b5.* Journal of Molecular Liquids, 2009. **147**(1-2): p. 139-144.
3. Ho, B.K. and D.A. Agard, *Conserved tertiary couplings stabilize elements in the PDZ fold, leading to characteristic patterns of domain conformational flexibility.* Protein Sci, 2010. **19**(3): p. 398-411.
4. Kidd, B.A., D. Baker, and W.E. Thomas, *Computation of conformational coupling in allosteric proteins.* PLoS Comput Biol, 2009. **5**(8): p. e1000484.
5. Kong, Y. and M. Karplus, *Signaling pathways of PDZ2 domain: a molecular dynamics interaction correlation analysis.* Proteins, 2009. **74**(1): p. 145-54.
6. Nguyen, P.H., P. Derreumaux, and G. Stock, *Energy flow and long-range correlations in guanine-binding riboswitch: a nonequilibrium molecular dynamics study.* J Phys Chem B, 2009. **113**(27): p. 9340-7.
7. Ota, N. and D.A. Agard, *Intramolecular signaling pathways revealed by modeling anisotropic thermal diffusion.* J Mol Biol, 2005. **351**(2): p. 345-54.
8. Sharp, K. and J.J. Skinner, *Pump-probe molecular dynamics as a tool for studying protein motion and long range coupling.* Proteins, 2006. **65**(2): p. 347-61.
9. Kinney, J.B., et al., *Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence.* Proc Natl Acad Sci U S A, 2010. **107**(20): p. 9158-63.
10. van Opijnen, T., K.L. Bodi, and A. Camilli, *Tn-seq: high-throughput parallel sequencing for fitness and genetic interaction studies in microorganisms.* Nat Methods, 2009. **6**(10): p. 767-72.
11. Mulder, F.A. and M. Filatov, *NMR chemical shift data and ab initio shielding calculations: emerging tools for protein structure determination.* Chem Soc Rev, 2010. **39**(2): p. 578-90.
12. Hunter, C.A., M.J. Packer, and C. Zonta, *From structure to chemical shift and vice-versa.* Progress in Nuclear Magnetic Resonance Spectroscopy, 2005. **47**(1-2): p. 27-39.
13. Feng, W. and M. Zhang, *Organization and dynamics of PDZ-domain-related supramodules in the postsynaptic density.* Nat Rev Neurosci, 2009. **10**(2): p. 87-99.
14. Zhang, W., et al., *Citron binds to PSD-95 at glutamatergic synapses on inhibitory neurons in the hippocampus.* J Neurosci, 1999. **19**(1): p. 96-108.
15. Bolliger, M.F., et al., *Identification of a novel neuroligin in humans which binds to PSD-95 and has a widespread expression.* Biochem J, 2001. **356**(Pt 2): p. 581-8.
16. Li, Y., et al., *DHHC5 interacts with PDZ domain 3 of post-synaptic density-95 (PSD-95) protein and plays a role in learning and memory.* J Biol Chem, 2010. **285**(17): p. 13022-31.
17. Niethammer, M., et al., *CRIPT, a novel postsynaptic protein that binds to the third PDZ domain of PSD-95/SAP90.* Neuron, 1998. **20**(4): p. 693-707.
18. Doyle, D.A., et al., *Crystal structures of a complexed and peptide-free membrane protein-binding domain: molecular basis of peptide recognition by PDZ.* Cell, 1996. **85**(7): p. 1067-76.
19. Nourry, C., S.G. Grant, and J.P. Borg, *PDZ domain proteins: plug and play!* Sci STKE, 2003. **2003**(179): p. RE7.
20. Penkert, R.R., H.M. DiVittorio, and K.E. Prehoda, *Internal recognition through PDZ domain plasticity in the Par-6-Pals1 complex.* Nat Struct Mol Biol, 2004. **11**(11): p. 1122-7.
21. Hillier, B.J., et al., *Unexpected modes of PDZ domain scaffolding revealed by structure of nNOS-syntrophin complex.* Science, 1999. **284**(5415): p. 812-5.

22. Gallardo, R., et al., *Structural diversity of PDZ-lipid interactions.* ChemBioChem, 2010. **11**(4): p. 456-67.
23. Mishra, P., et al., *Dynamic scaffolding in a G protein-coupled signaling system.* Cell, 2007. **131**(1): p. 80-92.
24. Peterson, F.C., et al., *Cdc42 regulates the Par-6 PDZ domain through an allosteric CRIB-PDZ transition.* Molecular Cell, 2004. **13**(5): p. 665-676.
25. Stiffler, M.A., et al., *PDZ domain binding selectivity is optimized across the mouse proteome.* Science, 2007. **317**(5836): p. 364-9.
26. Tonikian, R., et al., *A specificity map for the PDZ domain family.* PLoS Biol, 2008. **6**(9): p. e239.
27. Wang, N.X., H.J. Lee, and J.J. Zheng, *Therapeutic use of PDZ protein-protein interaction antagonism.* Drug News Perspect, 2008. **21**(3): p. 137-41.
28. Dev, K.K., *Making protein interactions druggable: targeting PDZ domains.* Nat Rev Drug Discov, 2004. **3**(12): p. 1047-1056.
29. Coggins, B.E., R.A. Venters, and P. Zhou, *Radial sampling for fast NMR: Concepts and practices over three decades.* Progress in Nuclear Magnetic Resonance Spectroscopy.
30. Kupce, E. and R. Freeman, *Projection-reconstruction technique for speeding up multidimensional NMR spectroscopy.* J Am Chem Soc, 2004. **126**(20): p. 6429-40.
31. Balacco, G. and C. Cobas, *Automatic phase correction of 2D NMR spectra by a whitening method.* Magn Reson Chem, 2009. **47**(4): p. 322-7.
32. Coggins, B.E. and P. Zhou, *PR-CALC: a program for the reconstruction of NMR spectra from projections.* J Biomol NMR, 2006. **34**(3): p. 179-95.
33. Farmer, B.T., 2nd, et al., *Localizing the NADP+ binding site on the MurB enzyme by NMR.* Nat Struct Biol, 1996. **3**(12): p. 995-7.
34. Johnson, B.A., *Using NMRView to visualize and analyze the NMR spectra of macromolecules.* Methods Mol Biol, 2004. **278**: p. 313-52.
35. Johnson, B.A. and R.A. Blevins, *NMR View: A computer program for the visualization and analysis of NMR data.* J Biomol NMR, 1994. **4**(5): p. 603-14.

CHAPTER 3: A Heterogeneous Physical Architecture and Structural Modes Demonstrated in PDZ3

One goal of this project is to test the hypothesis that proteins have a heterogeneous structural architecture with a large number of weak interactions between amino acid residues and a smaller number of strong and cooperative interactions. The global chemical shift perturbation experiment, described methodologically in the previous chapter, is designed to test this hypothesis by systematically perturbing every residue in a protein and observing the effects. Given the review of the literature concerning functional and biophysical properties of proteins in the first chapter, several predictions can be made. First, I expect that each mutation will have an effect on its immediate local environment due to the change in mass and chemistry of the amino acid side chain. In addition, we also know that most mutations slightly degrade thermodynamic stability, and that the interactions of each amino acid with its local environment are, on average, favorable [1]. Thus, I also expect to see a perturbation to the local environment due to the likely disruption of optimized interactions between the wild-type amino acid and its local contacts. Second, I expect to observe structural heterogeneity in the molecule based on the heterogeneous response to mutation found in functional studies and based on the sparse patterns of coevolution revealed by SCA. By structural heterogeneity, I mean that many mutations will have small effects confined to their local environment while some mutations will have larger effects including residues outside their local environment. Third, I expect to see sets of systematically interacting residues. This prediction is based on the observation that focused testing of allosteric pathways has revealed that some residues interact cooperatively within proteins [2]. If cooperativity is present among residues within PDZ3, then this may manifest as sets of mutations that display similar patterns of perturbation in PDZ3 despite the mutations being different in location and character. And finally, if cooperativity is observed, I can compare the patterns of structural cooperativity to the coevolution patterns revealed by SCA. If networks of cooperative interactions are driving coevolution, then we might observe patterns of systematic perturbation that match patterns of

coevolution. Each of these predictions will now be addressed by analyzing the results of the global chemical shift perturbation assay when applied to PDZ3.

Raw Data

Performing the global chemical shift perturbation experiment on PDZ3 resulted in the measurement of the chemical shift change (relative to WT) at each observed ^{1}H, ^{15}N, and $^{13}C(O)$ nucleus in the protein as a result of a single mutation at each position. The data can be represented as a matrix of chemical shift change values where columns represent the perturbation profile of each mutation and the rows represent the nuclei at which the chemical shifts were measured. Since the experiment was done on each mutant and then repeated in the presence of saturating amounts of the CRIPT target peptide, there is both a "free" chemical shift perturbation matrix and a "peptide-bound" chemical shift perturbation matrix. The raw chemical shift perturbation matrices are shown in panels A-B of Figure 3-1 with the proton, carbon, and nitrogen nuclei separated along the vertical dimension. Since these nuclei have different ranges of observed chemical shifts, the nitrogen and carbon nuclei are normalized by standard scaling factors (σ_N = 0.17 and σ_C = 0.39) to compensate [3]. To simplify the display of the data and to more easily map the data onto the protein structure, chemical shift data from $^{13}C_{i-1}(O)^{1}H_{Ni}{}^{15}N_i$ nuclei in the same bonded spin spin system are combined using a root-mean-square (RMS) sum of the normalized chemical shift changes and assigned to the *i-th* residue. The RMS chemical shift perturbation matrix is shown in panels D-E of Figure 3-1. In essence, this RMS matrix describes the interaction of each residue in the protein (represented by mutations in each column) with all other residues in the protein (represented by the RMS chemical shift perturbation of an $^{13}C_{i-1}(O)^{1}H_{Ni}{}^{15}N_i$ spin system in each row). This chemical shift perturbation matrices in panels D-E of Figure 3-1 is conceptually analogous to a SCA matrix wherein each pixel represents the statistical interaction as a result of coevolution between two positions. However in this chemical shift perturbation experiment, each pixel represents the physical interaction between two positions rather than a statistical interaction.

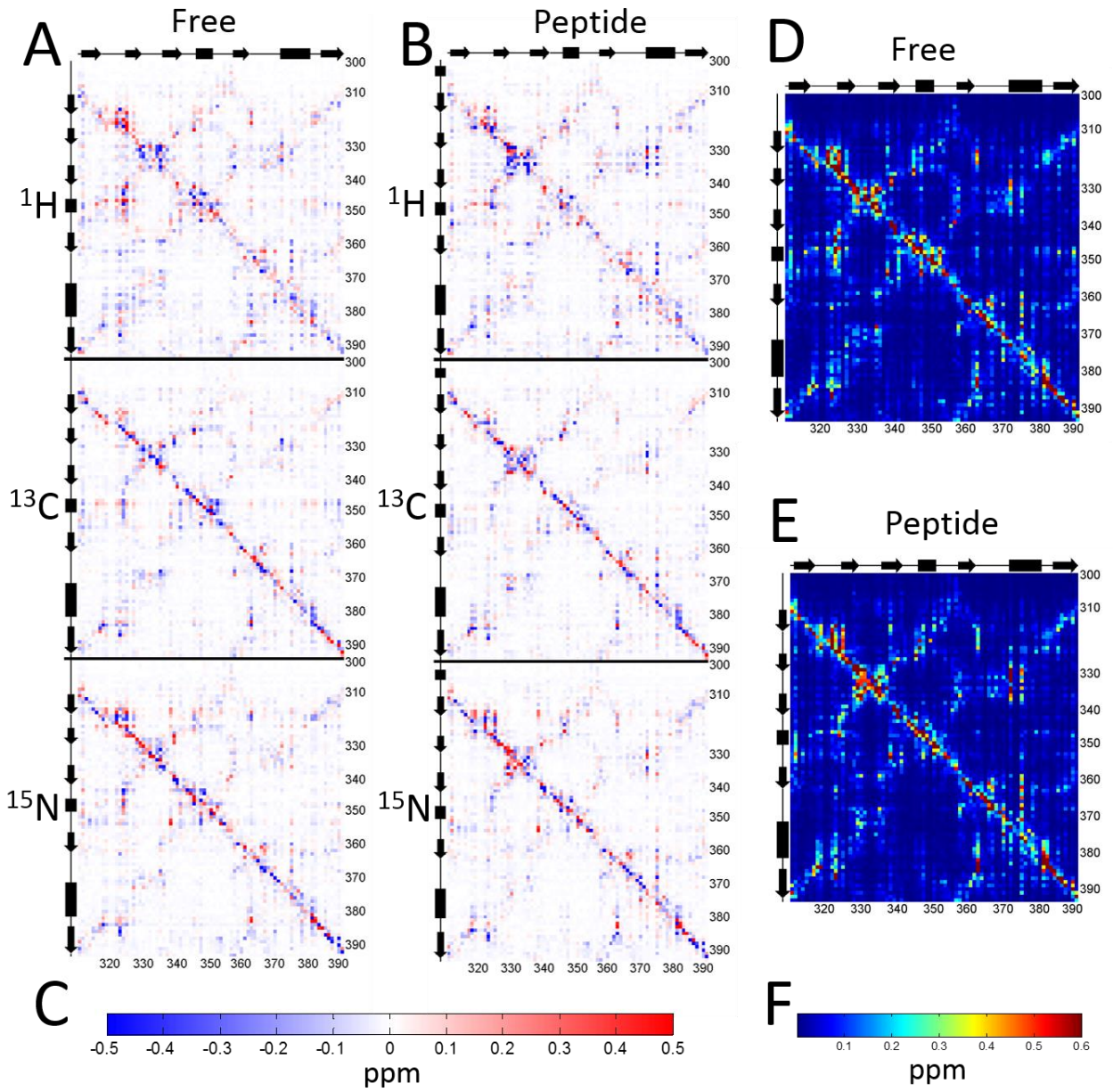


Figure 3-1: Chemical shift perturbation matrices for PDZ3.
A) Chemical shift perturbation matrix for free PDZ3. Chemical shift changes at proton (^{1}H), carbon (^{13}C), and nitrogen (^{15}N) nuclei are separated vertically. Secondary structure is shown on the top and left while sequence numbering of PDZ3 is shown on the bottom and right for reference. B) Chemical shift perturbation matrix of PDZ bound to CRIPT peptide. C) Colorbar indicates the range of values depicted in the chemical shift perturbation matrics in ppm normalized to proton chemical shifts. D & E) RMS representations of the free (D) and peptide-bound (E) chemical shift perturbation matrices. F) A colorbar indicates the range of values (ppm) depicted in the RMS matrices.

First-Order Observations

Before fully dissecting the results of the global chemical shift perturbation analysis, it is helpful to look at the chemical shift perturbation analysis of some single mutants to examine the underlying data. For each single mutant, the chemical shift change (relative to wild-type) is measured at all $C_{i-1}(O)H_{Ni}N_i$ bonded nuclei. Although chemical shifts were recorded for nearly all residues in the protein, the perturbation analysis is restricted to the non-proline residues with visible resonance peaks in the WT protein for residues 300-393. This restriction includes the canonical portion of the PDZ domain that can be sequence-aligned to other PDZ domains and results in 88 x 3 = 264 independent observations for each free PDZ3 mutant and 90 x 3 = 270 independent observations for each peptide-bound PDZ3 mutant. The peptide-bound spectra include observations at two additional residues in the $\beta 1 - \beta 2$ loop because resonance peaks for these residues were not visible in the free spectra due to chemical exchange effects. The C-terminal portion (residues 394-415) of the protein was not included in the analysis because it is not part of the canonical PDZ domain. The C-terminal residues form secondary structure elements (1 alpha helix and 2 beta strands) that pack against the PDZ domain. Some mutations near this interface have a strong effect on the extra C-terminal residues, so chemical shifts from these residues outside of the PDZ domain were excluded so as not to skew analysis of the chemical shift perturbation data.

As an example, consider the chemical shift perturbation as a result of the two mutations N326S and I327L as shown in Figure 3-2. For the purposes of illustration, the chemical shift changes at each bonded $C_{i-1}(O)H_{Ni}N_i$ nuclei are combined using the RMS distance and assigned to the *i-th* residue (in the same process as Figure 3-1 D-E) and residues with chemical shift change greater than 0.1 ppm are shown in spheres. As expected, both mutations perturb their local environment, however the magnitude and distribution of the perturbations is quite different. The mutation N326S has relatively little effect on PDZ3 with most residues having a very small chemical shift change and only two residues (both in close physical proximity to the mutation) showing a change greater than 0.1ppm. In fact, this mutation is typical of many mutations which have relatively few effects on other residues; 39 free mutations and 40

peptide-bound mutations perturb less than 10 residues by greater than 0.1 ppm RMS (Figure 2-6).

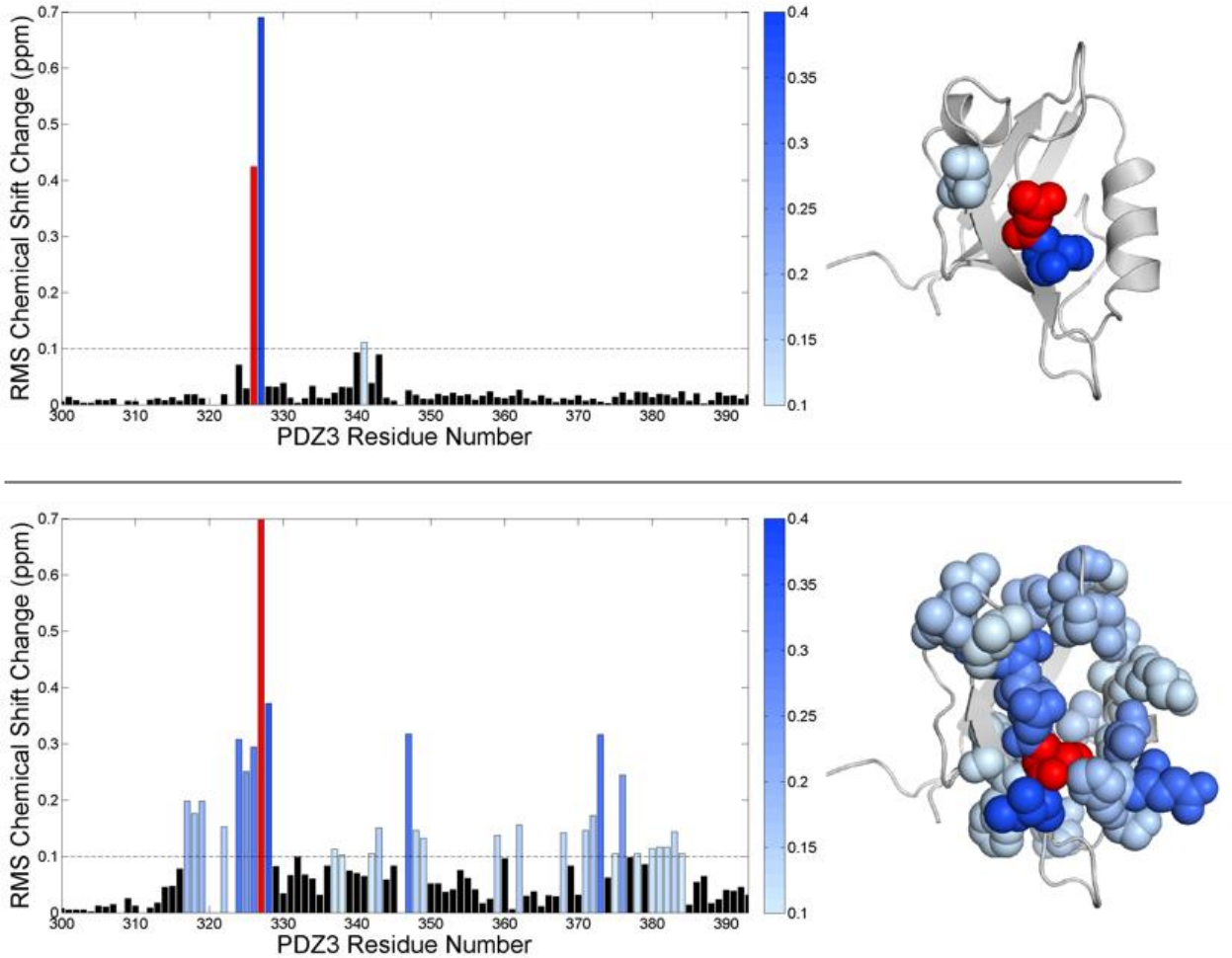


Figure 3-2: N326S and I327L mutations.
RMS chemical shift change is shown for the N326S (upper panel) and I327L (lower panel) mutation. The mutated residue is indicated in red. Positions for which the RMS chemical shift change is greater than 0.1ppm have been colored using the indicated color scale and the corresponding residue is depicted with spheres on the PDZ3 structure.

Contrast the effect of the N → S mutation at position 326 with the I → L mutation at the adjacent position 327 as shown in the bottom panel of Figure 3-2. The I327L mutant results in a significant chemical shift change for a much greater number of residues. These perturbed residues are clustered near the site of mutation but also extend to distant residues of the protein. These two mutations give a small example of the diversity of chemical

shift perturbation patterns seen in PDZ3. A structure-based representation of the chemical shift perturbations patterns for all free PDZ3 mutants is shown in Figure 3-3.

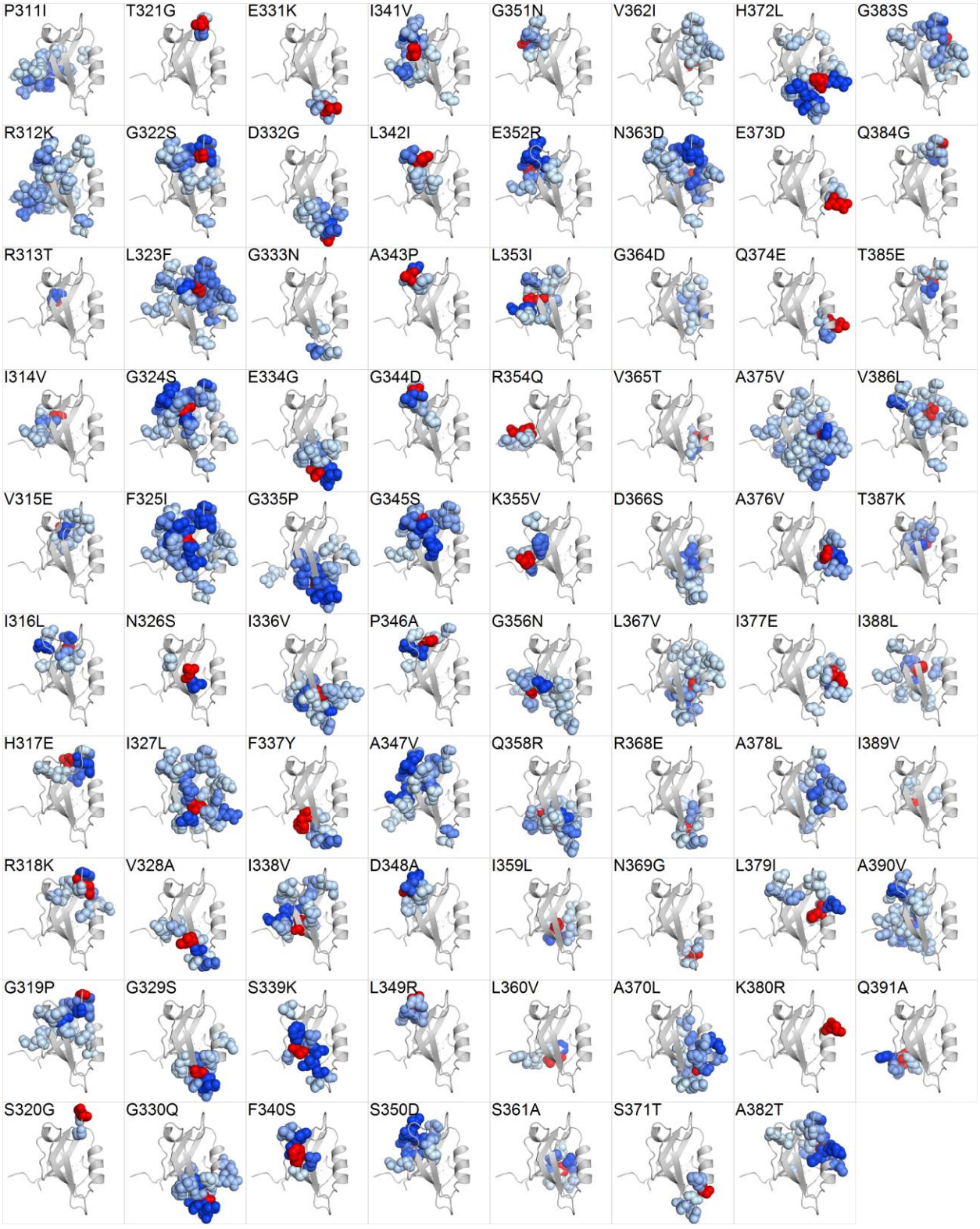


Figure 3-3: Chemical shift perturbations in PDZ3.

The gallery shown in Figure 3-3 allows us to compare our data with some of our initial hypotheses. First, an inspection of the chemical shift perturbation patterns projected onto the structure of PDZ3 quickly confirms that all mutations perturb their local environment regardless of whether the overall effect is large or small. Second, as shown in the original examples (N326S and I327L), mutations in PDZ3 have a wide range of effects from very little (R313T, S320G, T321G, K380R) to mutations that perturb many other residues such as F325I and A375V. In all cases though, there are some residues within the domain that are unperturbed or only minimally perturbed, indicating that the overall structure of the domain has been preserved. Upon closer inspection of the mutations, another key observation becomes apparent. Many perturbations propagate asymmetrically away from the mutation site; this holds true for mutations that result in both small and large numbers of perturbed residues. An illustrative example is provided by the neighboring mutations of F337Y and I338V in Figure 3-4. In the orientation shown, F337Y only affects residues toward the lower portion of the protein while I338V on effects other residues toward the upper portion of the protein. The stark difference in effect may not be completely unexpected as the sidechains of the two residues are oriented on opposite sides of a beta strand. However, they do illustrate that the perturbations do not propagate isotropically like an expanding shell from the point of mutation.

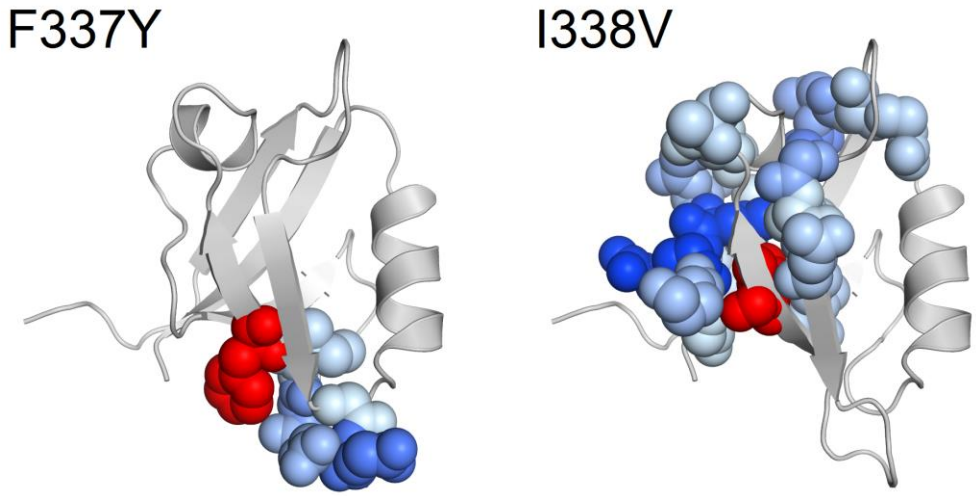


Figure 3-4: Mutations F337Y and I338V.
Mutations F337Y and I338V highlight the feature that some perturbations propagate strongly anisotropically away from the site of mutation.

