LOGIC AND MECHANISM OF AN EVOLUTIONARILY

CONSERVED INTERACTION IN PDZ DOMAINS

APPROVED BY SUPERVISORY COMMITTEE

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

Johann Deisenhofer, Ph.D.

Michael Rosen, Ph.D.

Hongtau Yu, Ph.D.

Acknowledgements

These have been happy years and I have many wonderful people to thank for making my graduate school experience so rich. First and foremost, I would like to express my deepest gratitude to my mentor, Dr. Rama Ranganathan. I came to his lab not only because of a keen interest in the problems he works on, but also because of an admiration for the way in which he works on them. In his lab, Rama has fostered an intellectual environment that I have benefited from tremendously – one that combines creativity, rigor and focus. I have particularly enjoyed the fearless approach to learning and applying new disciplines to the problems we work on. His joy and relentless energy for science have been infectious and inspirational. I cannot thank him enough for challenging me, for giving me a chance to work on a beautiful problem, and for his patient teaching through the years.

I have benefited immeasurably from the open and interactive spirit in our lab, a feature that derives from the lab members. They have all been an integral part of my graduate school experience – whether by answering a simple question or by providing critical analysis. I would like to give special thanks to Dr. Mark Wall for introducing me to the world of protein crystallography with the patience of a Jedi Knight. His thorough training formed a foundation that saw me through my thesis project. Mike Socolich deserves special mention not only for his unique ability to mete out abuse to graduate students, but also for his efforts in teaching me the basics of molecular biology. Not many people will come to the lab in the wee hours of the morning to help someone freeze crystals; Mike did that and other kind deeds and I am thankful for all of them. Dr. Steve Lockless and Dr. Rajul Jain each put together impressive bodies of work that formed the foundation for my thesis project. I thank them for their rigorous work and for the conversations I have had with them. I would also like to extend my appreciation to Dr. Bill Russ, Shan Mishra, and Dr. Gurol Suel for numerous conversations from which I have taken much insight and learning. In addition to being a stimulating scientific environment, our lab has also been a place of wonderful camaraderie and I will always remember the friendships I have formed through these years.

Through the course of my work I have also received significant help and training from many people not in our lab. I am extremely thankful to Mischa Machius, Diana Tomchick, and Hyock Kwon for training and helping during the several synchrotron trips I made through the course of my work; they also served as a reliable help-line for advice on numerous technical crystallography questions. Additionally, I must anonymously acknowledge the kind assistance of the synchrotron support staffs with whom I have worked. Celestine Thomas kindly provided training to perform ITC

experiments. I am deeply indebted to Dr. Michael Rosen and Dr. Gaya Amarasinghe for supporting my foray into the world of NMR dynamics. Both gave generously of their time and energy to teach me (and other lab members) and I sorely wish I had more time to work with them on these experiments.

I have had the benefit of interacting with an extremely supportive thesis committee including Dr. Johann Deisenhofer, Dr. Michael Rosen, and Dr. Hongtao Yu. They have helped create an environment that is at once approachable and scientifically intense. I thank them for this and for giving their time, energy, and insightful comments throughout the course of my work.

I am deeply thankful to the MSTP program for giving me the chance to study a fascinating problem in biology and to learn how medical problems are currently approached. I truly appreciate the support they have given me over the past five years. Special thanks go to Robin Downing and Stephanie Robertson for the tremendous work they to do keep the program running smoothly and for patiently handling my tardiness.

I am very fortunate to have had an extremely supportive group of friends through the trials and tribulations of graduate school. These friends have been with me through every step of this journey – from making the decision to apply to discussing results of experiments. I certainly hope and expect these friendships and conversations to continue long after graduate school.

I owe a debt to my parents, Ramesh and Savita Sharma, that I cannot express. They were my first mentors. My mother taught me to take joy in seeing the beauty of how nature works. My father gave me a love for math and science from childhood and laid the foundations that have served me so well through every phase of my schooling. They have sacrificed to provide every opportunity through my life and have even encouraged my endeavors that they did not fully understand. I thank them for their love, support, and trust. I thank my sister, Neha, for her incredible loyalty and consistent encouragement through the years. I am especially glad that she has come to believe research is cool. In my second year of graduate school, my wife and I got married and I gained a new family. I have loved spending time with my wife's parents – Yash and Rekha Puri, her brother – Sameer Puri, and her other family members and friends, boring them all with details of our work in the lab. I thank them all for their love, patience, and support.

Finally, it is impossible to say enough about my wife and biggest fan, Charu. While these graduate school years have occasionally been challenging for me, they have often been much harder on her. She has handled the ups and downs with tremendous patience and continues to be a source of

inspiration to me. Her support and excitement for my work has been unwavering. I cannot thank her enough for being who she is.

LOGIC AND MECHANISM OF AN EVOLUTIONARILY CONSERVED INTERACTION IN PDZ DOMAINS

by

ROHIT SHARMA

DISSERTATION / THESIS

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

June 2006

LOGIC AND MECHANISM OF AN EVOLUTIONARILY

CONSERVED INTERACTION IN PDZ DOMAINS

Publication No. _____

Rohit Sharma M.D., Ph.D.

The University of Texas Southwestern Medical Center at Dallas, 2006

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

Proteins are beautiful materials evolved to channel specific energetic perturbations into particular functions. At the core of virtually every biological process are two features of a protein: the energetic architecture and the mechanisms of energy propagation. Structural, dynamics, and mutagenesis experiments have revealed that anisotropy and cooperativity are common features of the energy propagation in proteins; however, a complete understanding of the patterns and mechanisms of energy propagation remain unclear from these studies.

Previous work in our lab developed a methodology, termed the Statistical Coupling Analysis (SCA), to estimate energetic interactions between residues in a protein from their statistical co-variation through evolution. The results of this algorithm revealed a small subset of the residues in a protein have significant energetic interactions and form a connected substructure in proteins and show excellent agreement with mutagenesis data in several systems.

Using the same fundamental concepts of the original SCA, we have developed an improved version of SCA. This new algorithm provides, for the first time, a global map of the co-evolutionary interactions between residues in a protein from a multiple sequence alignment. The results of the new SCA are consistent with the original method but produce values for all pairs of positions.

We then used the energetic map provided by SCA to understand the physical basis of specificity in the PDZ domain. The co-evolutionary energetic map of the PDZ domain predicts a long range interaction between position 372, a known specificity determinant that directly interacts with ligand, and position 322. Thermodynamic measurements in one PDZ domain reveal that position 322 modulates the specificity-determining interaction between 372 and its ligand contact. Structural studies show that flexibility at 322 is tuned to make conformational change on one side of the binding pocket sensitive to interactions at the distant specificity-determining contact. This designed mechanical coupling allows the domain to have AND gate-like behavior in screening for specific binding interactions. Understanding the logic and mechanism of a co-evolved interaction gives confidence in the ability of SCA to identify the functionally critical interactions in a protein, even when not structurally obvious.

Given the functional and structural relevance of SCA predictions, we next addressed the topology of the energetic map in proteins. Analysis of several structurally and functionally diverse proteins revealed several common striking features in their energetic maps. First, the highly co-evolved positions in a protein show a high degree of mutual co-evolution so that, together, they form a nearly completely co-evolved sub-cluster. Secondly, the pattern of energetic interactions in proteins is highly heterogeneous, and fit a power-law distribution where most residues have very few co-evolutionary links with other residues and a few residues have many co-evolutionary links. The data is very consistent with extensive mutagenesis studies in several systems. Together, these experiments begin to demonstrate that the contiguous networks identified by SCA reflect structural regions capable of cooperatively channeling energy to produce functionality.

Table of Contents

List of Figures

List of Tables

Abbreviations

A	Angstrom
Cdc42	cell division cycle 42
CRIB	Cdc42-Rac-Interactive Binding motif
DTT	dithiothreitol
GDP	Guanosine diphosphate
GFP	Green Fluorescent Protein
GPCR	G protein coupled receptor
GRK	G protein coupled receptor kinase
GTP	Guanosine triphosphate
hGH	human Growth Hormone
hGH-R	human Growth Hormone Receptor
HIV	Human Immunodeficiency Virus
IPTG	isopropyl-β-D-thiogalactopyranoside
ITC	Isothermal titration calorimetry
kcal	kilocalorie
K_d	Dissociation constant
MAP	Mitogen activated protein
MAPK	Mitogen activated protein-kinase
μcal	microcalories
μg	micrograms

ml	milliliters
μl	microliters
mM	millimolar
μM	micromolar
mol	mole
MSA	multiple sequence alignment
NHERF	Na^+/H^+ exchanger regulatory factor
nM	nanomolar
NMR	Nuclear magnetic resonance
ns	nanosecond
P_{WT}	Peptide of last 9 amino acids of CRIPT
P_{T7F}	Same peptide as P_{WT}, except with T7F mutation
P_{-2}	Refers to -2 position of peptide (P_0 refers to carboxy terminal residue)
PA	parent alignment
PDZ	PSD95, Discs-large, Zo-1
PDZ3	Third PDZ domain from PSD95
ps	picosecond
PSD	Post-synaptic density
PYP	Photoactive yellow protein
s	second
SA	subalignment
SCA	Statistical Coupling Analysis

SH2	Src Homology 2 domain
SH3	Src Homology 3 domain
U	Units

Chapter 1 Introduction

Through evolution proteins have achieved stability and functionality remarkably well-tuned for specific purposes. Studies of numerous systems have collectively demonstrated that proteins are tuned to respond to energetic perturbations in a specific manner. Such energetic perturbations may include, for example, substrate binding, covalent modification, or voltage changes. For example, allosteric signaling proteins reliably convert ligand binding into structural changes at a distant site [1]. Similarly, motor proteins use a mechanical amplifier to convert the energy released from breakdown of a high energy substrate in a catalytic core into motion [2]. Also, enzymes rely on the cooperative interaction of several amino acids to interact with specific substrates and efficiently catalyze complex reactions [3]. All these examples share one aspect of protein energetics: the collective interaction of specific sets of amino acids to deliver function.

Viewed from a materials science perspective, proteins display functional properties that classify them as very impressive ‘smart materials’ [4]. An emerging branch of materials science engineering, smart materials will be materials capable of both sensing environmental changes and performing a predetermined adaptive response. Engineers currently attempt to design smart materials with two components abundantly evident in biological systems: sensors to detect inputs from the environment and actuators to execute a specific physical change. For instance, the retinal chromophore of rhodopsin is a ‘sensor’ that is tuned so that light of a particular wavelength triggers its cis-trans isomerization [5]. This switch in the chromophore then initiates ‘actuator’ events: conformational changes in the protein ultimately cause structural changes in the cytoplasmic loops [5]. While functional characterizations of proteins clearly demonstrate

the efficiency of their sensors and actuators, a complete understanding of their underlying mechanism is lacking. Thus, while materials science engineers attempt to exploit the physical properties of various substances to design smart materials, protein research attempts to deconstruct the physical features of protein structures that endow them with 'smart' functionality.

Collectively, studies of numerous systems (some are discussed below) indicate that proteins attain their functionality through complex energetic interactions among a precise arrangement of atoms. Complete understanding of any function, whether it is binding, catalysis, or allostery, therefore depends on describing two features of the protein: 1) the energetic architecture of amino acid interactions in the protein, and 2) the mechanisms of energy propagation through the network of interatomic interactions. Motivated by this view of proteins, the goals of my research fall into three categories:

1) To develop a way of globally mapping the evolutionary constraint between residues as a surrogate for their energetic interactions.
2) To understand why and how residues in one protein, the PDZ domain, evolve and energetically interact.
3) To understand the topology of amino acid interactions in proteins.

Decades of research into this topic, generally referred to as the sequence-structure-function problem, have revealed several important general features regarding these energetic interactions. In the remainder of this chapter I will review several of these themes as they pertain to my research.

Insight from structures

Protein structures, either from X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy, have provided extremely useful insights into the mechanistic basis for function. For example, structures of individual enzymes provide clues about protein active sites. Enzyme-substrate analog structures have allowed the detailed description of several reaction coordinates [1]. Numerous structures of protein-protein complexes have revealed that interaction surfaces display geometric and chemical complementarities critical for specific interactions [1]. These examples highlight the value of structures in understanding the mechanistic role of individual amino acids. While structural studies are unable to directly report the energetic value of interactions among atoms, they have revealed important constraints on the underlying energetic architecture in proteins.

Conformational changes, flexibility, and stability

Comparison of active and inactive states of enzymes and signaling proteins has revealed that many proteins have the ability to undergo significant conformational changes in a highly coordinated manner. G protein activation provides a dramatic example. In the GDP-bound state G proteins are maintained in an inactive configuration [6]. Upon GTP binding, the switch I and II regions undergo conformational changes involving both significant displacements and disorder to order transitions. Together, the switch regions form the interaction surface for effector proteins and GTPase activators [6]. Signaling relies on the ability of the structure to channel energetic interactions reliably and efficiently into specific conformational changes. The essence of this ‘energy

channeling' is that the protein contains built-in mechanisms suited for propagating and delivering energy in a manner appropriate for triggering specific structural changes. Thus, nucleotide exchange can reliably trigger conformational change at a distance.

G protein conformational changes highlight the critical role of flexibility in protein structure and function. Studies of numerous proteins have shown that structurally disordered regions, usually in loops or linkers, are often sites of functional importance. In G proteins the switch regions are tuned to behave, as suggested by their name, like binary switches. Similarly, loops in antibodies and T-cell receptors provide conformational plasticity to permit promiscuous binding to antigens [7]. Furthermore, mutations to catalytic positions of T4 lysozyme, citrate synthase, staphylococcal nuclease, and other enzymes dramatically reduce enzymatic activity but increase thermal stability [8, 9]. Crystal structures of the T4 lysozyme mutants showed that the stabilized mutant proteins take a conformation similar to that of the enzyme-product complex – a structure with decreased flexibility [8]. All these examples highlight the importance of dynamics as well as structure. From an evolutionary point of view, proteins in nature can be seen as evolutionary solutions to two opposing tendencies: 1) rigidity, to achieve necessary stability and 2) flexibility, to make required conformational changes.

The balance of these opposing tendencies in proteins, first stated by Pauling and colleagues, is known as the 'stability-function hypothesis' [10]. Stability derives from a hydrophilic exterior, tight packing in the hydrophobic core, and minimal disordered loops [11]. However, structural elements that are critical for function often violate these patterns. Protein-protein interfaces often rely on surface exposure of hydrophobic patches [12]. Signaling proteins often utilize "disorder-to-order" changes to transfer energy [6]. Enzymes are built to complement the transition state of the reaction coordinate rather than

the ground state [9]. These examples indicate that there must be a well-tuned energetic architecture in proteins that represents the evolutionary solution to the stability-function balance. How has evolution organized energy in proteins to deliver both stability and dynamic function? Understanding this issue is a critical part of understanding protein mechanisms.

Structures reveal features of energy distribution

High resolution X-ray crystal structures have also provided some insight into parsing of energy in local regions of structures. A recent example came from a 0.83 A structure of α-lytic protease, an extremely stable enzyme released into the soil by *Lysobacter enzymogenes* [13]. Previous studies had shown that the α-lytic protease maximizes its longevity in the harsh extracellular environment through a large and highly cooperative kinetic barrier to unfolding ($t_{1/2}$ = 1800 years); it achieves the extremely stable folded state with the catalytic aid of an N-terminal pro region. After releasing all geometric constraints during refinement, the structure showed that Phe228, a position in the core of the C-terminal domain, was significantly distorted from planarity at an estimated energetic cost of 4.1 kcal/mol [13]. The authors suggest the strain in Phe228 stores energy “like the spring in a spring-loaded latch” and contributes to the cooperativity of unfolding [13]. Importantly, the structure shows that the geometric distortion of Phe228 results from steric interactions with neighboring positions constraining the ‘spring’, primarily Thr181 and Trp199. Furthermore, comparison of several bacterial serine protease sequences and structures revealed an intriguing correlation between the amino acid character of position 199 and the distortion at position 228. The alignment suggested that proteases with large cyclic residues at position 199 tend to have residues at position 228 with distorted geometry; these proteases accordingly

have larger pro regions presumably to catalyze the formation of the distorted geometry. The combination of structural and evolutionary data provides a striking example of evolutionary pressure to build and conserve a cooperative energetic architecture among several positions in order to maintain extreme stability.

While structures have provided enormous insights into how proteins achieve their function, several significant features of the physical nature of proteins are not revealed. First, structures do not allow the global energetic mapping among all atoms even with their exact positions. Second, the mechanisms of energy propagation among positions are not generally clear from structure alone. Lastly, protein function depends critically on the motion of atoms in the structure, a feature generally not apparent from structures. As the examples discussed above indicate, X-ray crystal structures provide hints at these critical issues and serve as extremely useful springboards for experiments focused on elucidating the energetic architecture and dynamic state of atoms in proteins.

Perturbation Analysis: Mutagenesis and Structures

Heterogeneous energy distribution and long-range effects

A powerful and commonly employed technique used to probe the energetic architecture of proteins is to measure the effect of perturbations to the system through site-directed mutagenesis [14]. With X-ray crystal structures as guides, measurements of the energetic effect of perturbations to a system have revealed several critical physical features in the energetics of amino acid interactions.

A seminal application of this strategy focused on understanding the source of the binding energy of human growth hormone (hGH) to its receptor (hGH-R) [15]. Alanine scanning mutagenesis of the approximately thirty receptor amino acids comprising a

majority of the interface showed that the positions did not contribute uniformly to the binding energy. Instead, two tryptophan residues packing with each other and comprising only a small fraction of the interface accounted for the majority of the binding energy, giving rise to the term binding “hot spot”. Interestingly, kinetic measurements suggested that the tryptophans are cooperatively positioned through the supportive packing of several other positions [16]. Thus, binding energy seems to be parsed in a highly heterogeneous way, with many residues having little contribution and a few making large contributions. Importantly, this heterogeneity is not obvious in crystal structures of the protein complex.

A particularly striking biological example of functional tuning through mutagenesis occurs in the evolution of the immune response to an antigen. In a phenomenon known as affinity maturation, each time the immune system encounters a particular antigen it produces antibodies that bind to that specific antigen with increasing affinity. This adaptive response depends on somatic hypermutation of antibody genes. Identification of the locations of mutations responsible for the improved affinity of one antibody-antigen interaction surprisingly revealed that the mutations occurred not in the interface, but at some distance from it [17]. This further demonstrates not only the heterogeneous nature of the underlying energetic architecture in proteins but also that this feature plays a critical role in the evolution of function.

The same heterogeneity of residue contribution seems to underlie energy transmission through proteins. In an effort to understand the physical mechanism by which energy propagates through proteins, several groups undertook a structural perturbation analysis based on the idea that the structural effects of a mutation should reflect the underlying energetic architecture. Structural studies of mutations in T4

lysozyme, gene V protein, lambda repressor, and other proteins showed that, in general, proteins have an inherent tolerance for perturbation at many sites that allows them to dissipate structural changes by distributing the effect over a large region [18-20]. Single site mutations often cause effects that are greatest near the site of mutation and rapidly decrease radially in all directions. For example, the structure of the Staphylococcal nuclease mutant V66K was almost unchanged from wild type except in the immediate vicinity of position 66 [21]. However, other studies have revealed, in some places, that structural effects of perturbations are context dependent, asymmetric, and sometimes cause effects over significant distance [22-25]. For example, the ribonuclease-S mutant M13F shows an unexplainable 1.5 A movement of a loop 20A away from the site of mutation [26]. Crystallography studies of hemoglobin [27], serine proteases [28], and dihydrofolate reductase [29] also show that ligand/substrate binding or mutation to these proteins induces long-range structural changes not predictable from ground state structures. Thus, both structural and thermodynamic perturbation experiments indicate energy distribution in proteins is highly heterogeneous and allows long-range propagation of perturbations.

Probing cooperativity through mutant cycles

Functional measurements have revealed that proteins have highly cooperative structures in which the energetic contribution of a set of positions is not simply the sum of their individual contributions. Analysis of single mutations, however, only reveals the contribution of individual amino acids to structure and function and does not reveal the energetic interactions among positions in a protein. Cooperative interactions among

amino acids can be estimated through a formalism known as thermodynamic mutant cycle analysis [30]. This method estimates the interaction between two positions by measuring the effect of a mutation at one position in two different conditions: 1) in a wild type background, and 2) in the background of a mutation at the second position (figure 1-1). The difference between these two energies, $\Delta\Delta G^{mut}$, defines the thermodynamic coupling between these two mutations and reports the extent to which the mutations feel one another. If the two mutations are completely independent of each other, the thermodynamic coupling energy is zero. This condition is also referred to as additivity, since the effect of this double mutant is the additive energetic effect of the single mutants. However, deviation from zero demonstrates that the mutations interact, although the mechanism of their interaction is not revealed. This condition is referred to as energetic non-additivity of the mutation pair. It is important to remember that this coupling energy ($\Delta\Delta G^{mut}$) is the coupling due to the mutations, and equals the native coupling energy only at the limit were the mutations are complete and pure loss of function. Nevertheless,

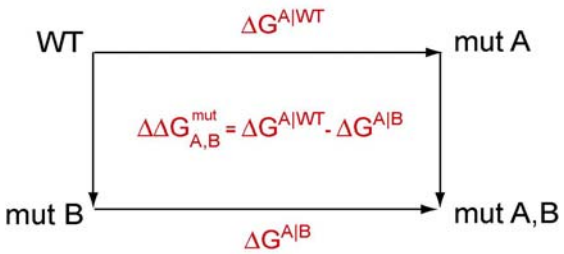


Figure 1-1. Thermodynamic mutant cycle analysis. The energetic interaction of sites can be estimated by measuring the effect of mutation A at one site in two different conditions: 1) WT background ($\Delta G^{A|WT}$), and 2) in the background of a mutation B at a second site ($\Delta G^{A|B}$). The difference of these two values ($\Delta\Delta G_{A,B}$) reports the extent to which the mutations feel each other and defines the thermodynamic coupling of the two mutations. If the effects of the mutations are completely independent, or additive, then $\Delta\Delta G^{mut} = 0\,kT*$. If, however, the effects are different then the mutations are said to be thermodynamically coupled to the extent given by $\Delta\Delta G^{mut}$.

with subtle mutagenesis, this approach can help probe the energetic constraints between sites or proteins.

Thermodynamic mutant cycle analysis has been used to dissect intra- as well as intermolecular interactions [31-33]. Though no experiment has totally saturated a protein with double mutant cycles, available data paint a picture of amino acid interaction that resembles results from studies of the protein-protein interaction surface. Most sites energetically couple only locally, while a few couple anisotropically at a distance. Data from numerous systems including hemoglobin, staphylococcal nuclease, tyrosyl-tRNA synthetase show that sparse long-range nonadditivity is a common feature of the energetic architecture of proteins [34, 35]. These results again highlight the highly anisotropic and cooperative nature of energy propagation in proteins.

All of these measurements are thermodynamic in nature and provide no mechanism for energy propagation between sites in a protein. A structural analogue of the thermodynamic cycle, termed the structure cycle, could give some insight into how two positions interact [30, 36]. Following the logic described above, this method compares the structural effect of a mutation on each atom of the protein in two different scenarios: 1) in a wildtype background, and 2) in the background of another mutation. While this method has only been applied in a few systems, the data show that mutation pairs that are thermodynamically independent are, as expected, also structurally additive [36]. Structures of cycles showing thermodynamic nonadditivity have identified particular structural regions that respond to a mutation differently in one background than another [19, 29, 37]. These results again support the view that the physical nature of proteins supports fracture-like propagation of energy.

While thermodynamic cycle analysis only reveals pairwise interactions, functional behavior may involve higher order cooperativity. Detection of higher order interactions can simply be envisioned as an expansion of the basic formalism described above. For example, three-way interactions could be interrogated with thermodynamic cubes and four-way interactions with hypercubes [34]. It quickly becomes apparent that, while thermodynamic and structural cycles are well-suited to study a limited set of positions, they are impractical for a complete mapping of even pairwise energetic interactions in a protein due to the massive numbers of mutants that would be required. In addition, the structure cycle analysis would require very high resolution data since very small displacements may account for significant thermodynamic differences. These limitations highlight the need for an alternate method to measure the global mapping of energetic interactions in the protein.

The core is critical for structure and function

Hydrophobicity vs. Packing

One of the observations from mutagenesis experiments of several systems has been that core positions are more sensitive to perturbation than surface positions [38]. Since these observations suggested that the core may encode structurally and functionally important information, many studies attempted to understand the energetic architecture of this region in more detail.

Protein structures show that cores generally consist of hydrophobic side chains with tight, jigsaw puzzle-like packing nearly as dense as organic molecule crystals [39]. This observation suggests the energy in the core may have at least two major sources:

hydrophobic interactions among side chains and the energy of packing. Analysis of sequence alignments revealed that hydrophobicity patterns are indeed one of the most conserved features in fold families [40, 41]. Furthermore, proteins are thought to derive their stability from the hydrophobic collapse of the core [40]. If hydrophobicity is the sole determinant of stability then mutations conserving the core hydrophobicity content should have similar effects on stability. Experiments testing this hypothesis in several systems including T4 lysozyme and gene V protein, however, showed clear context dependence of the effect of mutations on stability [19, 42, 43]. Furthermore, mutagenesis of multiple adjacent positions in the cores of Staphylococcal nuclease and lambda repressor showed that while nonpolar to nonpolar mutations were tolerated, few had close to wild type activity [44, 45]. Comparison of mutant sequences and their energetic effects showed that shape and total volume were constrained among those with wild type-like activity [45]. These experiments suggest that the specific packing of residues in the interior is at least as important as the hydrophobic effect and may even dominate the energetic stability of proteins [46]. The anisotropy of packing may be a plausible explanation for the heterogeneous and anisotropic pattern of free energy couplings in proteins.

Parsing of packing energy in protein cores

Packing is defined as the optimization of van der Waals interactions and the minimization of cavities [39]. It is thus a function of the sizes and shapes of the side chains and is closely related to the distribution of free energy in the protein. Tight packing optimizes attractive van der Waals interactions and therefore increases the

enthalpic contribution to the free energy. Tight packing also, however, minimizes the number of states and decreases entropy. Throughout the core, the balance of these two terms, enthalpy from tight packing and entropy from loose packing, is expected to determine the net free energy value of amino acid interactions [39]. This argument is not to discount other potential sources of free energy, but simply to state the dominant forces operating in the core.

What is the distribution of packing energy in proteins? Experiments in GroEL [47] and lambda repressor [48] have shown that proteins are not optimally packed, as originally suggested by comparisons of protein and organic crystal structures. Mutations in these and other systems have been found to improve packing and increase stability. This observation makes intuitive sense: an optimally packed protein would be less dynamic and less tolerant of mutation, features that would decrease function and evolvability. Nonetheless, proteins generally display a significant sensitivity to mutations in the core, especially to those that replace small side chains with larger ones [38]. Together, these observations indicate that proteins are not optimally packed but close to it.

To more carefully map the energetics of packing, several studies have used thermodynamic mutant cycle analysis. In one notable example, the cooperativity of packing in the core of Staphylococcal nuclease was dissected through an array of single, double, triple and quadruple mutants [43, 49-51]. The data showed, as expected, that adjacent positions formed highly cooperative arrangements. The packing of the core did not involve any tested higher order interactions; rather, it could be approximated as a series of short range pair-wise interactions. The authors of this work conclude that such pair-wise packing is a key selection factor in the evolution of proteins. A criticism of this work is that the coupling of adjacent sites may just reflect the average spatial correlation

distance of mutagenesis. Also, the failure to find high order couplings may just reflect the experimental limitation on the number of sites tested, and the fact that fold stability, not function, was assayed.

Structural studies of the core

Since packing is a function of shapes and volumes, an analysis of the structural effects of core mutations provides a particularly relevant approach to probe packing interactions in the core. In general, the cores showed plasticity, responding to minimize the volume changes of mutations. Backbone and side chains adjusted to minimize cavities or accommodate additional atoms; the structural adjustments were usually largest near the mutated site and decreased radially [39]. However, significant heterogeneities in the structural responses were observed. T4 lysozyme mutants showed some regions were less able to fill cavities than others suggesting local heterogeneity in rigidity [24]. Furthermore, long-range effects to mutations have also been observed; for example, the A98V mutation in the T4 lysozyme core causes structural changes 15 A away [52]. In thioredoxin, hydrogen exchange data showed that a core mutation causes a change in the dynamics at a distant site [53].

A further demonstration of heterogeneity in local packing comes from structural studies of mutants in green fluorescent protein (GFP). In an attempt to understand the mechanism of energetic cooperativity in cores of proteins, one group employed a structure cycle approach, solving structures of GFP mutants corresponding to three thermodynamic cycles [54]. Single mutations caused structural changes that overlapped in multiple regions. However, double mutant structures showed that thermodynamic

coupling resulted not from the entire region of overlap but from a specific structural subset [54]. Demonstration of such structural hotspots in GFP, a particularly rigidly packed protein, indicates that plasticity in the core allows for decomposition of energetic coupling.

Taken together, the mutagenesis and structural data argue for an interesting architecture in the protein core: many sites of weak, local coupling but a few sites of strong, propagated coupling. Such a heterogeneous architecture for packing is consistent with a structural basis for specific, coordinated, long-range interaction between functional surfaces. Evidence of regional differences in plasticity also hints at an often ignored dynamic dimension of the protein structure.

Dynamics are critical for function

Numerous lines of evidence indicate that proteins are highly dynamic materials and that the temporal scale of atomic fluctuations is a critical component of their energetic map. Early high temperature time-resolved X-ray crystallography experiments of myoglobin demonstrated that proteins are highly dynamical systems [55]. Structures of the inactive and active states of G protein (discussed above) emphasize how energetic perturbations at one site can cause significant changes in the dynamic state at a distant site. How does the jostling of atoms in one region propagate through the structure? Ultimately, a complete understanding of a protein should involve a description of the atomic trajectories as the protein performs its specific function. Motions of the atoms in a protein are critical for function though this dimension of protein behavior has been difficult to measure until recently. Recent developments in NMR spectroscopy have

allowed measurement of the rates of chemical exchange [56]. In general, atomic fluctuations occur on time scales ranging from femtoseconds to days. Mapping chemical exchange rates and their changes should provide significant insight into both the energetic map and mechanisms of energy propagation in proteins. Below, I review some recent work in relating protein dynamics to function.

Dynamics are heterogeneous

Several studies have attempted to characterize the fast time scale (picosecond to nanosecond) fluctuations of side chains. These fast motions are particularly interesting as they are related to the number of states explored by the protein. Thus, fast motions reflect, to some extent, the entropy of the protein in the folded state, often referred to as the residual entropy since it is significantly less than the entropy of the unfolded state [56]. ^{15}N relaxation measurements on numerous systems have generally revealed that the main chain atoms are essentially rigid [56]. This rigidity, however, seems to serve as a scaffold for more varied dynamics in side chains. Measurements of ^{13}C and ^{15}N relaxation suggest that side chains are often very dynamic and are sometimes decoupled from backbone motions [56].

One such relaxation experiment measured the fast time scale dynamics of calmodulin side chains. The data showed that the amplitudes of motion fell in a heterogeneous spectrum roughly divisible into three groups [57]. The authors also showed that retrospective analysis of previously published data found the same three categories of fast dynamics in other systems [57]. The observation of dynamic heterogeneity raises the possibility of addressing two important features of protein

dynamics: 1) the rigorous correlation between function and temporal fluctuations and 2) conservation of patterns of dynamics within protein families.

Dynamics correlate with function

NMR experiments have begun to probe the relationship between atomic fluctuations and function. Many studies have focused on correlations in changes in slow time scale (microsecond to millisecond) dynamics since this is the approximate time scale of many biological events. A natural system in which to probe the correlation of dynamics and function is an enzyme since these proteins have characteristic turnover rates that might be compared to the time scale of atomic fluctuations. One group measured backbone dynamics in human cyclophilin A (CypA), a peptidyl-prolyl cis/trans isomerase, in the presence and absence of substrate and found a physically connected set of amino acids with micro- to millisecond dynamics [58]. The authors point out that the positions in this "dynamic hotspot" are critical in binding substrate; importantly, the slow time scale of their chemical exchange correlates well with substrate turnover (~ 10,000 s^{-1}) [58].

Proteins involved in signaling have been selected to reliably transfer information and thus also represent particularly relevant systems to understand the link between dynamics and function. ^{15}N relaxation experiments of SpoOF, a response-regulator protein from *Bacillus subtilis*, identified a group of adjacent residues that experience slow time scale dynamics; intriguingly this 'hot spot' of slow dynamics involves the same residues that form a protein-protein interaction surface [59]. Similar ^{15}N relaxation experiments probed the change in backbone dynamics of NtrC, a member of the two-component system signaling family, upon activation by phosphorylation [60]. The data

suggest that, in the unphosphorylated state, the domain is in equilibrium between active and inactive conformations; phosphorylation shifts this equilibrium towards the active conformation. Interestingly, upon phosphorylation slow dynamics disappear in a region of the protein, a region known to undergo structural change upon activation. The data from these two systems begins to build the case that slow-time scale fluctuations are related to function.

Others maintain that the functionally relevant motions are not necessarily only in the slow time scale regime [56, 61, 62] . Instead, it is possible that the free energy that drives functional processes comes from changes in the fast time scale (ns) fluctuations that affect the entropy of the system. One study analyzed the change in side chain fast dynamics upon peptide binding to calmodulin [63]. The data showed that side chains throughout the protein undergo a rigidification upon ligand binding. The authors suggested this change in internal entropy may be a critical mechanism for modulating binding affinity or propagating energy in signaling proteins. The evidence for the correlation of both fast and slow dynamics with function suggests that it is likely that both are critical components of the underlying mechanism. Perhaps energetic perturbations at one site cause changes in fast time scale motions in adjacent regions that are, in turn, coupled to slow time scale motions in other regions. The precise parsing of dynamics, however, will likely differ from one protein family to another and even between members of the same protein family. Regardless, these initial insights motivate a rigorous study of dynamics, energetics, and function in one system.

Coupled dynamics correlate with function

As discussed above, structural and mutagenesis data clearly show that atoms in proteins act cooperatively. Such cooperative units likely have distinctive dynamical features. Many have suggested cooperative interactions will be reflected by correlated motion on a slow time scale since they involve larger, more massive, units moving in unison. Currently, NMR relaxation measurements only characterize individual side chain dynamics but do not directly show coupling of motion. One recent study proposed a method for detecting correlated fast internal motions through analysis of dynamical changes induced by mutations [64]. The strategy is based on the idea that if two bond vectors have coupled dynamics, then a subtle mutation that affects one is likely to similarly affect the other. When applied to ^{15}N relaxation of the immunoglobulin G-binding domain of *Streptococcal* protein G, the data suggested there was a "network of correlated motions in which the dynamics of residues on opposite sides of the protein are sensitive to each other by virtue of intervening noncovalent interactions." [64] While this work also suggests heterogeneity and cooperativity in the dynamical dimension of protein G, it has yet to be determined if the network of correlated motions in this case is functionally relevant.

Significant understanding of the functional role of correlated atomic motions has come through very high resolution X-ray crystal structures and molecular dynamics simulations. Collection of very high resolution diffraction data (better than 1.2 A) allows more detailed modeling of electron density with anisotropic B factors. By increasing the number of parameters describing electron density, anisotropic B factors allow depiction of electron clouds surrounding atoms as three dimensional ellipsoids rather than as spheres thus revealing the major axes of motion. Although no information about the time

scale of motion can be ascribed to these pictures, they give very useful insights into the co-variation of atomic motion. For example, a 0.82 A structure of photoactive yellow protein (PYP) in the dark-adapted ground state shows that a group of atoms in the active site show motion that anticipates the initial stages of double bond isomerization in the 4-hydroxycinnamic acid chromophore, the critical step in photoactivation of PYP [65]. Similar evidence for correlated motion has also been found in other proteins including HIV protease [66]. These concerted motions are yet another dramatic example of an innately biased energy landscape in which cooperative interactions of several residues are tuned for functionality.

Atomic trajectories from molecular dynamics simulations, if experimentally corroborated, can provide a powerful means of probing the link between synchronized dynamics and function. A recent study used normal mode analysis to understand the physical basis of α-lytic protease specificity [3]. Previous structural and mutagenesis experiments showed this enzyme derives its specificity from the interaction of a small residue (preferably Ala) on the peptide substrate with its primary specificity pocket. A simulation of the peptide-free state shows the atoms comprising its primary specificity pocket are involved in highly correlated motion such that the walls of the pocket vibrate in phase and maintain a constant pocket volume [3]. Interestingly, a mutation known to increase promiscuous activity was found to disrupt this correlated motion and allow out of phase vibrations giving the pocket the ability to accommodate substrates of different sizes. Thus, local energetic interactions have been tuned to modulate conformational plasticity and manifest as coordinated regional dynamics.

Another recent example demonstrating the importance of flexibility and coupled motion came in a study of Src, an allosterically regulated tyrosine kinase [67]. Src

consists of three domains: the Src homology SH2 and SH3 domains followed by a kinase domain. Previous work showed that phosphorylation of a tyrosine in the C-terminal tail of the protein results in an intramolecular association between the SH2 domain and the phosphorylated C terminal tail somehow causing a decrease in the kinase activity. To understand the physical basis of this allostery, one group used targeted molecular dynamics to simulate the atomic trajectories of the protein in several conformations. When the SH2 domain is bound to phosphorylated C terminal tail, simulations reveal strong correlated motion between the SH2 and SH3 domains. Furthermore, simulations suggested this correlated motion depends on the rigidity of the linker between the SH2 and SH3 domains. If true, loss of rigidity by mutation of critical linker positions to glycine would destroy correlated motion rendering the kinase insensitive to phosphorylation state. Indeed, yeast expression assays suggest such mutants are constitutively active, thus demonstrating the functional importance of appropriate tuning of hotspots of energy transfer. Overall, the model that is beginning to emerge is that proteins are dynamically heterogeneous, with certain regions undergoing concerted fluctuations that contribute to function.

Conclusions and previous work from the lab

While the ‘sensors’ and ‘actuators’ of any single protein have not been described to complete atomistic understanding, this brief review demonstrates several critical physical features that have emerged. First, the specific arrangement of atoms in a protein establishes a network of energetic interactions that dictate structure and function. Each protein has a particular pattern of energetic interactions between residues that represents

an evolutionary balance between the need to achieve requisite stability and to perform function appropriately. The interactions support cooperative behavior among atoms to create coordinated responses to energetic perturbation. Accordingly, the pattern of energetic interaction is highly heterogeneous and the structure supports long range transfer of energy.

In any protein system, applications of all the techniques discussed in the review above are focused on the same two issues stated at the outset of this chapter: 1) the energetic manifold, and 2) the mechanisms of energy transfer. However, while the methodologies have revealed important insights, they provide only partially complementary data sets. Mutagenesis experiments provide thermodynamic mappings but do not reveal mechanism; technical limitations preclude large scale application and limit us to regional and low-order studies. Structural studies provide the detailed three-dimensional organization of proteins that gives some insight into mechanism but are devoid of energetic interactions and dynamic information. NMR relaxation measurements have only recently been developed and promise to bridge the dynamic dimension of the protein with function but, so far, only through limited temporal and regional windows. Given their limitations, it is clear that an energetic mapping by these methods will, at best, be incomplete. In the absence of an energetic map these methods, even in combination, can only give limited understanding into the inner workings of the ‘sensors’ and ‘actuators’ of proteins.

Previous work from our lab sought to reveal a more complete characterization of the energetic architecture in proteins [68, 69]. Motivated by the idea that important interactions in proteins should be conserved through evolution, previous lab members developed an algorithm, termed the statistical coupling analysis (SCA), which extracts

amino acid coupling information from the evolutionary record of a protein family. The results of SCA show good agreement with experimental data in several protein families, verifying that it indeed captures critical physical interactions in the protein. This method and its results formed the basis for my thesis work. The next chapter describes an improved form of SCA that produces more complete and accurate measures of the co-evolution of positions in a protein. Chapter 3 presents thermodynamic and crystallography experiments focused on understanding the physical mechanism underlying a cooperative interaction among positions in the interface between a PDZ domain and peptide ligand. The data show that the coupled interaction is critical for maintaining binding specificity and evolvability; conformational flexibility is necessary to optimize the coupled energetics though it sacrifices binding affinity. Chapter 4 provides an analysis of the global topology of the energetic maps of several proteins as provided by SCA. Overall, my thesis work:

1) provides, for the first time, a hypothesis for the global architecture of amino acid interactions,
2) provides a mechanistic dissection to test this hypothesis through crystallographic and thermodynamic analysis in one protein, and
3) shows that this global architecture is conserved in many disparate protein families.

References

1. Carl Branden, J.T., *Introduction to Protein Structure*. 1999: Garland Publishers.
2. Vale, R.D. and R.A. Milligan, *The way things move: looking under the hood of molecular motor proteins.* Science, 2000. **288**(5463): p. 88-95.

3. Miller, D.W. and D.A. Agard, *Enzyme specificity under dynamic control: a normal mode analysis of alpha-lytic protease.* J Mol Biol, 1999. **286**(1): p. 267-78.
4. McCallister, *Introduction to Materials Science Engineering.*
5. Sakmar, T.P., et al., *Rhodopsin: insights from recent structural studies.* Annu Rev Biophys Biomol Struct, 2002. **31**: p. 443-84.
6. Sprang, S.R., *G protein mechanisms: insights from structural analysis.* Annu Rev Biochem, 1997. **66**: p. 639-78.
7. Yin, J., et al., *Structural plasticity and the evolution of antibody affinity and specificity.* J Mol Biol, 2003. **330**(4): p. 651-6.
8. Shoichet, B.K., et al., *A relationship between protein stability and protein function.* Proc Natl Acad Sci U S A, 1995. **92**(2): p. 452-6.
9. Fersht, A., *Structure and mechanism in protein science : a guide to enzyme catalysis and protein folding*. 1999, New York: W.H. Freeman.
10. Pauling L, C.D., Pressman D, Physiology Reviews, 1943. **23**: p. 203-219.
11. Richards, F.M., *Protein stability: still an unsolved problem.* Cell Mol Life Sci, 1997. **53**(10): p. 790-802.
12. Lo Conte, L., C. Chothia, and J. Janin, *The atomic structure of protein-protein recognition sites.* J Mol Biol, 1999. **285**(5): p. 2177-98.
13. Fuhrmann, C.N., et al., *The 0.83 A resolution crystal structure of alpha-lytic protease reveals the detailed structure of the active site and identifies a source of conformational strain.* J Mol Biol, 2004. **338**(5): p. 999-1013.
14. Wells, J.A., *Systematic mutational analyses of protein-protein interfaces.* Methods Enzymol, 1991. **202**: p. 390-411.
15. Clackson, T. and J.A. Wells, *A hot spot of binding energy in a hormone-receptor interface.* Science, 1995. **267**(5196): p. 383-6.
16. Clackson, T., et al., *Structural and functional analysis of the 1:1 growth hormone:receptor complex reveals the molecular basis for receptor affinity.* J Mol Biol, 1998. **277**(5): p. 1111-28.
17. Patten, P.A., et al., *The immunological evolution of catalysis.* Science, 1996. **271**(5252): p. 1086-91.
18. Xu, J., et al., *The response of T4 lysozyme to large-to-small substitutions within the core and its relation to the hydrophobic effect.* Protein Sci, 1998. **7**(1): p. 158-77.
19. Zhang, H., et al., *Context dependence of mutational effects in a protein: the crystal structures of the V35I, I47V and V35I/I47V gene V protein core mutants.* J Mol Biol, 1996. **259**(1): p. 148-59.
20. Eigenbrot, C. and A.A. Kossiakoff, *Structural consequences of mutation.* Curr Opin Biotechnol, 1992. **3**(4): p. 333-7.
21. Stites WE, A.G., Lattman EE, Shortle D, *In a Staphylococal Nuclease Mutant the Side-chain of a Lysine Replacing Valine 66 is Fully Buried in the Hydrophobic Core.* J Mol Biol, 1991. **221**: p. 7-14.
22. Vaughan, C.K., A.M. Buckle, and A.R. Fersht, *Structural response to mutation at a protein-protein interface.* J Mol Biol, 1999. **286**(5): p. 1487-506.
23. Eriksson, A.E., W.A. Baase, and B.W. Matthews, *Similar hydrophobic replacements of Leu99 and Phe153 within the core of T4 lysozyme have different structural and thermodynamic consequences.* J Mol Biol, 1993. **229**(3): p. 747-69.

24. Eriksson, A.E., et al., *Response of a protein structure to cavity-creating mutations and its relation to the hydrophobic effect.* Science, 1992. **255**(5041): p. 178-83.
25. Hellinga, H.W., R. Wynn, and F.M. Richards, *The hydrophobic core of Escherichia coli thioredoxin shows a high tolerance to nonconservative single amino acid substitutions.* Biochemistry, 1992. **31**(45): p. 11203-9.
26. Varadarajan, R. and F.M. Richards, *Crystallographic structures of ribonuclease S variants with nonpolar substitution at position 13: packing and cavities.* Biochemistry, 1992. **31**(49): p. 12315-27.
27. Paoli, M., et al., *Crystal structure of T state haemoglobin with oxygen bound at all four haems.* J Mol Biol, 1996. **256**(4): p. 775-92.
28. Perona, J.J., et al., *Structural origins of substrate discrimination in trypsin and chymotrypsin.* Biochemistry, 1995. **34**(5): p. 1489-99.
29. Brown, K.A., E.E. Howell, and J. Kraut, *Long-range structural effects in a second-site revertant of a mutant dihydrofolate reductase.* Proc Natl Acad Sci U S A, 1993. **90**(24): p. 11753-6.
30. Carter, P.J., et al., *The use of double mutants to detect structural changes in the active site of the tyrosyl-tRNA synthetase (Bacillus stearothermophilus).* Cell, 1984. **38**(3): p. 835-40.
31. Hidalgo, P. and R. MacKinnon, *Revealing the architecture of a K+ channel pore through mutant cycles with a peptide inhibitor.* Science, 1995. **268**(5208): p. 307-10.
32. Schreiber, G. and A.R. Fersht, *Energetics of protein-protein interactions: analysis of the barnase-barstar interface by single mutations and double mutant cycles.* J Mol Biol, 1995. **248**(2): p. 478-86.
33. Pineda, A.O., et al., *The thrombin epitope recognizing thrombomodulin is a highly cooperative hot spot in exosite I.* J Biol Chem, 2002. **277**(35): p. 32015-9.
34. Horovitz, A. and A.R. Fersht, *Strategy for analysing the co-operativity of intramolecular interactions in peptides and proteins.* J Mol Biol, 1990. **214**(3): p. 613-7.
35. LiCata, V.J. and G.K. Ackers, *Long-range, small magnitude nonadditivity of mutational effects in proteins.* Biochemistry, 1995. **34**(10): p. 3133-9.
36. Sandberg, W.S. and T.C. Terwilliger, *Engineering multiple properties of a protein by combinatorial mutagenesis.* Proc Natl Acad Sci U S A, 1993. **90**(18): p. 8367-71.
37. Vaughan, C.K., et al., *A structural double-mutant cycle: estimating the strength of a buried salt bridge in barnase.* Acta Crystallogr D Biol Crystallogr, 2002. **58**(Pt 4): p. 591-600.
38. Baldwin, E.P. and B.W. Matthews, *Core-packing constraints, hydrophobicity and protein design.* Curr Opin Biotechnol, 1994. **5**(4): p. 396-402.
39. Richards, F.M. and W.A. Lim, *An analysis of packing in the protein folding problem.* Q Rev Biophys, 1993. **26**(4): p. 423-98.
40. Dill, K., *Dominant Forces in Protein Folding.* Biochemistry, 1990. **29**: p. 7133-7155.
41. Bowie, J.U., et al., *Deciphering the message in protein sequences: tolerance to amino acid substitutions.* Science, 1990. **247**(4948): p. 1306-10.