Thus, we find that two types of heterogeneity are present in the chemical shift perturbation patterns: 1) evolutionarily conservative mutations have widely varying magnitudes (in terms of the number of residues perturbed); and 2) the effects of mutation can be propagated in strongly asymmetric patterns. Both of these findings indicate that some residues are more strongly physically coupled to other residues in the protein while others are only weakly coupled. The question remains, however, as to whether this heterogeneity of physical coupling is the result of systematic cooperative interactions between a subset of residues or whether the patterns of strong and weak interactions are largely random. In addition, if this heterogeneity is the result of cooperative interactions, are cooperative physical interactions important for function in PDZ3 and are these physical interactions the source of statistical coupling seen in the PDZ domain family?

Peptide-bound Datasets

To aid in addressing the question of the functional relevance of these chemical shift perturbation patterns, chemical shift data was also recorded for each mutation in the presence of a saturating amount of a target peptide. The CRIPT peptide (TKNYKQTSV) binds to PDZ3 in a groove created by the β2 strand and the α-2 helix with the terminal carboxylate and valine sidechain inserting into a pocket near the base of the $\beta1-\beta2$ loop (see Figure 2-1 and Figure 3-5). The structural, dynamic, and chemical shifts change of the wild-type protein as a result of peptide binding are shown in Figure 3-5. Upon peptide-binding, chemical shift changes are seen in many, but not all of the PDZ3 residues at the binding interface and in some residues that do not contact the peptide. Although only five residues of the 9-mer peptide are resolved in the crystal structure, the four unresolved residues are expected to be in contact with or in the vicinity of the $\beta2-\beta3$ loop, potentially explaining the chemical shift changes in that region. Another feature of peptide-binding in PDZ3 is a clamping of the $\beta1-\beta2$ loop (shown in the upper left panel of Figure 3-5). In free PDZ3, the $\beta1-\beta2$ loop has high conformational flexibility as evidenced by high B-factors in the X-ray crystal structure (Figure 3-5 upper center panel) and an absence of chemical shift peaks for residues 320, 321, and 323 due to chemical shift

exchange effects. When CRIPT peptide binds, however, the β1–β2 loop adopts a more clamped down position, B-factors are lowered (Figure 3-5 upper right panel), and peaks corresponding to residues 321 and 323 are now visible in the NMR spectrum due to reduced chemical exchange. A peak for residue 320 continues to be undetected in the peptide-bound spectrum, presumably due to chemical exchange.

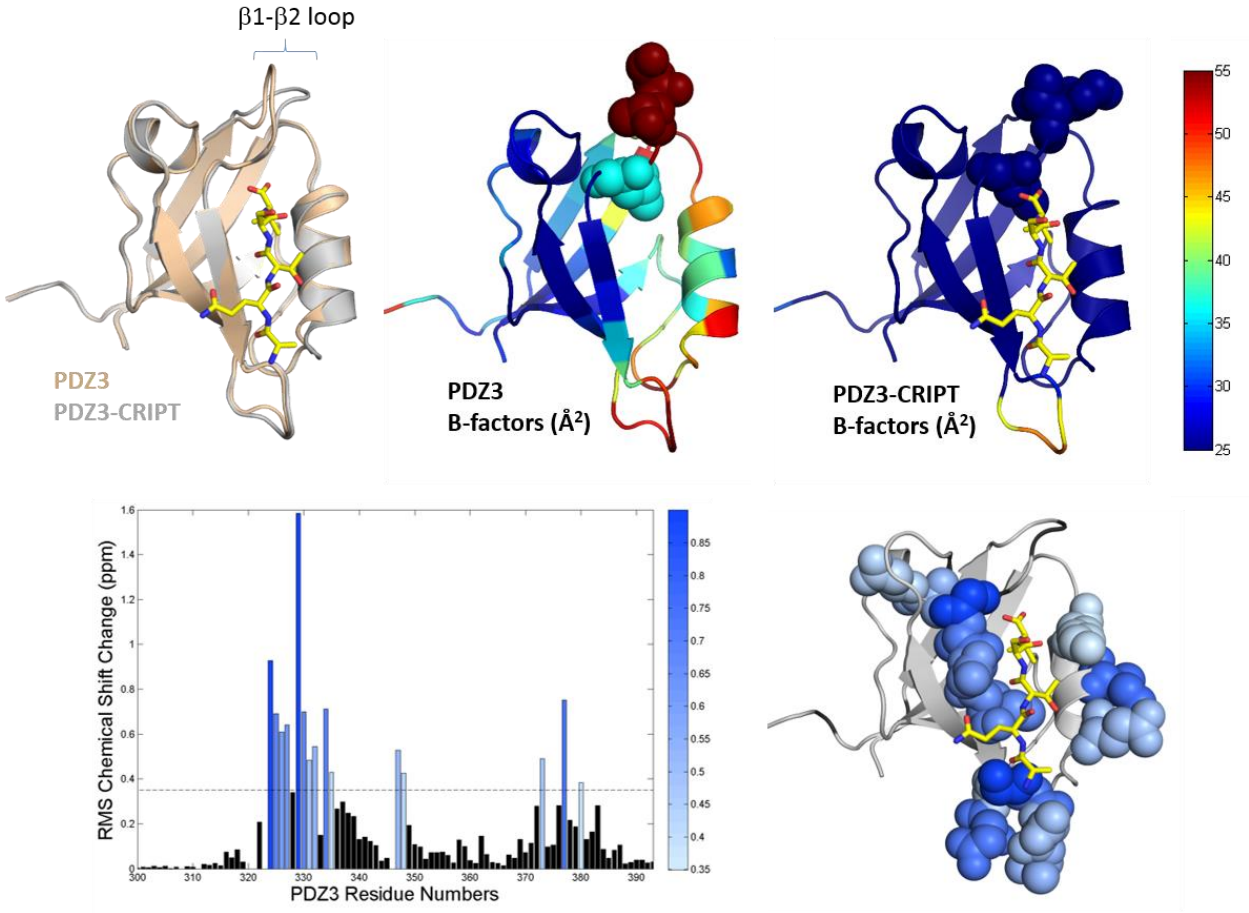


Figure 3-5: Structural and chemical shift perturbation due to peptide binding.
Upper row left: Structural change of PDZ3 upon peptide binding. Unbound PDZ3 is shown in tan color, peptide-bound PDZ3 is shown in grey, CRIPT peptide is shown in yellow, and the B1-B2 loops is indicated to show the clamping effect with peptide-binding. Upper row center: Free PDZ3 colored by B-factors with residues 320, 321, and 323 shown in spheres. The color scheme ranges from 25-55 Å^2. Upper row right: PDZ3 bound to CRIPT peptide colored by B-factors. Again, residues 320, 321, and 323 are shown in spheres. B-factors throughout the molecule are reduced compared to free PDZ3. Lower row left: Bar graph showing chemical shift changes upon peptide binding. Lower row right: Residues for which the RMS chemical shift change is greater than 0.35 ppm are shown as spheres and colored based on magnitude of chemical shift change as illustrated in the bar chart.

The next step in analyzing whether the heterogeneity of physical interactions is related to function is to determine whether the patterns of chemical shift perturbations are different in the functional state – peptide binding. As in the case of the apo protein, there is a wide range in the magnitude of chemical shift changes due to different mutations with some mutations having few effects and some mutations having more numerous and distributed chemical shift changes. In many cases, the pattern of chemical shift perturbations in the presence of peptide is very similar to the pattern in the absence of peptide with only subtle differences. However, in some cases, mutations can have a significantly different effect in the presence of peptide. An example is shown in Figure 3-6 where the I327L mutation has a much smaller perturbation in the presence of peptide than in the free state. The opposite effect is shown in Figure 3-7 and Figure 3-8 where the V328A and G329S mutations cause significantly greater chemical shift changes in the presence of CRIPT peptide. A full gallery of the chemical shift perturbations for each evolutionarily conservative single mutation in PDZ3 bound to CRIPT peptide is shown mapped onto the protein structure in Figure 3-9.

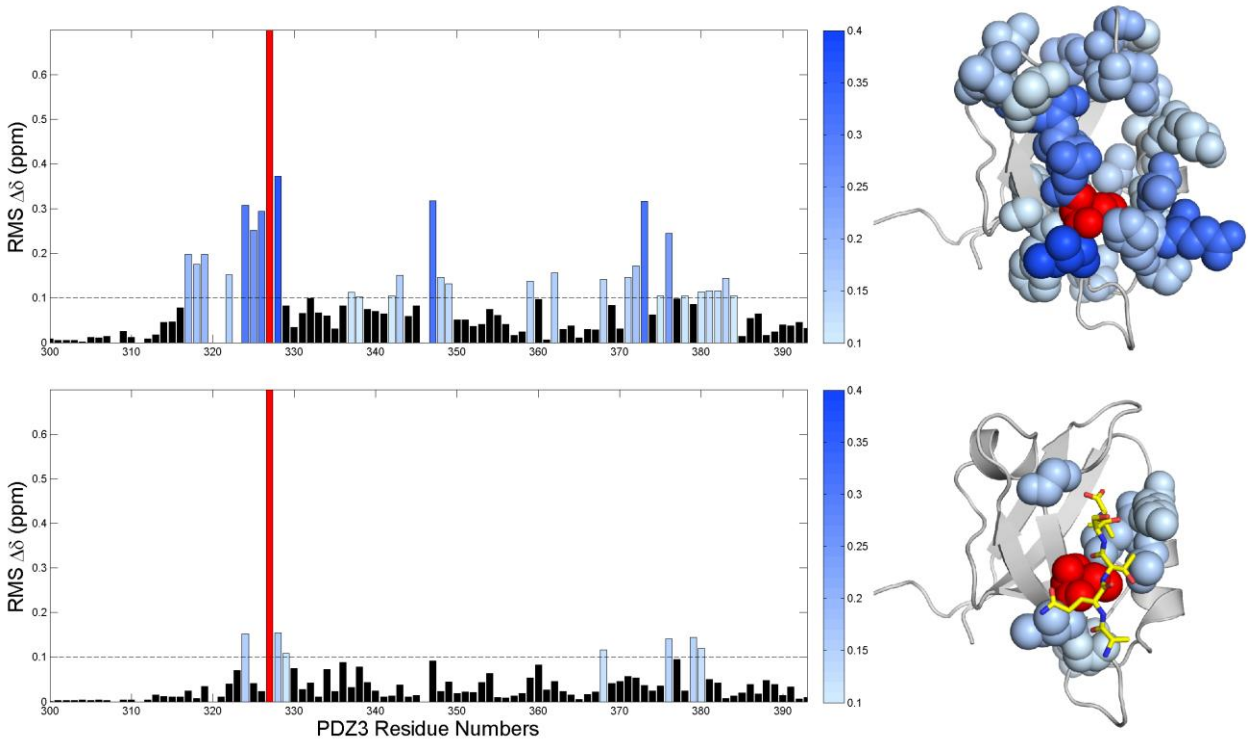


Figure 3-6: I327L mutation in the absence (top) and presence (bottom) of CRIPT peptide.

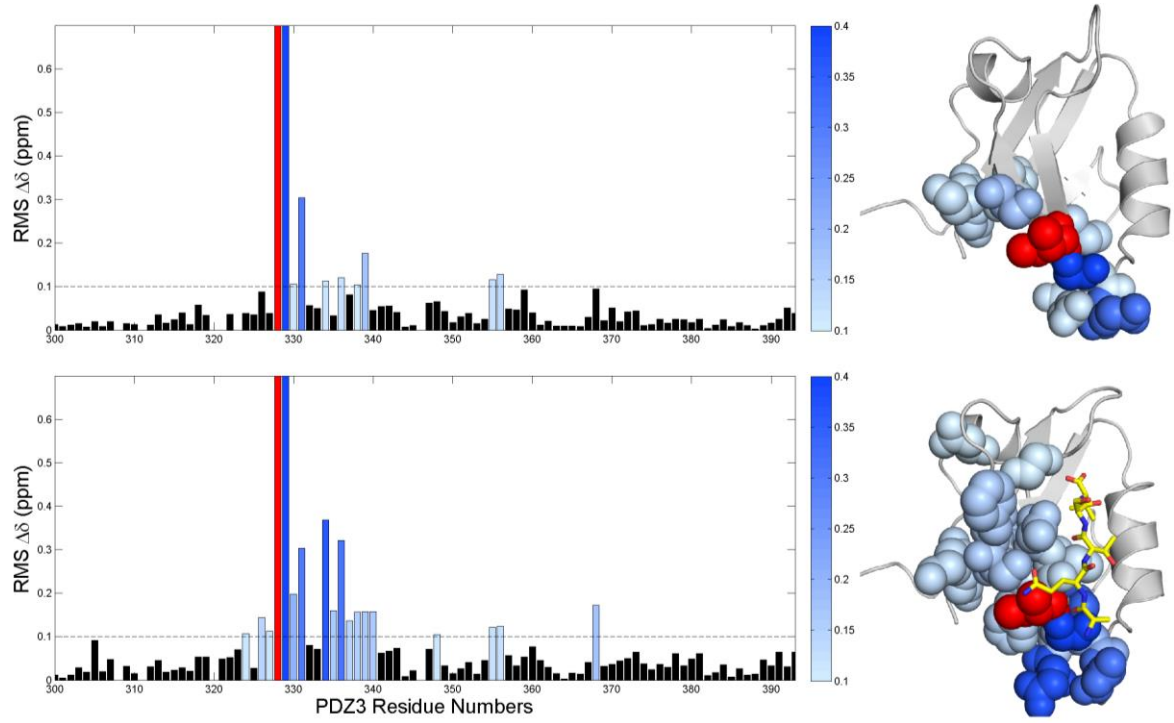


Figure 3-7: V328A mutation in the absence (top) and presence (bottom) of CRIPT peptide. Note that an isoleucine is shown based on the 1BFE and 1BE9 crystal structures, however a valine is present at position 328 in the construct used in this project.

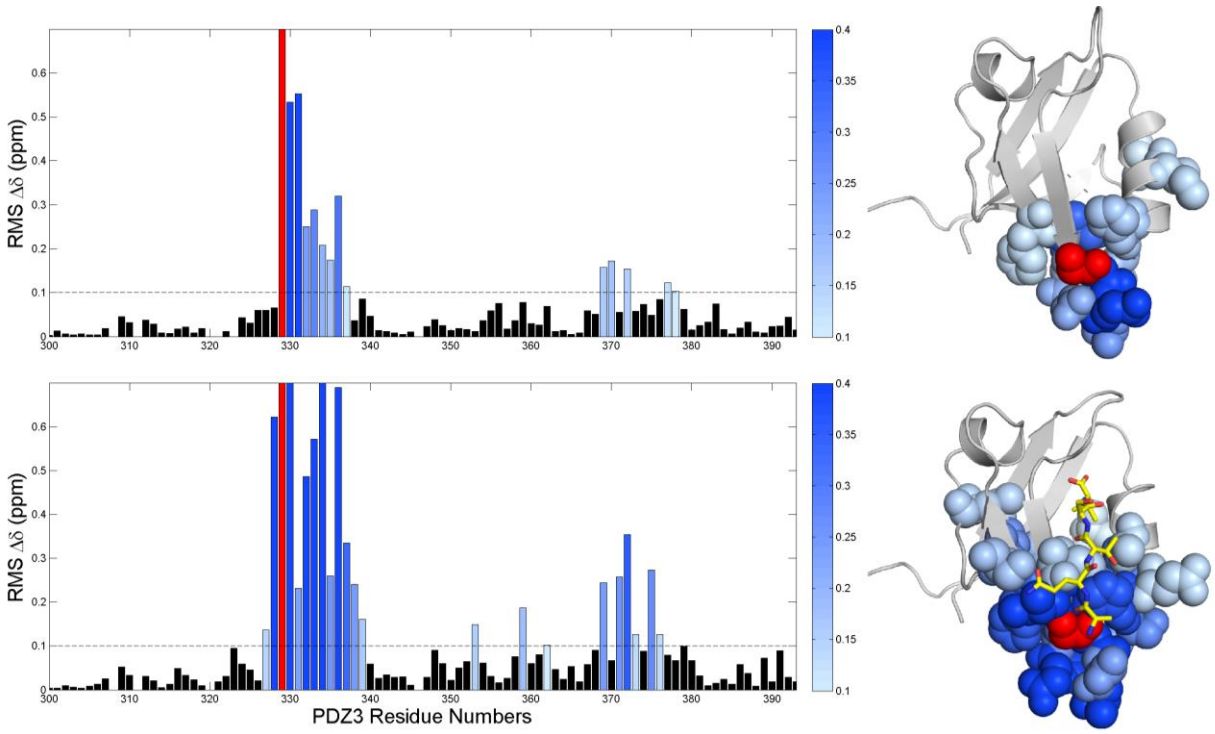


Figure 3-8: G329S mutation in the absence (top) and presence (bottom) of CRIPT peptide.

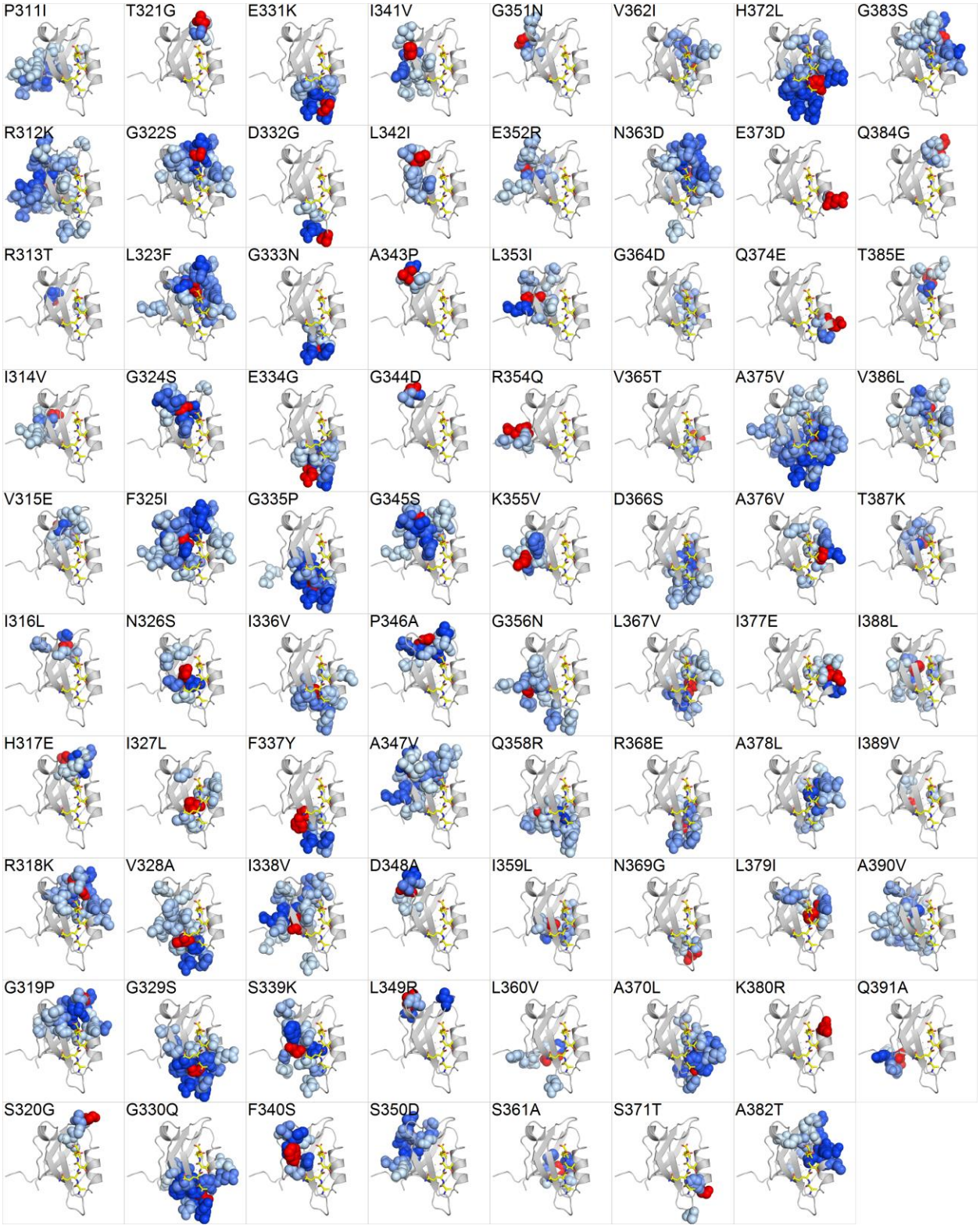


Figure 3-9: Chemical shift perturbations in PDZ3 in the presence of CRIPT peptide.

Evidence of cooperativity – structural modes in free PDZ3

If the architecture of PDZ3 contains cooperative networks of residues, then we would expect that a perturbation to one residue in such a network would be propagated to other coupled residues in that network. This architecture would cause mutations that perturb a cooperative network to have similar patterns of chemical shift perturbation since they are perturbing the same strongly interacting residues. Inversely, if cooperative networks do not exist, then the chemical shift perturbation patterns would be expected to be uncorrelated or, more specifically, to look like the spatial organization of the protein – residues close to the site of mutation are perturbed while residues far from the site of mutation are not.

One way to identify residues that may belong to a coupled network is to find mutations that produce chemical shift perturbation patterns that are similar to each other, but different from other mutations. Principal component analysis (PCA) is a useful way of identifying such patterns in high dimensional datasets by expressing the data in a form where it is easier to visualize the similarities and differences in the data. PCA transforms high dimensional datasets by finding a new coordinate set to capture the maximum amount of variance in the fewest number of orthogonal dimensions. In practice, PCA is applied by projecting a high dimensional dataset onto a lower dimensional set of coordinates while retaining the most possible information about the data. In applying PCA to the chemical shift perturbation data, I treated the rows of the matrix (nuclei where chemical shifts are measured) as independent observations and the columns (mutations) as separate trials. I performed PCA on the chemical shift perturbation matrix with the proton, nitrogen, and carbon chemical shifts separated along the vertical axis and only considered the magnitude of the chemical shift changes and not the sign. I chose to keep the proton, nitrogen, and carbon chemical shifts separate (rather than combining them as in the RMS matrix) because the nuclei are sensitive to different mechanisms of perturbation within the structure. In fact, the relative independence of these nuclei is easily seen in Figure 3-10 which shows scatter plots of chemical shift perturbation at nuclei that are in the same spin system. The proton, nitrogen, and carbon chemical shifts are largely uncorrelated with correlation coefficients of 0.24, 0.23, and 0.28 for the ^{1}H-^{15}N, ^{1}H-^{13}C, and ^{15}N-^{13}C nuclei within the same spin system. I chose to work with the

unsigned chemical shift changes because it was more intuitive to ask whether different mutations perturb the same nucleus rather than try to ask whether different mutations create identical chemical environments at the observed nucleus. Unsigned chemical shift change data also has a distinct advantage in that it is less sensitive to errors in peak matching. Errors usually arise when a spin system is strongly perturbed; if the chemical shift change data is unsigned, then an error might result in the chemical shift change being slightly larger or slightly smaller than the true value but still significantly different from zero. However, if the chemical shift data is signed and a peak matching mistake is made, then the chemical shift change could be a large positive or large negative value which could then be strongly anti-correlated with a similar perturbation at that nucleus.

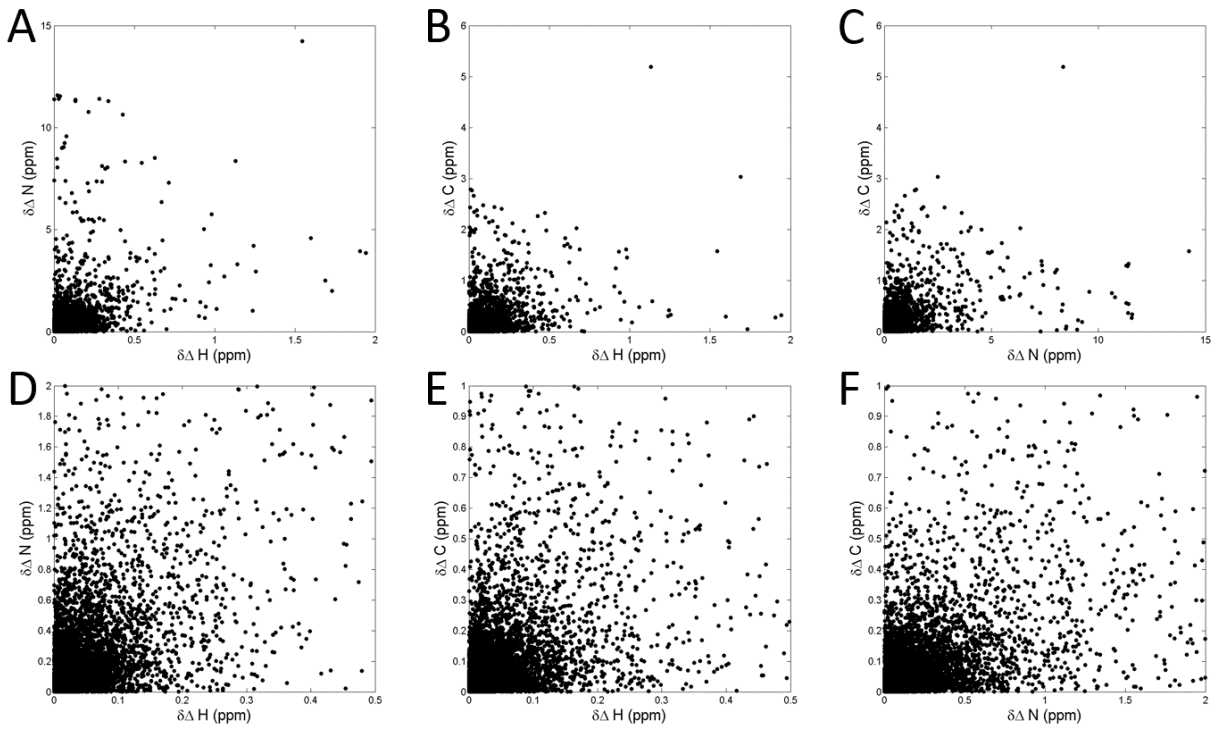


Figure 3-10: Proton, nitrogen, and carbon chemical shifts changes from the same spin system are not strongly correlated.
Panels A-C show scatter plots of chemical shifts for each pair of ^{1}H-^{15}N, ^{1}H-$^{13}C(O)$, and ^{15}N-$^{13}C(O)$ nuclei that are part of the same $C_{i-1}(O)H_{Ni}N_i$ spin systems. Panels D-F show enlargements of the central regions of panels A-C.

PCA involves finding the eigenvectors and eigenvalues of the covariance matrix of a data matrix (in this case, the chemical shift perturbation matrix). The normalized eigenvectors of the covariance matrix are called principal components (PC's), and the corresponding eigenvalues represent weights of how much variation each PC captures of the original data matrix. By examining the PC's corresponding to the largest eigenvalues, we can represent the most variation of our original dataset in the fewest dimensions. The principal components were calculated for the unsigned free chemical shift perturbation matrix, and the amount of variance captured in each principal component is shown in Figure 3-11. A vertical red line is shown to indicate the largest PC weight (on average) from vertically randomized matrices containing the same values as the free chemical shift perturbation matrix. Any eigenvalue larger than this cutoff is very unlikely to occur randomly and represents a significant feature in our dataset. For the free chemical shift perturbation matrix, three PC's are significant (collectively capturing 28% of the variance in the dataset), and all of the mutations are mapped onto these top three PC's in Figure 3-12.

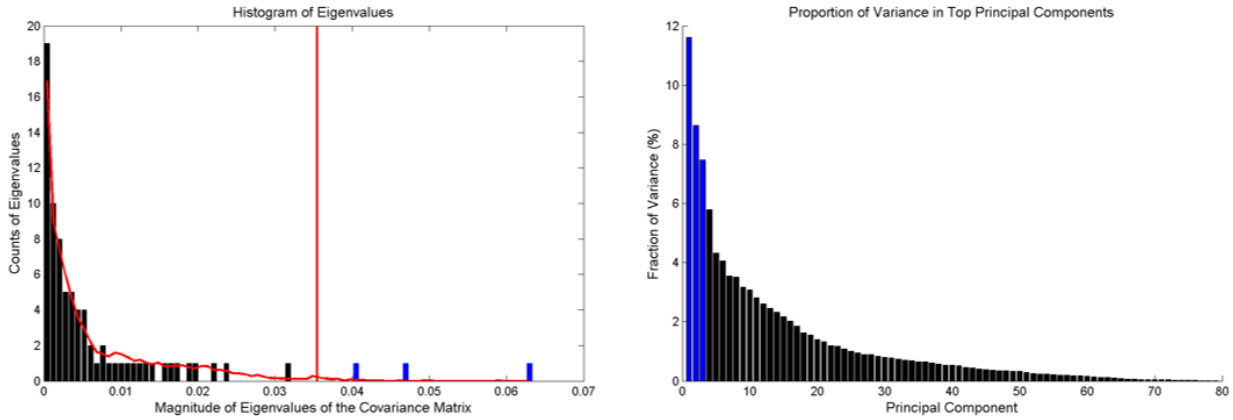


Figure 3-11: The proportion of variance captured in principal components of the free chemical shift perturbation matrix.
Left: A histogram of eigenvalues of the covariance matrix of the unsigned chemical shift perturbation matrix. The red contour represents an average eigenvalue spectrum of a randomized data matrix containing the same values as the unsigned chemical shift perturbation matrix. The vertical red line shows the average magnitude of the largest eigenvalue from such a randomized matrix. The top three eigenvalues are shown in blue. Right: Bar-graph showing the amount of variance captured by each principal component. The top three eigenvalues are shown in blue and collectively capture 28% of the variance in the dataset.

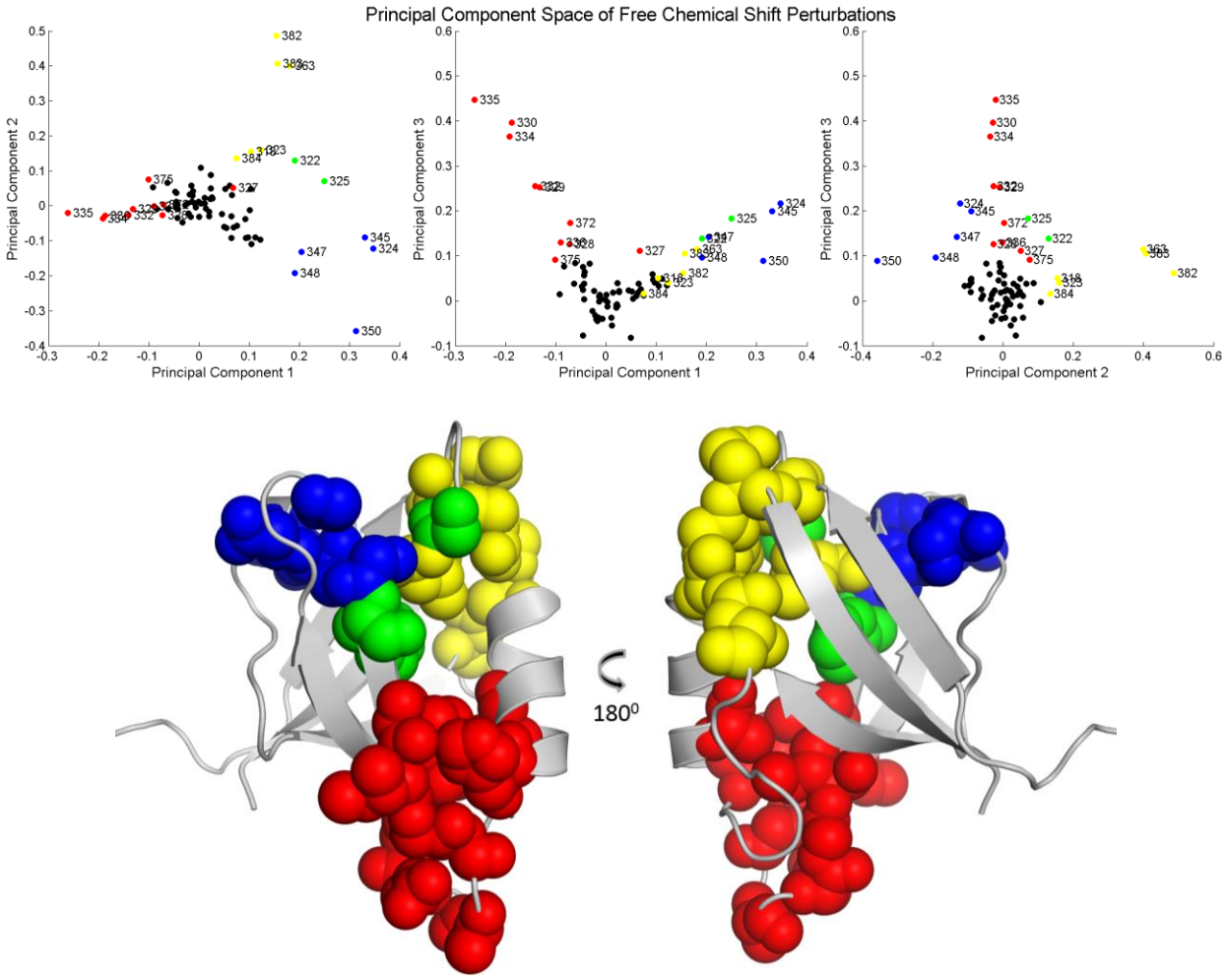


Figure 3-12: Chemical shift perturbation profiles projected in the PC space defined by the top 3 PC's. Structural modes are identified based on similarity of mutation profiles.
Top: Scatter plots of the mutations from the free chemical shift perturbation dataset projected onto the top 3 PC's. The mutations are grouped into three structural modes according to how they cluster in the PC space. Mutations belonging to the three structural modes are colored (1) blue, (2) yellow, and (3) red. Mutations at positions 322 and 325 are shown in green because they have significant projections in the PC space similar to structural modes 1 and 2. Bottom: Residues corresponding to the free structural modes identified by PCA are shown as spheres.

Mutations that are close together in the PC space have similar perturbation patterns. By grouping mutations that have maximal projections along similar axes of variation in the PC space, we can identify mutations that have similar features in their patterns of perturbation. For the free dataset, we find three main groups of mutations that are colored 1) blue – positive

projection along PC 1 and negative projection along PC 2, 2) yellow – positive projections along PC 1 and PC2, and 3) red – positive projection along PC 3 and projection of less than 0.1 along PC 1. In addition, two mutations at positions 322 and 325 are colored green to indicate a strong projection along PC 1, but neither a strong positive or negative projection along PC 2 – suggesting that these mutations share characteristics of the blue and yellow groups.

If I examine the location of these mutations on the structure of PDZ3 (bottom panel of Figure 3-12), I find that each group is physically contiguous within the structure. I call these groupings of mutations structural modes because they elicit a similar structural response within the protein. The blue mode consists of residues in and around the $\alpha 1$ helix while the yellow mode occupies the $\beta 1 - \beta 2$ loop region with the green residues having similar features of both of these two modes. The red mode consists of residues from the $\beta 2 - \beta 3$ loop region on the opposite side of PDZ3. These structural modes were identified by similar patterns of chemical shift change which is shown more clearly in Figure 3-13 where the RMS chemical shift change profile and structure representation of significant chemical shift changes are grouped by structural mode. These representations make it easier to see how the chemical shift perturbation patterns are similar within each structural mode, but largely distinct between structural modes. The similarities within and differences between structural modes suggest that the involved residues are physically coupled to each other through a cooperatively interacting group of residues that interact more strongly within this group than they do with residues outside of this group. In addition, because these structural modes were identified by patterns of maximal variation in all of the chemical shift perturbation data, they represent the most strongly interacting groups of residues in PDZ3.

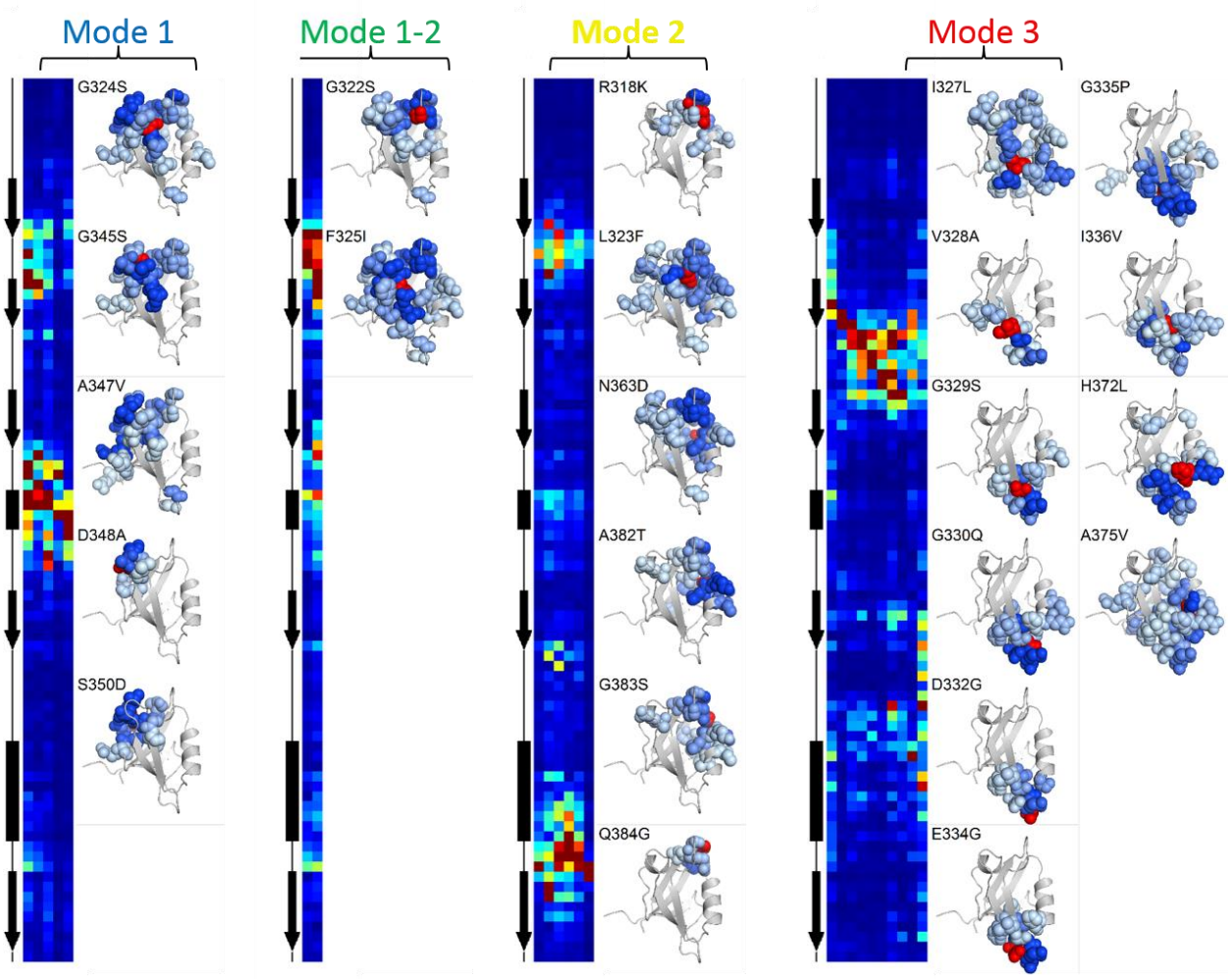


Figure 3-13: Structural modes of free PDZ3.
The perturbation profiles of mutations belonging to each structural modes assigned in Figure 3-12 are shown in RMS matrix format along with a structural representation of each mutation where residues with a chemical shift change > 0.1ppm are depicted as spheres.

Evidence of cooperativity – structural modes in peptide-bound PDZ3

The location of the structural modes is also significant; the residues in these modes are mostly located at the peptide binding interface while the opposite side of the beta sheet contains none of these residues. In fact, many of the residues that make key interactions with the peptide participate in these modes. This finding suggests that these structural modes may be involved in peptide binding, which we investigated by repeating the global perturbation analysis in the presence of peptide.

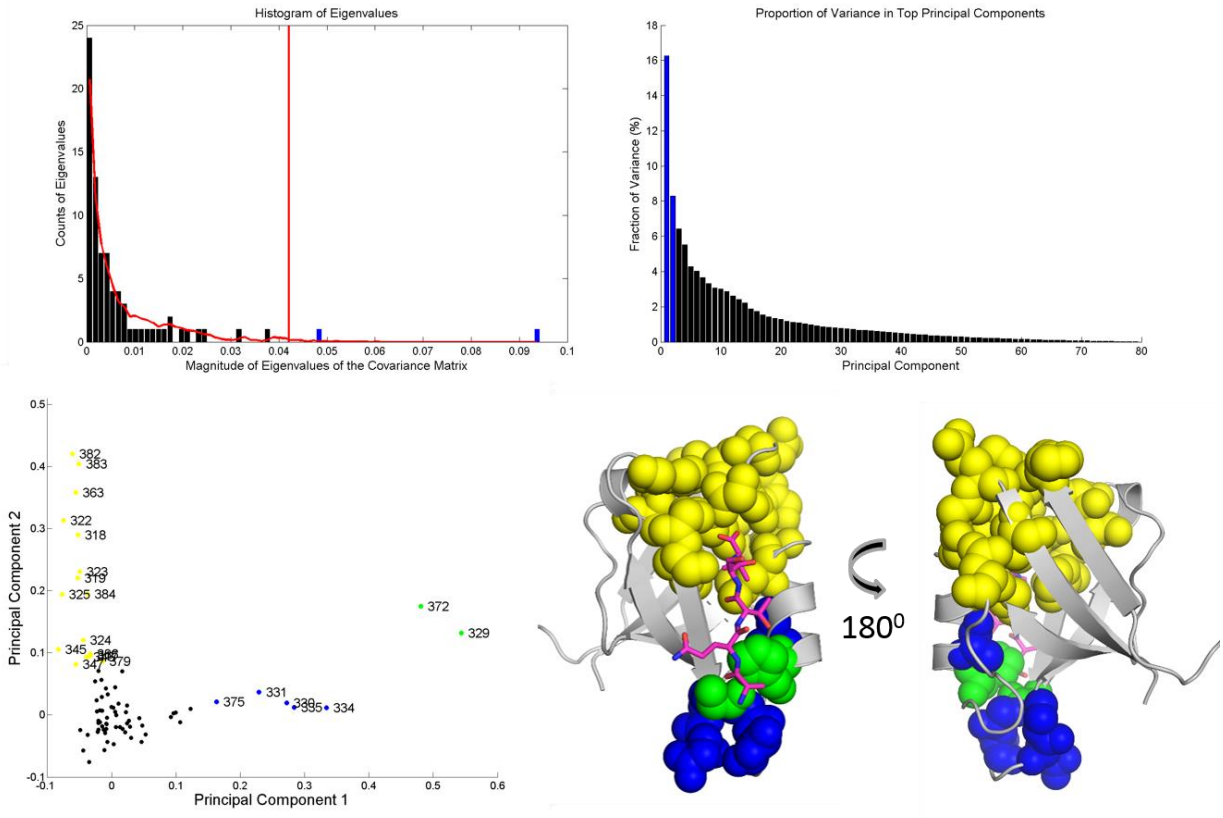


Figure 3-14: PCA of peptide-bound chemical shift perturbation matrix.
Top left: A histogram of eigenvalues of the covariance matrix of the unsigned peptide-bound chemical shift perturbation matrix. The red contour represents an average eigenvalue spectrum of a randomized data matrix containing the same values as the unsigned peptide-bound chemical shift perturbation matrix. The vertical red line shows the average magnitude of the largest eigenvalue from such a randomized matrix. The top two eigenvalues are shown in blue. Top right: Bar-graph showing the amount of variance captured by each principal component. The top two eigenvalues are shown in blue and collectively capture 25% of the variance in the dataset. Bottom left: Peptide-bound chemical shift perturbation patterns are shown in the space defined by the first two PC's. Significant projections along the first and second PC are colored blue and yellow, respectively. Positions 329 and 372 have significant projections along both PC's and are colored green. Bottom right: The structural modes are mapped onto the PDZ3 structure as spheres, and the CRIPT peptide is shown in magenta sticks. Image based on PDB 1BE9.

The result of PCA on the peptide-bound chemical shift perturbation dataset is shown in Figure 3-14. In this case, only two eigenvalues are significant, and thus the mutation data is only mapped onto the top two PC's. Here, there are only two main axes of variation – one with a strong projection along PC 1 and the other with a strong projection along PC 2. Again, there

are two mutations that have strong projections along both main axes, but these two positions are different from the two "bridging" mutations found in the free dataset. The strongest structural mode is now located in the $\beta2–\beta3$ loop while the other significant structural mode is a coalescence of two structural modes found the in the free protein – the $\alpha1$ helix + the $\beta1–\beta2$ loop + the residues lining the floor of the peptide binding pocket that contact the most C-terminal peptide residue. An interesting feature is that positions 329 and 372 have projections along both of the first two PC's. Although these two mutation have a much stronger projection along the first PC and thus are more similar to mutations in the first structural mode, close inspection does reveal perturbations at position 323 and some positions in the $\alpha2$ helix and $\alpha2$-$\beta6$ loop that are similar to perturbation patterns seen in the second structural mode. The likely source for these features is physical coupling through the peptide which makes important contacts with the base of the $\beta1–\beta2$ loop (position 323) and the $\alpha2$ helix. The perturbation patterns for each mutant belonging to the significant structural modes of peptide-bound PDZ3 are depicted in RMS matrix and in structure representation in Figure 3-15.

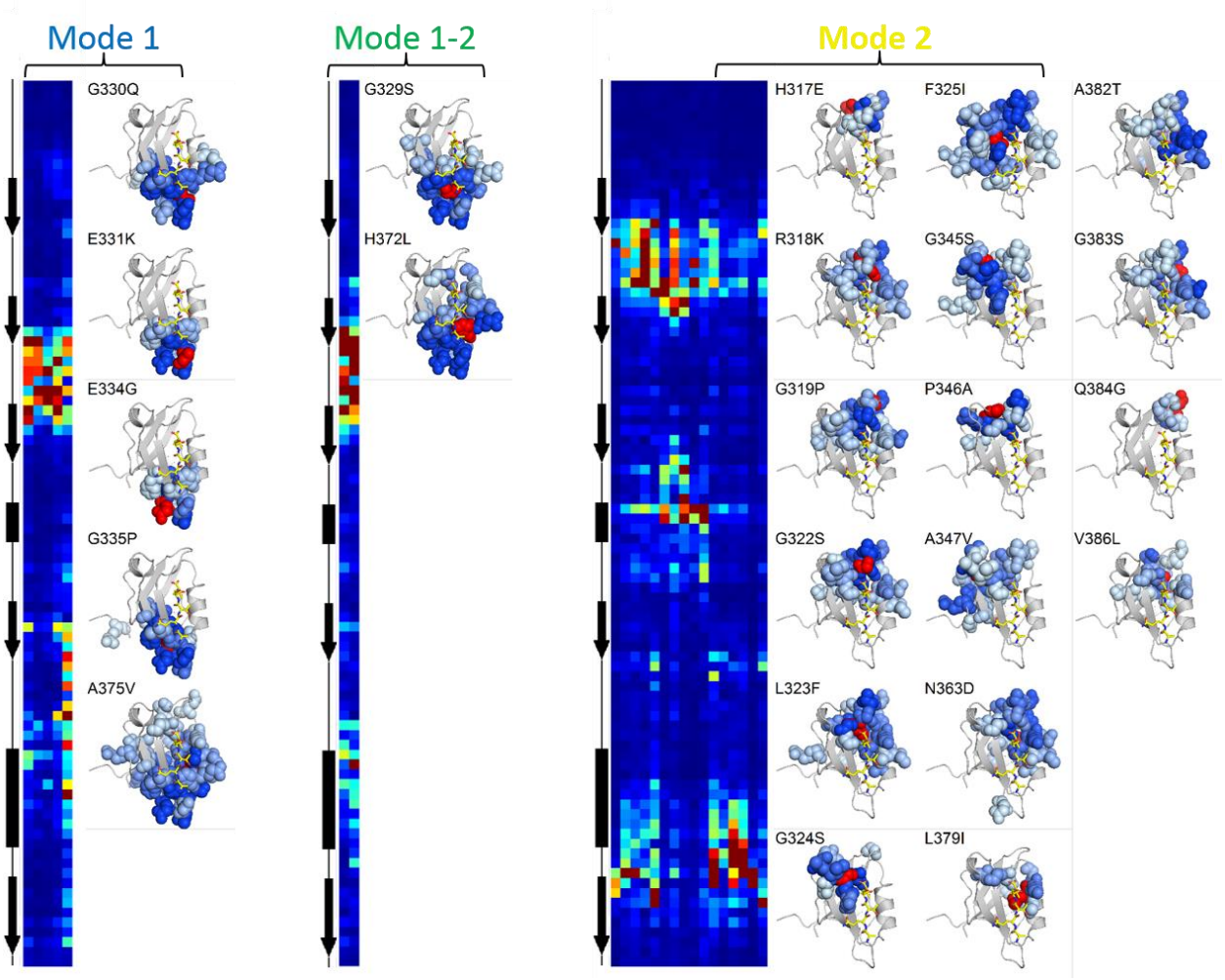


Figure 3-15: Structural modes of peptide-bound PDZ3.
The perturbation profiles of mutations belonging to each structural modes assigned in Figure 3-14 are shown in RMS matrix format along with a structural representation of each mutation where residues with a chemical shift change > 0.1ppm are depicted as spheres.

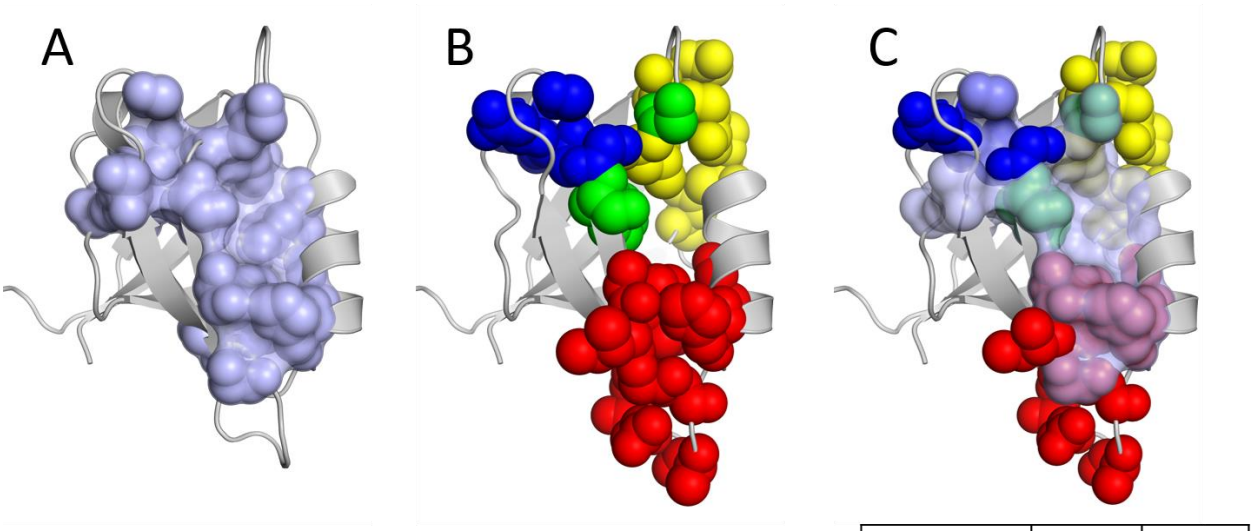
Peptide binding reorganizes the structural modes in PDZ3 by strengthening interactions within the $\beta2$–$\beta3$ loop and between the $\alpha1$ helix and the $\beta1$–$\beta2$ loop. This increase in physical cooperativity is consistent with the overall rigidification of the molecule upon peptide binding as seen by a decrease in B-factors in the crystal structure – especially in the $\beta1$–$\beta2$ loop (see Figure 3-5). This engagement and modulation of the structural modes in the presence of peptide argues that these modes are involved in peptide binding and that cooperative interactions of the involved residues may contribute to the affinity or specificity of the domain toward potential ligands. It is also interesting to note that most of the residues comprising the

structural modes are very similar in the presence and absence of peptide with only 6 out of 23 residues differing between the two sets. This feature suggests that structural modes are both an inherent feature of the protein and important for the function of the PDZ domain.

Structural modes likely contribute to statistical coupling

The global chemical shift perturbation experiment reveals that there is significant heterogeneity of the physical interactions in PDZ3 and that there are groups of residues with cooperative interactions that I describe as structural modes. These finding are consistent with predictions detailed earlier in Chapter 1 based on previous experiments and SCA. I also find that the structural modes appear to be important for peptide binding – the conserved function of PDZ domains and the most obvious evolutionary constraint contributing to statistical coupling in the PDZ family. Thus, I suspect that cooperative physical interactions in PDZ domains are a central mechanism driving statistical coupling in the PDZ family.

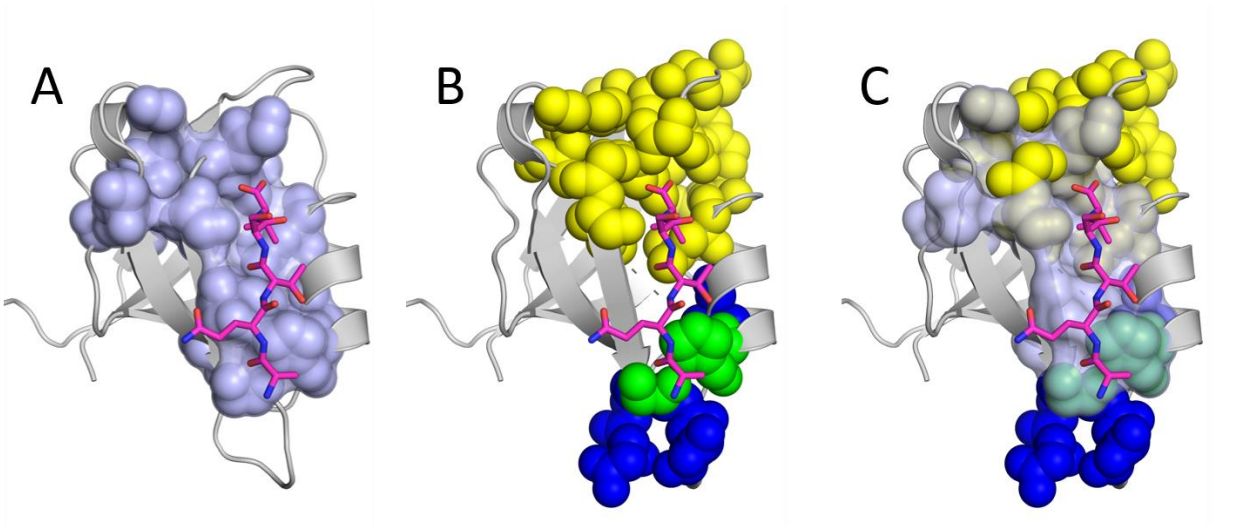
The most obvious way to test this hypothesis is to look for a correlation between the residues that participate in structural modes and the residues that comprise SCA sectors. The simplest way to compare the two sets of residues is to perform a Fisher's exact test to determine whether the two classifiers (structural modes and statistical sectors) are associated. SCA was performed on a 240 sequence alignment of PDZ domains and SCA sector residues were identified as described in the SCA 4.0 toolbox (available from the Ranganathan lab). The 15 residues comprising the SCA sectors are shown in Figure 3-16 (A) as spheres on the structure of PDZ3. For free PDZ3, 23 residues with the largest contributions to the strongest three PC's are considered to represent the structural modes. Figure 3-16 shows the representation of the structural modes and the construction of the 2 x 2 contingency table necessary for calculating the Fisher's exact test. The structural modes and SCA sector are significantly correlated with a p-value of 0.0009, meaning that the overlap between residue sets is statistically significant, and it would be very unlikely to find this level of similarity if these two sets of residues had been selected randomly.