42. Baldwin, E., et al., *Thermodynamic and structural compensation in "size-switch" core repacking variants of bacteriophage T4 lysozyme.* J Mol Biol, 1996. **259**(3): p. 542-59.
43. 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.
44. Lazar, G.A. and T.M. Handel, *Hydrophobic core packing and protein design.* Curr Opin Chem Biol, 1998. **2**(6): p. 675-9.
45. Pakula, A.A. and R.T. Sauer, *Amino acid substitutions that increase the thermal stability of the lambda Cro protein.* Proteins, 1989. **5**(3): p. 202-10.
46. Ratnaparkhi, G.S. and R. Varadarajan, *Thermodynamic and structural studies of cavity formation in proteins suggest that loss of packing interactions rather than the hydrophobic effect dominates the observed energetics.* Biochemistry, 2000. **39**(40): p. 12365-74.
47. Wang, Q., A.M. Buckle, and A.R. Fersht, *Stabilization of GroEL minichaperones by core and surface mutations.* J Mol Biol, 2000. **298**(5): p. 917-26.
48. Lim, W.A., et al., *The crystal structure of a mutant protein with altered but improved hydrophobic core packing.* Proc Natl Acad Sci U S A, 1994. **91**(1): p. 423-7.
49. Chen, J. and W.E. Stites, *Higher-order packing interactions in triple and quadruple mutants of staphylococcal nuclease.* Biochemistry, 2001. **40**(46): p. 14012-9.
50. Chen, J. and W.E. Stites, *Energetics of side chain packing in staphylococcal nuclease assessed by systematic double mutant cycles.* Biochemistry, 2001. **40**(46): p. 14004-11.
51. Holder, J.B., et al., *Energetics of side chain packing in staphylococcal nuclease assessed by exchange of valines, isoleucines, and leucines.* Biochemistry, 2001. **40**(46): p. 13998-4003.
52. Daopin, S., et al., *Structural and thermodynamic analysis of the packing of two alpha-helices in bacteriophage T4 lysozyme.* J Mol Biol, 1991. **221**(2): p. 647-67.
53. De Lorimier, R., H.W. Hellinga, and L.D. Spicer, *NMR studies of structure, hydrogen exchange, and main-chain dynamics in a disrupted-core mutant of thioredoxin.* Protein Sci, 1996. **5**(12): p. 2552-65.
54. 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.
55. Frauenfelder, H., S.G. Sligar, and P.G. Wolynes, *The energy landscapes and motions of proteins.* Science, 1991. **254**(5038): p. 1598-603.
56. Wand, A.J., *Dynamic activation of protein function: a view emerging from NMR spectroscopy.* Nat Struct Biol, 2001. **8**(11): p. 926-31.
57. Lee, A.L. and A.J. Wand, *Microscopic origins of entropy, heat capacity and the glass transition in proteins.* Nature, 2001. **411**(6836): p. 501-4.
58. Eisenmesser, E.Z., et al., *Enzyme dynamics during catalysis.* Science, 2002. **295**(5559): p. 1520-3.
59. Feher, V.A. and J. Cavanagh, *Millisecond-timescale motions contribute to the function of the bacterial response regulator protein Spo0F.* Nature, 1999. **400**(6741): p. 289-93.
60. Volkman, B.F., et al., *Two-state allosteric behavior in a single-domain signaling protein.* Science, 2001. **291**(5512): p. 2429-33.

61. Cooper, A. and D.T. Dryden, *Allostery without conformational change. A plausible model.* Eur Biophys J, 1984. **11**(2): p. 103-9.
62. Wand, A.J., *On the dynamic origins of allosteric activation.* Science, 2001. **293**(5534): p. 1395.
63. Lee, A.L., S.A. Kinnear, and A.J. Wand, *Redistribution and loss of side chain entropy upon formation of a calmodulin-peptide complex.* Nat Struct Biol, 2000. **7**(1): p. 72-7.
64. Mayer, K.L., et al., *Covariation of backbone motion throughout a small protein domain.* Nat Struct Biol, 2003. **10**(11): p. 962-5.
65. Getzoff, E.D., K.N. Gutwin, and U.K. Genick, *Anticipatory active-site motions and chromophore distortion prime photoreceptor PYP for light activation.* Nat Struct Biol, 2003. **10**(8): p. 663-8.
66. Reiling, K.K., et al., *Anisotropic dynamics of the JE-2147-HIV protease complex: drug resistance and thermodynamic binding mode examined in a 1.09 A structure.* Biochemistry, 2002. **41**(14): p. 4582-94.
67. Young, M.A., et al., *Dynamic coupling between the SH2 and SH3 domains of c-Src and Hck underlies their inactivation by C-terminal tyrosine phosphorylation.* Cell, 2001. **105**(1): p. 115-26.
68. Lockless, S.W. and R. Ranganathan, *Evolutionarily conserved pathways of energetic connectivity in protein families.* Science, 1999. **286**(5438): p. 295-9.
69. Suel, G.M., et al., *Evolutionarily conserved networks of residues mediate allosteric communication in proteins.* Nat Struct Biol, 2003. **10**(1): p. 59-69.

Chapter 2 Measuring Evolutionary Coupling in Proteins

Introduction

Proteins achieve their structure and function through characteristic patterns of energetic interactions among their amino acid residues. As argued in chapter 1, a finely balanced and evolutionarily specified energetic architecture endows proteins with both necessary stability and the ability to reliably channel energetic perturbations into specific functions. Descriptions of the patterns of energetic interactions between amino acid residues are at the heart of understanding any function, whether binding, catalysis, or allosteric activation. While structural, mutagenesis, and dynamics experiments have provided invaluable hints about the energetic topology in local regions of proteins, they have generally not yielded a complete picture of the important interactions.

Work from our lab has attempted to globally map the energetic interactions between amino acids in proteins through a sequence-based statistical method known as the statistical coupling analysis (SCA) [1-3]. SCA estimates the thermodynamic interactions between sites in a protein by measuring the strength of their co-variation through evolution. Application of this method to several protein families revealed a surprising result: subsets of highly co-evolving residues formed physically connected networks including surface and core positions. Importantly, the residues identified by SCA in several protein families also showed strong correlation with functional data [4, 5]. As these results formed the foundation for my thesis work, I will briefly review the statistical coupling method and its results. In its original form, the results of this method, while in close agreement with functional data, did not give a completely global mapping.

After discussing this limitation, I will describe an improved form of the analysis. This new formalism extracts more information from a protein alignment and produces co-evolution measurements for all pairs of positions.

Statistical coupling analysis: site-specific perturbation

Measuring evolutionary energy at each site

In theory, comprehensive thermodynamic mutant cycle analysis applied to all pairs of positions in a protein would give an estimate of the complete map of pairwise interactions; in practice, however, the massive number of mutations required for such a data set renders such an analysis impractical to execute. In addition, mutagenesis only estimates interaction energies by introducing perturbations at sites. No real knowledge of the native interaction energies is given. A potential alternate approach is suggested by considering evolution as a large-scale experiment in mutagenesis. Proteins found in nature, after all, are evolutionary solutions to the structure-function problem achieved through systematic random mutagenesis with selection for function. A sequence alignment of homologous proteins, if diverse and containing many sequences, is a representative ensemble of sequences coding for a common overall structure and function. The distribution of amino acids in these sequences should then reflect their common physical constraints. The foundation of SCA rests on two simple concepts [2]. First, functionally important sites should be constrained through evolution and will thus show amino acid frequencies that differ from their mean amino acid frequencies in all proteins. As a corollary, positions that are not functionally constrained should have frequency

distributions that approach the mean. Second, energetic coupling between two sites, whether important for stability or function, should force these sites to co-evolve. Such co-evolution should be measurable in an alignment of sufficient size and diversity. These two ideas guide the definition of the statistical parameters used to measure conservation and co-variation in a protein alignment.

At the core of SCA is the view of a sequence alignment as a statistical ensemble near equilibrium. That is, the sequences: 1) have undergone sufficient mutagenesis to have randomized sites that are unconstrained, and 2) comprise a reasonable sampling of this diversity. If so, a deviation in the probability distribution of amino acids between two sites corresponds to a statistical free energy difference in a state space of all possible amino acid distributions. Consider the observed frequency of an amino acid *x* at a site *i*, $p_{x,i}$. Given the frequency of amino acid *x* in all proteins is p_x, the probability of getting $p_{x,i}$, denoted P_i^x, is given by the binomial density function:

$$P_i^x = \frac{N!}{n_x!(N-n_x)!} p_x^{n_x} (1-p_x)^{N-n_x} \qquad \text{(Eq. 2-1)}$$

Here *N* is a normalized number of sequences in the alignment and n_x is the normalized number of sequences with amino acid *x* at position *i*. Using this relation we can also determine the probability of the reference state; that is, the probability of getting amino acid *x*, P_{MSA}^x, at the mean frequency observed in the multiple sequence alignment (MSA), $p_{x,MSA}$. The use of the binomial density function has two purposes: 1) to quantitatively account for cases where a frequency of an amino acid is zero, and 2) to represent the intuitive notion that the evolutionary significance of changes in observed amino acid frequencies should be greater as the conservation of an amino acid increases. Thus, the

probability density ratio of an amino acid frequency changing from 0.6 to 0.65 should be greater than a change from 0.1 to 0.15 given a mean frequency of 0.05.

With the probabilities of the observed (P_i^x) and reference states (P_{MSA}^x) we can calculate the statistical energy difference between them using the Boltzmann distribution, which provides a relationship between the energy difference between two states and their probabilities:

$$\Delta G_i^x = kT^* \ln \frac{P_i^x}{P_{MSA}^x} \qquad x = \text{ala,arg,asp,...,val} \qquad \text{(Eq. 2-2)}$$

The statistical energy differences between the observed and reference states can be calculated for all amino acids at a site to give a twenty element statistical energy vector for each site:

$$\Delta \vec{G}_i = \left[\Delta G_i^{ala}, \Delta G_i^{\arg}, ...\right] \qquad \text{(Eq. 2-3)}$$

The magnitude of the vector in equation 2-3 defines the conservation parameter for site *i*:

$$\Delta G_i^{stat} = \left|\Delta \vec{G}_i\right| = kT^* \sqrt{\sum_x \left(\ln \frac{P_i^x}{P_{MSA}^x}\right)^2} \quad x = \text{ala,arg,asp,...,val} \qquad \text{(Eq. 2-4)}$$

This parameter has arbitrary units of kT* and is a measure of the total statistical energy at each site. At unconserved sites ΔG_i^{stat} approaches zero and increases with conservation.

This formalism can be used to calculate the probability for all amino acids at all sites, giving a 20 x *m* matrix of probabilities for the alignment, $\mathbf{P}_{PA}$, where *m* is the number of positions in the alignment and PA denotes parent alignment. These probabilities can be used to determine a 20 x *m* matrix of amino acid statistical energies, $\mathbf{\Delta G_{PA}}$. Calculation of the magnitude of the 20 element vector for each site gives a vector of *m* conservation values. For example, figure 2-1A shows the 94 conservation values for

a PDZ domain (PSD95, Discs large, Zo-1) alignment with 240 sequences and 94 positions. Mapping the values onto a representative PDZ domains structure (PDZ3 from PSD95, figure 2-1B) shows that highly conserved positions identify the active site of the protein. This definition of evolutionary energy provides a measure of the total constraint at a site, that is, the conservation of a site.

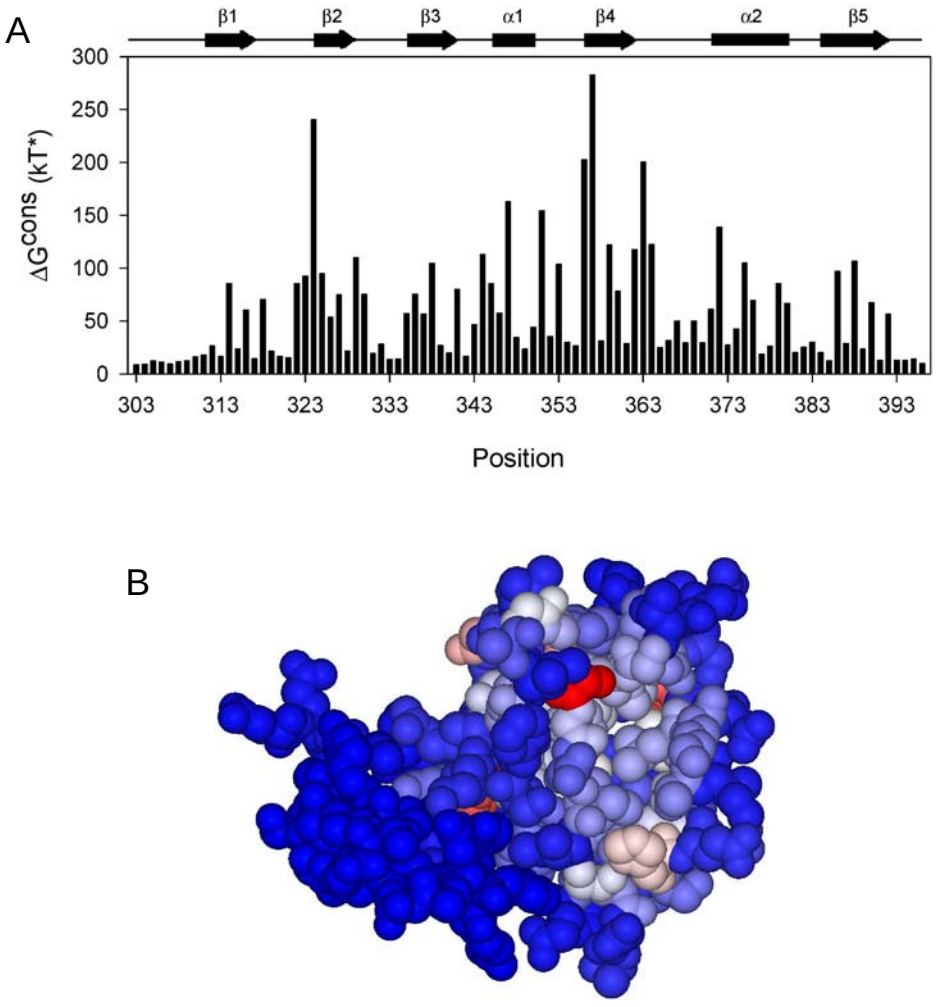


Figure 2-1 PDZ domain conservation energies. A) Conservation energies were calculated as described using an alignment with 240 sequences and 94 positions. The positions are numbered according to PDZ3 from PSD95 (PDB accession: 1BE9). The secondary structure of this domain is indicated above the graph. Clearly, most positions show low conservation and a subset rise above the noise. B) Conservation values mapped colorimetrically from blue (low) to red (high) on CPK rendition of PDZ3 shows that most conserved position occur in the active site of the protein.

Measuring evolutionary coupling by site-specific perturbation

Conservation reports the evolutionary energy at one site and is commonly used to indicate structural or functional importance. However, conservation at one site may not be independent of conservation at other sites. A measure of co-evolution between pairs of sites is interesting since it may represent the physical constraints between sites. Conservation analysis does not provide information about the co-conservation of positions through evolution. To reveal these interactions, SCA uses frequency perturbations in an approach that is a statistical analog of the thermodynamic mutant cycle analysis. The basic experiment is to perturb the amino acid distribution at a specific site and calculate the change in the conservation energy at all other sites. Specifically, a perturbation to the amino acid frequency distribution is made by extracting only sequences with a particular amino acid at a position *j*. The resulting subalignment must have sufficient size and diversity to be representative of the parent alignment [3] and still be at statistical equilibrium. Observed and reference probabilities can be calculated for each amino acid at each position of this subalignment exactly as described above and are denoted $P^x_{i|\delta j}$ and $P^x_{MSA|\delta j}$ respectively. Similarly, the Boltzmann equation can again be used to determine conservation energies for all amino acids at all sites of the subalignment. This gives the conservation energy vectors for site *i* in two different states: 1) in the parent alignment and 2) in the background of a perturbation at site *j*. The extent to which the distribution at site *i* depends on the distribution at *j* is simply captured by the magnitude of the difference between these two conservation energy vectors. This defines the statistical coupling between sites *i* and *j* and is expressed:

$$\Delta\Delta G^{stat}_{i,j} = \left|\Delta\vec{G}_i - \Delta\vec{G}_{i|\delta j}\right| = kT^* \sqrt{\sum_x \left(\ln \frac{P^x_{i|\delta j}}{P^x_{MSA|\delta j}} - \ln \frac{P^x_i}{P^x_{MSA}} \right)^2} \qquad \text{(Eq. 2-5)}$$

$$x = \text{ala}, \text{arg}, \text{asp}, ..., \text{val}$$

A useful vectorial depiction of the statistical perturbation experiment is shown in figure 2-2. The axes of this twenty-dimensional space measure the statistical energies of each amino acid at a position; for visualization, the first three amino acid dimensions are shown in the figure. A twenty-dimensional vector of statistical energies for a position (equation 2-3) in an alignment can be imagined as a vector in this space. Statistical

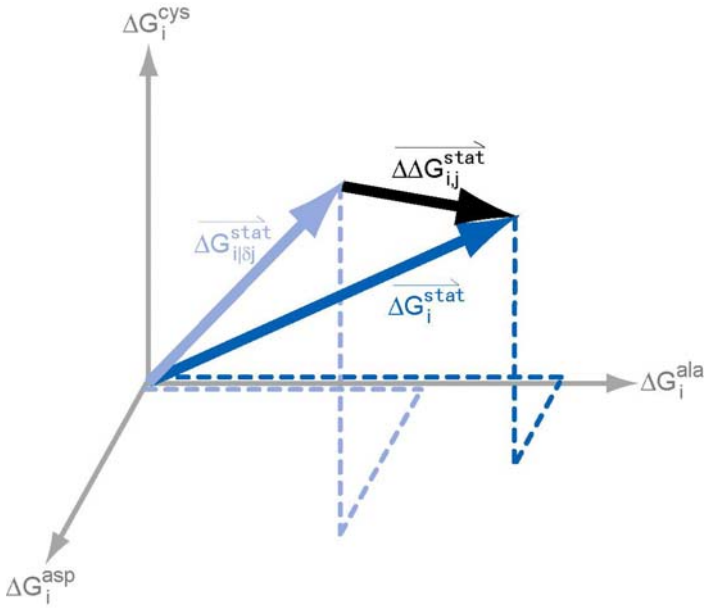


Figure 2-2. Vectorial representation of SCA by site specific perturbation. Axes represent three of twenty amino acid dimensions. The dashed lines represent component amino acid conservation energies for respective solid blue vectors. The solid dark blue vector represents the conservation energy vector for site *i* in the parent alignment. The light blue vector represents the conservation energy vector for the same site in the background of a statistical perturbation at site *j*. The difference in the two, depicted as the black vector, represents the statistical coupling of positions *i* and *j*. (Adapted from [1]).

coupling between sites *i* and *j* simply measures the displacement in the vector for site *i* caused by a frequency perturbation at site *j*.

Statistical coupling can be calculated for all sites and reports the extent to which the frequency at site *i* depends on the perturbation made at site *j*. Note that statistical perturbations can only be made when subalignments are representative of the parent

alignment [3]; due to the limitations of the subalignment size and diversity not all sites meet this criterion. Thus, a complete data set by this method consists of an *m* x *q* matrix of statistical coupling values, where *m* is the number of positions and *q* is the number of perturbations (and $q < m$). For example, 37 of 94 PDZ domain alignment positions meet the criteria for perturbation. Each column of the 94 x 37 **ΔΔG** matrix (figure 2-3, A-B) represents the statistical coupling energies to all positions in the protein for an individual

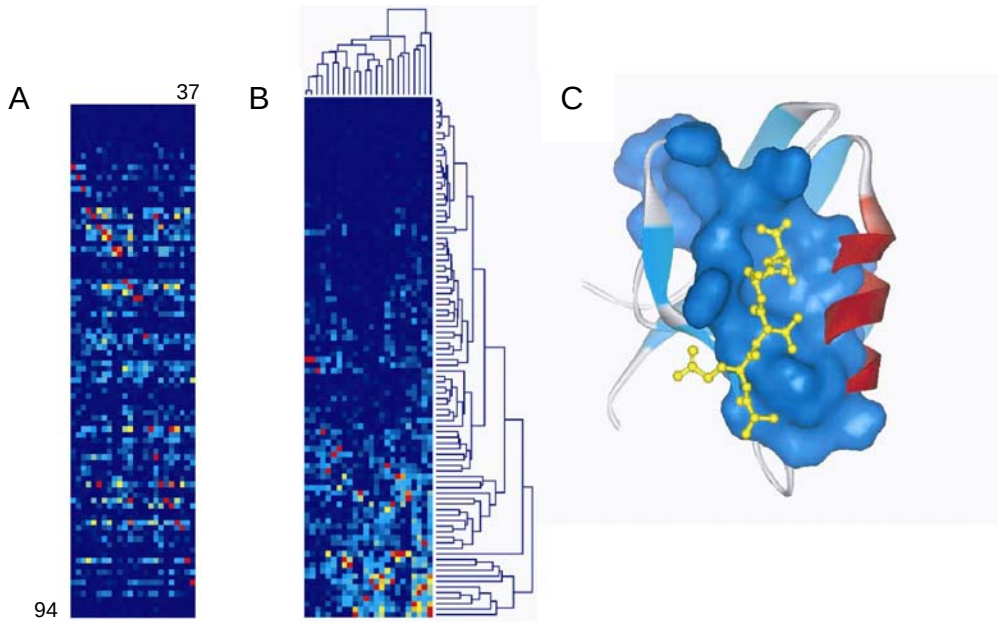


Figure 2-3. SCA results for PDZ domain by site specific perturbation. A) Rows represent 94 positions in the PDZ domain and columns represent 37 different perturbations. Statistical coupling values represented as gradient from blue (low) to red (high). B) Two dimensional clustering identifies a cluster of highly co-evolving positions. C) Mapped onto the structure of PSD95-PDZ3, these positions form a connected unit that includes the peptide binding pocket, part of the core, and positions on the back site of the domain. Co-crystalized peptide is shown in

statistical perturbation experiment at some position in the alignment. Two dimensional iterative clustering of this matrix extracts patterns of energetic interactions from this alignment (figure 2-3, B) and identifies a cluster of highly co-evolving positions.

Application of this method to the PDZ domain, G-protein coupled receptor, hemoglobin, and serine protease alignments revealed a common finding [3]. In all

families small subsets of highly co-evolving positions form structurally contiguous networks that include part of the protein core and link known functional sites (figure 2-3, C). The results of this method show the power of SCA to identify functionally important interactions. However, the $\Delta\Delta G^{stat}$ matrix shows two important practical issues related to the method of statistical perturbation. First, as noted above, the number of perturbations is limited and therefore prevents calculation of statistical couplings between all pairs of residues. Second, each perturbation is different in magnitude (since frequencies of amino acid at sites vary) and hence statistical coupling values between two positions are not reciprocally symmetric if interrogated by perturbations at both sites. That is, perturbation of *i* causes an effect on *j* which is not necessarily the same as how perturbation of *j* causes an effect on *i*. The incompleteness and asymmetry resulting from this methodology complicate analysis of the topology of the energetic interactions (to be discussed in chapter 4) and hence necessitate an alternate method to extract evolutionary couplings.

Measuring statistical coupling with small perturbations

Overview of method

The effects of the two fundamental postulates guiding the development of the statistical coupling analysis can be imagined in the twenty dimensional amino acid space described above. Energetically important positions will have frequency distributions that differ from the mean and, by definition, will have conservation energy vectors with larger magnitudes. Furthermore, energetic coupling between two positions should cause their respective conservation energy vectors to become correlated as mutations and selections

occur through the course of evolution. In its original form, SCA reveals this correlation by measuring the effect of a relatively large statistical perturbation at one position on the frequency distribution at all other positions. This approach, however, sacrifices much information in the alignment. By neglecting very low or high amino acid frequencies the original SCA provides no information about pairs of positions with low coupling. Furthermore, differences in the magnitudes of each perturbation results in asymmetric coupling energies between two positions.

An alternative approach is to measure the effects of many trials of much smaller statistical perturbations to the entire alignment. A small fluctuation in the frequency profile at a site can be imagined as a minor deflection in its corresponding conservation energy vector (figure 2-4A). A small perturbation to the entire alignment should cause a small deflection in the conservation energy vector for every position. For example, consider the effect of randomly selecting 50% of the alignment over many trials. If two positions *i* and *j* are evolutionarily coupled, their conservation energy vectors, $\Delta\vec{G}_i^{stat}$ and $\Delta\vec{G}_j^{stat}$ respectively, should fluctuate in a correlated manner through the course of many such small perturbations (figure 2-4, A). Such correlation is the result of amino acid co-evolution, where the component amino acid conservation energies of two coupled positions, ΔG_i^x and ΔG_j^x (where x = ala, arg, …, val), are coupled. In other words, the fluctuations in amino acid conservation energies, given by the difference between the parent and post-perturbation energies and depicted as $\Delta\Delta G_i^x$ in figure 2-4, A , should co-vary. Thus, a comparison of the trajectories of the amino acid conservation energy fluctuations through many small fluctuation experiments should reveal evolutionary

coupling. This approach is methodologically distinct, but conceptually identical, to that presented by Lockless and Ranganathan [2].

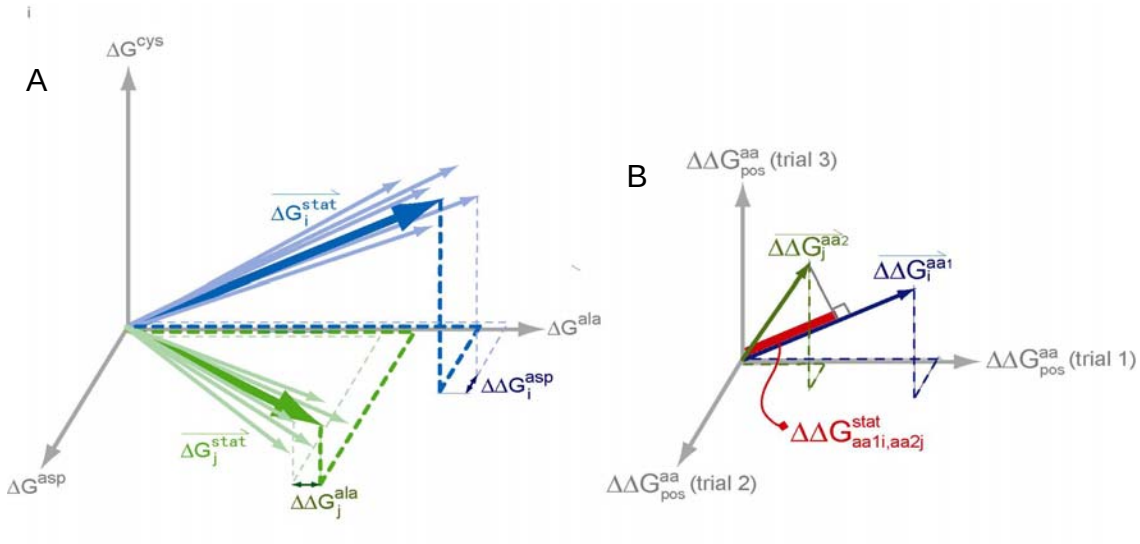


Figure 2-4 Vectorial representation of SCA by small perturbation. A) Axes represent 3 of 20 amino acid dimensions. Dashed lines represent component amino acid conservation energies for corresponding solid vectors. Solid thick blue and green vectors represents conservation energy vectors in the parent alignment for position *i* and *j*, respectively. Solid thin vectors represent small perturbations to corresponding conservation energy vectors. Differences between perturbed and parent vectors can exist in any of twenty amino acid dimensions ($\Delta\Delta G_{pos}^{aa}$). If positions are coupled, these fluctuations should correlate. B) Experiment space in which each axis represents a unique perturbation. Dashed lines represent components for thick vectors and simply measure corresponding $\Delta\Delta G_{pos}^{aa}$. Thus, solid lines trace vector for each amino acid at each position. Co-variation between two amino acids at two positions, $\Delta\Delta G_{aa1i,aa2j}^{stat}$, is represented as the dot product of their corresponding vectors.

How can we compare trajectories caused by small perturbations? To extract co-variation of amino acid deviations we can imagine an experiment space where each axis represents one trial of introducing a random fluctuation to the entire alignment (figure 2-4, B). For any particular amino acid at any position, the amino acid-specific energetic perturbation, $\Delta\Delta G_i^x(\text{trial t})$, caused in a particular fluctuation trial is measured on the corresponding trial axis. Repeating the experiment T times creates a T-dimensional experiment space in which we can trace the trajectory of energetic perturbations for each

amino acid at each position with the vector $\Delta\Delta\vec{G}_i^x$. A quantitative measure of co-variation between two amino acids at two positions is the dot product of their corresponding vectors in this experiment space (figure 2-4, B). The projection of one experiment vector on another is a measure of statistical inter-dependence. Thus, two independently fluctuating sites will produce experiment vectors that must be orthogonal and, hence, have a dot product of zero. Note that the Pearson correlation coefficient differs from this measurement of correlation since it only measures the cosine of the angle between the two experiment vectors. The dot product, however, includes the magnitudes of the vectors ($\vec{A} \bullet \vec{B} = \left|\vec{A}\right|\left|\vec{B}\right|\cos\theta$) and thus weights for their conservation. Furthermore, by incorporating directionality, the dot product can account for both correlated and anti-correlated changes: fluctuations that move in the same direction together will have large positive dot products while those that move in opposite directions will have large negative dot products. This analysis increases accuracy by measuring coupling at the level of each amino acid at each position and only requires a means of making small perturbations.

Small fluctuations through random perturbation

To explain the method, I will introduce a specific set of positions to be followed throughout the sections below. Consider the PDZ domain alignment with 240 sequences and 94 positions. Amino acid frequency distributions at three conserved positions (325, 359 and 372) are graphed in figure 2-5. The central process in the new method is to make small perturbations to the entire alignment and see if fluctuations in such amino acid frequencies tend to correlate.

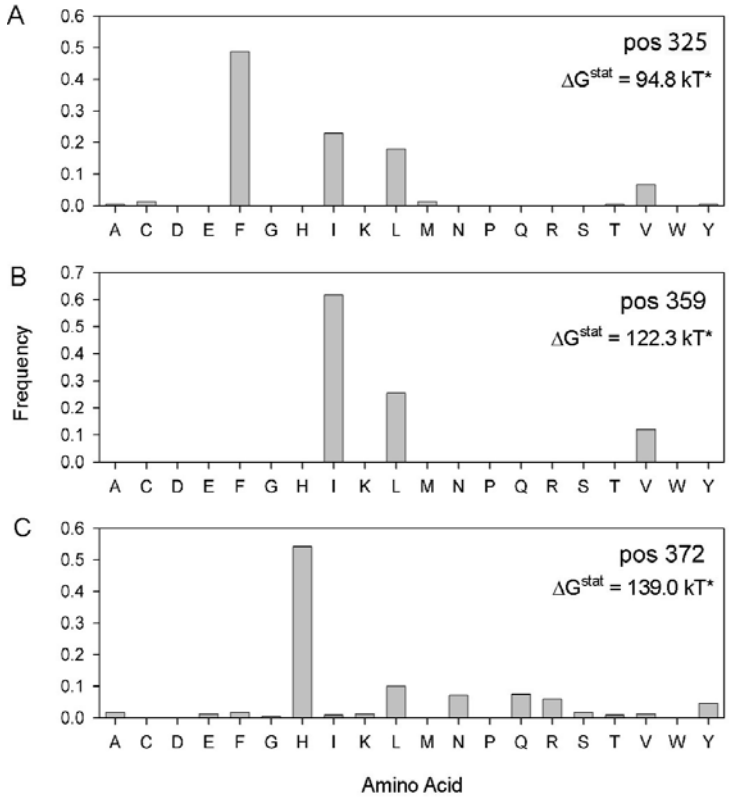


Figure 2-5 Amino acid frequency distributions at three conserved positions in the PDZ domain alignment. A) The distribution at position 325 shows this position is dominated by Phe, Ile, and Leu giving it a $\Delta G^{stat} = 94.8\text{kT}^*$. B) The distribution at position 359 shows only three amino acids (Ile, Leu, and Val) giving it a $\Delta G^{stat} = 122.3\text{kT}^*$. C) The distribution at position 372 shows approximately half of the sequences have His at this site giving it a $\Delta G^{stat} = 139.0\text{kT}^*$. Note that the y axes are scaled differently.

One simple way to make small fluctuations in a statistical ensemble is through random perturbation, for example, by randomly selecting 50% of the alignment. On average, subalignments made by such random selection over many trials will have exactly the same frequency distribution as the parent alignment. Thus, the average energetic effect in 1000 trials of selecting 50% of sequences in the PDZ alignment is zero (figure 2-6, A and D). However, in any one such trial there will be small deviations from the parent alignment frequency distribution and, hence, the standard deviation of the change in conservation of an amino acid at a site is non-zero (figure 2-6, D).

The energetic effect of the random perturbation to the alignment can be calculated using the formalism described above. The 20 x *m* conservation energy matrix for the random subalignment, $\Delta\mathbf{G}_{SA}$, provides an energetic measure of its state. The energetic deviation between the states of the parent alignment and the random subalignment is thus the difference of their respective conservation energy matrices:

$$\Delta\Delta\mathbf{G}_t = \Delta\mathbf{G}_{PA} - \Delta\mathbf{G}_{SA} \qquad \text{(Eq. 2-6)}$$

This matrix captures the energetic deviation for each amino acid at each position in trial t. Note that these are the deviations plotted on the axes of the experiment space described above (figure 2-3B). Each random perturbation trial generates a unique $\Delta\Delta\mathbf{G}_t$ matrix; repeated T times, these two dimensional matrices comprise a 20 x *m* x T three dimensional experiment matrix denoted $\Delta\Delta\mathbf{G}$. The record of fluctuations for each amino acid at each position in this matrix can thus be seen as coordinates for a unique vector in the T-dimensional experiment space (see figure 2-9).

Applying this methodology to a PDZ domain alignment with 94 positions based on random selection of 50% of the sequences over 1000 trials produces a 20 x 94 x 1000 $\Delta\Delta\mathbf{G}$ matrix. Thus, the experimental trajectory for each amino acid at each position is

described by a 1000 element vector. For example, the values in the ΔΔG vectors for three conserved amino acids, Phe325, His372, and Ile359, are graphed in figure 2-6, A-C. The adjacent histograms (figure 2-6, D, E, F, respectively) of these values show Gaussian distributions centered on zero, consistent with the random nature of the fluctuations. Thus, the experiment space contains a unique fluctuation vector for each amino acid with magnitudes proportional to their conservation energies.

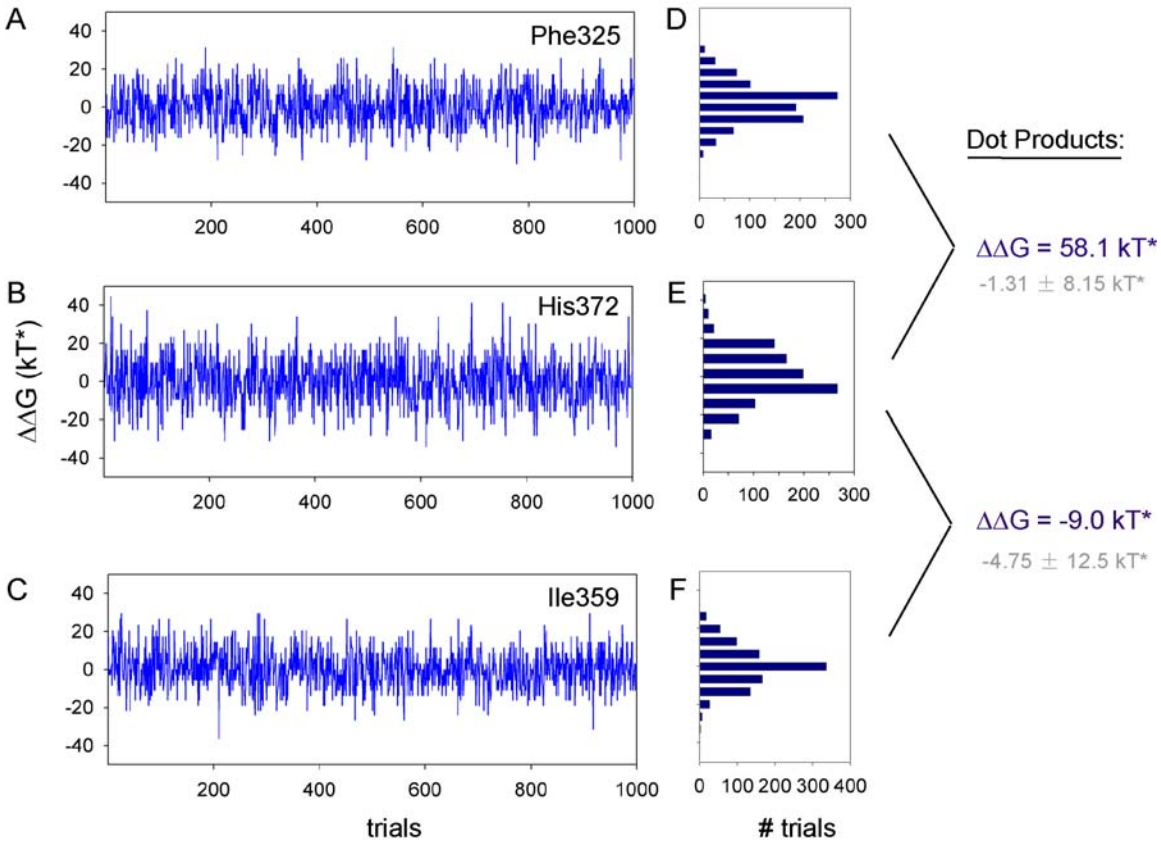


Figure 2-6. Random selection results. Each experiment consists of 1000 trials of randomly selecting 50% of the parent alignment. A,B,C) Graphs show ΔΔG fluctuation vectors for Phe325, His372, and Ile359, respectively. D,E,F) Histograms of corresponding fluctuation vectors show Gaussian distributions centered at zero, consistent with random nature of perturbation. Dot products of Phe325 and His372 show statistical coupling 58.1 kT*. Vertical scrambling of the alignment gives -1.31 ± 8.15 kT* which suggest these amino acids have significantly co-evolved ($p<10^{-10}$). His372 and Ile359 show evolutionary coupling of -9.0 kT* and random statistical coupling of -4.75 ± 12.5 kT*, suggesting these amino acids are evolutionarily independent.

Dot products of pairs of vectors give the extent of their co-variation through T trials of random fluctuation. Because the fluctuations are very small and subject to random noise, the probability of correlated fluctuation on any one trial is very low; however, over many trials of random perturbation amino acid distributions of evolutionarily coupled positions should show co-variation. On the other hand, evolutionarily independent sites will have completely independent trajectories. Note that since the measurement is based on stochastic perturbations, increasing the number of trials increases the signal of the coupling measurement. As is standard practice in measurement of signal in a stochastic system, the coupling measurement should be normalized by the number of trials.

Each pair of amino acids at each position has a unique statistical coupling energy. For example, using the PDZ domain data set of 1000 trials of 50% random selection, the dot product of the vectors for Phe325 and His372 show a coupling energy of 58.1 kT* (figure 2-6). To determine the significance of this coupling energy we randomly scrambled each column of the alignment and calculated coupling for the same amino acid pair. Since the amino acid frequency at each site is unchanged by this process, the conservation at each position stays the same; however, the correlation between sites is scrambled causing the statistical coupling in the alignment to be removed. One-hundred trials of this randomization showed a coupling between Phe325 and His372 of -1.31 ± 8.15 kT*, indicating the observed co-variation is very significant ($p<10^{-10}$). On the other hand, the native experiment vectors for His372 and Ile359 show a coupling energy of -9.0 kT*. Random scrambling of the matrix gives a coupling of -4.75 ± 12.5 kT* for His372 and Ile359, suggesting they are not significantly evolutionarily coupled ($p = 0.73$).

Calculating dot products for all pairs of amino acids at all PDZ positions gives a matrix that is 94 x 94 x 20 x 20 and captures all coupling information in the alignment. While perturbation of the alignment by random selection is mathematically valid, complete extraction of co-evolution information from the alignment would require an extremely large number of trials. Instead, we found that a simpler method to extract the same information from the alignment is to measure the effect of removing only one sequence at a time.

Making small fluctuations through single sequence elimination

As a practical matter, the same process as above can be achieved through single sequence elimination. In the limit of random elimination, the smallest possible perturbation to an alignment with N sequences is simply the removal of one sequence. In this approach, subalignments are made by throwing out one sequence from the alignment. The small perturbation approach described above can then be conducted using these subalignments of N-1 sequences. Since there are only N sequences, there are N possible subalignments and N trials (as opposed to T trials in the random selection method above) in this method. Importantly, each sequence in the alignment is treated as an experimental trial, creating an experiment space with N dimensions.

Proceeding as above, a unique $\Delta\Delta G_n$ matrix can be calculated for each sequence in the alignment. Together, these form a three-dimensional $\Delta\Delta G$ experiment matrix that is 20 x *m* x N in size. Each amino acid at each position therefore has an experiment vector defined by a sequence of N values. For example, application of the single sequence elimination calculation for all 240 sequences in the PDZ alignment generates a 20 x 94 x

240 **ΔΔG** matrix. The graph shown in figure 2-7A plots the $\Delta\Delta G_{325}^{Phe}$ values for Phe325. Position 325 is highly conserved with 117 Phe in 240 sequences, giving an amino acid conservation energy of -90.2 kT* for Phe325. The graph in 2-7A shows $\Delta\Delta G_{325}^{Phe}$ oscillates between one of two values. This is rooted in the fact that single sequence elimination has only one of two effects on the conservation energy of Phe at 325. If the sequence removed had a Phe at 325, the frequency of Phe in the subalignment will be smaller and closer to random causing a decrease in ΔG_{325}^{Phe} to -89.6 kT*, giving a $\Delta\Delta G_{325}^{Phe}$ of approximately 0.6 kT* (figure 2-7B). However, if the sequence removed does not have a Phe at 325, the resulting subalignment will be even more enriched with Phe causing an increase in ΔG_{325}^{Phe} to -90.8 kT*, giving a $\Delta\Delta G_{325}^{Phe}$ of approximately -0.6 kT* (figure 2-7B). Thus, the vector represented graphically in figure 2-7A is an energetic

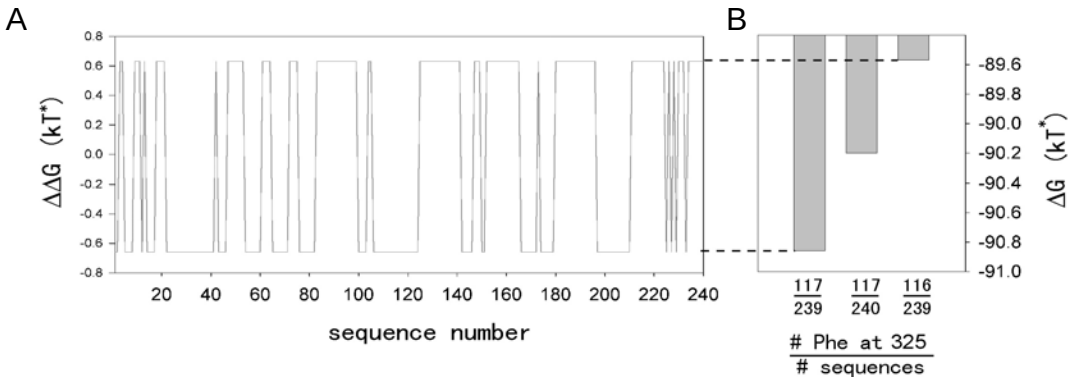


Figure 2-7. Single sequence elimination fluctuation vector for Phe325. A) ΔΔG vector oscillates between one of two values through the 240 sequences in the PDZ domain alignment, depending on whether or not the sequence removed had a Phe at position 325. B) The parent alignment has 117 Phe at 325 out of 240 sequences. If the removed sequence did not have a Phe at 325, the subalignment is further enriched for Phe and its conservation energy moves away from zero. If, however, the sequence removed had a Phe at 325, the subalignment is closer to the reference state and the energy moves closer to zero.

profile for Phe at position 325: it simply marks presence or absence with a quantal change in conservation energy. The vector depicting Phe325 in the 240-dimensional experiment space of the PDZ domain alignment is represented by this specific set of values.

The degree of evolutionary coupling is expressed as the dot product of the fluctuation vectors for two amino acids:

$$\Delta\Delta G^{n1,n2}_{x1i,x2j} = \Delta\Delta G^{n1}_{x1i} \bullet \Delta\Delta G^{n2}_{x2j} = \sum_{N} (\Delta\Delta \vec{G}_{x1i})(\Delta\Delta \vec{G}_{x2j}) \qquad \text{(Eq. 2-8)}$$

For example, figure 2-8 shows graphs of ΔΔG vectors for three representative amino acids at three PDZ domain positions: Phe325, His372, Ile359. Careful inspection of the

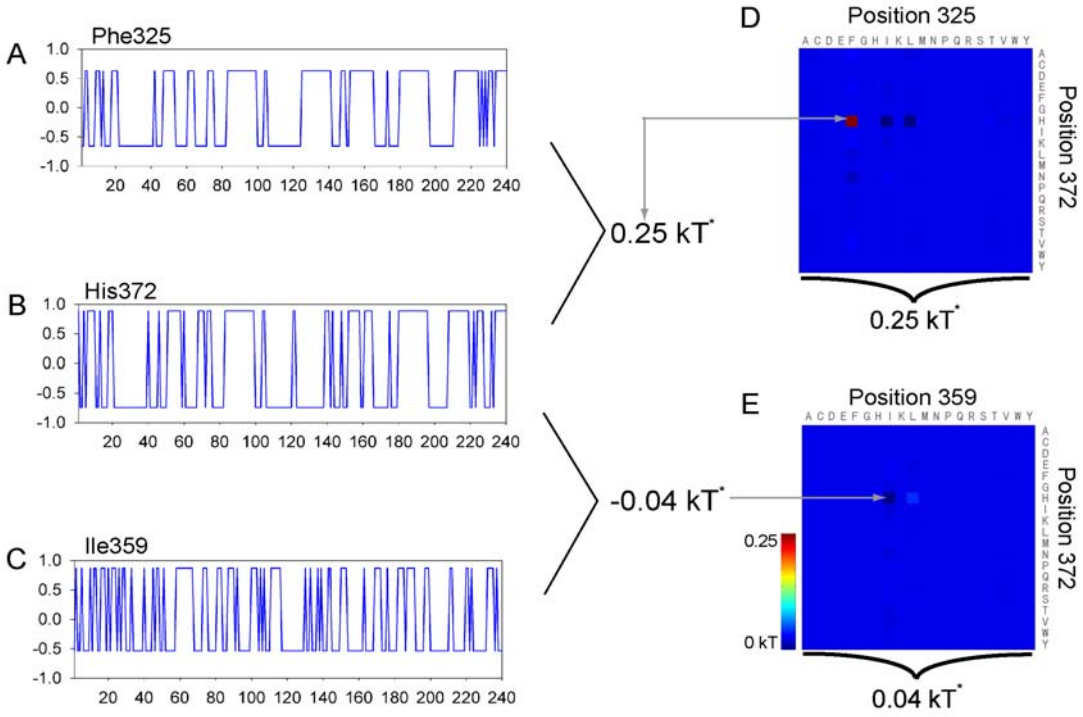


Figure 2-8. Single sequence elimination at three sites. Vectors for (A) Phe325 and (B) His372 appear correlated and indeed have a significant ΔΔG = 60.2 kT*. However, vectors for (B) His372 and (C) Ile359 do not appear correlated and have a low coupling of ΔΔG = -9.6 kT*. Each of these values are only one element in 20 x 20 matrices that capture the coupling between each pair of positions (D,E). The magnitude of these 400 values gives the statistical coupling between the corresponding pair of positions.

graphs reveals that Phe325 and His372 have very correlated trajectories as *n* goes from 1 to 240 (compare figure 2-8, A and B). The dot product of these two vectors is 0.25 kT*. To determine the significance of this value we performed the same vertical scrambling experiment described above and found that the expected random coupling for the amino acid frequencies at these positions is 0.0004 ± 0.04 kT*, suggesting that these positions have indeed significantly co-evolved ($p < 10^{-10}$). However, the trajectories of His372 and Ile359 do not appear correlated (compare figures 2-8, B and C). Correspondingly, their dot product and randomized dot product is -0.04 kT* and 0.003 ± 0.04 kT*, indicating their evolutionary coupling is insignificant (p = 0.22).