P = 0.0009	SCA Yes	SCA No
Str. Mode Yes	10	13
Str. Mode No	5	51

Figure 3-16 Structural modes in free PDZ3 correlate with PDZ SCA sector.
A) The SCA sector for the PDZ domain family is depicted on PDZ in purple spheres. B) Structural modes are displayed as spheres as in Figure 3.8. C) Overlap of SCA (purple surface) and structural modes.

A similar analysis was performed for the peptide-bound chemical shift perturbation data, and 23 residues with the strongest contributions to the top two structural modes were considered. Figure 3-17 compares the structural modes in the presence of peptide to the SCA sector residues and sets up the 2 x 2 contingency table for Fisher's exact test. Once again, a significant correlation is found between the structural modes and the SCA sector with a p-value of 0.0009. I conclude that the structural modes in both the free and peptide-bound PDZ3 are significantly correlated with the SCA sector identified in the PDZ domain family.



SCA Sector
15 positions

Structural Modes
23 positions

P = 0.0009	SCA Yes	SCA No
Str. Mode Yes	10	13
Str. Mode No	5	51

Figure 3-17: Structural modes in peptide-bound PDZ3 correlate with PDZ SCA sector.
A) SCA sector for the PDZ domain family is depicted on PDZ in pink spheres. B) Overlap of SCA sectors and structural modes – SCA is shown in pink surface. C) Cooperative structural modes are displayed as spheres as in Figure 3.9.

Structural modes in a single protein are not expected to correspond exactly with SCA sectors

Although the association between structural modes in PDZ3 and the SCA sector derived from the MSA of PDZ domains is significant, the correspondence is not perfect. There are some residues in the SCA sector that are not part of structural modes, and there are some residues in the structural modes that are not part of the SCA sector. Both of these differences are to be expected.

In the first case, we find only a minority of SCA residues (3 out of 15) that are not part of either the free or peptide-bound structural modes. SCA sectors are identified by statistical coupling that is subject to all evolutionary pressures on the individuals of a protein family. In PDZ domains, we have evidence to believe that many of the SCA sector positions are important for peptide binding, but other evolutionary pressures could also play a role. Statistical coupling

could arise from pressures to maintain fold stability, from coupling to allow allosteric regulation at sites away from the peptide binding surface, or from cooperativity to perform a function other than canonical peptide binding. Furthermore, these properties that are under evolutionary selection may not be present in all individuals in the protein family. For instance, we generally consider PDZ domains to act as independent protein-protein binding modules, but there are examples such as the Par6 PDZ – CRIB interaction where a PDZ domain is allosterically regulated at a site distant from the peptide binding site [4]. Other examples include the occurrence of tandem PDZ domains in multi-domain proteins where the interaction of the tandem PDZ domains appears to be important for their function [5]. Finally, there are also reports of PDZ domains binding phospholipids at surface locations outside of the peptide binding site [6], but this is not known to be a pervasive property throughout the domain family. Any of these extra functions or regulation could give rise to statistical coupling even if these features are only present in some, but not all members of the PDZ domain family. Thus, some SCA sector residues could be a result of evolutionary pressures that do not constrain PSD95 PDZ3 or are not dependent on cooperative physical interactions, and thus would not participate in structural modes.

The presence of residues in structural modes that are not part of SCA sectors is also completely expected. SCA sectors represent evolutionary constraints that are conserved in a significant number of the members of a protein family. Each individual PDZ domain will have some idiosyncrasies that are not shared by a significant number of other individual domains – this may be partly due to neutral sequence variation and partly due to functional variation such as differences in the specificity preference for target ligands. In addition, many residues that participate in structural modes, but are not found in the SCA sector, are located in the $\beta1-\beta2$ and $\beta2-\beta3$ loops. These loops have been shown to be important for peptide binding, and residues at the base of these loops are key members of the SCA sector. However the number of residues in the loops varies significantly in the MSA of the PDZ domain family as does the identity of the residues. This lack of amino acid conservation and inability to align the residues lead to a lack of covariation and statistical coupling in addition to any inherent lack of functional specificity. Unsurprisingly though, the loops tend to move as a mechanical unit, and

perturbations that affect one residue in the loop tend to affect many residues in the loop. Thus, although these loops act as mechanically coupled and functionally important units, they are not strong features of SCA. However, residues at the bases of loops do tend to show strong coevolution in SCA suggesting that these could be key residues for controlling their spatial orientation and/or conformational dynamics. One final discrepancy between the SCA and the structural modes has a likely explanation. Position 324 participates in structural modes of both free and peptide-bound PDZ3. This position is so strongly conserved in the 240 sequence PDZ MSA (96% glycine) that covariation with other residues is impossible to detect. Thus, it is not part of the SCA sector, despite its likeliness to be functionally important since it has extreme sequence conservation, it coordinates the terminal carboxylate of the peptide ligand, and it anchors the $\beta1-\beta2$ loop that clamps down with peptide binding. If position 324 were considered to be part of the SCA sector, the p-values for the correlation between structural modes and SCA sectors would improve to 0.00026 for both the free and peptide-bound cases.

Conclusions

A global perturbation analysis of PSD95 PDZ3 has revealed significant information about the architecture of this PDZ domain. By observing chemical shift changes as a result of mutation at all positions in the protein, I was able to record the physical interactions between all pairs of residues. The chemical shift perturbation patterns resulting from mutation are strongly heterogeneous with respect to the number of perturbed nuclei and to the anisotropic nature of how perturbations are propagated through the structure. This heterogeneity gives rise to structural modes in the protein consisting of groups of residues with strongly cooperative interactions. These structural modes occur at functionally important sites in the PDZ domain, exist in the free state, and are engaged in and subtly reorganized in the presence of peptide ligand.

When these structural modes are compared to SCA sector residues identified by an evolutionary analysis of the PDZ domain family, a significant correlation exists between the residues that comprise the structural modes and the SCA sectors. This correlation is present in the free PDZ3 domain and persists when the protein is bound to the CRIPT peptide with a

subtle reorganization of the involved residues. These results argue that structural modes in individual PDZ domains may contribute significantly to the statistical coupling found in the PDZ domain family.

At first glance, this may not seem like a particularly profound statement, but I would argue that it augments a hypothesis about the fundamental nature of proteins in a way that has never been possible. As I detailed in the introductory chapter, the scientific community has come to realize that proteins do not behave as a homogeneous collection of atoms. However, it was not until the advent of SCA and other evolutionary covariation methods that we could explain and more importantly, predict some of the heterogeneous phenomena seen in proteins. The hypothesis that subsets of cooperatively-interacting, co-evolving residues mediate protein function while many other protein sites are only weakly coupled and easily tolerate mutations represents a fundamental change in the way proteins are understood, and it has significant implications for both basic and applied biological science.

When SCA was first published, the observation that coevolving residues formed a contiguous set of residues in the protein structure led to the statement that "Sets of interacting residues form connected pathways through the protein fold that may be the basis for efficient energy conduction within proteins" [7]. Although in multiple model systems, statistically coupled residues have been confirmed to be energetically coupled and functionally important, it has not been possible to show the physical mechanism underlying the statistical coupling. As in the statement above, the hypothesis was always that "Sets of [physically] interacting residues …" create statistical coupling, but in over a decade following the original paper, no experiment has been able to provide direct *physical evidence* to support this statement.

This global perturbation analysis in a PDZ domain does not necessarily prove the statement either, but it does offer an unbiased interrogation of physical interactions in a protein that are very clearly consistent with an overall heterogeneous architecture and with the pattern of statistical coupling seen in PDZ domain. I would argue that this is currently the best set of physical data with which to vet the degree to which the patterns of strong and weak physical interactions in a protein correspond to statistical coupling in that protein family. I do not believe that it is the definitive and last experiment to be done concerning this hypothesis,

but I do believe that it represents a step forward in our understanding of fundamental design principles of proteins and to some degree validates the hypothesized mechanism underlying SCA. As I will discuss in the following section, I think that the results of this project will give direction to further experiments, may aid in the conceptual and practical advancement of computer simulations and models, and I hope, will inspire creative ways to apply these concepts to solve important problems in biology and medicine.

Methods

Chemical Shift Difference Measurement

Mutant peak lists were assigned via methods described in Chapter 2. The chemical shift difference was then determined for each nucleus by subtracting the WT chemical shift from the mutant chemical shift. Due to the significant difference in the ppm range of chemical shifts for ^{1}H, ^{15}N, and ^{13}C nuclei, the chemical shifts for ^{15}N and ^{13}C nuclei are scaled by previously determined scaling factors: $\sigma_N = 0.17$ and $\sigma_C = 0.39$ as is customary in the literature. To simplify visualization of the chemical shift changes, proton, nitrogen, and carbon chemical shifts of the same spin system ($^{13}C_{i-1}(O)^{1}H_{Ni}{}^{15}N_i$) are combined by calculating the root-mean-square distance of the three chemical shifts: $\delta\Delta = \mathrm{sqrt}(\Delta H^2 + \Delta N^2 + \Delta C^2)$. The RMS chemical shift changes are used to generate figures showing residues perturbed by each mutation or by peptide binding. The RMS chemical shift changes are not used for the principle component analysis.

Principal Component Analysis

As described in the text, PCA was applied to the matrix of all free or peptide-bound chemical shift changes where columns represent each mutation and rows represent the nucleus being observed. For the PCA analysis, nuclei from the same spin systems are kept separate as independent observations (ie. RMS combination was not performed). Chemical shifts of ^{15}N and ^{13}C nuclei are scaled as above, and the absolute value of chemical shift changes was used for all calculations. PCA was performed in Matlab using standard functions. A covariation matrix of the chemical shift perturbation matrix was calculated, and eigenvectors and eigenvalues of the covariation matrix were determined. The number of significant eigenvectors to consider was determined by calculating eigenvalues from randomized matrices that contain the same values as the chemical shift matrix, but vertically randomized within each column. Eigenvalues were calculated from 100 such randomized matrices, and the mean of the largest eigenvalue from each matrix was calculated. Any eigenvalue from the chemical shift perturbation matrix greater than the mean of the largest eigenvalues from the randomized

matrices was considered to be significant. The choice of structural mode assignment was subjective based on the clustering of mutations in the principal component analysis.

Statistical Coupling Analysis

SCA was performed on the 240 sequence PDZ alignment provided in Appendix 1: using the SCA 4.0 toolbox available from the Ranganathan Lab and described elsewhere [8].

Structure Images

All structures shown in figures were generated using the PyMOL Molecular Graphics System [9]. PDB files 1BFE and 1BE9 were used to generate figures without and with CRIPT peptide ligand respectively.

References

1. Sanchez, I.E., et al., *Point Mutations in Protein Globular Domains: Contributions from Function, Stability and Misfolding.* Journal of Molecular Biology, 2006. **363**(2): p. 422-432.
2. Sadovsky, E. and O. Yifrach, *Principles underlying energetic coupling along an allosteric communication trajectory of a voltage-activated K+ channel.* Proceedings of the National Academy of Sciences of the United States of America, 2007. **104**(50): p. 19813-19818.
3. Farmer, B.T., 2nd, et al., *Localizing the NADP+ binding site on the MurB enzyme by NMR.* Nat Struct Biol, 1996. **3**(12): p. 995-7.
4. Peterson, F.C., et al., *Cdc42 regulates the Par-6 PDZ domain through an allosteric CRIB-PDZ transition.* Molecular Cell, 2004. **13**(5): p. 665-676.
5. Hillier, B.J., et al., *Unexpected modes of PDZ domain scaffolding revealed by structure of nNOS-syntrophin complex.* Science, 1999. **284**(5415): p. 812-5.
6. Gallardo, R., et al., *Structural diversity of PDZ-lipid interactions.* ChemBioChem, 2010. **11**(4): p. 456-67.
7. Lockless, S.W. and R. Ranganathan, *Evolutionarily conserved pathways of energetic connectivity in protein families.* Science, 1999. **286**(5438): p. 295-9.
8. Halabi, N., et al., *Protein sectors: evolutionary units of three-dimensional structure.* Cell, 2009. **138**(4): p. 774-86.
9. Schrodinger, LLC, *The PyMOL Molecular Graphics System, Version 1.3r1*. 2010.

FUTURE DIRECTIONS

The experiments in this dissertation serve a dual purpose. The first involves refining our understanding of the general architecture of proteins – what are the general principles connecting the structural features of proteins to their functional and evolutionary properties. The second goal is much more focused; it is to identify a physical mechanism mediating the statistical covariation observed between residues in a protein via SCA. I think this project has succeeded on both fronts; yet despite being a "global" test, I see it as a starting point in both fields.

The results of this global chemical shift perturbation analysis provide a very visual representation of the heterogeneity of strong and weak physical interactions within a protein and support the hypothesis that a heterogeneous architecture exists and is important for protein function. While this project provides novel and independent support of this hypothesis, a single experiment cannot lay claim to the generality of this feature for all proteins. Generalizing this hypothesis will require data on the physical interactions within multiple proteins and multiple protein families. Given the labor-intensive nature of this experiment, I doubt that it will be used in its current form to characterize proteins *en masse*. It does however, conceptually bridge the gap between what I expect to be the two most common and most fruitful future lines of research in this area.

First, I expect that current leap forward in DNA sequencing technologies will promote many large-scale thermodynamic mutant cycle-style experiments where the assay will involve cell selection or cell sorting and the quantification will be sequencing counts. These experiments have the potential to not only thoroughly explore the sequence space surrounding proteins, they will also have the power to interrogate higher order coupling between residues and thoroughly test the hypothesis that high-level cooperativity is present and important for function. Although design of my perturbation experiment does not allow for the direct interrogation of cooperativity between residues, it appears likely that the observed structural modes are a reflection of residue cooperativity, and I expect that future sequence-based experiments will provide a definitive answer. Actually, these sequence-based experiments are

already underway in our labs and others, and I think that creativity in system and assay design along with intellectual focus in posing questions will determine who makes the most of these new capabilities.

The second line of research, I believe, is still some years away. Ultimately, I think that computer simulations are the only way to massively and broadly test hypotheses regarding the architecture of proteins and the detailed mechanisms of function. Currently, however, I have not seen evidence that any computational approach is sufficiently accurate to blindly trust its results. I do believe, though, that my collection of chemical shift perturbations may offer a unique dataset on which to test and develop simulations of proteins that are near their ground state. Combined with functional data from Richard McLaughlin in our lab, we have a comprehensive dataset of mutations linked to highly precise structural information (chemical shifts), functional effects (peptide affinities), and thermodynamic stability that could be useful to anyone developing equilibrium or non-equilibrium simulation methods. Personally, it is my intuition that refining structural predictions and simulations against chemical shift information may prove to be more productive than crystal structure coordinates because the chemical shifts may be more precise, more sensitive, and more reflective of the physical conditions in solution.

Our lab has long argued that one advantage of studying statistical coupling is that it is sensitive to many evolutionary pressures. Although we have now definitively seen that different evolutionary pressures can give rise to separate SCA protein sectors [1], I also suspect that multiple factors may contribute to what appear to be single sectors. The deconvolution of factors influencing statistical coupling is very valuable because it would allow us to better predict protein properties from sequence data and potentially allow for the design of protein sequences with precisely defined properties. This dissertation project provides evidence that physical interactions contribute to SCA sectors, but it is still unclear as to what quantity of SCA information is accounted for by this mechanism. In addition, it is not known how much of the physical interactions and patterns of heterogeneity are dictated by the conserved tertiary structure of the protein family rather than the specific sequence-function relationship of individual proteins. While I think that it will be difficult to determine all of the contributions to

statistical coupling, (especially since we are not even sure what the full set of evolutionary pressures to consider is) I think that it should be possible to better determine the influence of functional constraints on the physical interactions in proteins independent of structure. One approach to this question would be to start with a protein family and evolve (under random mutagenesis) two independent sets of protein sequences based on different functional constraints. One set would be required to remain folded and thermodynamically stable while the other set would need to retain a function (such as peptide binding) without any explicit stability requirement. With a sufficiently large and diverse set of sequences and long evolution times, SCA could be performed on the two sets of sequences separately and the statistical coupling necessary for folding/stability versus function should be visible. This type of information could make it easier to interpret the results of SCA as well as aid in choosing SCA information to use for downstream projects such as synthetic protein design and screening for allosteric regulators of protein function.

The design of novel proteins based on SCA information was initially very successful [2], and although there have been some promising results, it has proven to be much more difficult to extend this methodology to larger protein systems [unpublished data, Ranganathan Lab]. I think much of this difficulty in larger protein design is due to the fact that information obtained from SCA may be a convolution of multiple evolutionary pressures regarding the function of the protein as well as constraints from the tertiary structure, idiosyncratic couplings due to incomplete divergence of naturally extant sequences, and individual sequence optimization for thermal stability. Any effort to deconvolve SCA information could bring useful advances to clinically significant biological efforts including a better understanding of clinically significant genetic sequence variations and the design of protein therapeutics with more favorable properties such as higher binding affinities, increased stability, or decreased off-target effects.

At the conclusion of this work, I have developed a novel methodology intended to address a very specific (and I feel very important) question in biology. I do not expect this method to become a common investigation of structural biology, and in fact, I would not be surprised if this experiment is never repeated in this format. However, I do think that I have provided strong physical evidence to support a very important hypothesis regarding a general

property of proteins and protein evolution. I have argued that proteins have purposefully evolved a heterogeneous physical architecture (that is not detectable by visual inspection or most other experiments to date) that is essential for protein function and necessary for the protein to remain robust and evolvable. I have presented a completely unbiased interrogation of a model protein system that shows physical heterogeneity that corresponds (with high statistical significance) to the coevolution revealed by SCA. This result should give us confidence in the power of SCA to reveal meaningful and useful information about the physical and functional properties of proteins. In addition, I am all the more encouraged to design future experiments to dissect the information provided by SCA and to further look for ways to apply SCA to meaningful problems in biology. I hope that my work will encourage other investigators and scientists to do the same.

Missed Opportunities

At the conclusion of my data collection, and in the process of analyzing data and preparing reports, I had several ideas about ways to improve the experiments that I was not able to implement. The first and most interesting idea would have been to look at the chemical shift information of the peptide itself. In unpublished data from the Ranganathan lab, we have been able to investigate SCA for alignments of PDZ domains with known target ligands, enabling us to look at statistical coupling between residues in the PDZ domain and residues in the peptide. The data showed interesting patterns of statistical coupling with some residues in the peptide being highly coupled to specific residues in the PDZ domain. I believe that it would have been feasible to produce ^{13}C- and ^{15}N-labeled peptide, and that I could have used this labeled peptide to obtain the chemical shifts of the peptide residues in the presence of each PDZ mutation. These chemical shifts could be compared to the peptide chemical shifts in the presence of WT PDZ3 to determine interactions between the PDZ residues and the peptide residues. Although another member of the lab has already determined the peptide binding constant for each mutant, this experiment may have been able to provide higher resolution information about how each mutation changes PDZ3's interactions with the peptide and how residues in PDZ3 interact through residues in the CRIPT peptide. In addition, one could

compare the chemical shift change patterns of the peptide for each mutant to the statistical couplings of PDZ domains with target peptides. Confirming coevolution between domains and between proteins has been a focus of the lab, and it would be attractive to have physical evidence showing this interaction as well.

I also would have made another small modification to my data acquisition scheme to reduce any possible errors in my dataset. I collected HN(CA)CO spectra and made explicit peak assignments for every mutant for which there was a discrepancy between the numbers of observed and expected peaks. In retrospect, however, I would also like to have obtained HN(CA)CO spectra for the other few datasets which had a large number of significant chemical shift perturbations. As shown in Figure 2-6, spectra with larger number of chemical shift changes were more prone to assignment errors. However, as the data stands, I do not think the results of conclusions would change significantly by assigning more spectra. I have shown that the vast majority of spectra are likely to have very few errors, and I have taken steps to mitigate any existing assignment errors by using the absolute value of the chemical shift changes in my analysis which prevents strong anti-correlations that could arise from peak assignment errors. As I stated earlier, though, the choice and number of NMR experiments was limited by available spectrometer time, and many decisions were made to optimize the utility and quality of the massive amount of NMR data that was collected.

Finally, I will comment on the selection of NMR experiment. When the project was initially conceived, I obtained two dimensional ^{1}H-^{15}N HSQC spectra, but quickly realized that the peak density in the spectra was too great to permit detailed comparisons between WT and mutant spectra. I then switched to HNCO spectra to obtain an added dimension of peak separation, but also with the knowledge that an HNCO spectrum has a very good signal-to-noise ratio, has good dispersion in the $^{13}C(O)$ dimension, and also has one peak per residue which facilitates use of the projection-reconstruction method. In retrospect, however, I wonder if an HNCA experiment would have been feasible. HNCA spectra have a slightly lower signal-to-noise ratio, but probably still sufficient for my experiment. Also, HNCA spectra include the Cα resonance from the *i* and *i-1* residues giving two peaks per residue which may have been sufficient to determine explicit peak assignments for each mutant with a single spectrum. The

major drawback, however, would be that each spectrum would be twice as crowded as an HNCO spectrum, which may have been difficult to resolve with a projection reconstruction experiment and may have required significantly longer data acquisition times. The HNCA spectrum would also have provided different data – Cα resonances rather than C(O) resonances – and I do not know whether that information would have been more or less useful.

References

1. Halabi, N., et al., *Protein sectors: evolutionary units of three-dimensional structure.* Cell, 2009. **138**(4): p. 774-86.
2. Reynolds, K.A., et al., *Evolution-based design of proteins.* Methods Enzymol, 2013. **523**: p. 213-35.