For each pair of positions in an alignment we can calculate the dot products of all possible pairs of amino acids, giving a 20 x 20 matrix of 400 amino acid-pair specific coupling energies. Mathematically, this is simply represented as matrix multiplication of the two 20 x N matrices for the two sites (see figure 2-9). The coupling information in the entire alignment is therefore represented in an N x N x 20 x 20 matrix, where N=240 for the PDZ domain alignment. For example, figure 2-8D colorimetrically shows the elements of the 20 x 20 amino acid coupling matrix corresponding to positions 329 and 372; one of these 400 values corresponds to the coupling between Phe329 and H372 discussed above. The coupling between Phe329 and His372 clearly dominates these couplings. The corresponding matrix for positions 359 and 372 (figure 2-8E), however, shows very low coupling energies. The magnitude of the 400 amino acid coupling energies corresponding to a pair of positions defines their statistical coupling energy:

$$\Delta\Delta G_{i,j}^{stat} = \sqrt{\sum_{400} \Delta\Delta G_{aai,aaj}} \qquad \text{(Eq. 2-8)}$$

For example, positions 325 and 372 show significant evolutionary coupling of 0.25 kT* ($p < 0.0001$) while positions 359 and 372 have very low coupling of 0.04 kT* ($p = 0.28$) (figure 2-9). Thus, the statistical couplings between all pairs of positions in the alignment are represented by an N x N matrix. Note that the 20 x 20 matrix corresponding to the *i*,*j* position pair is exactly the same as the transpose of that for the *j*,*i* pair. Because this method quantifies the effect of energetic perturbations at the level of individual amino acids, it is possible to calculate statistical couplings between all pairs of positions and the resulting matrix is completely symmetric.

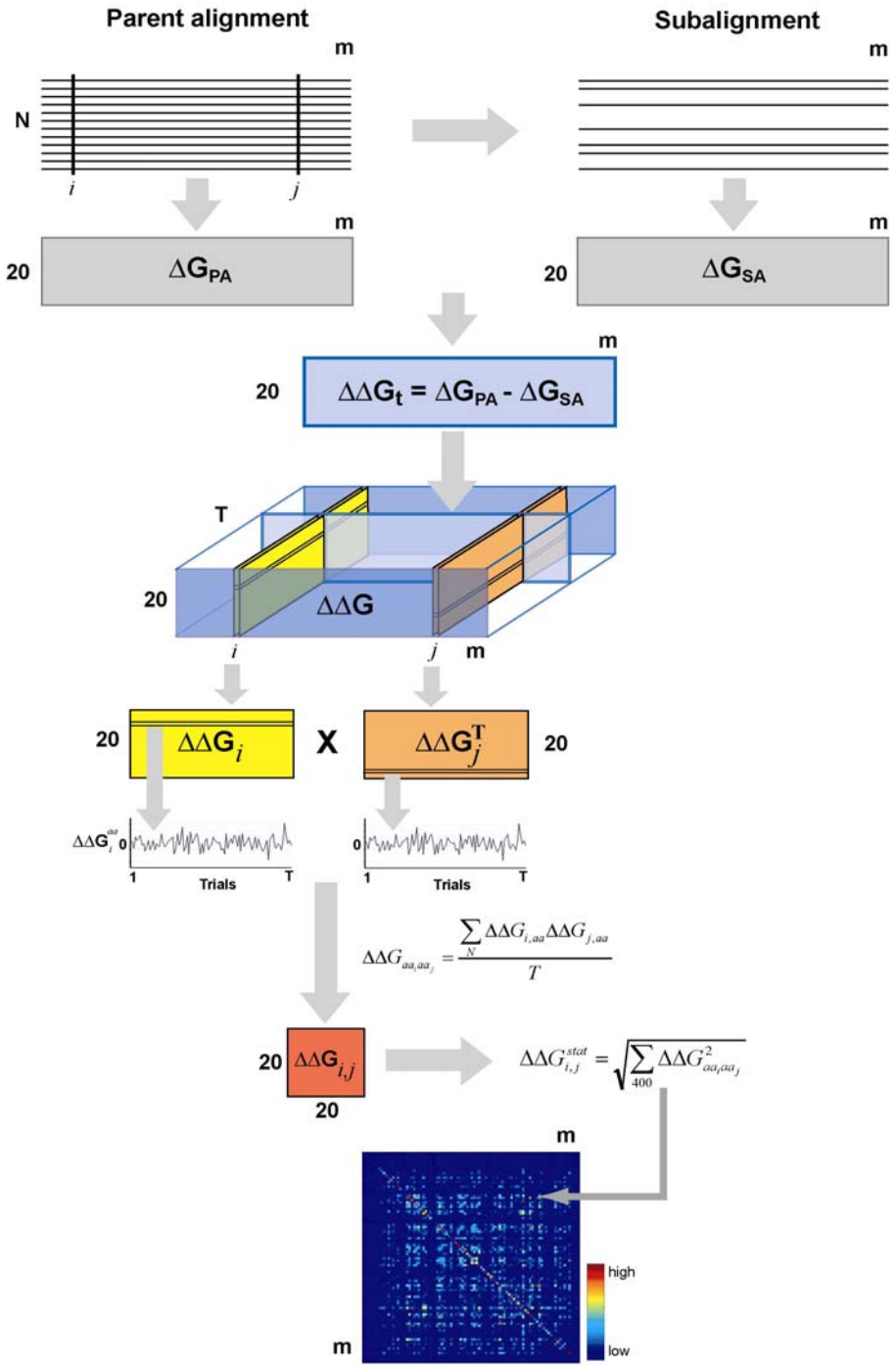


Figure 2-9. Overview of SCA by small perturbation. Small perturbations can be made by either random selection or single sequence elimination (random selection is depicted above). Conservation energy matrices can be calculated for both the parent alignment (ΔG_{PA}) and subalignment (ΔG_{SA}). The difference of these defines the $\Delta\Delta G_t$ matrix, where t denotes the particular trial. Repeating this for T trials generates a three dimensional $\Delta\Delta G$ matrix that is 20 x m x T (note that in single sequence elimination, T=N). Coupling between two amino acids at two sites is simply the dot product of their ΔΔG fluctuation vectors. Thus, coupling between two sites, *i* and *j*, is simply represented as the matrix multiplication of their corresponding 20 x T ΔΔG matrices. This gives a 20 x 20 matrix for each pair of positions that contains the coupling energies between each pair of amino acids at sites *i* and *j*. The magnitude of these 400 values defines the statistical coupling energy between sites *i* and *j*.

Statistical coupling analysis of PDZ domain

For the PDZ domain alignment with 240 sequences and 94 positions, analysis based on single sequence elimination gives a symmetric 94 x 94 statistical coupling matrix, colorimetrically represented in figure 2-10A. This matrix is a global representation of the co-evolutionary relationships between all pairs of positions in the PDZ alignment. Clearly, most positions show very little coupling to other positions, suggesting evolutionary independence, while a subset shows strong coupling to a few positions.

To understand the patterns of evolutionary interactions, the data in the coupling matrix can be imagined as a network represesentation in which vertices are PDZ positions. Between every pair of vertices is an edge representing corresponding evolutionary coupling energy, the value of which is an element in the coupling matrix. To identify the significant evolutionary interactions in the PDZ domain we plotted a histogram of the coupling values (figure 2-9B). The graph shows that coupling values show a highly skewed distribution with low coupling between most pairs and high coupling between a small subset which is well-described by a log-normal distribution:

$$f = ae^{-\frac{1}{2}\left(\frac{\ln\frac{x}{x_0}}{\sigma}\right)^2} \qquad \text{(Eq. 2-9)}$$

Here σ is the standard deviation and x_o is the mean. Log-transformation of the x-axis of a log-normal distribution produces a normal distribution. Display of the data in this manner makes the distribution easier to appreciate and is shown in the right panel of figure 2-9, B. Using the standard deviation from the fit (given in figure) we can establish an energetic threshold for significant evolutionary interaction. For example, at a 3.5σ (28 kT*) cutoff

to the PDZ domain matrix, we find 33 couplings out of 4371. Paring down the graph representation to only these edges reveals that these couplings connect 19 positions (20 % of PDZ positions) into a nearly completely connected subset (figure 2-9C).

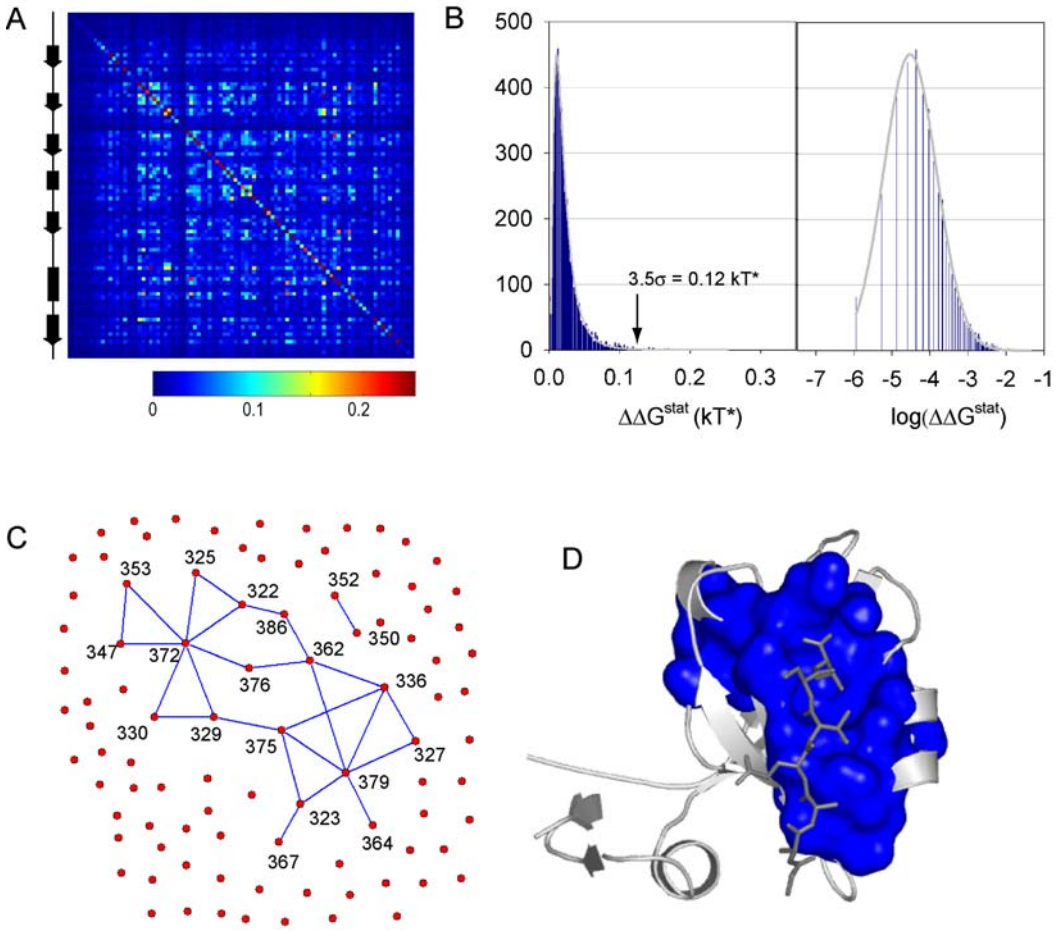


Figure 2-10. SCA analysis of PDZ domain. (A) The single sequence elimination method yields a 94 x 94 DDG matrix for the PDZ domain alignment. (B) The distribution of coupling energies in this global co-variation analysis fits a log-normal distribution. (C) Graph representation of the interactions 3.5σ above the mean reveals a nearly completely connected subgraph. (D) Positions in this highly co-evolving network form a connected unit when mapped on the structure of a representative PDZ domain (PSD95-PDZ3 with co-crystallized peptide from C-terminus of CRIPT shown in teal).

Importantly, the 19 positions identified by this method are in agreement with those identified by the original SCA. Mapping these residues on the structure of a representative PDZ domain (PDZ3 from PSD95) shows a completely connected structural network that includes the majority of the binding site, part of the core, and several distant residues on the backside of the protein. Previous work from our lab showed that the statistical coupling energies (using the original method) showed excellent agreement with thermodynamic coupling [2].

Analysis of coevolution in alignments of other protein families, including GPCRs, G proteins, hemoglobin, WW domains, and ligand binding domains, further emphasize these themes. In each family, the single sequence elimination SCA reveals small subsets of mutually co-evolving positions that formed connected units when mapped on their respective structures. The coupled positions were nearly identical to those already identified by the original SCA. This previous work also showed that these sets of coupled positions have excellent correlation with a large body of published functional data for these systems [2-5].

Conclusion

In this chapter, I have described an alternate approach to SCA that provides, for the first time, a completely global map of co-evolution between pairs of positions in which coupling between pairs of positions is reciprocal. A critical feature common to this new SCA method and its original formulation is their view of the alignment as a statistical ensemble. This view allows the use of Boltzmann statistics to define the conservation energy, an energetic measure that lies at the core of both approaches and is

the source of the sensitivity of both methods. By focusing on each amino acid at each position, the new method allows a finer parsing of conservation energy than the original method. As a consequence, it is possible to extract a global map of statistical coupling energies that is completely symmetric.

The results of both SCA methods are very consistent. In all protein families studied both methods identify the same set of highly co-evolving positions. Graph representation of the networks emphasizes two critical and surprising features. First, there is sparseness in the pattern of significantly co-evolving positions. Most sites are evolutionarily independent and only a small subset show significant evolutionary interaction. Secondly, these subsets of positions are not scattered throughout the protein; instead, they are organized into a highly inter-linked network to form a nearly completely connected subgraph. This highly improbable arrangement of the energetic topology must endow some evolutionary advantage to the protein. Mapping of the coupled positions onto structures consistently reveals physically contiguous networks that link distant sites on the protein through a subset of core positions. Highly coupled positions identified in GPCRs, hemoglobin, and serine proteases showed strong correlation with published functional data [3]. Recent work in the G protein [4] and ligand binding domain [5] demonstrated that the highly coupled pathways identify the core allosteric mechanism in these proteins. These results from proteins with diverse functions suggest the coupled networks of residues contain the core energetic interactions that enable protein function.

The analysis presented here focuses on the high coupling values in the heavy tail region of the log-normal distribution. However, this is not to discount the information contained in the low coupling values – those below 3.5σ in PDZ domains. As stated previously, in all protein families studied so far the coupling energies in the tail of the

distribution repeatedly form connected structural networks that connect functional surfaces. While data shows that these high couplings contain the core functional processes in proteins, it is unlikely that they operate in complete isolation from the remainder of the protein. It is possible that a subset of the low coupling energies form a necessary bridging framework that connects the highly-coupled core functional unit to the remainder of the protein. Furthermore, while the discussion above and in the following chapters focuses on the functional role of coupling energies, the co-evolutionary information is likely to embed function as well as stability since both are evolutionarily selected properties. Recent work in our lab has tested this by making two categories of synthetic (not found in nature) WW domain sequences with a Monte Carlo algorithm: 1) sequences with the same conservation pattern as natural sequences, and 2) sequences with the same coupling pattern as natural sequences [1]. Folding studies of these novel proteins demonstrated that the coupling information is necessary and sufficient to build folded WW domains [1]. Ongoing experiments in our lab are attempting to parse the co-evolutionary interactions that are necessary for folding from those necessary for endowing functionality.

The complete mapping of evolutionary couplings provides the foundation to address two general questions. First, since the evolutionary analysis and mutagenesis experiments are intrinsically thermodynamic measurements, these analyses do not provide insight into the underlying mechanism of interaction between positions. In the next chapter I will present a combination of crystallographic and thermodynamic studies of the PDZ domain that are focused on understanding a set of interactions predicted by SCA. The fourth chapter takes a graph theoretic approach to analyze the topology of the network and its functional implications.

Materials and Methods

Statistical coupling analysis. All alignments used in these analyses were provided from members of the Ranganathan lab and were prepared as described in [3]. Statistical coupling analyses, curve fitting, and clustering was performed with MATLAB (version 6.5.0.180913a (R13), Natick, MA). The choice of cutoff value depends on the distribution of $\Delta\Delta G^{stat}$ values which depends on the size and diversity of the alignment. The MATLAB code used is given in Appendix A.

References

1. Lockless, S.W., *Networks of Evolutionarily Coupled Residues*, in *Thesis*. 2002.
2. Lockless, S.W. and R. Ranganathan, *Evolutionarily conserved pathways of energetic connectivity in protein families.* Science, 1999. **286**(5438): p. 295-9.
3. Suel, G.M., et al., *Evolutionarily conserved networks of residues mediate allosteric communication in proteins.* Nat Struct Biol, 2003. **10**(1): p. 59-69.
4. 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.
5. Shulman, A.I., et al., *Structural determinants of allosteric ligand activation in RXR heterodimers.* Cell, 2004. **116**(3): p. 417-29.

Chapter 3 Logic and Mechanism of an Evolutionarily Conserved Interaction in a PDZ Domain

Introduction

Statistical coupling analysis results provide a critical piece of the structure-function puzzle. Mutagenesis studies in numerous systems have shown that the networks of highly co-evolving residues identified by SCA capture energetic interactions necessary for function. [1-3] These findings motivate an atomic level description of the mechanisms that underlie evolutionary coupling. A complete description of the atomic events that allow energy to propagate among the atoms in this network should bring us closer to a complete explanation of how functionality is achieved. However, the problem is complex. Though SCA can identify amino acid interactions, it does not tell us about mechanism.

A clue about mechanism does emerge from the striking physical connectedness of these networks. Long-range energy transfer may occur through chains of anisotropic local interactions. In other words, these contiguous networks may reflect structural regions capable of converting and propagating energy between adjacent residues, ultimately channeling it into functional events such as binding, catalysis or long-range signaling. Ample evidence shows that proteins are indeed structurally inhomogeneous. Crystallographic studies often show that mutations induce atomic displacements that propagate anisotropically, often resulting in changes at very distant sites [4]. NMR dynamics experiments have revealed hotspots of slow time scale dynamics that correlate with known regional and temporal functional importance [5, 6].

In the light of the SCA results, these observations raise two questions. First, what physical properties characterize statistically coupled networks? Second, what is the mechanistic basis of their cooperative energetic interactions? In this chapter I will describe a combination of thermodynamic and crystallographic experiments I have performed to address these issues.

The experiments focus on the dissection of a set of functionally important interactions in the binding interface of a PDZ domain and peptide. Tuned molecular interfaces such as these are common and conserved features of protein-protein interaction domains, enzymes, and signaling proteins. PDZ domains therefore represent an excellent model system to study this fundamental phenomenon with a combination of co-variation analysis and biophysical experiments.

Background

Interfaces must balance affinity and specificity

Each protein-protein binding interface has a characteristic specificity and affinity evolved to suit a particular function. In one elegant example, recent work showed that the behaviors of multi-domain proteins depend on the binding affinities of regulatory domains for their respective regulatory molecules [7]. This group showed that modular binding domains, like PDZ or SH3 domains, could be combined with catalytic domains to create novel proteins with sophisticated behaviors. Importantly, these behaviors, like allostery and signal integration, could be adjusted by simple changes in the individual domains. For example, one protein consisted of a PDZ domain, an SH3 domain and a VCA domain that stimulates actin polymerization; in the inhibited state the PDZ and SH3

domains bind internal autoinhibitory sequences and prevent actin polymerization. Activation of the protein required input of both the PDZ ligand and the SH3 ligand and hence the protein acts as a regulatory AND gate. However, a mutation in the PDZ domain that causes a ten fold decrease in binding affinity converted the protein to an OR gate in which input of either ligand was sufficient for activation. This example illustrates how simple changes at one molecular interface can significantly affect the properties of a signaling molecule and the behavior of an entire signaling pathway.

The fidelity of a signaling pathway depends on the reliable sequential transfer of information through a specific set of components. Minimization of inappropriate cross-talk between components of different pathways requires that proteins avoid non-specific interactions. In recent demonstrations of this design principle, two groups studied the specificity of the SH3 domain from the yeast protein Sho1p, a membrane protein that initiates the yeast hyperosmotic stress response. A critical interaction in this MAP kinase (MAPK) pathway is the binding of the Sho1 SH3 domain to a prototypical PXXP motif in Pbs2, a MAPKK. One group sought to determine how specifically the Pbs2 binding motif interacts with the Sho1 SH3 domain among all 27 SH3 domains found in yeast. [8]. The data showed that the peptide ligand (from Pbs2) bound only to the Sho1 SH3 domain with no cross-reactivity to any other yeast SH3 domain. Importantly, the ligand did interact with non-yeast SH3 domains. Together, these observations indicate the complement of yeast SH3 domains has evolved under negative selection. In other words, the affinities between yeast SH3 domains and their respective ligands have evolved to minimize cross-reactivity. To test the importance of specificity, they introduced a mutation in Pbs2 that binds the Sho1 SH3 with slightly higher affinity but also cross-reacts with other yeast SH3 domains. Interestingly, the promiscuous mutant strain could

not adapt as well to high-stress growth conditions and was out-competed by the wild type strain. In total, these data indicate that reliable output from the hyperosmotic stress response pathway depends critcally on the strength of the interaction between the Sho1 SH3 and Pbs2.

Another group addressed the importance of the strength of this interaction for the reliability of the hyperosmolarity response pathway [9]. To do this, they made yeast strains in which the Sho1 SH3 domain was replaced with mutants that bound the Pbs2 motif with a range of weaker affinities; these strains were assayed for their response to a hyperosmolar environment. In vivo measurements showed a linear correlation between the strength of the interaction and the output of the hyperosmolarity response pathway: weaker interactions gave smaller outputs and stronger inteactions gave larger outputs. Interestingly, weak interactions also correlated with increasing inappropriate activation of the related mating pheromone response pathway, also a MAPK pathway. These two sets of experiments demonstrate that loss of binding specificity at even one step in a signaling pathway can, ultimately, lead to a fitness defect for the cell. Interactions among amino acids involved in protein-protein interactions must have mechanisms that achieve a functional balance between specificity and affinity in the face of competing substrates.

General mechanisms of specificity and affinity

Structural and mutagenesis experiments probing numerous protein-protein complexes have revealed several important principles underlying binding energy. Structures of complexes have shown that specificity relies on geometric and chemical complementarity of contacting surfaces [10] . For example, groups of residues in a

protein may form a pocket, such as the S1 pocket of α-lytic protease, suited to accommodate only a specific residue [11]. In some systems, such pockets become hotspots for tumorigenic mutations [12, 13]. While these structural studies have, in general, only addressed the contribution of individual residues, binding measurements suggest that more complex mechanisms can significantly tune binding energy. Studies of the human growth hormone receptor and antibody-antigen complexes show that residues in the interface do not contribute equally to the binding energy; a few residues comprising only a small fraction of the interface are hotspots that account for most of the binding energy [14]. To add to this complexity, multiple residues, often distantly positioned, can interact cooperatively to modulate the binding energy [15-17]. Thus, a complete understanding of binding energy requires a mechanistic description of all cooperative interactions that affect binding.

Energetic networks in proteins retain evolvability

In addition to meeting functional requirements, proteins must also maintain an ability to adapt to a changing environment, a capacity referred to as evolvability. While the property of evolvability is typically applied by evolutionary biologists to the level of organisms, it is also relevant and abundantly evident in proteins. Indeed, since the fundamental level of mechanistic action in evolution is the protein, it should not be surprising to observe the feature of evolvability in proteins. In the face of changing environmental pressures, a protein with greater evolvability would endow an organism with a fitness advantage. In general, evolvability is defined [18] as the "capacity to generate heritable, selectable phenotypic variation." From this perspective, the

evolvability of a protein can be seen as the efficiency with which it can be converted through mutations to perform a new function; a highly evolvable protein would require very few mutations to switch to a new function. Thus, while performing a particular function, the energetic architecture of a protein must also maintain a functional plasticity that allows rapid adaptability.

While no experiments we are aware of have explicitly assessed the evolvability of proteins, mutagenesis data from several systems clearly demonstrates that the energetic framework of proteins is highly adaptable. For example, one group recently showed that as few as 18 mutations to a ribose binding protein could convert it to a triose phosphate isomerase, a particularly dramatic change in function [19]. Importantly, this novel protein had significant enzymatic activity and was capable of supporting growth of bacteria. In a second example, one group used phage display libraries of mutant SH3 domains to screen for domains capable of binding to two ligands, a Src-binding peptide and an Abl-binding peptide [20]. Analysis of the isolated domains indicated that only two or three substitutions could cause a dramatic change in binding specificity. Similarly, mutation of only one position in PDZ and WW domains significantly shifts the binding specificity [21, 22]. These results are consistent with and suggest an energetic logic for the phenomena of binding hotspots. A focus of binding energy in a protein-protein interface may allow rapid change in binding specificity through mutagenesis of hotspot residues only. Together, these examples indicate that the energetic architecture in proteins must not only allow the cooperative interactions necessary for function, but must also maintain functional plasticity. While mutagenesis studies have revealed these fundamental features of the energetic framework, they have not explained the mechanistic basis for how amino acids in a protein achieve these properties. The PDZ domain is a

protein-protein interaction domain with well established class specificity and a known crystal structure; it represents an excellent model system in which to understand how collective interactions among multiple positions can tune binding energy [23, 24].

Evolutionary and thermodynamic coupling in the PDZ domain

PDZ domain background

PDZ domains are protein-protein interaction modules approximately 90 amino acids long and typically bind to the C terminal 4-5 amino acids of target proteins [23]. Named after the first three proteins in which they were observed (PSD95, Discs large, Zo-1), PDZ domains have since been found to be well-represented in *Caenorhabdatis elegans*, *Drosophila melanogaster*, and mammalian genomes [23]. PDZ-containing proteins serve as scaffolds to assemble supramolecular complexes in specific subcellular locations [23]. Such scaffolding proteins often contain multiple PDZ domains or combinations of PDZ and SH3 domains. By co-localizing the components of a pathway, such scaffolding proteins are thought to dramatically enhance the speed and reliability of signaling [23].

The crystal structure of PDZ3 from PSD95 bound to a peptide ligand (hereafter protein and peptide are referred to as PDZ3 and P_{WT}, respectively) shows that the protein takes an approximate β-sandwich fold with the peptide binding in a pocket formed by the βB strand and αB helix (figure 3-1A). This and other PDZ-ligand structures have revealed several common structural features of PDZ-ligand interfaces [24]. First, the carboxy terminus of the peptide forms a hydrogen bonding interaction with the

carboxylate-binding loop between βA and βB. Second, the P_{-2} position of the peptide (peptide numbering convention refers to the carboxy terminus amino acid as P_0 and counts backwards from this residue) of the peptide interacts with the αB1, the first side chain of the αB helix. Several mutagenesis studies have found this interaction to be critical in determining binding specificity [23, 25]. Accordingly, this interaction has been used to organize PDZ-ligand interactions into three general classes [23, 26]. PDZ3 is an example of a class I domain in which His at αB1 forms a hydrogen bond with a Ser or Thr at the P_{-2} position (figure 3-1A). In class II domains hydrophobic amino acids at both αB1 and P_{-2} positions form hydrophobic interactions [26]. Class III domains have a negatively charged amino acid at αB1 interacting with an acidic moiety at the P_{-2} position [26]. The correlation between amino acids at the αB1 and P_{-2} positions suggest this interaction is an energetic hotspot strongly dictating specificity.

Thermodynamic analysis of a PDZ hotspot

The strength of the interaction between the αB1 and P_{-2} positions can be estimated using thermodynamic mutant cycle analysis [27]. In this method the energetic effect of a mutation at site *i* is measured in two different conditions: 1) in a wild type background, and 2) in the background of a mutation at another site *j*. The difference between these two energetic effects is defined as the thermodynamic coupling energy, $\Delta\Delta G_{i,j}^{mut}$, between sites *i* and *j* and reports the extent to which these two mutations energetically interact. For example, the H372Y mutation in PDZ3 converts the αB1 position from a class I to a class II amino acid. Binding measurements of this mutant to P_{WT} by isothermal titration calorimetry (ITC) show that H372Y destabilizes binding by 3.25 ± 0.09 kcal/mol (figure

3-1B), reflecting a nonspecific interaction. However, in the background of a T7F peptide mutation that converts the P_{-2} position to a class II amino acid (mutated peptide referred to as P_{T7F}), the H372Y mutation only destabilizes binding by 0.54 ± 0.09 kcal/mol (figure 3-1B). The significantly smaller energetic effect simply reflects that the compensatory mutation at the P_{-2} position essentially completely converts the interaction to a class II interaction. The 2.71 ± 0.09 kcal/mol thermodynamic coupling of these mutations is large and highlights the role of this interaction in discriminating between specific and non-specific binding interactions.

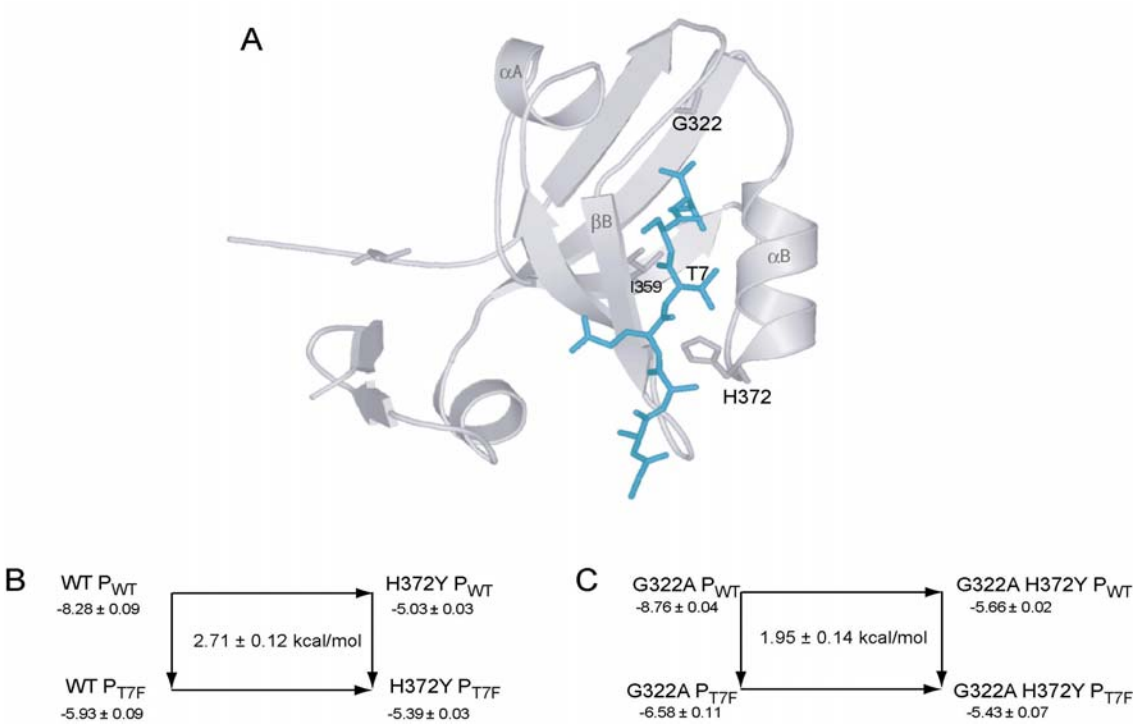


Figure 3-1 PDZ3 structure and thermodynamics. A) Crystal structure of PSD95-PDZ3 bound to peptide highlights several critical interactions. The contact between H372 and T7, in this case through a hydrogen bond, is a specificity defining interaction in PDZ domains. The carboxy terminus of the peptide interacts with the carboxylate binding loop. B) Thermodynamic mutant cycle analysis reveals that mutations at the contacting residues H372 and T7 are, as expected, strongly coupled. C) However, in the background of a mutation at 322 in the carboxylate binding loop on the other side of the binding pocket, the coupling decreases by 0.8 kcal/mol, indicating a three-way energetic coupling between these positions.

Importantly, this finding is consistent with the role of this interaction in at least two class I PDZ domains as a site of binding regulation through covalent modification. Protein kinase A mediated phosphorylation of the serine at the P_{-2} position of the inwardly rectifying potassium channel Kir2.3 inhibits binding to a class I PDZ domain in postsynaptic density-95 protein (PSD-95) [28]. In a similar mode of regulation, phosphorylation of the serine in the P_{-2} position of β2-adrenergic receptor by the serine-threonine kinase (GRK-5) inhibits binding to a class I PDZ domain in Na^+/H^+ exchanger regulatory factor (NHERF) [29]. These examples illustrate a general energetic principle in regulation of binding interactions: binding regulation often occurs through perturbation, in this case via covalent modification, at a focus of binding energy. These data further support the claim that the strong thermodynamic coupling between αB1 and P_{-2} measured in the PDZ3-peptide interaction reflects an evolutionarily conserved binding hotspot.

Knowledge of this thermodynamic coupling does not reveal the potentially complex mechanism by which these two mutations interact. Binding studies in other systems suggests that the energy likely reflects perturbation of not only the contact sites but also energetic interactions among other amino acids in the protein. However, identification of all other PDZ3 positions involved in tuning this interaction is not possible from the crystal structure, and thermodynamic mutant cycle analysis of all pairs of positions is impractical.

SCA reveals energetic interactions

If SCA truly maps the propagation pattern of energetic perturbations then the global PDZ domain co-evolutionary mapping should predict positions that influence the interaction between 372 and P_{-2}. The statistical couplings for position 372 are represented in the row corresponding to this position in the 94 x 94 matrix of PDZ domain statistical couplings (figure 3-2A). The bar graph of these values in figure 3-2B reveals that most positions in the domain are evolutionarily independent of position 372 while only a few show significant statistical coupling. Since each of these evolutionarily coupled positions may interact with position 372 through unique mechanisms, we chose to first focus on only one significant predicted interaction. Application of a 3.5σ (0.12kT*) energetic cutoff (discussed in chapter 2) identifies seven positions that have significantly co-evolved with position 372 (figure 3-2C). Position 322 shows significant mutual statistical coupling with position 372 ($\Delta\Delta G^{stat} = 0.14\,\mathrm{kT}^*$, p = 0.0002) and is particularly interesting because of its position on the opposite side of the binding pocket in the carboxylate binding loop (figure 3-1A). The significant co-evolution of these positions predicts that position 322 energetically interacts with position 372.

To test the role of position 322 as a modulator of coupling energy we determined the thermodynamic coupling of the H372Y and P_{T7F} mutations in the background of a glycine to alanine mutation at position 322 (figure 3-1C). By analogy, figures 3-1B and C can be seen as opposite faces of a thermodynamic cube that compare the coupling energy of H372Y and T7F in the wild type and G322A backgrounds. Binding energies (figure 3-1C and table 3-1) show that, in the background of G322A, thermodynamic coupling between H372Y and T7F falls to 1.95 ± 0.14 kcal/mol. This 0.8 ± 0.18 kcal/mol

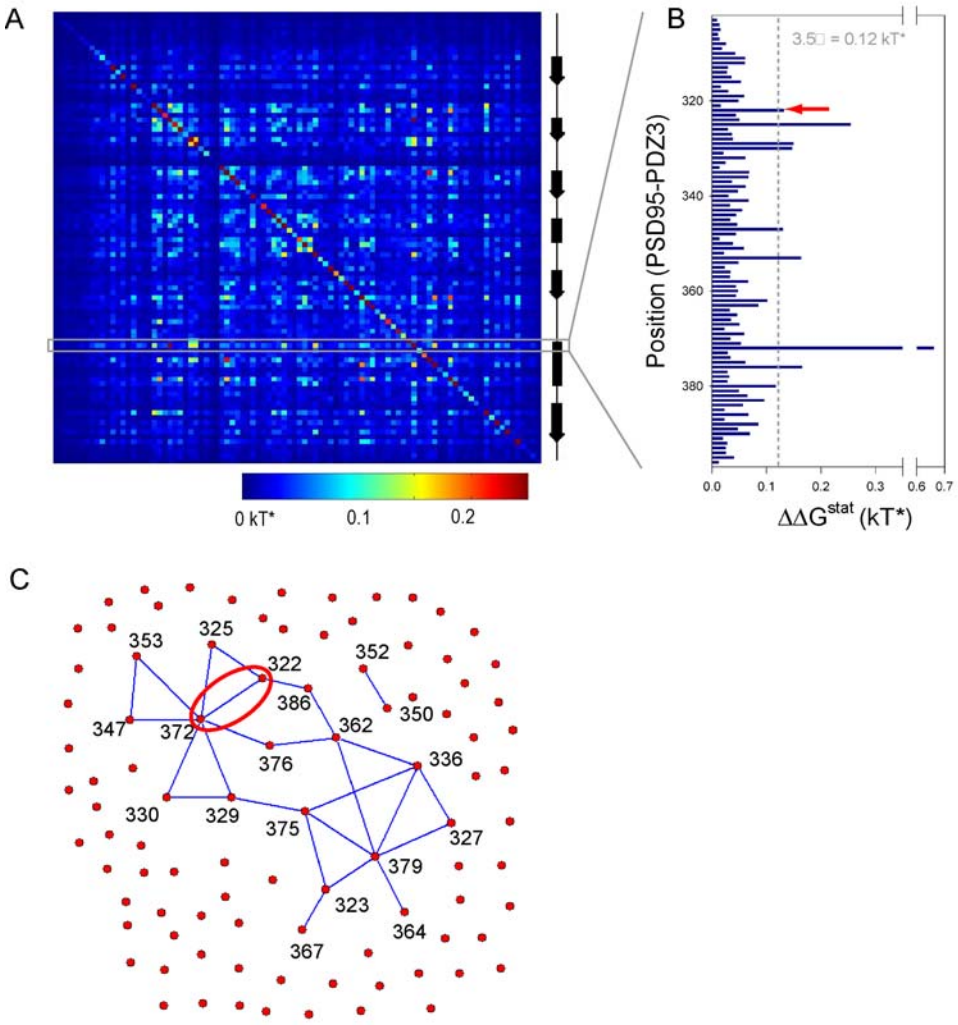


Figure 3-2. SCA analysis of PDZ domain and position 372. A) SCA analysis gives a 94 x 94 matrix containing the global map of pairwise interactions (chapter 2). Secondary structure and position numbering are from PDZ3 from PSD-95. B) Bar graph of values in row corresponding to position 372 shows significant (greater than 3.5σ, as indicated by dased line) co-evolution with 7 positions. C) Network graph shows all significant statistical couplings in PDZ domain. The results of this analysis suggest positon 322 is energetically coupled to position 372.

decrease in thermodynamic coupling reveals an energetic interaction among these three amino acids, supporting the SCA prediction. In the wild type state PDZ3 is tuned to distinguish between specific and nonspecific peptides with a coupling energy of 2.71 kcal/mol. However, position 322 on the opposite side of the binding pocket decreases the

strength of this discrimination. Thus, long range cooperativity plays a role in PDZ domain function.

Thermodynamic analysis shows interaction important for evolvability

Thermodynamic measurements also reveal a critical role for position 322 in the evolvability of PDZ3 specificity. Dissociation constants of wild type protein to the two peptides reveal that PDZ3 is tuned to favor the class I P_{WT} over the class II P_{T7F} by 50 fold (table 3-1). As stated above, the H372Y mutant shows a significant switch in specificity favoring the class II peptide over the class I by almost 2 fold. This single mutation tips the binding affinity balance and begins to convert the protein to a class II PDZ domain, revealing an innate evolutionary plasticity. We can define the evolvability quantitatively with a score:

$$Evolvability = \frac{\frac{K_d^{classI,mut}}{K_d^{classII,mut}}}{\frac{K_d^{classI,WT}}{K_d^{classII,WT}}} \qquad \text{(Eq. 3-1)}$$

This evolvability score captures the ability of a mutation to switch binding specificity from class I to class II. For H372Y, this ratio (1.8/0.02, see table 3-1) gives an evolvability score of 90.1, consistent with the well-established role of this position as a specificity determinant. In fact, binding studies of the class III PDZ domain in neuronal nitric oxide synthase (nNOS) showed that a tyrosine to histidine mutation at the αB1 position converts its binding specificity to that of a class I PDZ domain [25]. Thus, position 372 is a determinant of binding specificity and evolvability.

In contrast to H372Y, the G322A mutation alone does little to affect binding specificity. Similar to the wild type protein, this mutant has an approximately 30 fold stronger binding affinity for class I peptide over class II peptide (table 3-1), giving the G322A mutation an evolvability score of only 1.5. However, in the background of G322A, the effect of H372Y is significantly altered. While H372Y shows clear preference for P_{T7F}, the H372YG322A mutant shows a nearly equal preference for P_{WT} and P_{T7F} (table 3-1). The evolvability score of H372Y in a G322A background is 22.7. Thus, the G322A mutation reduces the class-switching potential of the H372Y mutation. Since G322A specifically influences the energetics of position 372, we conclude that this long range interaction contributes to the evolvability (or plasticity) of the PDZ domain. The fact that this amino acid combination is so highly co-selected in evolution is an indication of its functional relevance. If the interaction between positions 322 and 376 is indeed engineered by evolution as suggested by SCA and thermodynamic measurements, we should be able to uncover a physical mechanism for this interaction. Such a demonstration would add to the confidence in our claim that this is a selected interaction in PDZ domains.

Protein	Peptide	ΔH (kcal/mol)	TΔS (kcal/mol)	ΔG (kcal/mol)	K_d (μM)	$\frac{K_d(P_{WT})}{K_d(P_{T7F})}$
WT	P_{wt}	-8.6 ± 1.6	-0.4 ± 1.6	-8.3 ± 0.1	0.87 ± 0.1	0.02
	P_{T7F}	-3.6 ± 1.4	2.2 ± 1.5	-5.9 ± 0.1	44.8 ± 5.8	
H372Y	P_{wt}	-2.0 ± 0.3	3.0 ± 0.4	-5.0 ± 0.03	205.8 ± 10.2	1.8
	P_{T7F}	-0.4 ± 0.1	5.0 ± 0.1	-5.4 ± 0.03	111.8 ± 5.3	
G322A	P_{wt}	-10.5 ± 1.9	-1.7 ± 1.9	-8.8 ± 0.04	0.4 ± 0.2	0.03
	P_{T7F}	-4.9 ± 0.3	1.7 ± 0.4	-6.6 ± 0.11	15.3 ± 2.8	
H372YG322A	P_{wt}	-3.7 ± 0.2	2.0 ± 0.2	-5.7 ± 0.02	70.7 ± 2.3	0.68
	P_{T7F}	-2.0 ± 3.2	3.4 ± 0.4	-5.4 ± 0.07	104.7 ± 11.8	

Table 3-1. Isothermal titration calorimetry binding measurements. Each value represents the average ± standard deviation of three measurements.

Structural mechanism of high-order coupling

Structural role of position 322

A clue to the mechanism of position 322 comes from a comparison of the binding energies in the two thermodynamic cycles (figures 3-1, B-C and table 1). The G322A mutation stabilizes binding of wild type and almost mutant all proteins to both P_{WT} and P_{T7F} peptides. For example, WT binds P_{WT} with a binding energy of -8.28 ± 0.1 kcal/mol (K_d = 0.87 ± 0.1 μM) while G322A binds the same peptide with ΔG = -8.78 ± 0.04 kcal/mol (K_d = 0.38 ± 0.2 μM). To understand the mechanism by which G322A stabilizes binding, we solved the X-ray crystal structures of the WT and G322A proteins, both peptide-free and bound to P_{WT}. The protein was expressed as N-terminal GST fusion protein in E. coli, purified over GST affinity chromatography, and crystallized in sodium citrate. Structures were solved using rigid body refinement and the Ramachandran plot showed no outliers for any structure. These and other structures discussed below were solved under isomorphous conditions to ensure that observed atomic displacements were not the consequence of differences in crystal contacts (see methods and Table 3-2 at end of chapter).

To determine peptide induced conformational changes to the WT protein, we overlayed the free and P_{WT}-bound WT structures by least squares minimization of Cα positions and calculated the displacement of each atom. Upon peptide binding, the major atomic displacements in the WT protein occur in the carboxylate binding loop (figures 3-4, A-C). Specifically, the vector diagram (figure 3-4C) shows that, upon P_{WT} binding, the loop moves to an orientation we refer to as ‘clamped down’. This involves a shift of 4.8 A in the end of the loop towards the α2 helix (figure 3-3A). Additionally, comparison of

carboxylate binding loop B factors in these two structures shows peptide binding induces a disorder-to-order conformational change in this region (figure 3-3B).

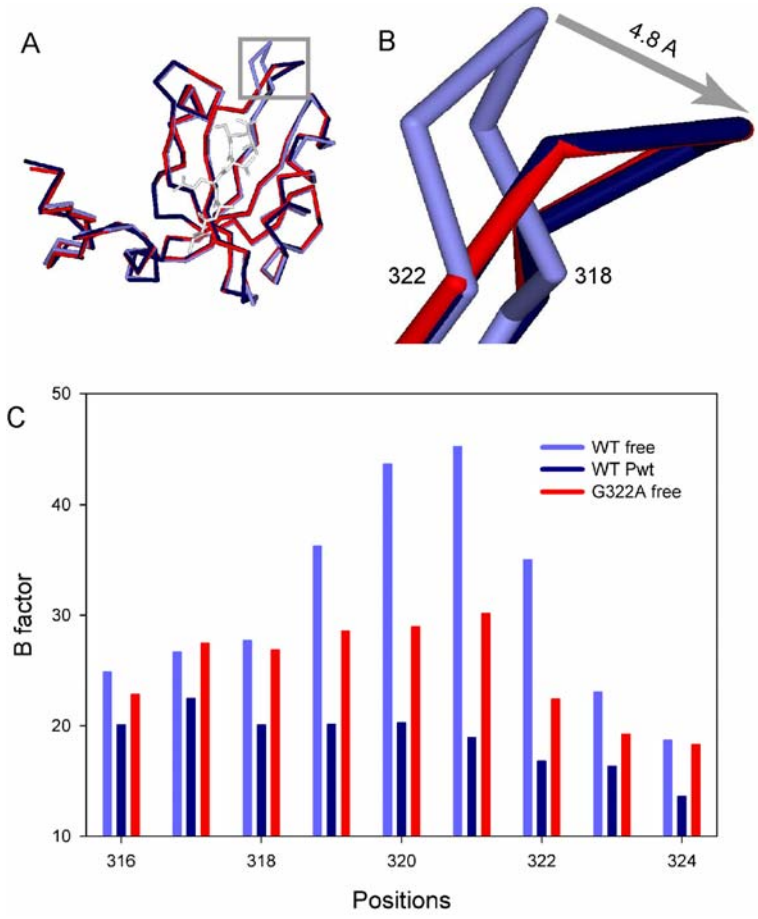


Figure 3-3 Disorder to order conformational change in carboxylate binding loop. A) Overlay of Cα traces for three structures (WT free in light blue, WT bound to P_{wt} in dark blue and white respectively, and G322A free in red). B) Zooming in on the carboxylate binding loop, we see the peptide-induced conformational change in the WT structures, referred to here as 'clamped down' (compare light and dark blue Cα traces). The free G322A structure (red) shows that it is in the clamped down even in the absence of peptide. C) Bar graph of B factors of Cα atoms at indicated positions shows that, in addition to displacement, there is a disorder-to-order transition upon peptide binding to WT protein (compare light and dark blue bars). The G322A mutation itself induces this transition even in the absence of peptide binding (red bars).

The picture of conformational changes induced by peptide binding to G322A is clearly different. An overlay of G322A structures in the free and P_{WT} bound states shows essentially no atomic displacements upon binding to P_{WT} (figures 3-4, D-F, table 3-2). The loop is already in the clamped down conformation in the G322A mutant in the absence of peptide. This uncoupling of peptide binding to conformational change suggests the G322A mutation has stabilized a high-affinity, clamped-down conformation in the absence of peptide.

To determine how position 322 controls conformational change in the carboxylate binding loop we solved and compared the peptide-free WT and G322A structures (figures 3-4, G-I, table 3-2). These structures reveal the G322A mutation itself induces the same clamping down conformational change in the carboxylate binding loop normally observed with peptide binding to the WT PDZ3. Comparison of B factors shows that the G322A mutation also causes a disorder to order conformational change in the carboxylate binding loop in the absence of peptide (figure 3-3, A and B). Thus, in the peptide-free state, the G322A mutation appears to redistribute the protein conformational ensemble to a bound-state conformation. This structural change provides an explanation for the increased affinity of G322A-containing proteins for both P_{wt} and P_{T7F} peptides. The structures suggest stabilization of the loop by the G322A mutation decreases the entropic cost associated with clamping the loop upon peptide binding to the WT protein.