Appendix 1: Multiple Sequence Alignment of the PDZ domain family

GI Accession #	PDZ Domain Protein Sequence
627585	YSFVTEENTFEVKLFK----NSSGL-GFSFSREDNL-------IPEQINASIVRVKKLFPG-QPAAESG-KIDVGDVILKVNG----ASLKG-LS--QQEVISALRGT-------APEVFLLLCRPPPG
515031b	IVSSPEREITLVNLKKD---AKYGL-GFQIIGGEKMGRL----------DLGIFISSVAPG-GPADFHG-CLKPGDRLISVNS----VSLEG-VS--HHAAIEILQNA-------PEDVTLVISQPKEK
4507137	LPEALLLQRRRVTVRKA---DAGGL-GISIKGGRENK-------------MPILISKIFKG-LAADQTE-ALFVGDAILSVNG----EDLSS-AT--HDEAVQVLKKT-------GKEVVLEVKYMKDV
4502129	IHFSKSENCKDVFIEKQ---KGEIL-GVVIVESGWGS-----------ILPTVIIANMMHG-GPAEKSG-KLNIGDQIMSING----TSLVG-LP--LSTCQSIIKGLE-----NQSRVKLNIVRCPPV
3127043	EATLKQLDSIHVTILHKE--EGAGL-GFSLAGGAD------------LENKVITVHRVFPN-GLASQEG-TIQKGNEVLSING----KSLKG-AT--HNDALAILRQA------RDPRQAVIVTRRTTV
266646	GVQQIQPNVISVRLFKR---KVGGL-GFLVKERV--------------SKPPVIISDLIRG-GAAEQSG-LIQAGDIILAVND----RPLVD-LS--YDSALEVLRGI-------ASETHVVLILRGPE
2351794	GVQQIQPNVISVRLFKR---KVGGL-GFLVKERV--------------SKPPVIISDLIRG-GAAEQSG-LIQAGDIILAVNG----EDLSS-AT--HDEAVQVLKKT-------GKEVVLEVKYMKDV
2134506	VESSAEATVYTVTLEK----MSAGL-GFSLEGGKGS----------LHGDKPLTINRIFKG-AASEQSE-TIQPGDEILQLAG----TAMQG-LT--RFEAWNTIKAL------PDGPVTIVIRRKSLQ
3953613	IHFSNSENCKELQLEKH---KGEIL-GVVVVESGWGS-----------ILPTVILANMMNG-GPAARSG-KLSIGDQIMSING----TSLVG-LP--LATCQGIIKGLK-----NQTQVKLNIVSCPPV
1706530	TDSTMSLNIITVTLNME---KYNFL-GISIVGQSNER-----------GDGGIYIGSIMKG-GAVAADG-RIEPGDMLLQVND----INFEN-MS--NDDAVRVLRDIV----HKPGPIVLTVAKCWDP
1588680	LPEALLLQRRRVTVRKA---DAGGL-GISIKGGRENK-------------MPILISKIFKG--LMDQTE-ALFVGDAILSVNG----EDLSS-AT--HDEAVQVLKKT-------GKEVVLEVKYMKDV
1486367	LEDFELEVELLITLIKS---EKGSL-GFTVTKGNQ-------------RIGCYVHDVIQD--PAKSDG-RLKPGDRLIKVND----TDVTN-MT--HTDAVNLLRG--------SKTVRLVIGRVLEL
1401051	TDSTMSLNIITVTLNME---KYNFL-GISIVGQSNER-----------GDGGIYIGSIMKG-GAVAADG-RIEPGDMLLQVND----MNFEN-MS--NDDAVRVLRDIV----HKPGPIVLTVAKCWGP
1256761a	RAISLEGEPRKVVLHK----GSTGL-GFNIVGGEDGE-------------GIFVSFI------ADLSG-ELQRRKQILSVNG----IHLPG-DS--HEQALP-LKGA-------GQTVTIIAQYQPED
2947232b	GFASHSLQTSDVVIHRK---ENEGF-GFVIISSLNRPES------GSTITVXHKIGRIIDG-SPADRCA-KLKVGDRILAVNG----QSIIN-MP--HADIVKLIKDA-------GLSVTLRIIPQEEL
3878084d	LNSAGPSGSYDVLLHRN---ENDGF-GFVLMSSQHKN-------------GSTVGQIQPG-SPAARCG-RLSVGDRVIAVNG----IDILS-LS--HPDTISLIKDS-------GLSVRLTIAPPNTA
4995819	LYSDCIIEDKTVVLQKK---DNEGF-GFVLRGAKADTPIEEFT-PTPAFPALQYLESVDEG-GVAWQAG--LRTGDFLIEVNN----ENVVK-VG--HRQVVNMIRQG-------GNHLVLKVVTVTRN
1890856b	AGQVVHTETTEVVLTAD---PVTGF-GIQLQGSVFATET---------LSSPPLISYIEAD-SPAERCG-VLQIGDRVMAING----IPTED-ST--FEEANQLLRDS-----SITSKVTLEIEFDVAE
1498137a	VGDRTTWEYHTVAVTRV---PGYGF-GIAVSGGRDNPHF--------ANGDPSIAVSDVLKG-GPAEDRL---QVNDRIISVNG----VSLEN-VE--YATAVQVLRDS-------GNTVQLVVKRRVPL
3033501b	MEELTIWEQHTATLCRD---PRRGF-GIAISGGRD------------RASGSVVVSDVVPG-GPA--DG-RLQTGDHVVMVNG----VSMES-VT--STFAIQILKTC-------TKLANITVKRPRKI
3875228	NWESTKFELIDVALHRD---PALGL-GITVAGYVHKKE----------EIGGIFVKSLVPR-SAASSSG-VIKVHDLILEVNG----TTLEH-MS--HADSVRTLVKS-------GDQVKLKLVRFPLS
5453992	IVSSPEREITLVNLKKD---AKYGL-GFQIIGGEKMGRL----------DLGIFISSVAPG-GPADLDG-CLKPGDRLISVNS----VSLEG-VS--HHAAIEILQNA-------PEDVTLVISQPKEK
915210c	IASSPEREITLVNLKKD---AQYGL-GFQIIGGEKMGRL----------DLGVFISSVTPG-GPADLDG-CLKPGDRLISVNS----VSLEG-VS--HHAAVEILQNA-------PEDVTLVISQPKEK
1232104	IVSSPEREITLVNLKKD---PKHGL-GFQIIGGEKMGRL----------DLGVFISAVTPG-GPADLDG-CLKPGDRLISVNS----VSLEG-VS--HHAAVDILQNA-------PEDVTLVISQPKEK
3880014	QESVPLEALTVVEIEK----TSKGF-GFNIVGGTDN--------PHFVGDIGIYVSSV--N-SESKSYG-VVRTGDKILSFDG----IDMTY-KT--HDEAVEVFRSV-----KIGHVAKMLIDREYLH
3874621	AAGHETNIARILVIPR----GVKGF-GFILRGAKHVAMPLNFE-PTAQVPALQFFEGVDMS-GMAVRAG--LRPGDYLLEIDG----IDVRR-CS--HDEVVEFIQQA-------GDTITLKVITVDVA
1176422	STKNRWQLVGPVHMTR----GEGGF-GFTLRGDSPV-----------------LIAAVVPG-GQAESAG--LKEGDYIVSVNG-----QPCKWWK--HLEVVTQLRSM------GEEGVSLQVVVTQAL
2702347a	ENRLPDYQEQDIFLWR----KETGF-GFRILGGNEP-------------GEPIYIGHIVPL-GAADTDG-RLRSGDELICVDG----TPVIG-KS--HQLVVQLMQQA-----AKQGHVNLTVRRKVVF
4838485	DSSGPDYKELDVHLRR----MESGF-GFRILGGDEP-------------GQPILIGAVIAM-GSADRDG-RLHPGDELVYVDG----IPVAG-KT--HRYVIDLMHHA-----ARNGQVNLTVRRKVLC
3878084b	QYNQKPSDLITVSLIR----KPVGF-GFRLLGGVES-------------KTPLSVGQIVIG-GAAEEDG-RLQEGDEIVEIDG----HNVEG-AS--HSEAVVLLEAAA-----QNKHVKLIVRRPSRT
2702347b	PQAAQEQDFYTVELER----GAKGF-GFSLRGGREY-------------NMDLYVLRLAED-GPAERCG-KMRIGDEILEING----ETTKN-MK--HSRAIELIKNG-------GRRVRLFLRRGDGS
2947252	YRQPQDFDYFTVDMEK----GAKGF-GFSIRGGREY-------------KMDLYVLRLAED-GPAIRNG-RMRVGDQIIEING----ESTRD-MT--HARAIELIKSG-------GRRVRLLLKRGTGS
3878084c	DRMSMNGNLIDVTLER----GTKGF-GFSIRGGQEFG------------SMPLFVLRIADD-GPAKADG-RLQVGDQLTTING----QSTKG-MS--HDDAIRIIKQH--------TMVNLTVLRNRLP
2702347	SSIATQPELITVHIVK----GPMGF-GFTIADSPGGGGQR--------------VKQIVD--SPRCRG---LKEGDLIVEVNK----KNVQA-LT--HNQVVDMLIECP-----KGSEVTLLVQRGGLP
2947232	SSGATQAELMTLTIVK----GAQGF-GFTIADSPTGQR---------------VKQILD--IQGCPG---LCEGDLIVEINQ----QNVQN-LS--HTEVVDILKDCP-----IGSETSLIIHRGGFF
3878084	YAAAKSRDLHEIDIFK----GSEGF-GFTIADNLNGQR---------------IKKILFP-SQCPN----LMEGDTIVELDG----RNVRP-IP--HTQLVDMLRERP-----IGYRGKLVVKRGSPK
3874215	AASSSTAPSKTITIRK----GPFGF-GFTLKSVRVYLGE-----HSEYYTIEHIVTAVVEG-SPAFHAN--LQAEDMITHVNG----HPVHN-LT--HPQLMHRLLAN-------GNELILRLVPLANT
2695620	NPSELKGKFIHTKLRK----SSRGF-GFTVVGGDEP-------------DEFLQIKSLVLD-GPAALDG-KMETGDVIVSVND----TCVLG-HT--HAQVVKIFQSIP-----IGASVDLELCRGYPL
3327224	DASQLKGTFLSTTLKK----SNMGF-GFTIIGGDEP-------------DEFLQVKSVIPD-GPAAQDG-KMETGDVIVYINE----VCVLG-HT--HADVVKLFQSVP-----IGQSVNLVLCRGYPL
3878084a	DPARLGGELISTKIVK----GAKGL-GFTLIGNDSSS-----------RGDEFIQVKSVLSG-GPAAANG-VLRSGDILVRVNG----RLLLG-AT--QKEACDVFVAIP------VNEAVDIQVCRGYEL
3876327	PQIIFNPRPHVVKVVK----SETGF-GFNVKGQVSEGGQLRSL-NGQLYAPLQHVSAVLRR-GAADQAG--LRKGDRILEVNG----LNVEG-ST--HRKVVDLIKNG-------GDELTMIVISVEDP
1203931	ETFLENATRQVVIVKK----PDSGF-GLSIKGGSENAQN-----------MPIVISKIFKG-LPADECG-ELFIGDAIVEVNG----ISIEG-QS--HDEVVNMLKSS-------GDQVTLGVRHFTHM
4506509	LPGPSPPRVRSVEVAR----GRAGY-GFTLSGQAPC-----------------VLSCVMRG-SPADFVG--LRAGDQILAVNE----INVKK-AS--HEDVVKLIGKC-------SGVLHMVIAEGVGR
2500169	QSGPAPPRVRSVEVAR----GRAGY-GFTLSGQAPC-----------------VLSCVMRG-SPADFVG--LRAGDQILAINE----INVKK-AS--HEDVVKLIGKC-------SGVLRMVISEGSSH
4504703	---LLPHQPRIVEMKK----GSNGY-GFYLRAGSE--------------QKGQIIKDIDSG-SPAEEAG--LKNNDLVVAVNG----ESVET-LD--HDSVVEMIRKG-------GDQTSLLVVDKETD
2224573	SCQIIPPAPRKVEMRRD---PVLGF-GFVAGSEKPV-----------------VVRSVTPG-GPSE--G-KLIPGDQIVMIND----EPVSA-AP--RERVIDLVRSC-------KESILLTVIQPYPS
3123565	RYADLPGELHIIELEK----DKNGL-GLSLAGNKDRS------------RMSIFVVGINPE-GPAAADG-RMRIGDELLEINN----QILYG-RS--HQNASAIIKTA-------PSKVKLVFIVQSLG
2959979c	RYGTLTGQLHMIELEK----GHSGL-GLSLAGNKDRT-----------RMSVFIVGIDPT-GAAGRDG-RLQIADELLEING----QILYG-RS--HQNASSIIKCA-------PSKVKIIFIRNADH
3123565a	TCPIVPGQEMIIEISK----GRSGL-GLSIVGGKDTPLN------------AIVIHEVYEE-GAAARDG-RLWAGDQILEVNG----VDLRN-SS--HEEAITALRQT-------PQKVRLVVYRDEAH
5031715	IFAHVKGQRKEVEVFK----SEDAL-GLTITDN---------------GAGYAFIKRIKEG-SVIDHIH-LISVGDMIEAING----QSLLG-CR--HYEVARLLKELP-----RGRTFTLKLTEPRKA
2462851	EKRVERLELFPVELEK----DSEGL-GISIIGMGAGAD-------MGLEKLGIFVKTVTEG-GAAHRDG-RIQVNDLLVEVDG----TSLVG-VT--QSFAASVLRNT-------KGRVRFMIGRERPG
2623757	EKRVEKLELFPVELEK----DEDGL-GISIIGMGVGAD-------AGLEKLGIFVKTVTEG-GAAQRDG-RIQVNDQIVEVDG----ISLVG-VT--QNFAATVLRNT-------KGNVRFVIGREKPG
1703566	ERRLERMDLFEVDLEK----GAEGL-GVSIIGMGVGAD-------SGLEKLGIFVKSITPG-GAVHRDG-RIRVCDQIVSVDG----KSLVG-VS--QLYAANTLRST-------SNRVTFTIGREQNL
1094005	YSFVTEDNTFEVKLFK----NSSGL-GFSFSREDNL-------IPEQINGSIVRVKKLFPG-QPAAESG-KIDVGDVILKVNG----APLKG-LS--QQDVISALRGT-------APEVSLLLCRPAPG

```
515031a         YSFVTEENTFEVKLFK----NSSGL-GFSFSREDNL-------IPEQINASIVRVKKLFAG-QPAAESG-KIDVGDVILKVNG----ASLKG-LS--QQEVISALRGT-------APEVFLLLCRPPPG
915210b         YSFVTEENTFEVKLLK----NSSGL-GFSFSREDNV-------IPEQMNTSIVRVKKLFPG-QPAAESG-QIDVGDVILKVNG----ASLKG-LS--QQEVISALRGT-------SPEVSLLLCRPPPG
886895          TTALLLKIIFEVKLFK----NSSGL-GFSFSREDNL-------IPEQINGSIVRVKKLFPG-QPAAESG-KIDVGDVILKVNG----APLKG-LS--QQDVISALRGT-------APEVSLLLCRPAPG
2959979b        PLAMWEAGIQAIELEK----GSRGL-GFSILDYQDP---------IDPANTVIVIRSLVPG-GIAEKDG-RLFPGDRLMFVND----INLEN-ST--LEEAVEALKGA------PSGMVRIGVAKPLPL
3875228a        GLAVWNCVPLVIHLCK----DSRGL-GFSIVDYKDP---------THRDESVIVVQSLVPG-GVAQADG-RVVPGDRLLFVNN----HDLSN-SR--HPVPLQVRKLC-------GLVQLNNIESFIL
5031791         ELALWSPEVKIVELVK----DCKGL-GFSILDYQDP---------LDPTRSVIVIRSLVAD-GVAERSG-GLLPGDRLVSVNE----YRLDN-TS--LAEAVEILKAV------PPGLVHLGICKPLVE
627585a         SSPPKPGDIFEVELAK----NDNSL-GISVTVLFDKGG-----VNTSVRHGGIYVKAVIPQ-GAAESDG-RIHKGDRVLAVNG----VSLEG-AT--HKQAVETLRNT-------GQVVHLLLEKGQSP
5453992a        SSPPKPGDIFEVELAK----NDNSL-GISVTGG----------VNTSVRHGGIYVKAVIPQ-GAAESDG-RIHKGDRVLAVNG----VSLEG-AT--HKQAVETLRNT-------GQVVHLLLEKGQSP
915210          SSPPKPGDIFEVELAK----NDNSL-GISVTGG----------VNTSVRHGGIYVKAVIPK-GAAESDG-RIHKGDRVLAVNG----VSLEG-AT--HKQAVETLRNT-------GQVVHLLLEKGQSP
2118000         ASPPKPGDTKEVELAK----TDGSL-GISVTGG----------VNTSVRHGGIYVKAIIPK-GAAESDG-RIHKGDRVLAVNG----VSLEG-AT--HKQAVETLRNT-------GQVVHLLLEKGQVP
4759306         AASEGHSHPRVVELPK----TDEGL-GFNVMGGKEQ-------------NSPIYISRIIPG-GVAERHG-GLKRGDQLLSVNG----VSVEG-EH--HEKAVELLKAA-------KDSVKLVVRYTPKV
2623836         AASEGHAHPRVIELPK----TNEGL-GFNVMGGKEQ-------------NSPIYISRMXPG-GVADRHG-GLKRGDQLLSVNG----ISVES-EH--HERAVELLKLA-------QGTVKLVVRYTPRI
1685067         AAAEGHAHPRVIELPK----TDQGL-GFNVMGGKEQ-------------NSPIYISRIIPG-GVADRHG-GLKRGDQLIAVNG-----NVEA-EC--HEKAVDLLKSA-------VGSVKLVIRYMPKL
1478493         VESTAEATVCTVTLEK----MSAGL-GFSLEGGKGS-----------LHGDKPLTINRIFKG-AASEQSE-TVQPGDEILQLGG----TAMQG-LT--RFEAWNIIKAL------PDGPVTIVIRRKSLQ
3127039         VDSTAEATVCTVTLEK----MSGGL-GFSLEGGKGS-----------LQGDKPLTINRIFKG-AASEQSE-TVQPGDEILHLAG----TAMQG-LT--RFEAWNIIKAL------PDGPVTIVIKRKSMQ
3127037         VESSAEATVYTVTLEK----MSAGL-GFSLEGGKGS-----------LHGDKPLTINRIFKG-AASEQSE-TIQPGDEILQLAG----TAMQG-LT--RFEAWNIIKAL------PDGPVTTVIRRKSLQ
2735710         SGDSTEATVHTVTLEK----TSAGL-GFSLEGGKGS-----------LLGDKPLTVNRIFKG-AASEQSE-TIQPGDEILHLAG----TAVQG-LT--RFEAWNVIKTL------PDGPVTIVIRRRSVQ
2224541         GRSVAVHDALCVEVLK----TSAGL-GLSLDGGKSS-----------VTGDGPLVIKRVYKG-GAAEQAG-IIEAGDEILAING----KPLVG-LM--HFDAWNIMKSV------PEGPVQLLIRKHRNS
5174575         QPLRKEPEIITVTLKK-----QNGM-GLSIVAAKG-----------AGQDKLGIYVKSVVKG-GAADVDG-RLAAGDQLLSVDG----RSLVG-LS--QERAAELMTRT-------SSVVTLEVAKQGAI
2555013         QPLRKEPEVITVTLKK-----QNGM-GLSIVAAKG-----------AGQDKLGIYVKSVVKG-GAADVDG-RLAAGDQLLSVDG----RSLVG-LS--QERAAELMTRT-------SSVVTLEVAKQGAI
1362604         SNKLPQPELQLIKLHK----NSNGM-GLSIVASKG-----------AGQEKLGIYIKSVVPG-GAADADG-RLQAGDQLLRVDG----QSLIG-IT--QERAADYLVRT-------GPVVSLEVAKQGAI
1517938         AEEDFTREPRKIILHK----GSTGL-GFNIVGGEDGE-------------GIFVSFILAG-GPADLSG-ELRRGDRILSVNG----VNLRN-AT--HEQAAAALKRA-------GQSVTIVAQYRPEE
424013          GEEDIPREPRRIVIHR----GSTGL-GFNIVGGEDGE-------------GIFISFILAG-GPADLSG-ELRKGDQILSVNG----VDLRN-AS--HEQAAIALKNA-------GQTVTIIAQYKPEE
2497505         GDDEITREPRKVVLHR----GSTGL-GFNIVGGEDGE-------------GIFISFILAG-GPADLSG-ELRKGDRIISVNS----VDLRA-AS--HEQAAAALKNA-------GQAVTIVAQYRPEE
2228746a        GDDEITREPRKVVLHR----GSTGL-GFNIVAGEDGE-------------GIFISFILAG-GPADLSG-ELRKGDRIISVNS----VDLRA-AS--HEQAAAALKNA-------GQAVTIVAQYRPEE
1517940         RAISLEGEPRKVVLHK----GSTGL-GFNIVGGEDGE-------------GIFVSFILAG-GPADLSG-ELQRGDQILSVNG----IDLRG-AS--HEQAAAALKGA-------GQTVTIIAQYQPED
1049459b        APIAIPLEPRPVQLVK----GQNGL-GFNIVGGEDNE-------------PIYISFVLPG-GVADLSG-NVKTGDVLLEVNG----VVLRN-AT--HKEAAEALRNA-------GNPVYLTLQYRPQE
399393b         STEDITREPRTITIQK----GPQGL-GFNIVGGEDGQ-------------GIYVSFILAG-GPADLGS-ELKRGDQLLSVNN----VNLTH-AT--HEEAAQALKTS-------GGVVTLLAQYRPEE
3043690         KDRPYVEEPRHVKVQK----GSEPL-GISIVSGEK---------------GGIYVSKVTVG-SIAHQAG--LEYGDQLLEFNG----INLRS-AT--EQQARLIIGQ---------QCDTITILAQYNP
3875228e        VVFLPTHLERTVKLQK----GALPL-GAVLDGDKDKGV------------NGCVVKSICGK-KAVALDG-RIQVGDFITKINT----ESLRN-VT--NSQARAILKRTN----LVGTFCNVTYITSADA
2370149         GSDSSLFETYNVELVRK---DGQSL-GIRIVGYVG--------TSHTGEASGIYVKSIIPG-SAAYHNG-HIQVNDKIVAVDG----VNIQG-FA--NHDVVEVLRNA-------GQVVHLTLVRRKTS
2959979a        TQKNEESETFDVELTK----NVQGL-GITIAGYIG---------DKKLEPSGIFVKSITKS-SAVELDG-RIQIGDQIVAVDG----TNLQG-FT--NQQAVEVLRHT-------GQTVRLTLMRKGAS
1504000         TDSTMSLNIITVTLNME---KYNFL-GISIVGQSNER-----------GDGGIYIGSIMKG-GAVAADG-RIEPGDMLLQVNE----INFEN-MS--NDDAVRVLREIV----HKPGPITLTVAKCWDP
4758216         TDSTMSLNIITVTLNME---KYNFL-GISIVGQSNER-----------GDGGIYIGSIMKG-GAVAADG-RIEPGDMLLQVND----MNFEN-MS--NDDAVRVLRDIV----HKPGPIVLTVAKCWDP
4758218         SDSAMSLNIITVTLNME---KYNFL-GISIVGQSNER-----------GDGGIYIGSIMKG-GAVAADG-RIEPGDMLLQVKE----INFEN-MS--NDDAVRVLREIV----HKPGPITLTVAKCWDP
930347          TDSTMSLNIITVTLNME---RHHFL-GISIVGQSNDR-----------GDGGIYIGSIMKG-GAVAADG-RIEPGDMLLQVND----VNFEN-MS--NDDAVRVLREIV----SQTGPISLTVAKCWDP
516485          TDSTMSLNIITVSINME---AVNFL-GISIVGQSNRG-----------GNGGIYVGSIMKG-GAVALDG-RIEPGDMILQVND----VNFEN-MT--NDEAVRVLREVV----QKPGPIKLVVAKCWDP
1199661         --------------NSE---KYNFL-GISIVGQSNER-----------GDGGIYIGSIMKG-GAVAADG-RIEPGDMLLQVNE----INFEN-MS--NDDAVRVLREIV----HKPGPITLTVAKCWDP
1166632         TESSMSLDVITVNLNMD---TVNFL-GISIVGQTSNC-----------GDNGIYVANIMKG-GAVALDG-RIEAGDMILQVNE----TSFEN-FT--NDQAVDVLREAV----SRRGPIKLTVAKSFEN
2104785         YKEEDVCDTFTIELQLQKR-PGKGL-GLSIVGKRN--------------DTGVFVSDIVKG-GIADADG-RLMQGDQILMVNG----EDVRH-AT--QEAVAALLKCS-------LGAVTLEVGRVKAA
2959979g        YKEEDVCDTFTVELQKR---PGKGL-GLSIVGKRN--------------DTGVFVSDIVKG-GIADADG-RLMQGDQILMVNG----EDVRN-AT--QEAVAALLKCS-------LGTVTLEVGRIKAA
3108057b        YRDEENLEVFLVDLQKK---TGRGL-GLSIVGKRS--------------GSGVFISDIVKG-GAADLDG-RLIRGDQILSVNG----EDMRH-AS--QETVATILKCV-------QGLVQLEIGRLRAG
3875228d        LDPTQIYNIFEIDLVKK---TGRGL-GISIVGRKN--------------EPGVYVSEIVKG-GLAESDG-RLMTGDQILEVNG----KDVRG-CM--QEDVAAMLKTI-------TGKVHLKTTENNND
3641615         TAEIKPNKKILIELKV----EKKPM-GVIVCGGKNN-----------HVTTGCVITHVYPE-GQVAADK-RLKIFDHICDINGTP--IHVGS-MT--TLKVHQLFHTT------YEKAVTLTVFRADPP
1418684         IIEVVPGRKIVIEVKT----DKKPL-GVIVVGGKNN-----------YVKTGCIITHIYPE-GVIAEDK-RLKIFDHIIQVNGKE--VQCEA-MT--TLKVHQLFYTL------YEKIVTIQVYRADPP
806292          LEDFELEVELLITLIKS---EKGSL-GFTVTKGNQ-------------RIGCYVHDVIQD--PAKSDG-RLKPGDRLIKVND----TDVTN-MT--HTDAVNLLRAA-------SKTVRLVIGRVLEL
515031          LEDFELEVELLITLIKS---EKASL-GFTVTKGNQ-------------RIGCYVHDVIQD--PAKSDG-RLKPGDRLIKVND----TDVTN-MT--HTDAVNLLRAA-------SKTVRLVIGRVLEL
915210a         LEDFELEVELLITLHKS---EKGSL-GFTVTKGNQ-------------SIGCYVHDVIQD--PAKSDG-RLRPGDRLIKVND----TDVTN-MT--HTDAVNLLRRA-------PRTVRLVLGRVLEL
1094005a        LEDSELEVELLITLVKS---EKGSL-GFTVTKGSQ-------------SIGCYVHDVIQD--PAKGDG-RLKAGDRLIKVND----TDVTN-MT--HTDAVNLLRAA-------PKTVRLVLGRILEL
3641615b        KQGTAGELIHMVTLDKT---GKKSF-GICIVRGEVKDSP-------NTKTTGIFIKGIVPD-SPAHLCG-RLKVGDRILSLNG----KDVRN-ST--EQAVIDLIKEA-------DFKIELEIQTFDKS
1418684c        QLNASSGQVQSVTLDKT---GKKSF-GLSIVRGEARDGS---------NSKGIFIKGIVPD-SPGHLCG-KIKVGDRLLTLNG----KDVRD-AT--EPEVINLIKQA-------GSKIDLELQTYGSE
2104785a        IFPDDLGPPQSKTITLDR--GPDGL-GFSIVGGYG----------SPHGDLPIYVKTVFAK-GAAAEDG-RLKRGDQIIAVNG----QSLEG-VT--HEEAVAILKRT-------KGTVTLMVLSLHDG
1469876         LPPESPGPLRQRHVACLAR-SERGL-GFSIAGGKGS-------TPYRAGDAGIFVSRIAEG-GAAHRAG-TLQVGDRVLSING----VDVTE-AR--HDHAVSLLTAA-------SPTIALLLEREAGG
2497500         VMRRKPPAEKVMEIKLIK--GPKGL-GFSIAGGVGN--------QHIPGDNSIYVTKIIEG-GAAHKDG-RLQIGDKILAVNS----VGLED-VM--HEDAVAALKNT-------YDVVYLKVAKPSNA
297480          VMRRKPPAEKVMEIKLIK--GPKGL-GFSIAGALGT--------SIIPGDNSIYVTKIIEG-GAGHKDG-RLQIGDKILAVNS----VGLED-VM--HEDAVAALKNT-------YDVVYLKVAKPSNA
2497501         VMRRKPPAEKIIEIKLIK--GPKGL-GFSIAGGVGN--------QHIPGDNSIYVTKIIEG-GAAHKDG-RLQIGDKILAVNS----VGLED-VM--HEDAVAALKNT-------YDVVYLKVAKPSNA
2228746         VKRRKLASEKIMEIKLIK--GPKGL-GFSIAGGIGN--------QHIPGDNSIYVTKIIEG-GAAHKDG-KLQIGDKLLAVNS----VCLEE-VT--HEEAVTALKNT-------SDFVYLKVAKPTSM
2497506         VRRRQPPPETIMEVNLLK--GPKGL-GFSIAGGIGN--------QHIPGDNSIYITKIIEG-GAAQKDG-RLQIGDRLLAVNN----TNLQD-VR--HEEAVASLKNT-------SDMVYLKVAKPGSL
2497503         VRRRRPILETVVEIKLFK--GPKGL-GFSIAGGVGN--------QHIPGDNSIYVTKIIDG-GAAQKDG-RLQVGDRLLMVNN----YSLEE-VT--HEEAVAILKNT-------SDVVYLKVGKPTTI
```

```
1256761          ILRRRPILETVVEIKLFK--GPKGL-GFSIAGGVGN--------QHIPGDNSIYVTKIMDG-GAAQKDG-RLQVGDRLLMVNN----YSLEE-VT--HEEAVAILKNT-------SDVVYLKVGKPTTI
2497505a         YVKRRKAFRKNHEIKLIK--GPKGL-GFSIAGGVGN--------QHIPGDNSIYVTKIIEG-GAAHKDG-KLQIGDKLLAVNS----VCLEE-VT--HEEAVTALKNT-------SDFVYLKAAKPTSM
399393           ARDSAASGPKVIEIDLVK--GGKGL-GFSIAGGIGN--------QHIPGDNGIYVTKLTDG-GRAQVDG-RLSIGDKLIAVRTNGSEKNLEN-VT--HELAVATLKSI-------TDKVTLIIGKTQHL
1049459          VGQYRSTRPNTSVIDLVK--GARGL-GFSIAGGQGN--------EHVKGDTDIYVTKIIEE-GAAELDG-RLRVGDKILEVDH----HSLIN-TT--HENAVNVLKNT-------GNRVRLLIQQGTGA
4758162          YVNGTDADYEYEEITLER--GNSGL-GFSIAGGTDN--------PHIGDDSSIFITKIITG-GAAAQDG-RLRVNDCILQVNE----VDVRD-VT--HSKAVEALKEA-------GSIVRLYVKRRKPV
399393a          DSVNGDDSWLYEDIQLER--GNSGL-GFSIAGGTDN--------PHIGTDTSIYITKLISG-GAAAADG-RLSINDIIVSVND----VSVVD-VP--HASAVDALKKA-------GNVVKLHVKRKRGT
1049459a         VIDDHGRKWELENIVLEK--GHTGL-GFSITGGMDQ--------PTEDGDTSIYVTNIIEG-GAALADG-RMRKNDIITAVNN----TNCEN-VK--HEVAVNALKSS-------GNVVSLSLKRRKDE
5031791b         TLPETVCWGHVEEVELIN--DGSGL-GFGIVGGK----------------TSGVVVRTIVPG-GLADRDG-RLQTGDHILKIGG----TNVQG-MT--SEQVAQVLRNC-------GNSVRMLVARDPAG