Why would evolution design a peptide-induced conformational change in the carboxylate binding loop that sacrifices binding affinity? As discussed above, recent studies comparing yeast SH3 domain specificities have shown that evolution has not simply optimized binding affinity at protein-protein interfaces. Rather, as found in PDZ3, binding interactions have been designed to achieve specificity for partners in the face of

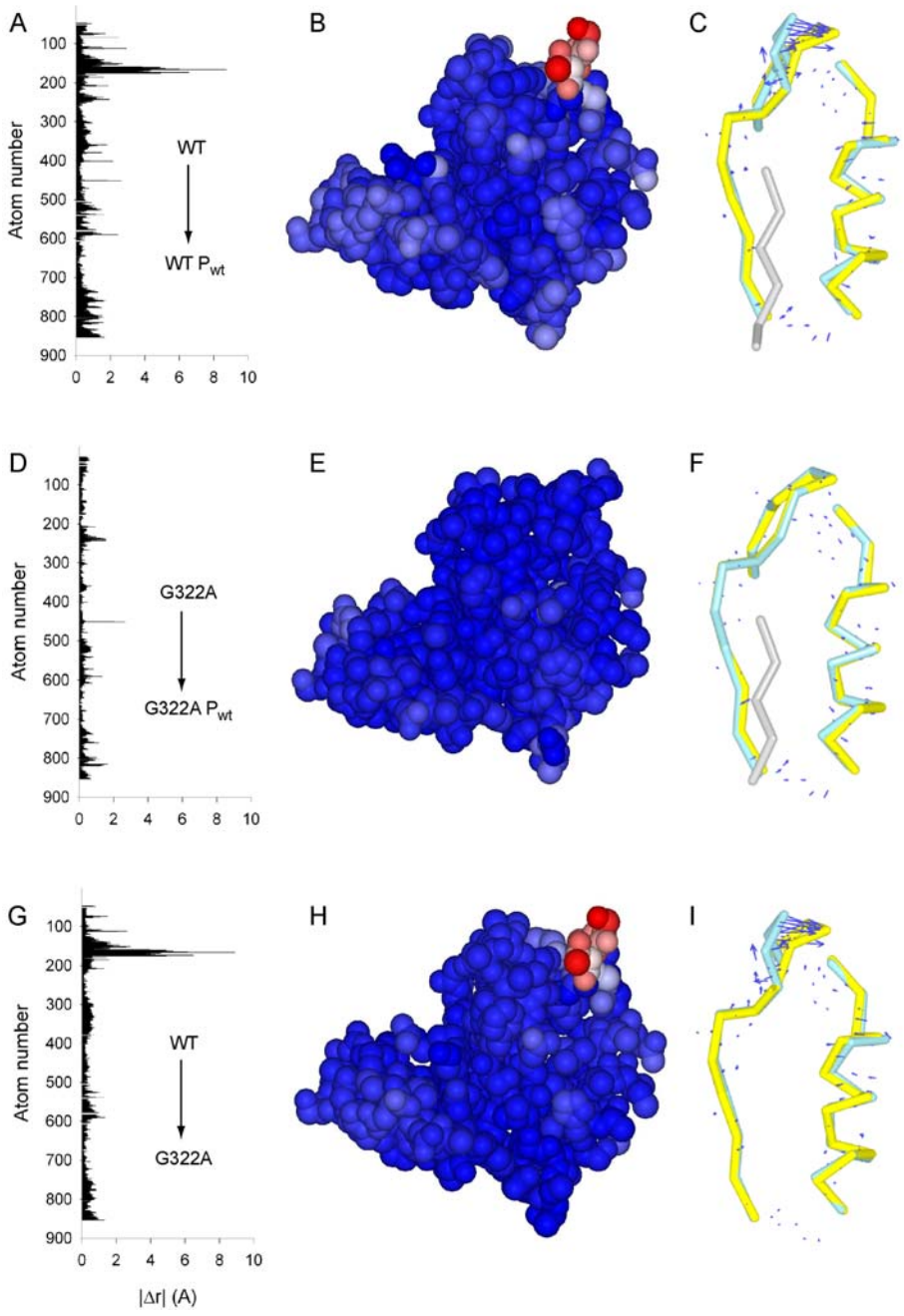


Figure 3-4. **Position 322 controls the conformational change of PDZ3 to peptide binding.** All panels depict differences between a reference structure and a perturbed state structure. Bar graphs (A, D, G) show displacements of protein atoms (numbered according to PDB) in response to a perturbation. Colorimetric representation of these displacements (B, E, H) are shown from blue (low) to red (high) on a CPK model of the reference structure. Vector diagrams (C ,F, I) show least squares overlay of C trace of reference (blue) and perturbed states (protein in yellow and peptide in gray). Vectors are drawn from atomic centroid in reference state to the corresponding centroid in the perturbed state. Upon P_{wt} binding to WT PDZ3 (A-C) the carboxylate binding loop clamps down. However, G322A shows no significant atomic displacements in response to Pwt binding (D-F). This occurs because the G322A mutation itself induces essentially the same clamping conformational change in the carboxylate binding loop (G-I).

competing ligands. In addition, since evolvability is a key characteristic of natural proteins, perhaps the co-selection between positions 322 and 372 contributes to the facility of class-switching by the PDZ domain. Either way, it is clear that understanding the role of G322 involves understanding its long-range cooperativity with the specificity/evolvability determinant, the αB1-P_{-2} interaction.

Structural analysis of double mutants

Recent work has shown that the mechanism underlying thermodynamic coupling between two mutations can be revealed through analysis of the structural analog of the thermodynamic cycle [30-32]. Comparison of the structural effects of a mutation in two different backgrounds can provide a physical mapping of how two mutations structurally interact; this structural interaction can be correlated with their energetic coupling. To understand the physical basis of the high order coupling among these positions, we solved the structures of WT and H372Y proteins in three states: peptide-free, bound to P_{WT}, and bound to P_{T7F}.

Structural comparison of P_{WT}-bound WT and H372Y structures revealed that H372Y causes two major structural effects (figure 3-5, A-C). First, tyrosine at 372 appears to prevent the N-terminal end of the peptide from binding though the C-terminal amino acid and carboxyl terminus are bound as in the WT structure. By disrupting several protein-peptide contacts, a tyrosine at position 372 seems to actively select against a class I ligand, reflected in the substantially weaker binding between H372Y and P_{WT} (K_d = 205.8 ± 10.2 μM). Second, the structural comparison shows an unexpected long-range physical consequence of H372Y. The interaction of H372Y and P_{WT} leaves the

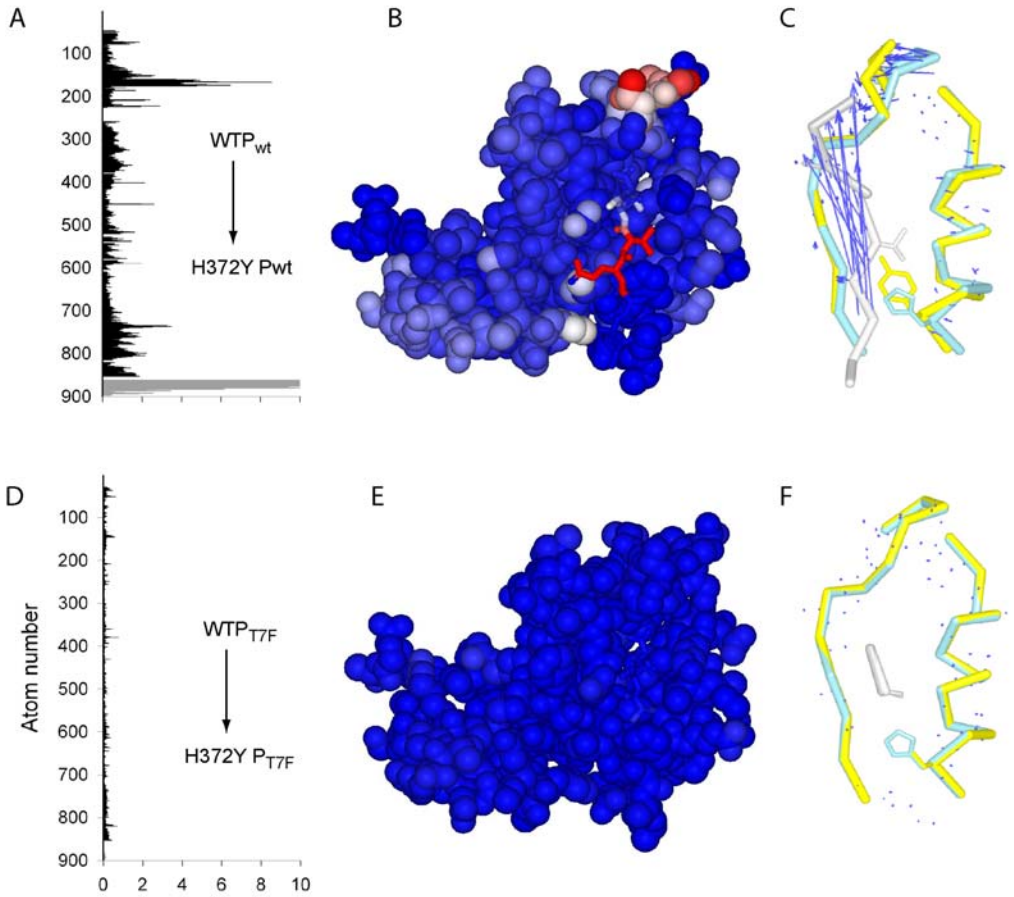


Figure 3-5. Mutations H372Y and T7F structurally interact through both local and propagated atomic displacements. Bar graphs (A, D) show atomic displacements (r) in protein (black bars) and peptide (gray bars) atoms induced by H372Y mutation in P_{wt} and P_{T7F} backgrounds, respectively. Atom numbering follows PDB file numbering. Colorimetric representations of displacements (B, E) are shown from blue (low) to red (high) on CPK models of protein, with bound peptide shown as stick model. Vector diagrams (C, F) show least squares overlay of C trace of reference (protein in blue and peptide in white) and perturbed (protein in yellow and peptide in gray) states. Vectors are drawn from atomic centroid in reference state to the corresponding centroid in the perturbed state. H372Y has two major effects in a P_{wt} background: the mutation prevents the N-terminal end of the peptide from binding and leaves the carboxylate binding loop is in the unclamped conformation. However, in the PT7F background, H372Y has essentially no structural effect.

carboxylate binding loop in the unclamped conformation suggesting the nonspecific binding between H372Y and P_{WT} is not sufficient to induce conformational change in the loop. This demonstrates a remarkable finding: coordination of the terminal carboxylate is perhaps necessary, but certainly not sufficient to cause clamping down of this loop. Achieving the clamped down state also requires the αB1-P_{-2} interaction. In this sense, the carboxylate binding loop acts like an AND gate, clamping down only if the C-terminal site and the P_{-2} site of the ligand are recognized correctly.

The dramatic structural effects of H372Y shown in figure 3-5C reveals the mechanism of class II specificity and a common design principle underlying specific protein-protein interactions. As previously observed in SH3 domains, binding specificity depends on a balance of forces in the binding interface [33]. The structure of H372Y bound to peptide shows that the side chain of the carboxy terminal valine of the peptide forms a presumably favorable van der Waals interaction with the hydrophobic binding pocket. However, the tyrosine at position 372 clearly makes an extremely disruptive interaction with the peptide and significantly reduces the binding affinity for P_{wt}.

If the carboxylate binding loop is an AND gate as described above, then restoring the αB1-P_{-2} interaction in the H372Y mutant should restore loop clamping. Comparison of structures of WT and H372Y bound to P_{T7F} peptide shows that the class-switching mutation on the peptide almost completely structurally compensates for the H372Y mutation (figures 3-5, D-F). In the background of the T7F mutation, H372Y has little effect on the peptide or protein. This makes structural sense since the tyrosine at 372 and phenylalanine of P_{T7F} can now form class II hydrophobic interactions. The compensated interaction of this contact is apparently sufficient to allow the carboxylate binding loop to adopt a clamped-down conformation.

Thus, we now see the structural logic of energetic communication between positions 322 and 372. The αB1-P_{-2} interaction triggers both local and non-local (carboxylate binding loop) structural changes. Position 322 is a determinant of the non-local effect. The cycle of structures reveals that the H372Y and T7F mutations interact through their compensatory effects in two regions of the PDZ domain: proximally in the peptide and distally in the carboxylate binding loop. From these data it is easy to imagine how perturbation of the long-range mechanical coupling between these two structural regions could be used to modulate binding affinity. Evidence from two PDZ domain systems, NHERF and PSD-95, shows that phosphorylation of the P_{-2} position of the respective binding partners inhibits binding [28, 29]. In the light of these structural observations, it is likely that phosphorylation in these systems disrupts interactions at the αB1-P_{-2} contact in the respective binding interfaces and prevents the long-range mechanical coupling necessary for binding. Thus, it may be that effective regulatory control by phosphorylation also requires the thermodynamic coupling with the carboxylate binding loop modulated by G322. Overall, these data strongly provide a mechanistic underpinning for thermodynamic and evolutionary coupling of positions 322 and 372. The fact that these interactions are at long range and are not obvious in any previous PDZ structural studies highlights the value of the SCA in identifying such residue pairs.

As a further test of our hypothesis, we sought to demonstrate that mutation at position 322 selectively interferes with the non-local structural effects of the αB1-P_{-2} interaction. Given the ability of 322 to control carboxylate binding loop conformational change, we predicted that in the background of G322A mutation the effect of H372Y would prevent structural communication between position 372 and the carboxylate

binding loop. In other words, the structural changes induced by H372Y would only extend to the peptide and not to the carboxylate binding loop. To test this, we solved the structures for the same cycle in the presence of the G322A mutation (table 3-2). As

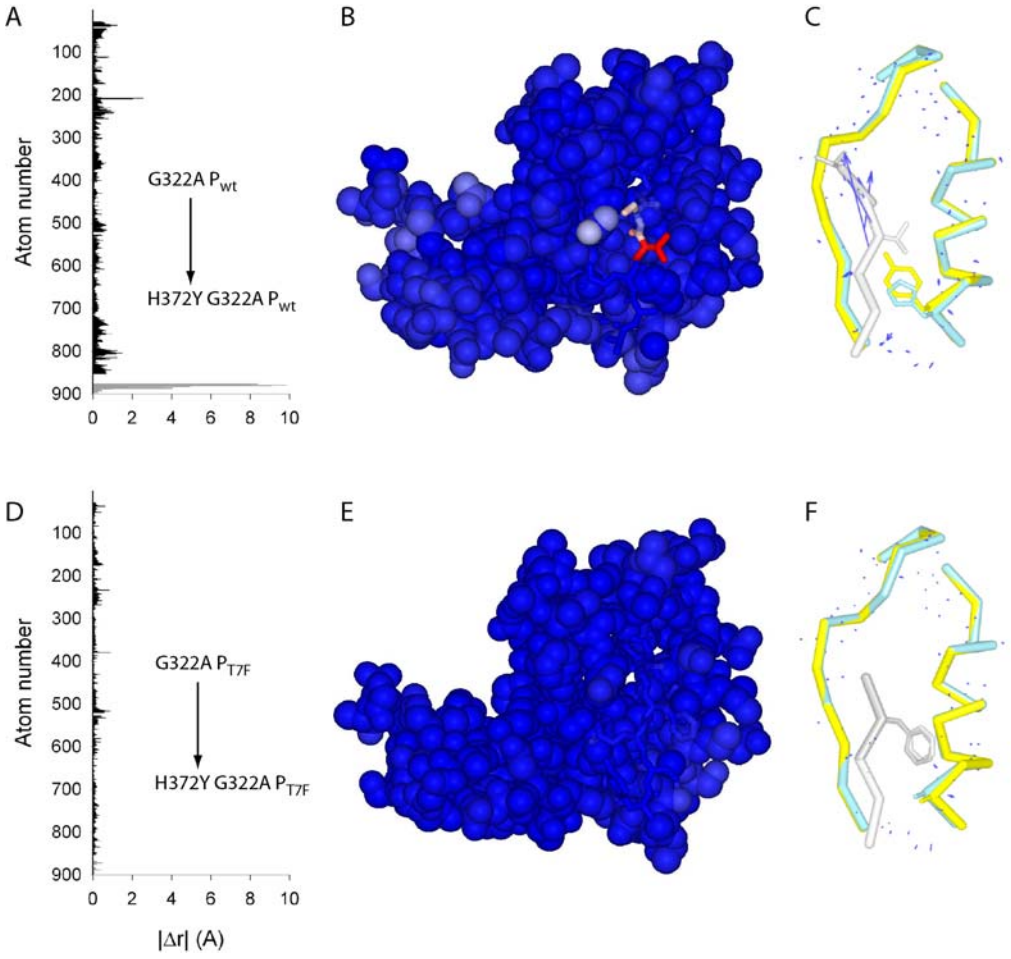


Figure 3-6. G322A uncouples carboxylate binding loop conformational change from peptide binding. Bar graphs (A, D) show atomic displacements in protein (black bars) and peptide (gray bars) atoms induced by H372Y mutation in G322A P_{wt} and G322A P_{T7F} backgrounds, respectively. Atom numbering follows PDB file numbering. Colorimetric representations of displacements (B, E) are shown from blue (low) to red (high) on CPK models of protein, with bound peptide shown as stick model. Vector diagrams (C, F) show least squares overlay of Ca trace of reference (protein in blue and peptide in white) and perturbed (protein in yellow and peptide in gray) states. Vectors are drawn from atomic centroid in reference state to the corresponding centroid in the perturbed state. H372Y has only one structural consequence in the G322A P_{wt} background: it distrupts the N-terminal end of the peptide from binding (A-C). The T7F mutation compensates for this local structural displacement (D-F).

predicted, comparison of structures of G322A and H372YG322A bound to P_{WT} show that H372Y only ejects the N-terminal end of the peptide and has no effect on the carboxylate binding loop (figures 3-6, A-C). Thus, the G322A mutation completely structurally uncouples the long- and short-range effects of the H372Y mutation. As before, in the background of the T7F mutation, H372Y has no structural effects (figures 3-6, D-F).

The combination of structural and thermodynamic data indicate that *the flexibility of the carboxylate binding loop (with glycine at position 322) is tuned to make its physical response sensitive to the interaction between positions 372 and* P_{-2}. This mechanical coupling is a built-in mechanism for screening for specific interactions and for optimizing plasticity of the PDZ domain, but comes at the cost of decreased binding affinity. By optimizing this long-range coupling, the designed flexibility improves the role of position 372 in two regards: 1) as an evolutionary hotspot such that mutations at this position cause a significant class redistribution of the binding partner profile, and 2) as a regulatory hotspot, such that phosphorylation of substrates interacting with position 372 may permit larger free energy destabilization.

Balance of stability and function

The demonstration of the functional importance of G322 for regional instability is not novel. This concept, originally stated as the 'stability-function hypothesis' by Pauling, is evident in numerous systems [34]. For example, comparisons of thermophilic, mesophilic, and psychrophilic enzymes show that their stabilities are tuned for specific operating temperatures [35]. Additionally, enzymes are tuned to stabilize the transition state of a reaction coordinate and thus are necessarily not optimally stable in their ground

states [36]. These examples highlight the fact that regional instabilities are necessary compromises in the evolutionary balance of stability and function. The thermodynamic and structural dissection of the set of interactions in PDZ3 described above not only clearly demonstrates another solution of this balance, but also shows how regional inhomogeneities may be coupled to other energetic interactions in the protein.

Tuning the physical properties of flexible regions represents a means to modulate function. For example, the physical basis for the improved affinity of the G322A mutation illustrates a structural mechanism also observed in antibody maturation. Recent structural comparisons revealed that the unbound conformation of the mature antibody closely resembles the antigen-bound conformation of the germline antibody [37]. This suggests the mutations acquired during maturation shift the conformational equilibrium such that the mature antibody exists in an ensemble that more closely matches the bound-state conformation. This decrease in flexibility sacrifices the binding repertoire of the antibody but produces improved complementarity to a specific antigen and hence enhances the binding affinity of that specific interaction. Similarly, the G322A mutation pre-organizes the binding site such that it more closely complements peptide and enhances binding. However, this improved binding weakens a built-in mechanism for selecting specific interactions.

Par-6 PDZ domain shows allosteric regulation involving loop

In light of the results discussed above, modulation of carboxylate binding loop flexibility can be imagined as a means to regulate ligand binding. A combination of structural and thermodynamic experiments studying the regulation of a PDZ domain in

Par-6 adds to the diversity of roles for the carboxylate binding loop [38]. Par-6 is a conserved protein involved in establishing cell polarity and contains a PDZ domain adjacent to a CRIB motif; these domains bind the Rho GTPase Cdc42 in a GTP-dependent manner. Structural studies of the complex showed that Cdc42 contacts the PDZ domain in two regions: 1) along the β1 strand and 2) in the α1 helix [39]. Intriguingly, the α1 helix contains several residues (PDZ3 numbering: 345, 350, 351, 352) in the PDZ statistically coupled network and is located on the backside of the PDZ domain.

Binding measurements showed that the affinity of Par-6 PDZ domain for peptide ligand improves 13 fold in the presence of Cdc42 [38]. That is, Cdc42 binding to the back of the domain allosterically controls binding at the active site. What is the mechanism for this regulation? The P171G mutation in the Par6 PDZ domain was found to uncouple this long-range regulation [38]. Interestingly, position 171 of the Par-6 PDZ domain corresponds to PDZ3 position 322. The location of this position in the carboxylate binding loop suggested a potential role for Cdc42 in regulating binding interaction. To address this, the authors compared the mobilities of the carboxylate binding loop from an NMR structure of the free Par-6 PDZ domain and a crystal structure of the Cdc42-bound Par-6 PDZ domain. Though a comparison of mobility from these different techniques is tenuous, the authors claim that upon Cdc42 binding, the carboxylate binding loop of the Par-6 PDZ domain becomes more ordered. In addition to providing the first demonstration of allosteric regulation of PDZ domain binding, this work showed remarkable consistency with already published SCA results. Note that in the Par-6 PDZ domain, this position is a proline. It is possible that proline, rather than glycine, provides a rigidity in this domain necessary to transmit energy of Cdc42 binding

into ordering of the carboxylate binding loop. In combination with the structural analysis of PDZ3 described above, these two case studies indicate that PDZ3 position 322 acts as a critical energetic relay point. Flexibility at this position is tuned to create appropriate energetic coupling of distant structural elements. Together, these two studies provide evidence for the majority of the PDZ domain SCA network: the Cdc42-Par6 work connects the back side (αA) to the carboxylate binding loop and our work connects the αB1-P_{-2} interaction to the carboxylate binding loop.

This mechanistic insight demonstrates the power of SCA to identify important interactions among positions even when acting at long range and not structurally obvious. However, these results only represent a part of a complete atomistic understanding of how PDZ3 modulates its binding. The structural analysis of energetics above only considered the atomic displacements caused by mutation. However, other processes, such as dynamical fluctuations, are likely to significantly contribute to binding energetics. Additionally, the statistically coupled network identified numerous other highly co-evolving interactions that remain unexplained by these structures. Each of these interactions may have a unique mechanistic explanation. In the remainder of this chapter I will discuss experiments which we conducted to address other interactions in PDZ domains.

Understanding the effect of V386I

One of the hallmark results of SCA is that, in all proteins analyzed, the structurally contiguous networks connect distant sites through several intervening van der Waals contacts. What mechanisms allow energetic interaction between such positions?

One possibility is that atoms in statistically coupled units are tightly packed to create a micro-region of increased rigidity. Such a region of relative solidity would permit efficient, anisotropic propagation of energy between distant sites. We reasoned that such solid-like regions should display distinctive mechanical features such as a tendency to anisotropically propagate physical displacements. To test this in PDZ3, we measured the structural and thermodynamic effects of a mutation at Val386. Position 386 is located on the backside of the protein with the side chain facing towards the core. SCA suggests that this position is most strongly coupled to positions 322, 362, and 345 (figure 3-2C).