2959979f         AHSNPTHWQHVETIELVN--DGSGL-GFGIIGGK----------------ATGVIVKTILPG-GVADQHG-RLCSGDHILKIGD----TDLAG-MS--SEQVAQVLRQC-------GNRVKLMIARGAVE
1657758          GQRSMDGYPEQFCVRIE---KNPGL-GFSISGGISGQGNP-----FKPSDKGIFVTRVQPD-GPAS--N-LLQPGDKILQANG----HSFVH-ME--HEKAVLLLKSF-------QNTVDLVIQRELTV
2613002          -VTAVVQRVEIHKLRQG---ENLIL-GFSIGGGIDQDPSQNP-FSEDKTDKGIYVTRVSEG-GPAEIAG--LQIGDKIMQVNG----WDMTM-VT--HDQARKRLTKR------SEEVVRLLVTRQSLQ
1145730          GEAGASPPVRRVRVVKQ---EAGGL-GISIKGGRENR-------------MPILISKIFPG-LAADQSR-ALRLGDAILSVNG----TDLRQ-AT--HDQAVQALKRA-------GKEVLLEVKFIREV
2133805          DAEPGNGCKPTVRIVKQ---EAGGL-GISIKGGRENH-------------MPILISKIFRG-LAAEQSR-LLFVGDAILSVNG----TDLRD-AT--HDQAVQALKKT-------GKEVVLEVKYLKEV
2134823          VPESISNQKRGVKVLKQ---ELGGL-GISIKGGKENK-------------MPILISKIFKG-LAADQTQ-ALYVGDAILSVNG----ADLRD-AT--HDEAVQALKRA-------GKEVLLEVKYMREA
1363070          LPEALLLQRRRVTVRKA---DAGGL-GISIKGGRENK-------------MPILISKIFKG-LAADQTE-ALFVGDAILSVNG----EDLSS-AT--HDEAVQALKKT-------GKEVVLEVKYMKEV
3876554          EQNEAEAEKRTVRVVKY---DGNGL-GISIKGGRDNN-------------MPIVISKIFKG-MAADQAG-ELFLDDVIISVNG----ENLLD-AS--HEEAVRALKRA-------GRVVDLQVQYRRED
1469876a         EPARIEEEELTLTILR----QTGGL-GISIAGGKGS-------TPYKGDDEGIFISRVSEE-GPAARAG--VRVGDKLLEVNG----VALQG-AE--HHEAVEALRGA-------GTAVQMRVWRERMV
5031791a         PNFSHWGPPRIVEIFRE---PNVSL-GISIVGGQTVIKRLKN----GEELKGIFIKQVLED-SPAGKTN-ALKTGDKILEVSG----VDLQN-AS--HSEAVEAIKNA-------GNPVVFIVQSLSST
2959979e         AAYSSWSQPRRVELWRE---PSKSL-GISIVGGRGMGSRLSN----GEVMRGIFIKHVLED-SPAGKNG-TLKPGDRIVEVDG----MDLRD-AS--HEQAVEAIRKA-------GSPVVFMVQSIVNR
3875228c         VRSKYWGEARTVTLVRE---PNKSF-GISIVGGRVEVSQKGGLPGTGNTVCGIFIKSVLPN-SPAGRSG-QMNMGDRVISVND----VDLRD-AT--HEQAVNAIKNA-------SNPVRFVLQSLHTN
3108057a         PQKCTEEEPRTVEIIRE---LSDAL-GISIAGGKGS----------PLGDIPIFIAMIQAN-GVAARTQ-KLKVGDRIVSING----QPLDG-LS--HTDAVNLLKNA-------FGRIILQVVADTNI
226930           KKSQGVGPIRKVLLLKE---DHEGL-GISITGGKEHG-------------VPILISEIHPG-QPADRCG-GLHVGDAILAVNG----VNLRD-TK--HKEAVTILS---------QQRGEIEFEVVYVA
3881858          RKRNGVGKSRKVVLSKH---PHEGL-GISITGGSEHA-------------LPIVISEIQPG-QPADRCG-QVFVGDAILSVNG----YDLRT-VK--HQEAVDILSGQ-------VQQGDLDLELVFVC
1469876b         RRDPAPPGLRELCIQKA---PGERL-GISIRGGARGHAGNP----RDPTDEGIFISKVSPT-GAAGRDG-RLRVGLRLLEVNQ----QSLLG-LT--HGEAVQLLRSV-------GDTLTVLVCDGFEA
2959979d         ALASEIQGLRTVEIKKG---PADAL-GLSIAGGVGS----------PLGDVPIFIAMMHPN-GVAAQTQ-KLRVGDRIVTICG----TSTDG-MT--HTQAVNLMKNA-------SGSIEVQVVAGGDV
3168891          NLAAGTQNMHTIRIQKD---DTGKL-GLSFAGGTSNDPAPNS-----NGDSGLFVTKVTPG-SAAYRCG--LREGDKLIRAND----VNMIN-AS--QDNAMEAIKKR--------ETVELVVLRRSPS
1363206          EEDKLGIPTVPGKVTLQKD-AQNLI-GISIGGGAQYC-------------PCLYIVQVFDN-TPAALDG-TVAAGDEITGVNG----KSIKG-KT--KVEVAKMIQEV-------KGEVTIHYNKLQAD
3881198          EEDRLGMRIQSETIELTKD-EKGVV-GISIGGGGPYC-------------PCVYVVQVFDK-SPAFKDG-RIRCGDEIVAING----ITVKG-ER--KSAVAQLIQVS-------LNPVKITINKLEEA
1176780          ECLSIAVELHKQEVIDAHGQVTIRV-GFKIGGGIDQDPTKAPF---KYPDSGVYITNVESG-SPADVAG--LRKHDKILQVNG----ADFTM-MT--HDRAVKFIKQ--------SKVLHMLVARADLP
2498801          RSAEELRRAELVEIIVETEAQTGVS-GFNVAGG---------------GKEGIFVRELRED-SPAAKSL-SLQEGDQLLSAR-----VFFEN-FK--YEDALRLLQCA------EPYKVSFCLKRTVPT
1709902          KSSSPVGEDQVVTIKMRPD-RHGRF-GFNVKGGADQN-------------YPVIVSRVAPG-SSADKCQPRLNEGDQVLFIDG----RDVST-MS--HDHVVQFIRSARSG--LNGGELHLTIRPNVYR
5031791c         QQMAQGRQIEYIDIERP---STGGL-GFSVVALRSQN----------LGKVDIFVKDVQPG-SVADRDQ-RLKENDQILAINH----TPLDQNIS--HQQAIALLQQT-------TGSLRLIVAREPVH
2959979k         KSMAQGRHVEIFELLKP---PCGGL-GFSVVGLRSEN----------RGELGIFVQEIQEG-SVAHRDG-RLKETDQILAING----QVLDQTIT--HQQAISILQKA-------KDTIQLVIARGSLP
5231271          LTPRRSRKLKEVRLDRL---HPEGL-GLSVRGGLEFG-------------CGLFISHLIKG-GQADSVG--LQVGDEIVRING----YSISS-CT--HEEVINLIRTK--------KTVSIKVRHIGLI
5031979a         MFTPEQIMGKDVRLLRIK--KEGSL-DLALEGGVDSP------------IGKVVVSAVYER-GAAERHG-GIVKGDEIMAING----KIVTD-YT-LAEADAALQKAWN----QGGDWIDLVVAVCPPK
1890856c         RQSIPEEFKGSTVVELMKK-EGTTL-GLTVSGGIDKD-------------GKPRVSNLRQG-GIAARSD-QLDVGDYIKAVNG----INLAK-FR--HDEIISLLKNV-------GERVVLEVEYELPP
3127039a         LDEATLKQLDSIHVTILHKEEGAGL-GFSLAGGAD------------LENKVITVHRVFPN-GLASQEG-TIQKGNEVLSING----KSLKG-TT--HNDALAILRQA------REPRQAVIVTRKLTP
2224541a         EAKAQSENEEDVCFIVLNRKEGSGL-GFSVAGGTD------------VEPKSITVHRVFSQ-GAASQEG-TMNRGDFLLSVNG----ASLAG-LA--HGNVLKVLHQA------QLHKDALVVIKKGMD
732430           ASSQPAKPTKVTLVKSR---KNEEY-GLRPASH-------------------IFVKEISQD-SLAARDG-DIQEGDVVLKING----TVTEN-MS--LTDAKTLIERS-------KGKLKMVVQRDERA
4507517          ASSQPAKPTKVTLVKSR---KNEEY-GLRLASH-------------------IFVKEISQD-SLAARDG-NIQEGDVVLKING----TVTEN-MS--LTDAKTLIERS-------KGKLKMVVQRDERA
1536970          GQPDSDRPIGVLLMKSK---ANEEY-GLRLGSQ-------------------IFIKQMTRT-ALATKDG-NLHEGDIILKING----TVTEN-MS--LTDARKLIEKS-------RGKLQLVVLRDSKQ
4759342a         EPRGRPGPIGVLLMKSR---ANEEY-GLRLGSQ-------------------IFVKEMTRT-GLATKDG-NLHEGDIILKING----TVTEN-MS--LTDARKLIEKS-------RGKLQLVVLRDSQQ
3033501          RQDVHMRPVKSVLVRRT---ESEEF-GVTLGSQ-------------------IFIKHITDS-GLAARNR-GLQEGDLILQING----VSSEN-LS--LSDTRRLIEKS-------EGKLTLLVLRDRGQ
1346574          INGAELSRMREVAFEKN---QSEPL-GVTLKLN---------------DKQRCSVARILHG-GMIHRQG-SLHEGDEIAEING----KSVAN-QT--VDQLQKILKET-------NGVVTMKIIPRPQS
105150           VKGQEVRKVRLIQFEKV---TEEPM-GITLKLN---------------EKQSCTVARILHG-GMIHRQG-SLHVGDEILEING----TNVTN-HS--VDQLQKAMKET-------KGMISLKVIPNQQS
3641615a         KRYNMMKDLRRIEVQRD---ASKPL-GLALAGHKDRQ------------KMACFVAGVDPN-GALGSVD--IKPGDEIVEVNG----NVLKN-RC--HLNASAVFKNV------DGDKLVMITSRRKPN
1418684b         KRYNTMRDLKKLEIVRP---TNTAL-GLALAGHSDRQ------------KMGCFVAGVNTS-GPLASVD--IKSGDEILEVNG----TVLKN-RC--HLNASVIFKNI------DGERLVLITSRRKPN
2625023          IHFSKSENCKDVFIEKQ---KGEIL-GVVIVESGWGS-----------ILPTVIIANMMHG-GPAEKSG-KLNIGDQIMSING----TSLVG-LP--LSTCQSIIKGLK-----NQSRVKLNIVRCPPV
5031585          IHFSNSENCKELQLEKH---KGEIL-GVVVVESGWGS-----------ILPTVILANMMNG-CPAARSG-KLSIGDQIMSING----TSLVG-LP--LATCQGIIKGLK-----NQTQVKLNIVSCPPV
3169807          DHFCNSQNCREVCIQKR---PGEGL-GVALVESGWGS-----------LLPTAVIANLLHG-GPAERCG-ALSIGDRVTAING----TSLVG-LS--LAACQAAVREVR-----RHSSVTLSIIHCPPV
3874209          EMFAKKETQKEVVVPKK---AGEPL-GIVVVESGWGS-----------MLPTVVLAHMNPV-GPAAHSN-KLNIGDQIINING----ISLVG-LP--LSAAQTQIKNMK-----TATAVRMTVVSTPPV
1890856a         DSVATASGPLLVEVAKT---PGASL-GVALTTSVCCNKQ------------VIVIDKIKSA-SIADRCG-ALHVGDHILSIDG----TSMEY-CT--LAEATQFLANT-------TDQVKLEILPHHQT
2137012          HPERRELRPRLCAMKK----GPNGY-GFNLHSDKS--------------RPGQFIRAVDPD-SPAEASG--LREQDRIVEVNG----VCVEG-KQ--HGDVVTAIKAG-------GDEAKLLVVDKETD
1644404          KSHLRELRPRLCTMKK----GPNGY-GFNLHSDKS--------------KPGQFIRAVDPD-SPAEASG--LRAQDRIVEVNG----VCMEG-KQ--HGDVVSAIKGG-------GDEAKLLVVDKETD
4759140          HPEQRELRPRLCTMKK----GPSGY-GFNLHSDKS--------------KPGQFIRSVDPD-SPAEASG--LRAQDRIVEVNG----VCMEG-KQ--HGDVVSAIRAG-------GDETKLLVVDRETD
2198849          SGPLRELRPRLCHLRK----GPQGY-GFNLHSDKS--------------RPGQYIRSVDPG-SPAARSG--LRAQDRLIEVNG----QNVEG-LR--HAEVVASIKAR-------EDEARLLVVDPETD
2137012a         DAAAGAPLPRLCCLEK----GPNGY-GFHLHGEKG--------------KVGQFIRLVEPG-SPAEKAG--LLAGDRLVEVNG----ENVEK-ET--HQQVVSRIRAA-------LNAVRLLVVDPDTD
1644404a         DAAAGEPLPRLCCLEK----GPNGY-GFHLHGEKG--------------KVGQFIRLVEPG-SPAEKSG--LLAGDRLVEVNG----ENVEK-ET--HQQVVSRIRAA-------LNAVRLLVVDPETD
2198849a         MAAPEPLRPRLCRLVR----GEQGY-GFHLHGEKG--------------RRGQFIRRVEPG-SPAEAAA--LRAGDRLVEVNG----VNVEG-ET--HHQVVQRIKAV-------EGQTRLLVVDQETD
```

```
3873819          HIPSDVTPPRLCVVEKLN--GENEY-GYNLHAEKG---------------RGQFVGTVDPD-SPAERGG--LITGDRIFAVNG----HSIIG-EN--HKKVVERIKAN-------PNRCEMLVISEEGA
3876279          PTDAMPYLPRLAELNKGT--PDQEF-GFNLHAERG---------------RGHFIGTVDAG-GIGEKAG--LEAGQRIVGVNG----QLIYP-TTG-HKEVVALIKKD--------TMKTTLLVASEDV
4505703          TTEEVDHKPKLCRLAK----GENGY-GFHLNAIRG--------------LPGSFIKEVQKG-GPADLAG--LEDEDVIIEVNG----VNVLD-EP--YEKVVDRIQSS-------GKNVTLLVCGKKAY
2331224a         --MASTFNPRECKLSKK---EGQNY-GFFLRIEKD--------------TDGHLVRVIEEG-SPAEKAG--LLDGDRVLRIKR----VFVDK-EE--HAQVVDLVRKS-------GNSVTLLVLDGDSY
1083418          LPALGSLRPPIIIHRA-----GKKY-GFTLRAIRVYMGD------TDVYTVHHMVWHVEDG-GPASEAG--LRQGDLITHVNG----EPVHG-LV--HTEVVELVLKS-------GNKVSISTTPLENT
4589590          SPAVSGLRSPITIQRS-----GKKY-GFTLRAIRVYMGD------TDVYSVHHIVWHVEEG-GPAQEAG--LCAGDLITHVNG----EPVHG-MV--HPEVVELILKS-------GNKVAVTTTPFENT
2224547          SAASASPHQPIVIHSS-----GKNY-GFTIRAIRVYVGD------SDIYTVHHIVWNVEEG-SPACQAG--LKAGDLITHING----EPVHG-LV--HTEVIELLLKS-------GNKVSITTTPFENT
1925010          PFTRFLEPSRLAALRRGTAGSVTGV-GLEITYDG-------------GSGKDVVVLTPAPG-GPAEKAG--ARAGDVIVTVDG----TAVKG-LS--LYDVSDLLQGE------ADSQVEVVLHAPGAP
2245133          PFTRFLEPGKFKSLRSGTQGAVTGV-GLSIGYPT----------ASDGPPAGLVVISAAPG-GPANRAG--ILPGDVIQGIDN----TTTET-LT--IYDAAQMLQGP------EGSAVELAIRSGPEG
1296805          PYTRFLSSSDFSKMSKY---DMTGI-GLNIREI-----------PDDNGSLRLVVLGLILD-GPANSAG--VRQGDELLSVNG----SDVRG-KS--AFDVSSMLQGP------KETFVTIKVKHGNCG
2224621          IGRVILNKRTTMPKDS-----GALL-GLKVVGGKMTD----------LGRLGAFITKVKKG-SLADVVG-HLRAGDEVLEWNG----KPLPG-AT--NEEVYNIILES-------KSEPQVEIIVSRPI
2852638          ----------SVPRDS-----GAML-GLKVVGGKMTE----------SGRLCAFITKVKKG-SLADTVG-HLRPGDEVLEWNG----RLLQG-AT--FEEVYNIILES-------KPEPQVELVVSRPI
1086750          KLIGHMILHRTENSAA-----NGDL-GLKIVGGRRTD----------TGKLGAFITQVKPG-SVADTIG-RLRPGDEVVEWNG----QSLQN-AT--YEQVYDSIAAS-------RYDTSVELIVSRSA
2738915          GSQRRYIGVMMLTLSP-----SILA-ELQLREPS-----------FPDVQHGVLIHKVILG-SPAHRAG--LRPGDVILAIGE----QMVQN-----AEDVYEAVRT--------QSQLAVQIRRGREV
2228536          ETKRRYIGVMMLTLTP-----QHPA-ELKLRDPS-----------FPDVSYGVLIHKVIIG-SPAHQAG--LKAGDVVLEING----QATRR-----AEDVYEAVRT--------QQSLALLVRRSYDT
3043642          ARIKITRDSKDHTVS------GNGL-GIRIVGGKEIPGH--------SGEIGAYIAKILPG-GSAEQTG-KLMEGMQVLEWNG----IPLTS-KT--YEEVQSIISQQ-------SGEAEICVRLDLNM
1572787          KRILLTRSYKHHNIY-------NDL-GVRVVGGKRQM----------NGELSAYVSQLHST-ANNQTLG-QIKIGDEVVEWNG----ILLRG-KT--FEEVERIVNKS-------HGEIEMVIRTYKNP
3641615c         FIFDQFPKARTVQVRK-----EGFL-GIMVIYGK-----------HAEVGSGIFISDLREG-SNAELAG--VKVGDMLLAVNQ----DVTLE-SN--YDDATGLLKRA-------EGVVTMILLTLKSE
1418684a         FLFEQYAKARSVQVKK-----EGFL-GIMVIYGK-----------HVEVGNGIFISDLREE-SNAMLAG--LKVGDMLLAVNK----DVCVE-SN--YDEAVALLKRA-------EGIVNLVVLTLKTE
732430a          EDGILRPSMKLVKFRK-----GDSV-GLRLAGGND---------------VGIFVAGVLED-SPAAKEG--LEEGDQILRVNN----VDFTN-II--REEAVLFLLDL-------PKGEEVTILAQKKK
4507517a         KMGFLRPSMKLVKFRK-----GDSV-GLRLAGGND---------------VGIFVAGVLED-SPAAKEG--LEEGDQILRVNN----VDFTN-II--REEAVLFLLDL-------PKGEEVTILAQKKK
1839162          NEAIYGPNTKMVKFKK-----GDSV-GLRLAGGND---------------VGIFVAGIQEG-TSAEQEG--LQEGDQILKVNT----QDFRG-LV--REDAVLYLLEI-------PKGETVTILAQSRA
1536970a         DEAIYGPNTKMVRFKK-----GDSV-GLRLAGGND---------------VGIFVAGIQEG-TSAEQEG--LQEGDQILKVNT----QDFRG-LV--REDAVLYLLEI-------PKGEMVTILAQSRA
3033501a         EDRGYSPDSRVVRFHK-----GTTI-GLRLAGGND---------------VGIFVSGVQEG-SPADGQG--IQEGDQILQVND----VPFRN-LT--REEAVQFLVAL-------PPGEEVELVTQRNE
5453714          ------MSNYSVSLVG-----PAPW-GFRLQGGKD-------------FNMPLTISSLKDG-GKAAQAN--VRIGDVVLSIDG----INAQG-MT--HLEAQNKIKGC-------TGSLNMTLQRASAA
3851178          ------MSNYNVSLVG-----PAPW-GFRLQGGKD-------------FNMPLTISSLKDG-GKASQAH--VRIGDVVLSIDG----ISAQG-MT--HLEAQNKIKAC-------TGSLNMTLQRASAA
4885207          ------MDSFKVVLEG-----PAPW-GFRLQGGKD-------------FNVPLSISRLTPG-GKAAQAG--VAVGDWVLSIDG----ENAGS-LT--HIEAQNKIRAC-------GERLSLGLSRAQPV
3138926          -------MPQNVVLPG-----PAPW-GFRLSGGID-------------FNQPLVITRITPG-SKAAAAN--LCPGDVILAIDG----FGTES-MT--HADAQDRIKAA-------SYQLCLKIDRAETR
3138922a         -------MPQNVVLPG-----PASW-GFRLSGGID-------------FNQPLVITRITPG-SKAEAAN--LCPGDMILAIDG----FGTES-MT--HADAQDRIKAA-------SYQLCLKIDRAETR
2773060a         -------MPQTVILPG-----PAPW-GFRLSGGID-------------FNQPLVITRITPG-SKAAAAN--LCPGDVILAIDG----FGTES-MT--HADAQDRIKAA-------AHQLCLKIDRGETH
1565269          -------MTHSVTLRG-----PSPW-GFRLVGGRD-------------FSLPLTISRVHAG-SKAALAA--LCPGDLIQAING----ESTEL-MT--HLEAQNRIKGC-------HDHLTLSVSRPENK
4506531          -------MPHSVTLRG-----PSPW-GFRLVG-RD-------------FSAPLTISRVHAG-SKASLAA--LCPGDLIQAING----ESTEL-MT--HLEAQNRIKGC-------HDHLTLSVSRPEGR
1710304          -------MTHAVTLRG-----PSPW-GFRLVGGRD-------------FSAPLTISRVHAG-SKAALAA--LCPGDSIQAING----ESTEL-MT--HLEAQNRIKGC-------HDHLTLSVSRPENK
1705900          ------MTTQQIVLQG-----PGPW-GFRLVGGKD-------------FEQPLAISRVTPG-SKAAIAN--LCIGDLITAIDG----EDTSS-MT--HLEAQNKIKGC-------VDNMTLTVSRSEQK
1905874          ------MTTQQIDLQG-----PGPW-GFRLVGRKD-------------FEQPLAISRVTPG-SKAALAN--LCIGDVITAIDG----ENTSN-MT--HLEAQNRIKGC-------TDNLTLTVARSEHK
4502175          TRAADGGRLVEVQLSG-----GAPW-GFTLKGGRE-------------HGEPLVITKIEEG-SKAAAVD-KLLAGDEIVGIND----IGLSG-FR--QEAICLVKGS--------HKTLKLVVKRRSEL
2911719          AAKAKWRQVVLQKASR-----ESPL-QFSLNGGSEKG-------------FGIFVEGVEPG-SKAADSG--LKRGDQIMEVNG----QNFEN-IT--FMKAVEILRN--------NTHLALTVKTNMFK
2224567          AAKAKRRLMTLTKPSR-----EAPL-PFILLGGSEKG-------------FGIFVDSVDSG-SKATEAG--LKRGDQILEVNG----QNFEN-IQ--LSKAMEILRN--------NTHLSITVKTNLFV
3138922          NYFEHKHNIRPKPFII----PGRTS-GCSTPSGID-----------------CGSGRSTPS-SVSTVS--TICPGDL--KVAA-----KMAPNIP--LEMELPGVKIVH-----AQFNTPMQLYSDDNI
2773060          NYFEHKHNIRPKPFVI----PGRSS-GCSTPSGID-----------------CGSGRSTPS-SVSTVS--TICPGDL--KVAA-----KLAPNIP--LEMELPGVKIVH-----AQFNTPMQLYSDDNI
595790           SHLPHTVTLVSIPASAH---GKRGL-SVSIDPPHGPPG------CGTEHSHTVRVQGVDPGCMSPDVKN-SIHVGDRILEING----TPIRN-VP--LDEIDLLIQET-------SRLLQLTLEHDPHD
4505001          SHLPHTVTLVSIPASSH---GKRGL-SVSIDPPHGPPG------CGTEHSHTVRVQGVDPGCMSPDVKN-SIHVGDRILEING----TPIRN-VP--LDEIDLLIQET-------SRLLQLTLEHDPHD
2257461          SRSPHTVTLVSLPAS------DGKR-GLSVSITP----------SCAEHSHTVRVTELDADFLGPDIQS-SIHIGDRILEING----TPIRS-VP--LDEIDVLIQET-------SRLLQLTIEHDPHE
1330390          TESSMGLEVITVRLNL----ETIPL-GMTPSGHTNAR-----------GDAGLYVGDIQDR-GAVALDG-RIDIGDMIVGINE----ISLGN-YS--NKEAVQLLREAV-----QRQYLTLTIAKTGDP
1890856          EQESSGAIIYTVELKR----YGGPL-GITISGTEE-------------PFDPIIISSLTKG-GLAERTG-AIHIGDRILAINS----SSLKG-KP--LSEAIHLLQMA-------GETVTLKIKKQTDA
2959979h         TLQSMSQEAFERTVTIAK---GSSSL-GMTVSANKDG-------------LGVIVRSIIHG-GAISRDG-RIAVGDCILSINE----ESTIS-LT--NAQARAMLRRHS----LIGPDIKITYVPAEHL
1666538          ALSPSGASRFEIVIPFINGSSSAGL-GVSLKARVSKKS------NGSKVDCGIFIKNVMHG-GAAFKEG-GLRVDDRIVGVED----IDLEP-LD--NREAQAALAKKLKEVGMISSNVRLTISRYNEC
2959979i         RIMGINYEIVVAHVSKFS--ENSGL-GISLEATV----------------GHHFIRSVLPE-GPVGHSG-KLFSGDELLEVNG----INLLG-EN--HQDVVNILKEL-------PIDVTMVCCRRTVP
3875228f         SRIGDDIEIIAAVVKPDRQSVDGGL-GISLEGTVDVLNGAQL-------CPHHYIESIRQD-GPVAKTK-VLQAGDELLQVNH----SPLYG-ES--HVTVRQALTRAV----HSGAPVTLIVARRSQH
1498137          YEERQSAEPRFISFQK-----EGSV-GIRLTGGNE----------AGIFVTAVQPG-SPASLQG--LMPGDKILKVND----MDMNG-VT--REEAVLFLLSL-------Q--DRIDLIVQYCK
3875228b         KYDSDGGELVLVACER----PDGGL-GISLAGNKDRD------------KQNVFVVNVRPS-CPLA-----IRPGDELLEING----RLLNK-IS--HVAASAVVRECC----DQHQNIEIVLRRRNGA
2959979j         DLSSLTN-VYHLELPK----DQGGL-GIAICE-ED-------------TLNGVTIKSLTER-GGAAKDG-RLKPGDRILAVDD----ELVAG-CP--IEKFISLLKTA-------KTTVKLTVGAENPG
3879915          EFLNEVEGKVGVQLRGV---DIGGL-GFNIQGN---------------MNEGIFVKEIISK-GIAEQCG-NILVGDKIKSLT-----INFEN-MV--YEDAVTLLSYS------SPYKVKLELERKLSD
3043690a         SLGGKVVTPLHINLSG-----QKDS-GISLENG-----------------VYAAAVLPG-SPAAKEG-SLAVGDRIVAING----IALDN-KS--LNECESLLRSCQ-----DSLTLSLLKVFPQSS
189262           GVQQIQPNVISVRLFKR---KVGGL-GFLVKERV---------------SKPPVIISDLIRG-GAAEQSG-LIQAGDIILAVNG----RPLVD-LS--YDSALEVLRGI-------ASETHVVLILRGPE
3041879          FHLIPDGEITSIKINRAD--PSESL-SIRLVGGSETPLV------------HIIIQHIYRD-GVIARDG-RLLPGDIILKVNG----MDISN-VP--HNYAVRLLRQP-------CQVLRLTVLREQKF
2224701          ASETTGLVQRCVIIQK----DQHGF-GFTVSGDRIV-----------------LVQSVRPG-GAAMKAG--VKEGDRIIKVNG----TMVTN-SS--HLEVVKLIKSG--------AYVALTLLGSSPS
2760368          LALPKNFQYLTLTVRK-----DSNGY-GMKVSGDNPV----------------FVESVKPG-GAAEIAG--LVAGDMILRVNG----HEVRL-EK--HPTVVGLIKAS--------TTVELAVKRSQKL
5031979          SPGNRENKEKKVFISLV---GSRGL-GCSISSGPIQK-------------PGIFISHVKPG-SLSAEVG--LEIGDQIVEVNG----VDFSN-LD--HKEAVNVLKN--------SRSLTISIVAAAGR
```

```
5032083          RRAEIKQGIREVILCKD---QDGKI-GLRLKSIDN----------------GIFVQLVQAN-SPASLVG--LRFGDQVLQING----ENCAG-WS--SDKAHKVLKQAF------GEKITMTIRDRPFE
2331224          FETAAPPAPGSSDQEG-----QQWL-RFLSEAGPE--------------QKGQIIKDIEPG-SPAEAAG--LKNNDLVVAVNG----ESVEA-LD--HDGVVEMIRNG-------GDQTTLLVLDKEAD
3874414          REGYVYELATLVWVQNG---PKLGL-GIKHFQN-----------------RVLVSRVDPG-SLAEK---CLVLGDHLCDVDG----IPVSD-----KDVARDLLVKNIQ----EKGKVTFVVERPDSI
2088778          KKELAGPSSAEDYFVRK---TNGRL-GLTIYAHNDD---------------GVIRAEVRGVTSFAPR---CAQVGDSVVAVDS----ELISS-VR--NASDVEKLLRI-------GKVIHLRRKTPLTP
3879448          SVNSGLPRILEIYLPMK---NVPYL-GLSVCTI----------------DGHIFVSEIAPE-GAVEKDG-RVNVGDQILQVNR----VSFEE-LS--GPQAVRSLREAA----SSKRPITLYISKFARG
3123565b         DDAELQKYSKLLPIH------TLRL-GVEVDSFD----------------GHHYISSIVSG-GPVDTLG-LLQPEDELLEVNG----MQLYG-KS--RREAVSFLKEV-------PPPFTLVCCRRLFD
630714           PLILYACFIESALLRRRS--DNINW-GLNIQSS---------------YRGVHVISEIKEG-SPADACT-KIDAGDEILMING----RTVVG-WD--LTSVVQQVGAL------DVLELSLIVKRRPRE
1666538a         ENEKQLGIEVNAVFDE-----SSELPGTSEPTKL----------------SSVQIMKIEDG-GRIAKDG-RIRVGDCIVAIDG----KPVDQ-MSIIRVRASISDLAAV-----TSRPVTLIINRSLES
2388583          GIVANVIFAYAIIFTQ-----VVSV-GLPVQES----------------FPGVLVPDVKSF-SAASRDG--LLPGDVILAVDG----TELSNSGSDSVSKVVDVVKRNP------EHNVLLRIERGKES
```

Appendix 2: List of Mutations in PDZ3

P311I
R312K
R313T
I314V
V315E
I316L
H317E
R318K
G319P
S320G
T321G
G322S
L323F
G324S
F325I
N326S
I327L
V328A
G329S
G330Q
E331K
D332G
G333N
E334G
G335P
I336V
F337Y
I338V
S339K
F340S
I341V
L342I
A343P
G344D
G345S
P346A
A347V
D348A
L349R
S350D
G351N
E352R
L353I
R354Q
K355V
G356N
Q358R
I359L
L360V
S361A
V362I
N363D
G364D
V365T
D366S
L367V
R368E
N369G
A370L
S371T
H372L
E373D
Q374E
A375V
A376V
I377E
A378L
L379I
K380R
A382T
G383S
Q384G
T385E
V386L
T387K
I388L
I389V
A390V
Q391A

Appendix 3: Sample NMRPipe Processing Script for 2-D PR-HNCO data

```
#!/bin/csh

var2pipe -in ./$1 -noaswap -aqORD 1 \
  -xN  1366           -yN  256  \
  -xT  683            -yT  128  \
  -xMODE  Complex     -yMODE  Complex  \
  -xSW  8000          -ySW  2000  \
  -xOBS  599.772      -yOBS  0.0  \
  -xCAR  4.765        -yCAR  0.0  \
  -xLAB  H1           -yLAB  tilt  \
  -ndim  2            -aq2D  States  \
| nmrPipe -fn POLY -time \
| nmrPipe -fn SP -off 0.35 -end 0.95 -pow 1.5 -c 1.0 \
| nmrPipe -fn ZF -zf 1 -auto \
| nmrPipe -fn FT -verb \
| nmrPipe -fn PS -p0 170.0 -p1 -44.0 -di \
| nmrPipe -fn EXT -x1 6.5ppm -xn 11ppm -sw \
| nmrPipe -fn TP \
| nmrPipe -fn LP -fb \
| nmrPipe -fn SP -off 0.5 -end 1.0 -pow 2 -c 0.5 \
| nmrPipe -fn ZF -zf 1 -auto \
| nmrPipe -fn FT -verb \
| nmrPipe -fn PS -p0 2.0 -p1 0.0 -di \
| nmrPipe -fn POLY -auto \
| nmrPipe -fn TP \
| nmrPipe -fn POLY -auto \
-out $1.ft2 -verb 2 -ov
```

Appendix 4: MATLAB code for automatic phase correction

autophase_pdz_prhnco_white.m

```
function [optimizephase] = autophase_pdz_prhnco_white(debug)
%AUTOPHASE_PDZ_PRHNCO_WHITE tries to automatically find optimal phasing for 2D spectra used for
PRHNCO experiments
%   [phases,bestphase,singlebestphase] =
autophase_pdz_prhnco_white(startbestphase,startincrement,debug)
%   [phases,bestphase,singlebestphase] = autophase_pdz_prhnco_white([0 0 0 0],90,1)
%
%INPUTS
%   startbestphase - starting phases, aplies to all spectra ... [xp0 xp1 yp0 yp1]
%   startincrement - starting increment (in degrees) to search over phase space ... use 90 if
unknown phases
%   debug - 0 or 1 flag determines whether to process all files or just one to debug programming
%
%NOTES
%   Start in a directory containing one folder for each 2D projection + folders "fidfiles" and
"pr"
%   Each projection directory must be named "pdz****"
%   Processing files in the fidfiles directory must be named "proc*"
%   Fid files in the fidfiles directory must be named "pdz*"
%
%DEPENDENCIES
%   System installed PRSP, nmrPipe - parallel_process_prhnco_files.m
%   parallel_process_prhnco_files.m
%   autophase_2d_spectrum.m
%   process_2d_nmrpipe_with_phase.m
%   twodautophasing_fmin_short.m
%   prcalc_prhnco_write_and_process.m
%   read_procpar.m
%   read_text_file_all_lines.m
%
%08/26/09
%Alan Poole, Ranganathan Lab, alanpoole@alumni.wfu.edu

%% Do inital processing and write procfiles
[p,param,commands] = parallel_process_prhnco_files(0,1)
!rm pr/*
copyfile('fidfiles','initialfidfiles');
!rm fidfiles/*
cd initialfidfiles;

%% Top Level - detect files and distribute processing
d = dir;
d = struct2cell(d);
fidfiles = d(1,strmatch('pdz_',d(1,:)));
procfiles = d(1,strmatch('proc',d(1,:)));
if numel(fidfiles) ~= numel(procfiles);
    disp('Number of procfiles does not match the number of fidfiles');
    return
end
nfiles = numel(fidfiles);
for ii = 1:nfiles
    if isempty(strfind(procfiles{ii},fidfiles{ii}))
        disp(sprintf('%s does not match %s',procfiles{ii},fidfiles{ii}))
        return
    end
end

if debug
    [optimizephase] = twodautophasing(procfiles{1},fidfiles{1});
%     cd ..
else
    try matlabpool open 2; catch; end;
    parfor ii = 1:nfiles;
        [optimizephase(ii,:)] = twodautophasing(procfiles{ii},fidfiles{ii});
    end
    try matlabpool close; catch; end;
```

```
    cd ..
end

cd pr/
prcalc_prhnco_write_and_process

end

%%
function [optimizephase] = twodautophasing(procfile,fidfilename)
% make folder for each fidfile and copy data and processing script
foldername = sprintf('folder_%s',fidfilename);
if exist(foldername)==7;
    rmdir(foldername,'s');
end
mkdir(foldername);
copyfile(procfile,sprintf('%s/%s',foldername,procfile));
copyfile(fidfilename,sprintf('%s/%s',foldername,fidfilename));
cd(foldername);

[optimizephase] = autophase_2d_spectrum(procfile,fidfilename,[0 0 0 0],90);

ftfilename = sprintf('%s.ft2',fidfilename);
copyfile(ftfilename,'../../pr/');
copyfile(procfile,'../../fidfiles/');
copyfile(fidfilename,'../../fidfiles/');
cd ..
rmdir(foldername,'s');
end
```

autophase_2d_spectrum.m

```
function [optimizephase,score,phases,bestphase,spectra,threshold,numwhites,scoreimage] =
autophase_2d_spectrum(procfile,fidfilename,startbestphase,startincrement)
%% Function to search over phase variable space

% initialize variables
phasestart = [0 -270 -90 0]; phaseend = [270 360 180 270]; increments = [90 45 15 5 2 1];
startat = find(increments == startincrement);
if startat > 1;
    phasestart = startbestphase - increments(startat-1);
    phasestart(2) = startbestphase(2) - 2*increments(startat-1);
    phaseend = startbestphase + increments(startat-1);
    phaseend(2) = startbestphase(2) + 2*increments(startat-1);
    increments = increments(startat:end);
end
bestphase = startbestphase;
% iterative grid search over successively smaller phase space with finer increments
count = 1;
spectra = cell(1,2*numel(increments));
threshold = zeros(1,2*numel(increments));
numwhites = cell(1,2*numel(increments));
phases = cell(1,2*numel(increments));
scoreimage = cell(1,2*numel(increments));
for ii = 1:numel(increments);
    fprintf('increment %g\n',increments(ii));
    % optimize x phases
    phases{count} = generatephases_x(phasestart,phaseend,increments(ii),bestphase(ii,:));
    disp(phases{count})
    disp(bestphase)
    [bestphase(ii,:),spectra{count},threshold(count),numwhites{count}] =
process_2d_nmrpipe_with_phase(phases{count},procfile,fidfilename);
    xsteps = (phaseend(1)-phasestart(1))/increments(ii)+1;
    ysteps = (phaseend(2)-phasestart(2))/increments(ii)+1;
    scoreimage{count} = reshape(numwhites{count},xsteps,ysteps);
    count = count+1;
    phasestart(1) = bestphase(ii,1) - increments(ii);
    phaseend(1) = bestphase(ii,1) + increments(ii);
    phasestart(2) = bestphase(ii,2) - 2*increments(ii);
    phaseend(2) = bestphase(ii,2) + 2*increments(ii);
```

```
    %optimize y phasest
    phases{count} = generatephases_y(phasestart,phaseend,increments(ii),bestphase(ii,:));
    disp(phases{count})
    disp(bestphase)
    [bestphase(ii,:),spectra{count},threshold(count),numwhites{count}] =
process_2d_nmrpipe_with_phase(phases{count},procfile,fidfilename);
    xsteps = (phaseend(3)-phasestart(3))/increments(ii)+1;
    ysteps = (phaseend(4)-phasestart(4))/increments(ii)+1;
    scoreimage{count} = reshape(numwhites{count},xsteps,ysteps);
    count = count+1;
    phasestart(3:4) = bestphase(ii,3:4) - increments(ii);
    phaseend(3:4) = bestphase(ii,3:4) + increments(ii);

    bestphase(ii+1,:) = bestphase(ii,:);
end

singlebestphase = bestphase(end,:);
[score,optimizephase,exitflag,output] =
twodautophasing_fmin_short(procfile,fidfilename,singlebestphase,threshold(end));

end

%% Generate phases varying 1st dimension
function [phases] = generatephases_x(phasestart,phaseend,increment,bestphase)
nums = (phaseend-phasestart)./increment+1;
nums(3:4) = 1;
total = prod(nums);
phases = zeros(total,4);
phases(:,3) = bestphase(3);
phases(:,4) = bestphase(4);
count = 1;
for ii = 1:nums(1);
    for jj = 1:nums(2);
        phases(count,1:2) = phasestart(1:2)+([ii jj]-1).*increment;
        count = count+1;
    end
end
end

%% Generate phases varying 2nd dimension
function [phases] = generatephases_y(phasestart,phaseend,increment,bestphase)
nums = (phaseend-phasestart)./increment+1;
nums(1:2) = 1;
total = prod(nums);
phases = zeros(total,4);
phases(:,1) = bestphase(1);
phases(:,2) = bestphase(2);
count = 1;
for ii = 1:nums(3);
    for jj = 1:nums(4);
        phases(count,3:4) = phasestart(3:4)+([ii jj]-1).*increment;
        count = count+1;
    end
end
end
```

parallel process prhnco files.m

```
function [p,param,commands] = parallel_process_prhnco_files(phases,fast1)
%PARALLEL_PROCESS_PRHNCO_FILES writes processing files, does initial processing and sets up
directory structure for PRHNCO file processing
%   [p,param,commands] = parallel_process_prhnco_files([0 0 0 0],1)
%
%INPUTS
%   phases - 0 for initial phases of [0 0 0 0]; otherwise, enter phases for initial processing -
[xp0 xp1 yp0 yp1]
%   fast1 = 1 for fast processing with no linear prediction.  0 for processing with LP
%
%NOTES
%   Start in a directory containing one folder for each 2D projection.
```

```
%   Each directory must be named "pdz****" and you cannot have other files folder that start with
"pdz*"
%   Free vs. peptide bound parameters are set by the presence of the string "pep" in the first
directory name
%
%DEPENDENCIES
%   System installed PRSP, nmrPipe
%   read_procpar.m
%   maybe more
%
%08/26/09
%Alan Poole, Ranganathan Lab, alanpoole@alumni.wfu.edu

%%
d = dir;
d = struct2cell(d);
% delete folders "pr" and "fidfiles" if they exist
if strmatch('pr',d(1,:),'exact'); rmdir('pr','s'); end;
if strmatch('fidfiles',d(1,:),'exact'); rmdir('fidfiles','s'); end;
% each projection should have a directory named "pdz_*"
filenames = d(1,strmatch('pdz_',d(1,:)));
n = numel(filenames);

% determine if free or peptide-bound dataset by presence of 'pep' in the directory name
if strfind(filenames{1},'pep');
    free1pep2 = 2;
else
    free1pep2 = 1;
end

if ~isequal(size(phases),[n*2-2 4])
    disp('No phases provided or incorrect size of phase variable');
end

% if slow processing with linear prediction, use parallel processing
if ~fast1
    pctconfig('hostname','localhost');
    try
        matlabpool
    catch
    end
end

parfor (ii = 1:n)
    cd(filenames{ii});
    nhsqcflag(ii) = ~isempty(strfind(filenames{ii},'nhsqc'));
    % get parameters from the procpar file
    [parameters] = read_procpar('procpar');
    if nhsqcflag(ii)
        parameters.pra = 90;
    end
    %% start printing processing file lines
    lines = {'#!/bin/csh' '' 'var2pipe -in ./$1 -noaswap -aqORD 1 \'};
    lines{end+1} = sprintf('  -xN\t\t%g\t\t-yN\t\t%g  \\',parameters.np,2*parameters.ni);
    lines{end+1} = sprintf('  -xT\t\t%g\t\t-yT\t\t%g  \\',parameters.np/2,parameters.ni);
    if nhsqcflag(ii)
        lines{end+1} = sprintf('  -xMODE\t%s\t\t-yMODE\t\t%s  \\','Complex','Rance-Kay');
    elseif parameters.pra == 90 && sum([numel(parameters.phase) numel(parameters.phase2)]) < 4;
        lines{end+1} = sprintf('  -xMODE\t%s\t\t-yMODE\t\t%s  \\','Complex','Rance-Kay');
    else
        lines{end+1} = sprintf('  -xMODE\t%s\t\t-yMODE\t\t%s  \\','Complex','Complex');
    end
    lines{end+1} = sprintf('  -xSW\t\t%g\t\t-ySW\t\t%g  \\',parameters.sw,parameters.sw1);
    lines{end+1} = sprintf('  -xOBS\t\t%g\t\t-yOBS\t\t0.0  \\',parameters.sfrq);
    lines{end+1} = sprintf('  -xCAR\t\t%g\t\t-yCAR\t\t%s  \\',4.77,'0.0');
    lines{end+1} = sprintf('  -xLAB\t\tH1\t\t-yLAB\t\ttilt  \\');
    lines{end+1} = sprintf('  -ndim\t\t2\t\t-aq2D\t\tStates \\');

    % processing area
    lines{end+1} = '| nmrPipe -fn POLY -time \';
    lines{end+1} = '| nmrPipe -fn SP -off 0.35 -end 0.95 -pow 1.5 -c 0.5 \';
```

```
    lines{end+1} = '| nmrPipe -fn ZF -zf 1 -auto \';
    lines{end+1} = '| nmrPipe -fn FT -verb \';
    % correct for too narraw sweep width for some peptide datasets
    if free1pep2 == 2 && parameters.sw < 9000;
        lines{end+1} = '| nmrPipe -fn CS -rs 1.0ppm -sw  \';
    end
    % use phases if provided, otherwise use 0,0 for x dim phases
    if isequal(size(phases),[n*2-2 4]);
        lines{end+1} = sprintf('| nmrPipe -fn PS -p0 %g -p1 %g -di
\\',phases(ii,1),phases(ii,2));
    else
        lines{end+1} = '| nmrPipe -fn PS -p0 0.0 -p1 0.0 -di \';
    end
    % set SW larger for peptide datasets - due to far downfield shift of res 27
    if free1pep2 == 1;
        lines{end+1} = '| nmrPipe -fn EXT -x1 6.5ppm -xn 11ppm -sw \';
    elseif free1pep2 ==2;
        lines{end+1} = '| nmrPipe -fn EXT -x1 6.5ppm -xn 12ppm -sw \';
    end

    lines{end+1} = '| nmrPipe -fn TP \';

    % begin processing of y dimension
    if fast1;
        % omit this line for no linear prediction for fast processing
    elseif nshqcflag(ii)
        lines{end+1} = '| nmrPipe -fn LP -fb -ord 32 \';
    elseif parameters.pra < 30;
        lines{end+1} = '| nmrPipe -fn LP -fb \';
    elseif parameters.pra > 70;
        lines{end+1} = '| nmrPipe -fn LP -fb -ord 32 \';
    else
        lines{end+1} = '| nmrPipe -fn LP -fb -ord 16 \';
    end
    % f1180 dependent offset for the window fuction
    if parameters.f1180 == 'y';
        lines{end+1} = '| nmrPipe -fn SP -off 0.5 -end 1.0 -pow 2 -c 1.0 \';
    else
        lines{end+1} = '| nmrPipe -fn SP -off 0.5 -end 1.0 -pow 2 -c 0.5 \';
    end
    lines{end+1} = '| nmrPipe -fn ZF -zf 1 -auto \';
    lines{end+1} = '| nmrPipe -fn FT -verb \';
    % use phases if provided or 0,0 f1180=n, or -90,180 for f1180=y
    if isequal(size(phases),[n*2-2 4]);
        lines{end+1} = sprintf('| nmrPipe -fn PS -p0 %g -p1 %g -di
\\',phases(ii,3),phases(ii,4));
    elseif parameters.f1180 == 'y';
        lines{end+1} = '| nmrPipe -fn PS -p0 -90.0 -p1 180.0 -di \';
    else
        lines{end+1} = '| nmrPipe -fn PS -p0 0.0 -p1 0.0 -di \';
    end

    lines{end+1} = '| nmrPipe -fn POLY -auto \';
    lines{end+1} = '| nmrPipe -fn TP \';
    lines{end+1} = '| nmrPipe -fn POLY -auto \';
    lines{end+1} = '-out $1.ft2 -verb 2 -ov';

    % write proc+ file
    if nhsqcflag(ii)
        procfilenameplus = sprintf('proc_%s.txt',filenames{ii});
    elseif parameters.pra == 0 || parameters.pra == 90
        procfilenameplus = sprintf('proc_%s.txt',filenames{ii});
    else
        procfilenameplus = sprintf('proc_%s_++.txt',filenames{ii});
    end
    fid = fopen(procfilenameplus,'w');
    for jj = 1:numel(lines);
        fprintf(fid,'%s\n',lines{jj});
    end
    fclose(fid);
    [s,r] = system(sprintf('chmod +x %s',procfilenameplus));
```

```
    %% Perform processing
    if nhsqcflag(ii)
        copyfile('fid',filenames{ii});
        [s,r] = system(sprintf('%s %s',procfilenameplus,filenames{ii}));
    elseif parameters.pra ~= 0 && parameters.pra ~= 90;
        procfilenameminus = sprintf('proc_%s_-+.txt',filenames{ii});
        fid = fopen(procfilenameminus,'w');
        for jj = 1:numel(lines);
            fprintf(fid,'%s\n',lines{jj});
        end
        fclose(fid);
        [s,r] = system(sprintf('chmod +x %s',procfilenameminus));

        % perform PRSP separation of PRHNCO projection into positive and negative angles
        [s,r] = system(sprintf('/usr/local/bin/prsp -s 3 2 fid %s',filenames{ii}));
        d = dir;
        d = struct2cell(d);
        temp_files = d(1,strmatch('pdz_',d(1,:)));

        [s,r]= system(sprintf('%s %s',procfilenameplus,temp_files{1}));
        [s,r]= system(sprintf('%s %s',procfilenameminus,temp_files[1]));

    % Next 2 if clauses do PRSP separation on 0 & 90 projections acquired with phase1 = 1,2
    elseif parameters.pra == 90 && sum([numel(parameters.phase) numel(parameters.phase2)]) == 4
%
        [s,r] = system(sprintf('/usr/local/bin/prsp -s 3 2 fid %s',filenames{ii}));
        delete(sprintf('%s_-+',filenames{ii}));
        movefile(sprintf('%s_++',filenames{ii}),filenames{ii});
        [s,r] = system(sprintf('%s %s',procfilenameplus,filenames{ii}));
    elseif parameters.pra == 0 && sum([numel(parameters.phase) numel(parameters.phase2)]) == 4
        [s,r] = system(sprintf('/usr/local/bin/prsp -s 3 2 fid %s',filenames{ii}));
        delete(sprintf('%s_-+',filenames{ii}));
        movefile(sprintf('%s_++',filenames{ii}),filenames{ii});
        [s,r] = system(sprintf('%s %s',procfilenameplus,filenames{ii}));
    else
        copyfile('fid',filenames{ii});
        [s,r] = system(sprintf('%s %s',procfilenameplus,filenames{ii}));
    end

    cd ..

    param{ii} = parameters;
    commands{ii} = lines;

end
%% Setup directory structure, move files, and collect most relevant parameters
if ~fast1
    try
        matlabpool close
    catch
    end
end

!mkdir pr
!mv pdz*/*.ft2 pr/
!mkdir fidfiles
!mv pdz*/pdz*.fid* fidfiles/
!mv pdz*/proc*.txt fidfiles/

for ii = 1:numel(param)
    p.pra(ii) = param{ii}.pra;
    p.nt(ii) = param{ii}.nt;
    p.sw1(ii) = param{ii}.sw1;
    p.sw(ii) = param{ii}.sw;
    p.ni(ii) = param{ii}.ni;
    p.dof(ii) = param{ii}.dof;
    p.dof2(ii) = param{ii}.dof2;
    p.sfrq(ii) = param{ii}.sfrq;
    p.dfrq(ii) = param{ii}.dfrq;
    p.dfrq2(ii) = param{ii}.dfrq2;
```

```
    p.gain(ii) = param{ii}.gain;
    p.array{ii} = param{ii}.array{1};
    p.f1180{ii} = param{ii}.f1180;
end

save parameters p
```

process_2d_nmrpipe_with_phase.m

```
function [bestphase,spectra,threshold,numwhites] =
process_2d_nmrpipe_with_phase(phases,procfile,fidfilename)
%% Function to process data with new phases and get spectrum
nphases = size(phases,1);
% read procfile to get processing script
lines = read_text_file_all_lines(procfile);
l = strmatch('| nmrPipe -fn PS',lines);
for ii = 1:nphases;
    writeprocfilephases(lines,l,phases(ii,:),procfile)
    % process with new phases
    [spectra(:,:,ii)] = process_get_spectrum(procfile,fidfilename);
    if ii == 1;
        tempspectra = spectra;
        spectra = zeros(size(spectra,1),size(spectra,2),nphases);
        spectra(:,:,1) = tempspectra(:,:,1);
    end
end
threshold = 1*mean(abs(spectra(:)));
whitespace = abs(spectra) < threshold;
nonwhitespace = abs(spectra) >= threshold;
spectra_threshold = spectra .* nonwhitespace;
posneg = squeeze(mean(mean(spectra_threshold))) > 0;
numwhites = posneg .* squeeze(sum(sum(whitespace)));
[c,maxwhiteindex] = max(numwhites);
bestphase = phases(maxwhiteindex,:);

end


%%
function [spectrum] = process_get_spectrum(procfile,fidfilename)

ftfilename = sprintf('%s.ft2',fidfilename);                          % currently, procfiles have
the output set to $1.ft2
[s,r] = system(sprintf('%s %s',procfile,fidfilename));
fid = fopen(ftfilename);                                              % read ft file into matlab
and store data
[header,count] = fread(fid,512,'float32');
xpoints = header(100); ypoints = header(99);
[spectrum,count2] = fread(fid,[xpoints ypoints],'float32');
fclose(fid);
% delete(ftfilename);
end


%%
function writeprocfilephases(lines,l,phases,procfile)
lines{l(1)} = sprintf('| nmrPipe -fn PS -p0 %1.1f -p1 %1.1f -di \\',phases(1),phases(2));
lines{l(2)} = sprintf('| nmrPipe -fn PS -p0 %1.1f -p1 %1.1f -di \\',phases(3),phases(4));
fid = fopen(procfile,'w');
for kk = 1:numel(lines);
    fprintf(fid,'%s\n',lines{kk});
end
fclose(fid);
[s,r] = system(sprintf('chmod +x %s',procfile));
end
```

twodautophasing_fmin_short.m

```
function [score,optimizephase,exitflag,output] =
twodautophasing_fmin_short(procfile,fidfilename,bestphase,threshold)


lines = read_text_file_all_lines(procfile);
l = strmatch('| nmrPipe -fn PS',lines);

[score(1)] = optimizewhitespace(bestphase,lines,l,procfile,fidfilename,threshold);

f = @(x)optimizewhitespace(x,lines,l,procfile,fidfilename,threshold);
options = optimset('Display','iter','TolX',1,'TolFun',1);
[optimizephase,score(2),exitflag,output] = fminsearch(f,bestphase,options);

% cd ..

end


%%
function [numnonwhites] = optimizewhitespace(phases,lines,l,procfile,fidfilename,threshold)
writeprocfilephases(lines,l,phases,procfile)
[spectrum] = process_get_spectrum(procfile,fidfilename);
whitespace = abs(spectrum) < threshold;
nonwhitespace = abs(spectrum) >= threshold;
spectrum_threshold = spectrum .* nonwhitespace;
posneg = squeeze(mean(mean(spectrum_threshold))) > 0;
numwhites = posneg .* squeeze(sum(sum(whitespace)));
numnonwhites = numel(spectrum)-numwhites;
end


%%
function [spectrum] = process_get_spectrum(procfile,fidfilename)

ftfilename = sprintf('%s.ft2',fidfilename);                          % currently, procfiles have
the output set to $1.ft2
[s,r] = system(sprintf('%s %s',procfile,fidfilename));
fid = fopen(ftfilename);                                              % read ft file into matlab
and store data
[header,count] = fread(fid,512,'float32');
xpoints = header(100); ypoints = header(99);
[spectrum,count2] = fread(fid,[xpoints ypoints],'float32');
fclose(fid);
% delete(ftfilename);
end


%%
function writeprocfilephases(lines,l,phases,procfile)
lines{l(1)} = sprintf('| nmrPipe -fn PS -p0 %1.1f -p1 %1.1f -di \\',phases(1),phases(2));
lines{l(2)} = sprintf('| nmrPipe -fn PS -p0 %1.1f -p1 %1.1f -di \\',phases(3),phases(4));
fid = fopen(procfile,'w');
for kk = 1:numel(lines);
    fprintf(fid,'%s\n',lines{kk});
end
fclose(fid);
[s,r] = system(sprintf('chmod +x %s',procfile));
end
```

prcalc prhnco write and process.m

```
function prcalc_prhnco_write_and_process
%AUTOPHASE_PDZ_PRHNCO_WHITE tries to automatically find optimal phasing for 2D spectra used for
PRHNCO experiments
%   [phases,bestphase,singlebestphase] =
autophase_pdz_prhnco_white(startbestphase,startincrement,debug)
%   [phases,bestphase,singlebestphase] = autophase_pdz_prhnco_white([0 0 0 0],90,1)
%
%INPUTS
%
```

```
%NOTES
%
%
%DEPENDENCIES
%
%09/01/09
%Alan Poole, Ranganathan Lab, alanpoole@alumni.wfu.edu

%% Load parameters
% Expect a single variable "p"
try
    load parameters.mat
catch
    load ../parameters.mat
end
% read all FT files
d = dir;
d = struct2cell(d);
ftfiles = d(1,strmatch('pdz_',d(1,:)));
ftfiles = ftfiles(strfindincell(ftfiles,'.ft2'));

for ii = 1:numel(ftfiles);
    fid = fopen(ftfiles{ii});
    [header{ii}] = fread(fid,512,'float32');
    xpoints(ii) = header{ii}(100); ypoints(ii) = header{ii}(99);
    xsw(ii) = header{ii}(101); ysw(ii) = header{ii}(230);
    [spectra{ii}] = fread(fid,[xpoints(ii) ypoints(ii)],'float32');
    fclose(fid);
end

%% set prhnco parameters
% N15 parameters
ix = p.pra == 90; nsw = max(p.sw1(ix));
nsize = 256;
nfrq = p.dfrq2(1);
nppm = p.dof2(1)/nfrq + 89;   % note this parameter is approximate - referencing is out of date
% C13 parameters
ix = p.pra == 0; csw = max(p.sw1(ix));
csize = 256;
cfrq = p.dfrq(1);
cppm = p.dof(1)/cfrq + 92;
% H1 parameters
% test that all x sw's are the same.
xsw2 = xsw - xsw(1);
xpoints2 = xpoints - xpoints(1);
tf = isequal(xsw2,zeros(size(xsw2))) && isequal(xpoints2,zeros(size(xpoints2)));
if ~tf
    disp('H Dim sweep widths or number of points are not equal')
    return
end
hsw = xsw(1);
hsize = xpoints(1);
hfrq = p.sfrq(1);
hppm = (header{1}(102)+header{1}(101)/2)/header{1}(120);

%% Match angles to ft filenames ... this may not be foolproof!!!!
nft = numel(ftfiles);
junk = char(ftfiles);
firstunmatch = find((sum(repmat(junk(1,:),nft,1) == junk) == nft) == 0);
prefix = ftfiles{1}(1:firstunmatch-1);
minusprefix = cellstr(junk(:,firstunmatch:end));
basicangles = p.pra;

% set angle = 90 for nhsqc if it exists
ixnhsqc = strfindincell(ftfiles,'nhsqc');
if ~isempty(ixnhsqc)
    angles(ixnhsqc) = 90;
    ix90 = find(basicangles == 90);
    basicangles = basicangles(setdiff(1:numel(basicangles),ix90(1)));
    minusprefix(ixnhsqc) = [];
end
```

```
if ~allsame(p.dof2)
    disp('Warning - nitrogen center (dof2) does not match for all files check NHSQC file if
used')
end

basicangles = sort(basicangles,'descend');
for ii = 1:numel(basicangles);
    anglematches{ii} = strfindincell(minusprefix,num2str(basicangles(ii)));
    angles(anglematches{ii}) = basicangles(ii);
    minusprefix(anglematches{ii}) = [];
end

for ii = 1:numel(angles);
    if strfind(ftfiles{ii},'-+')
        angles(ii) = -angles(ii);
    end
end


%% Get peak intensities
for ii=1:numel(ftfiles);
    peakthreshold = 5*mean(spectra{ii}(:));
    [numpeaks(ii),sumintensities(ii),intensities{ii}] =
nmrDrawpeaklist(ftfiles{ii},peakthreshold,1,1)
    while numpeaks(ii) > 500
        peakthreshold = peakthreshold*2;
        [numpeaks(ii),sumintensities(ii),intensities{ii}] =
nmrDrawpeaklist(ftfiles{ii},peakthreshold,1,1)
    end
    while numpeaks(ii) < 100
        peakthreshold = peakthreshold/2;
        [numpeaks(ii),sumintensities(ii),intensities{ii}] =
nmrDrawpeaklist(ftfiles{ii},peakthreshold,1,1)
    end
    sortintensities{ii} = sort(intensities{ii},'descend');
    topintensities(:,ii) = sortintensities{ii}(1:100);
end
scale = max(sum(topintensities)) ./ sum(topintensities);

%% Write File

fid = fopen(sprintf('prcalc_prhnco_controlfile_%s.txt',prefix),'w');
fprintf(fid,'# pr-calc control file for PSD95PDZ3 %s PRHNCO\n',prefix);
fprintf(fid,'prcalc version = 1\n\nexpt dims = 3\n');
fprintf(fid,'{ label = N15  sw = %g  size = %g  tilt = 1  sf = %4.3f  centerppm = %1.3f
}\n',nsw,nsize,nfrq,nppm);
fprintf(fid,'{ label = C13  sw = %g  size = %g  tilt = 1  sf = %4.3f  centerppm = %1.3f
}\n',csw,csize,cfrq,cppm);
fprintf(fid,'{ label = H1  sw = %4.3f  size = %g  tilt = 0  sf = %4.3f  centerppm = %1.3f
}\n\n',hsw,hsize,hfrq,hppm);
for ii = 1:numel(ftfiles);
   fprintf(fid,'proj %g  dims = 2  { sw = %g  angles = *, %g, 90 }  { angles = 90, 90, 0 }  scale
= %1.3f  file = %s\n',ii,ysw(ii),angles(ii),scale(ii),ftfiles{ii});
end
```

read_procpar.m

```
function [parameters] = read_procpar(filename)

fid = fopen(filename);
xx=1;
while 1
    tline = fgetl(fid);
    if ~ischar(tline),   break,   end
    procparlines{xx} = tline;
    xx = xx+1;
end
fclose(fid);
```

```
searchterm = {'np 7' 'sfrq 1 1' 'tof 5' 'ni 7' 'sw 1 1' 'sw1 1 1' 'sw2 1 1' 'phase 1 1' 'phase2
7' 'pra 1' 'nt 7' 'dof 5' 'dfrq 5' 'dfrq2 1' 'gain 1' 'array ' 'f1180 4' 'dof2 5'};
param = {'np' 'sfrq' 'tof' 'ni' 'sw' 'sw1' 'sw2' 'phase' 'phase2' 'pra' 'nt' 'dof' 'dfrq' 'dfrq2'
'gain' 'array' 'f1180' 'dof2'};

for ii = 1:numel(param);
    ix = strmatch(searchterm{ii},procparlines);
    if isempty(ix)
        continue
    end
    if strcmp('array',param{ii});
        a = textscan(procparlines{ix+1},'%n %s %*[^\n]');
        paramvalues{ii} = a[1];
    elseif strcmp('f1180',param{ii});
        a = textscan(procparlines{ix+1},'%n %s %*[^\n]');
        paramvalues{ii} = a[1]{1}(2);
    else
        a = textscan(procparlines{ix+1},'%n %n %*[^\n]');
        a = cell2mat(a);
        if a(1) == 2;
            a = textscan(procparlines{ix+1},'%n %n %n %*[^\n]');
            a = cell2mat(a);
            paramvalues{ii} = a(2:3);
        else
            paramvalues{ii} = a(2);
        end
    end
end

parameters.param = param;
parameters.paramvalues = paramvalues;
for ii = 1:numel(param);
    eval(sprintf('parameters.%s = paramvalues{%g};',param{ii},ii));
end
```

read_text_file_all_lines.m

```
function [lines] = read_text_file_all_lines(filename);

fid = fopen(filename);
lines = textscan(fid,'%s','Delimiter','\n');
lines = lines{1};
fclose(fid);
```

Appendix 5: Sample PR-Calc Control File for HNCO Projection Reconstruction

```
# pr-calc control file for PSD95PDZ3 PR HNCO
prcalc version = 1

expt dims = 3
{ label = N15     sw = 2200  size = 256   tilt = 1 sf=60.781 centerppm=120.0 }
{ label = C13    sw = 2000  size = 256   tilt = 1 sf=150.838 centerppm=176.0 }
{ label = H1    sw = 2703.125  size = 692  tilt = 0 sf=599.772 centerppm=8.75 }

proj  1  dims = 2 { sw = 2000 angles =  *,  00.00, 90.00 } { angles = 90.00, 90.00, 00.00 } scale = 2.0 file = pdz_mut71_0.fid.ft2
proj  2  dims = 2 { sw = 2000 angles =  *,  22.50, 90.00 } { angles = 90.00, 90.00, 00.00 } file = pdz_mut71_22.5.fid_++.ft2
proj  3  dims = 2 { sw = 2000 angles =  *, -22.50, 90.00 } { angles = 90.00, 90.00, 00.00 } file = pdz_mut71_22.5.fid_-+.ft2
proj  4  dims = 2 { sw = 2000 angles =  *,  45.00, 90.00 } { angles = 90.00, 90.00, 00.00 } file = pdz_mut71_45.fid_++.ft2
proj  5  dims = 2 { sw = 2000 angles =  *, -45.00, 90.00 } { angles = 90.00, 90.00, 00.00 } file = pdz_mut71_45.fid_-+.ft2
proj  6  dims = 2 { sw = 2200 angles =  *,  67.50, 90.00 } { angles = 90.00, 90.00, 00.00 } file = pdz_mut71_67.5.fid_++.ft2
proj  7  dims = 2 { sw = 2200 angles =  *, -67.50, 90.00 } { angles = 90.00, 90.00, 00.00 } file = pdz_mut71_67.5.fid_-+.ft2
proj  8  dims = 2 { sw = 2200 angles =  *,  90.00, 90.00 } { angles = 90.00, 90.00, 00.00 } scale = 1.8 file = pdz_mut71_90.fid.ft2
```

Appendix 6: Matlab script for automated residue assignments based on similarity to a similar spectrum

```
function [hnco,hncopeakmatch] = assign_without_replacement(hnco_assigned,hnco)
%ASSIGN_WITHOUT_REPLACEMENT Performs matching (without replacement) of unassigned HNCO peaks to a
set of assigned HNCO peaks
%   [hnco] = assign_without_replacement(hnco_assigned,hnco)
%
%INPUTS
%   hnco_assigned - data structure with the following fields
%       hncowt =
%           peak: [114x1 double]
%           Hshift: [114x1 double]
%           COshift: [114x1 double]
%           Nshift: [114x1 double]
%           volume: [114x1 double]
%           intensity: [114x1 double]
%           residue: [1x114 double]
%   hnco - data structure with all fields above except 'residue'
%
%CAUTION:  hnco_assigned and hnco need to be overlayed (aligned) using alignpeaklist2reference.m
first
%
%NOTES:  Use read_hnco_peaklist.m to get inputs
%
%DEPENDENCIES:  none
%
% Alan Poole, Ranganathan Lab
%%
count = 1;
hncoleft = hnco;

while numel(hnco_assigned.residue) > 0 && numel(hncoleft.peak) > 0;
    % calculate distances from every hnco_assigned peak to the closest hnco peaks
    for ii = 1:numel(hnco_assigned.residue);
        [dist(ii),hncopeaknum(ii)] = find_closest_peak_hnco(hnco_assigned.shifts(ii,:),hncoleft);
    end
    % find the closest pair of peaks
    [y,ix] = min(dist); ix = ix(1);
    % keep talley of matching peaks and distances
    hncopeakmatch(count,:) = [hnco_assigned.residue(ix) hncopeaknum(ix) dist(ix)];
    count = count +1;
    % update hnco_assigned to exclude the most recently matched peak
    hnco_assignedleft = setdiff(1:numel(hnco_assigned.residue),ix);
    hnco_assigned.residue = hnco_assigned.residue(hnco_assignedleft);
    hnco_assigned.shifts = hnco_assigned.shifts(hnco_assignedleft,:);
    % update hncoleft to exclude the most recently matched peak
    hncoix = setdiff(1:numel(hncoleft.peak),find(hncoleft.peak == hncopeaknum(ix)));
    if isempty(hncoix); disp(['All hnco peaks used up.  ' num2str(numel(hnco_assigned.residue)) '
assigned hnco_assigned peaks unmatched']); end;
    hncoleft.peak = hncoleft.peak(hncoix);
    hncoleft.shifts = hncoleft.shifts(hncoix,:);
    clear dist hncopeaknum y ix hnco_assignedleft hncoix
end

% hnco.residue contains the residue identity that each hnco.peak was matched to
[c,ia,ib] = intersect(hnco.peak,hncopeakmatch(:,2));
hnco.residue = NaN(size(hnco.peak));
hnco.residue(ia,1) = hncopeakmatch(ib,1);
% hnco.matchorder is the order in which the peaks were matched.  lower index peaks were closer to
reference peaks
hnco.matchorder(ia,1) = ib;

end

%% Included functions
function [dist,peaknumber,diff] = find_closest_peak_hnco(peakloc,peaklist2)
% modified on 10/14/09 to use H C N ordering and peaklist.shifts field
```

```
diff = [peaklist2.shifts(:,1) - peakloc(1) peaklist2.shifts(:,2) - peakloc(2)
peaklist2.shifts(:,3) - peakloc(3)];
normdiff = diff;
normdiff(:,2) = normdiff(:,2)*.39;
normdiff(:,3) = normdiff(:,3)*.17;

distances = sqrt(sum(normdiff'.^2));

[dist,index] = min(distances);
peaknumber = peaklist2.peak(index);

diff = diff(index,:);
end



function [aligned_peaklist,offset,dsum] =
alignpeaklist2reference(refpeaks,peaklist2align,fraction_peaks2use)
%ALIGNPEAKLIST2REFERENCE Aligns one HNCO peaklist to a reference HNCO peaklist
%   [aligned_peaklist] = alignpeaklist2reference(refpeaks,peaklist,0.5)
%
%INPUTS
%   refpeaks - reference peaklist data structure with the following required fields: peak,
Hshift, COshift, Nshift
%   peaklist - peaklist to align to refpeaks.  same requirement as above
%   fraction_peaks2use - [0 to 1] use the closest fraction of peaks for alignment.  [1] aligns
all peaks in refpeaks
%
%DEPENDENCIES
%   everything is included
%
% Alan Poole, Ranganathan Lab

%% find optimum overlay of the peaks
options = optimset('MaxFunEvals',1000000,'TolFun',1e-7,'TolCon',1e-7);
[offset,dsum,exitflag,output] = fmincon(@overlayhncopks,[0 0 0],[],[],[],[],[-1 -1 -1],[1 1
1],[],options);

% correct for offset to get optimum overlay
aligned_peaklist = peaklist2align;
aligned_peaklist.Hshift = peaklist2align.Hshift - offset(1);
aligned_peaklist.COshift = peaklist2align.COshift - offset(2);
aligned_peaklist.Nshift = peaklist2align.Nshift - offset(3);
aligned_peaklist.shifts = peaklist2align.shifts - repmat(offset,size(peaklist2align.shifts,1),1);

%% Nested Function
    function [dsum] = overlayhncopks(offsetguess)
        H = refpeaks.Hshift + offsetguess(1);
        CO = refpeaks.COshift + offsetguess(2);
        N = refpeaks.Nshift + offsetguess(3);

        numpeaks = numel(refpeaks.peak);
        dists = zeros(numpeaks,1);
        for ii = 1:numpeaks;
            [dists(ii)] = find_closest_peak_hnco([H(ii) CO(ii) N(ii)],peaklist2align);
        end

        [dists,ixdistsort] = sort(dists);
        numpeaks2use = floor(numpeaks*fraction_peaks2use);
        dists = dists(1:numpeaks2use);

        dsum = sum(dists);

    end
end

%% Included function
function [dist] = find_closest_peak_hnco(peakloc,peaklist2)

Hdiff = peakloc(1) - peaklist2.Hshift;
COdiff = peakloc(2) - peaklist2.COshift;
```

```
Ndiff = peakloc(3) - peaklist2.Nshift;

distances = sqrt(Hdiff.^2 + (0.17*Ndiff).^2 + (0.39*COdiff).^2);
[dist,index] = min(distances);
end
```

Appendix 7: Matlab script for calculating chemical shift change between mutant and wild-type HNCO spectra

```
function [data,dsum] =
hnco_perturbation(wtpeaklist,mutant_key,res2calculate,residue_number_offset,match_algorithm,use_a
ssignments,new_or_old_offsets)
%HNCO_PERTURBATION_NO_REPLACEMENT Calculate chemical shift perturbations
%from WT and mutant peaklists.
%
%INPUTS
%   wtpeaklist - filename of the wild-type dataset
%   mutant_key - text file correlating mutant numbers to residue numbers
%   res2calculate - residue numbers for which to calculate chemical shift perturbations
%   residue_number_offset - numerical offset to match PDB or other numbering convention
%   match_algorithm - 'replace' or 'no_replace'
%   use_assignmens - 1 to use assignments, 0 to not use them
%   new_or_old_offsets - 'new' or 'old' - 'new' will write a new offsets file
%
%NOTES:
% 1) Must be executed from directory with all peaklists
% 2) Mutant peaklists must be in form "pdz_mut##_*.xpk"
% 3) Directory must include "mutant_key"
%
%DEPENDENCIES:
%   read_hnco_peaklist.m
%   alignpeaklist2reference.m
%   assign_without_replacement.m
%
% Alan Poole, Ranganathan Lab, 09/24/09

%% Setup
% get mutant peaklist names
d = dir;
d = struct2cell(d);
ix = strmatch('pdz_mut',d(1,:));
peaklists = d(1,ix);
clear ix d

% read in wt peaklist - must be assigned
[hncowt] = read_hnco_peaklist(wtpeaklist,1);

% adjust for residues to calculate
[c,ia,ib] = intersect(hncowt.residue,res2calculate);
data.residues = hncowt.residue(ia) + residue_number_offset;
data.res2peakwt = hncowt.peak(ia)';
data.wtpeakloc = hncowt.shifts(ia,:);
clear c ib

%read mutant key to match mutant numbers to residue numbers
fid = fopen(mutant_key);
c = textscan(fid,'%n%s%n%s');
fclose(fid);

% extract mutant numbers & residues from peaklist filename & mutant key
for jj = 1:numel(peaklists);
    x = strfind(peaklists{jj},'mut') + 3;
    xx = strfind(peaklists{jj},'_');
    xxx = str2num(peaklists{jj}(x:xx(2)-1));
    data.mutnum(jj) = xxx;
    ix = find(c{1} == xxx);
    data.muts(jj) = c{3}(ix);
%     mutations{jj} = [num2str(c{3}(ix) + residue_number_offset) ' ' c{2}{ix} ' to ' c{4}{ix}];
    mutations{jj} = [c{2}{ix} num2str(c{3}(ix) + residue_number_offset) c{4}{ix}];
end

% sort data by ascending residue numbers
[mutations2,ix2] = sortrows(char(mutations));
peaklists = peaklists(ix2);
data.mutnum = data.mutnum(ix2);
```

```
data.muts = data.muts(ix2) + residue_number_offset;
data.mutations = mutations2;
clear ix2 mutations x xx xxx
data.assigned = zeros(size(data.muts));

if strfind(peaklists{1},'pep');
    freepep = 'pep';
else
    freepep = 'free';
end

% read offset file if not computing new offsets
if strcmp(new_or_old_offsets,'old')
    [offsets] = read_offsets(freepep);
end

%% Loop over all mutant peaklists
for jj = 1:numel(peaklists);
    disp(jj)
    disp(peaklists{jj})
    % read mutant peaklist
    hncopks2 = peaklists{jj};
    if use_assignments;
        [hnco2] = read_hnco_peaklist(hncopks2,1);
        hnco2.residue = hnco2.residue + residue_number_offset;
        % if >90 residues are assigned, set data.assigned(index) = 1
        if sum(~isnan(hnco2.residue)) > 90;
            data.assigned(jj) = 1;
        else
            [hnco2] = read_hnco_peaklist(hncopks2,0);
        end
    else
        [hnco2] = read_hnco_peaklist(hncopks2,0);
    end

    % use calculated offsets or find optimum overlay of the peaks
    if strcmp(new_or_old_offsets,'old')
        ix3 = offsets(:,1) == data.muts(jj);
        data.offset{jj} = offsets(ix3,2:4);
        hnco2.shifts = hnco2.shifts - repmat(data.offset{jj},size(hnco2.shifts,1),1);
    else
        [hnco2,offset,dsum(jj)] = alignpeaklist2reference(hncowt,hnco2,0.5);
        data.offset{jj} = offset;      clear offset
    end

    % use assignments in peaklist if available, otherwise make best guess by matching without
replacement
    if ~data.assigned(jj) && strcmp(match_algorithm,'no_replace')
        [hnco2] = assign_without_replacement(hncowt,hnco2);
        hnco2.residue = hnco2.residue + residue_number_offset;
    end


    for ii = 1:numel(data.residues);
        if isfield(hnco2,'residue') && ismember(data.residues(ii),hnco2.residue);
            ix = find(hnco2.residue == data.residues(ii));
            data.csd(ii,:,jj) = hnco2.shifts(ix,:) - data.wtpeakloc(ii,:);
            data.res2mutpeak(ii,jj) = hnco2.peak(ix);
        else
            [junk1,data.res2mutpeak(ii,jj),data.csd(ii,:,jj)] =
find_closest_peak_hnco(data.wtpeakloc(ii,:),hnco2);
        end
    end
end

data.csdnorm = data.csd; data.csdnorm(:,2,:) = data.csdnorm(:,2,:)*.39; data.csdnorm(:,3,:) =
data.csdnorm(:,3,:)*.17;
data.city = squeeze(sum(abs(data.csdnorm),2));
data.rms = squeeze(sqrt(sum(data.csdnorm.^2,2)));
data.allshifts =
reshape(data.csdnorm,size(data.csdnorm,1)*size(data.csdnorm,2),size(data.csdnorm,3));
```

```
if isfield(data,'assigned'); data.assigned = logical(data.assigned); end;

%% Offset file
if strcmp(new_or_old_offsets,'new');
    fid = fopen(sprintf('offsets_%s.txt',freepep),'w');
    for ii = 1:numel(data.muts);
        fprintf(fid,'%g %g %g
%g\n',data.muts(ii),data.offset{ii}(1),data.offset{ii}(2),data.offset{ii}(3));
    end
    fclose(fid);
end
end


%% Extra Functions
function [offsets] = read_offsets(freepep)
fid = fopen(sprintf('offsets_%s.txt',freepep),'r');
offsetscan = textscan(fid,'%f %f %f %f\n');
fclose(fid);
offsets = cell2mat(offsetscan);
end


function [dist,peaknumber,diff] = find_closest_peak_hnco(peakloc,peaklist2)
% modified on 10/14/09 to use H C N ordering and peaklist.shifts field

diff = [peaklist2.shifts(:,1) - peakloc(1) peaklist2.shifts(:,2) - peakloc(2)
peaklist2.shifts(:,3) - peakloc(3)];
normdiff = diff;
normdiff(:,2) = normdiff(:,2)*.39;
normdiff(:,3) = normdiff(:,3)*.17;

distances = sqrt(sum(normdiff'.^2));

[dist,index] = min(distances);
peaknumber = peaklist2.peak(index);

diff = diff(index,:);
end


function [structure] = read_hnco_peaklist(filename,assigned_flag)

fid = fopen(filename);
for ii = 1:6; header{ii} = fgetl(fid); end;
C = textscan(fid,'%f%s%f%*s%*s%*s%*s%*s %s%f%*s%*s%*s%*s%*s  %s%f%*s%*s%*s%*s%*s %f%f%*[^\n]');
fclose(fid);
nuclei = textscan(header{2},'%s%s%s');
for ii = 1:numel(nuclei); nuclei{ii} = nuclei{ii}{1}; end;
H1 = strmatch('H1',nuclei);
N15 = strmatch('N15',nuclei);
C13 = strmatch('C13',nuclei);
nuclei_cell_array = [3 5 7];
structure.peak = C{1};
structure.Hshift = C{nuclei_cell_array(H1)};
structure.COshift = C{nuclei_cell_array(C13)};
structure.Nshift = C{nuclei_cell_array(N15)};
structure.shifts = [structure.Hshift structure.COshift structure.Nshift];
structure.volume = C{8};
structure.intensity = C{9};


if assigned_flag;
    tempres = C{2};
    % here, I am assuming that the residue name formatting looks like {56.HN}
%     assigned_index = strfindincell(tempres,'HN');
    assigned_index = find(~cellfun(@isempty,strfind(tempres,'HN')));

    assigned_res = tempres(assigned_index);
    dots = strfind(assigned_res,'.');
```

```
    for ii = 1:numel(assigned_res);
        res2(ii) = str2double(assigned_res{ii}(2:dots{ii}));
    end
    structure.residue(1,1:numel(tempres)) = NaN;
    if numel(assigned_index); structure.residue(1,assigned_index) = res2; end
end;


function [data] = restrict_hnco_pert_data(data,residues,muts)
%RESTRICT_HNCO_PERT_DATA Restrict chemical shift perturbation data to defined residues and
mutants
%
%INPUTS
%   data - data structure produced by hnco_perturbation.m
%   residues - residue numbers of observed positions to keep
%   muts - residue numbers of mutants to keep
%
%Alan Poole, Ranganathan Lab, 03-02-2010
%%
[~,ia] = intersect(data.residues,residues);
[~,ib] = intersect(data.muts,muts);
data.residues = data.residues(ia);
data.res2peakwt = data.res2peakwt(ia);
data.wtpeakloc = data.wtpeakloc(ia,:);
data.muts = data.muts(ib);
data.mutnum = data.mutnum(ib);
data.mutations = data.mutations(ib,:);
data.assigned = data.assigned(ib);
data.offset = data.offset(ib);
data.res2mutpeak = data.res2mutpeak(ia,ib);
data.csd = data.csd(ia,:,ib);
data.csdnorm = data.csdnorm(ia,:,ib);
data.city = squeeze(sum(abs(data.csdnorm),2));
data.rms = squeeze(sqrt(sum(data.csdnorm.^2,2)));
data.allshifts =
reshape(data.csdnorm,size(data.csdnorm,1)*size(data.csdnorm,2),size(data.csdnorm,3));
```