To characterize the energetic effect of a perturbation at position 386 on binding, we measured binding energies of the V386I mutant to both class I and class II peptides by ITC. Both P_{WT} and P_{T7F} binding measurements show an approximately three fold

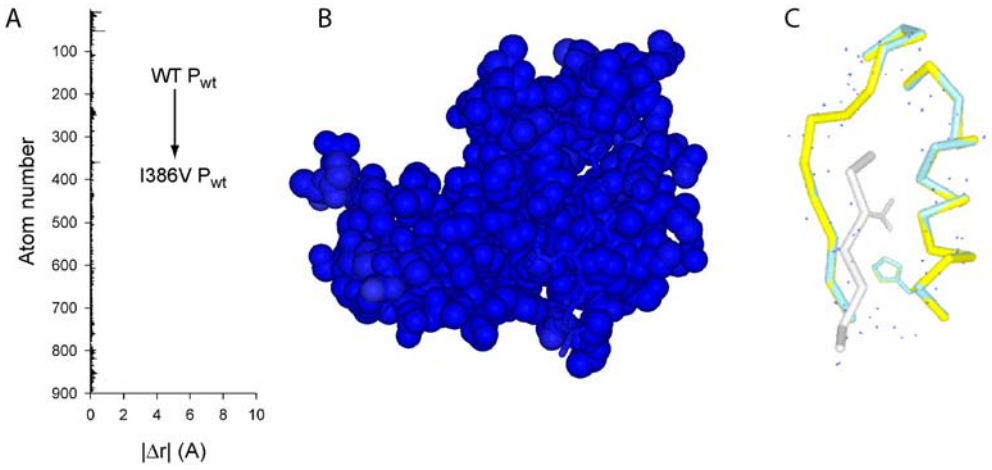


Figure 3-7. V386I has little structural effect. (A) Bar graph shows atomic displacements in protein (black bars) and peptide (gray bars) induced by the V386I mutation. Atom numbering follows PBD numbering. (B) Colorimetric representations of displacements are shown from blue (low) to red (high) on a CPK model of the WT protein with bound peptide shown as stick model. (C) Vector diagram show least squares overlay of Ca trace of WT (protein in blue and peptide in white) and V386I (protein in yellow and peptide in gray) states. Vectors are drawn from atomic centroid in WT Pwt state to V386I Pwt state. Clearly, V386I causes little detectable structural perturbation.

destabilization (-7.64 ± 0.12 kcal/mol and -5.41 ± 0.11 kcal/mol, respectively) relative to wild type PDZ3. To understand the structural basis of these destabilizations we solved the structures of V386I in complex with P_{WT} and in complex with P_{T7F}. Comparison of the WT and V386I structures bound to P_{WT} showed very little significant change between the two structures (figure 3-7). Thus, from this data set it is impossible to determine the mechanism underlying the destabilization caused by V386I.

The analysis of V386I highlights several critical issues in correlating structural and energetic effects of mutations. While atomic displacement may be indicative of a change in the energetic state between two structures being compared, no correlation can be made between the magnitude of displacement and the magnitude of the energetic effect. The physical basis of the small energetic effect of V386I may be at the limit of the resolution of structural analysis. Furthermore, as noted previously, crystal structures do not capture all critical physical features of proteins. The mutation may exert its influence on the dynamic features of the protein, such as coupled motions among binding pocket residues, which would not be observed in crystal structures.

In an attempt to further characterize the effect of V386I, we solved the peptide-free structure of the mutant. Though isomorphous with other structures analyzed, the peptide-free structure showed different crystal contacts as reflected by a significantly smaller unit cell size (87.0 A). Consequently, comparison with other structures would not only report energetic differences caused by the mutation but also effects induced by crystal packing. Given the small energetic and structural effects observed in the interaction between V386I and either peptide, we decided not to further pursue thermodynamic and structural studies of this mutant (though a structure of the double mutant H372Y V386I bound to P_{wt} was obtained).

Understanding the interaction between I359V and H372Y

In the thermodynamic and structural analysis of the interaction between positions 372, 322, and P_{-2}, we showed the power of SCA to predict a functionally important cooperative interaction. If the results of SCA indeed reflect the energetic interactions in proteins, then the converse should also apply; that is, interactions predicted to be weak should show thermodynamic and structural independence. We tested this by analyzing the energetic interaction between positions 359 and 372. Position 359 serves as a particularly appropriate control because it is located in the core of the protein and is highly conserved but statistically uncoupled to 372 (0.04 kT*, p = 0.38) and located in the core of the protein.

Thermodynamic cycle analysis based on ITC binding measurements with P_{WT} showed that the I359V mutation was indeed not thermodynamically coupled to H372Y. To confirm that these perturbations are structurally independent, we solved the P_{WT}-bound structures of I359V and the H372Y I359V double mutant (table 3-2). The I359V mutant bound to P_{WT} is essentially identical to wild type PDZ3 bound to P_{WT} (figure 3-8). As discussed above, H372Y has two structural effects: 1) proximally, it prevents the N terminal end of the peptide from binding and 2) distally, the carboxylate binding loop remains in the unclamped conformation. In the background of I359V, H372Y shows exactly the same effects, confirming that structural independence correlates with thermodynamic as well as evolutionary additivity. These findings are consistent with the observed correlation between structural and thermodynamic additivity in other systems [31, 32].

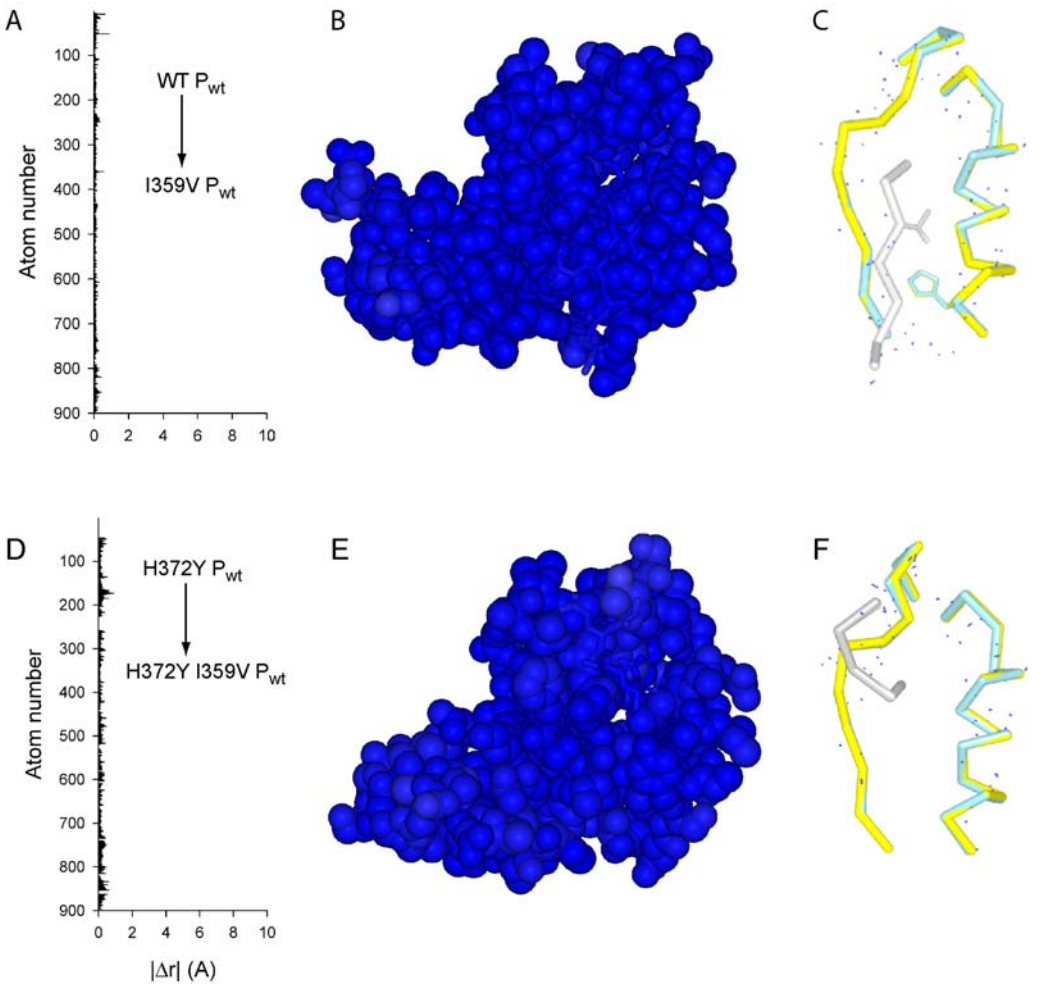


Figure 3-8. Structural interaction between H372Y and I359V. Figures show structural effects of H372Y in two different backgrounds: 1) WT bound to P_{WT} (A, B, C) and, 2) H372Y bound to P_{WT} (D, E, F). Bar graphs (A, D) show little structural change induced by I359V in either background. CPK renditions of PDZ domain (B, E) with colorimetric representation of atomic displacements from blue (low) to red (high) simply reflect that I359V causes minimal atomic displacements in either background. Vector plot (C) on overlayed Cα traces of WT (light blue) bound to P_{WT} (white) and I359V (yellow) bound to P_{WT} (gray) indicate no structural changes; similarly, there are no changes (F) between H372Y (light blue) bound to P_{WT} (white) and H372Y I359V (yellow) bound to P_{WT} (gray).

Conclusion

Summary of results

This chapter presents a set of experiments focused primarily on elucidating the mechanistic basis for coupled interactions in the interface between PDZ3 and a peptide ligand. Thermodynamic data reveal that the strength of the specificity-determining contact between H372 and P_{-2} depends on the presence of glycine at position 322, located on the opposite side of the binding pocket. Mutation of this glycine to alanine significantly reduces the strength of the coupling and, therefore, the ability to distinguish specific binding interactions. As a consequence, position 322 controls the evolvability of the PDZ domain. In a wildtype background, the H372Y mutation can flip the binding profile of the domain so it prefers class II ligands over class I ligands by two fold. However, in the G322A background, the evolutionary potential at position 372 is dramatically reduced; in this context the H372Y mutation creates a protein with essentially equal preference for both classes.

How does this coupled interaction happen? Structural studies of the mutants in complex with ligands demonstrate that glycine at position 322 endows the carboxylate binding loop with flexibility such that a conformational change in this region is sensitive to class-specific interactions at a distant site. The G322A mutation structurally uncouples conformational change in the loop from peptide binding. These findings provide a clear example of a regional structural inhomogeneity tuned to allow long-range cooperativity that optimizes both specificity and evolvability. Indeed, the role of position 322 as an evolutionarily conserved energetic lynchpin was also demonstrated by recent experiments that revealed its role in Par-6 allosteric regulation. Together, the data demonstrate the

power of SCA to identify functionally critical energetic interactions, even when not obvious from structure alone.

Physical evidence for pathways in other proteins

Based on the experiments described above, a combination of crystallography, NMR relaxation, and molecular dynamics experiments should be able to detect structural units critical for energy propagation. In fact, evidence for the presence of structurally inhomogeneous units tuned for energy propagation has been found in several systems. For example, NMR relaxation and molecular dynamics experiments suggest the presence of contiguous networks of atoms with correlated motion in dihydrofolate reductase and lactate dehydrogenase [40, 41]. The coupled motions occur on time scales ranging from femtoseconds to milliseconds and involve both active site and distant exterior residues; kinetic measurements of the effects of mutagenesis at these positions suggest the coupled motions are important for promoting catalysis. In a second example, structural studies of kinesin mutants attempted to understand the physical basis of its allosteric regulation [42]. Kinesin is a motor protein whose ATPase activity is enhanced when a distant surface region of the protein is bound to filament. Structures of kinesin mutants that uncoupled this allosteric regulation revealed structural changes that define a connected pathway from the microtubule binding site to the ATP binding site. The authors of this work suggest this pathway is a means of communicating the energy of binding directly to the active site and is therefore critical for motor movement. These and other examples [43, 44] indicate an emerging ability to rigorously characterize the internal physical features of proteins and how these features relate to functional behavior. While the available methods

partially complement one another, they cannot, even taken together, provide a complete mechanistic understanding. However, guided by the the energetic map provided by SCA, these techniques should ultimately provide significant insight into the physiochemical basis of function.

Future work to understand coupling

While the data presented in this chapter provide useful insights into mechanisms tuning binding energy in the PDZ domain, they do not provide a complete physiochemical understanding of how binding energy is tuned in PDZ3. The analysis shown here only focused on one interaction suggested by SCA. The network graph in figure 3-2C shows 25 other interactions each of which has its own unique physical mechanism. How do these other atomic interactions contribute to the binding energy? Furthermore, these experiments only address the correlation of atomic displacements with thermodynamic coupling and do not reveal the role of the dynamic state of the protein. Another physical change that occurs with peptide binding is a significant decrease in entropy (indicated by temperature factors) in several regions of the PDZ domain. These entropic changes may play a significant role in coupling distant structural elements. A careful characterization of changes to the dynamic state of the domain, at both slow and fast time scales, will be a critical step in achieving a more complete mechanistic understanding of function.

Recent work from several labs has made significant inroads to a characterization of the dynamical dimension of proteins. NMR relaxation experiments were used to study changes in fast time scale dynamics (ps-ns) in the PDZ domain from human tyrosine

phosphatase 1E induced by peptide binding. The data indicate that ligand binding induces changes in dynamics at positions in the binding pocket at distant sites; importantly, these positions show strong correlation with the highly coupled network identified by SCA [45]. Molecular dynamics experiments using novel protocols to optimize the signal-to-noise ratio have also been used to probe the energetic architecture of PDZ domains. The data suggest that energetic perturbations on the coupled network in PDZ3 tend to propagate preferentially through the network; furthermore, perturbations off the network dissipate locally [46].

The data presented in this chapter, both from work in our lab and other labs, indicates that the networks identified by SCA not only correlate strongly with function but also have distinct physical features tuned to allow cooperative energy propagation. At present, these results only present a small fraction of the physical mechanisms operating in the network. The initial glimpses of the PDZ energetic architecture presage a richness of physical interactions represented by the SCA matrix. How are the fundamental inter-atomic forces combined to create the links in each energetic network? Each link in the network likely has a unique solution to this question; it is also possible that a particular network link has a different physical explanation in each member of a protein family. An extensive combination of SCA results and biophysical techniques, such as NMR, crystallography, and molecular dynamics, should expose the physical principles underlying the sequence-structure-function problem in proteins. Given the observation that the SCA network correlates with physical cooperativity at a ‘microscopic’ level, it is also worth understanding the organization of the network at a more ‘macroscopic’ level. Specifically, how are the links arranged in the network? The next chapter presents an analysis of the toplogy of the energetic architecture as revealed by SCA.

Protein	Synch.	Res	% comp	R factor	Last shell R	Mos.	Unit cell length	Unique refl.	# atoms	R_{free}/R
WT	ALS, 8.2.1	1.95	99.3	0.052	0.314	0.749	90.03	9569	881	26.9/23.6
	APS, ID19	1.8	99.9	0.038	0.461	0.249	89.09	11775	917	26.8/25.3
WT P_{wt}	ALS, 8.2.1	1.45	100	0.049	0.496	0.369	89.67	22549	1032	23.7/21.5
	APS, ID19	1.58	99.9	0.039	0.443	0.314	89.18	19565	1080	24.6/21.9
	APS, ID19	1.58	100	0.049	0.437	0.272	89.16	17216	1062	24.9/22.1
	APS, BM19	1.53	99.1	0.060	0.507	0.357	89.35	18905	1044	24.3/22.5
WT P_{T7F}	APS, ID19	2.0	99.5	0.067	0.510	0.705	88.91	8567	963	26.4/22.8
	ALS, 8.2.1	1.6	99.4	0.045	0.456	0.324	89.17	16573	1001	24.6/22.8
H76Y	APS, ID19	2.1	99.9	0.058	0.435	0.401	89.28	7623	878	26.5/23.7
	APS, ID19	2.1	99.9	0.057	0.467	0.662	89.18	7757	867	29.8/26.6
	APS, ID19	2.0	100	0.057	0.448	0.159	89.18	8727	899	29.3/25.8
H76Y P_{wt}	APS, ID19	2.1	98.4	0.051	0.532	0.647	88.29	7751	901	28.7/24.9
	APS, ID19	2.0	98.8	0.051	0.513	0.183	88.54	8771	893	28.4/27.3
H76Y P_{T7F}	ALS, 8.2.1	1.5	98.5	0.059	0.430	0.527	90.05	20273	1021	26.0/24.1
G26A	APS, ID19	1.97	99.9	0.058	0.413	0.411	89.27	9149	966	26.3/23.1
	APS, ID19	1.9	99.7	0.064	0.479	0.334	89.16	10036	960	25.6/22.9
G26A P_{wt}	APS, ID19	1.58	100	0.377	0.356	0.356	89.74	17567	1060	23.9/21.8
	APS, ID19	1.63	99.9	0.044	0.462	0.493	89.26	15773	1050	23.9/21.3
	APS, ID19	1.53	99.9	0.054	0.496	0.210	89.36	19033	1070	22.7/19.6
G26A P_{T7F}	APS, ID19	1.65	99.6	0.053	0.411	0.311	89.20	15155	1084	25.2/22.0
V90I	ALS, 8.2.1	1.85	99.9	0.045	0.492	0.912	87.00	10150	866	29.6/25.4
V90I P_{wt}	ALS, 8.2.1	1.5	100	0.041	0.506	0.378	89.27	20164	1039	25.8/22.9
	APS, ID19	1.67	99.9	0.055	0.502	0.231	89.36	14730	1045	24.2/21.8
V90I P_{T7F}	APS, BM19	1.92	100	0.053	0.510	0.385	89.11	9805	925	29.5/27.6
I63V P_{wt}	APS, ID19	2.1	99.7	0.066	0.438	0.542	88.26	7274	975	28.2/27.0
	APS, ID19	1.65	99.8	0.037	0.309	0.398	89.31	15253	1063	24.2/22.0
	APS, BM19	1.55	99.8	0.053	0.431	0.193	89.61	18298	1016	26.8/24.0
I63V P_{T7F}	APS, BM19	2.15	99.7	0.078	0.450	0.477	88.97	7024	854	28.6/23.6
H76YG26A	APS, ID19	1.9	100	0.072	0.414	0.440	89.07	10076	918	28.6/23.6
H76YG26A P_{wt}	ALS, 8.2.1	1.65	99.7	0.042	0.333	0.710	89.39	15137	969	27.2/25.2
	APS, ID19	1.63	99.9	0.049	0.433	0.274	88.94	15632	1008	25.5/21.5
	APS, ID19	1.61	100	0.051	0.507	0.189	88.79	16122	999	27.3/23.9
H76YG26A P_{T7F}	APS, ID19	1.60	99.5	0.046	0.522	0.377	89.12	19056	1066	23.9/21.1
H76YV90I	APS, BM19	2.15	99.8	0.053	0.479	0.406	89.62	7092	862	29.2/27.6
H76YV90I P_{wt}	ALS, 8.2.1	1.77	99.0	0.043	0.509	0.656	88.96	12222	901	31.1/27.0
H76YI63V	APS, BM19	1.9	98.9	0.049	0.574	0.352	89.08	9958	867	29.8/28.4
H76YI63V P_{wt}	ALS, 5.0.1	1.82	99.9	0.052	0.495	0.445	88.49	11187	901	28.5/25.3
	APS, ID19	2.05	100	0.056	0.476	0.320	88.16	7859	870	27.5/26.6
H76YI63V P_{T7F}	ALS, 5.0.1	1.96	99.9	0.036	0.453	0.278	89.27	9255	866	28.3/27.4

Table 3-2 Crystallographic Data. For proteins where multiple structures were solved, the structure with the highest resolution was used for structural comparisons discussed in the text. (Abbreviations. Synch.: synchrotron where data was collected. Res: resolution. % comp: percent completeness. Mos.: mosaicity. Unique refl.: number of unique reflections).

Materials and Methods

Mutagenesis, expression, purification, crystallization. A pGEX4T-1 plasmid expressing a GST-PDZ3 fusion was obtain from Roderick MacKinnon. Site directed mutagenesis was carried out on PDZ3 of rat PSD-95 (residues 294-402) using standard polymerase chain reaction-based techniques. The domains were expressed as N-terminal glutathione S-transferase (GST) using the pGEX4T-1 vector in Escherichia coli [strain BL21(RP), Stratagene]. Cultures (1L) were grown in Terrific Broth to an optical density (600 nm) of 1.6 at 37C, induced for 4 hours at 25C with 500 μM isopropyl-β-D-thiogalactopyranoside, and then harvested by centrifugation. Pellets were resuspended in Buffer A (140 mM NaCl, 2.7 mM KCl, 10 mM Na_2HPO_4, 1.8 mM KH_2PO_4 (pH 7.3), 1.0 mM dithiothreitol (DTT)) with protease inhibitors (10 μg/ml pepstatins, 10 μg/ml leupeptins, 0.1 mM phenyl methyl sulphonylfluoride), lysed by sonication, and then centrifuged. Fusion protein was purified from the supernatent through GST affinity chromatography. The PDZ domains were cleaved off the resin through thrombin proteolysis (Sigma, 100U per 6 ml resin, 4 hr room temperature) and purified to homogeneity using a Mono Q HR5/5 (Amersham) column run with a linear gradient from low salt (1.0 mM DTT, 20 mM Tris-HCl (pH7.5)) to high salt (1.0M NaCl, 1.0 mM DTT, 20 mM Tris-HCl (pH 7.5)). The protein was dialyzed into 10 mM NaCl, 10 mM HEPES, 1.0 mM DTT (pH 7.2) and concentrated as necessary. For crystallization, the protein was concentrated to 33 mg/ml and either flash frozen or immediately used. Crystals of both peptide-free and bound PDZ proteins were grown in focused grid-screen trials with a range of Na Citrate (0.6 M to 1.1 M) and pH (7.0-7.6) conditions. In crystal trials of PDZ-peptide complex, dissolved peptide (either P_{wt} or P_{T7F}) was added to a 2:1 molar ratio with protein. Sitting

drops (4 μl) were set up in a 1:1 ratio with reservoir solution. Bipyrimidal crystals appeared within a week. Mutants containing the H372Y mutation required microseeding from WT crystals to initiate growth. Crystals were first stabilized in 0.9 M Na Citrate, 0.1 M HEPES (pH 7.4) and then cryoprotected by transferring into the same solution with progressively higher (up to 20%) glycerol concentration. An alternate and more efficient means of crystallization involved adding 20% glycerol to the crystallization trials solutions. However, this method required addition of microseeds to obtain crystals within a 3 to 5 days; without addition of microseeds crystals appeared variably in several weeks to months. Since this solution already contained 20% glycerol, cryoprotection only required incubation with well solution for 20 minutes. The crystals were frozen in propane. All crystals were screened at the home R-AXIS II or IV sources; only crystals showing diffraction better than 2.2 A and mosaicity less than 0.4 were saved and taken to the synchrotron for final data collection.

Binding measurements. Isothermal titration calorimetry (ITC) measurements were conducted at 25C using the VP-ITC microcalorimeter (MicoCal Inc.) by making 38 injections (8 μl each) of peptide ligand into PDZ protein. The peptides (P_{WT} is the C-terminal nine amino acids of CRIPT, N-TKNYKQTSV-C, and P_{T7F} simply mutates the antepenultimate position) were dissolved in 10 mM NaCl, 10mM HEPES (pH 7.2), 1.0 mM DTT. Concentration of peptide (0.5 mM to 2.8 mM) and protein (0.05 mM to 0.15 mM) in each run were determined from absorption at 280 nM. The ratio of peptide to protein concentrations was adjusted between 10:1 and 30:1 (depending on the dissociation constant) in order to reach saturation in the binding reaction. In all titrations, the reference power was 12.9 ucal/s and equilibration time was 180s between peptide

injections. Peaks were integrated and the titration curve was fit using Origin (MicroCal) assuming a 1:1 stoichiometry. The values given in Table 1 are averages and standard deviations from 3 measurements for each protein.

Data Collection, Structure Determination, and Analysis of Structures. Data were collected at synchrotrons indicated in Table 3-2. Diffraction data were indexed, integrated, and scaled with HKL2000. Structures were solved using the software Crystallography and NMR System (CNS). An initial model was obtained from rigid body refinement of the published PDZ3 structures (1BE9 and 1BFE). This model was then iteratively refined through rounds of simulated annealing, positional refinement, B-factor refinement, solvent modeling, and model building in O. A randomly selected set of reflections (5%) was flagged for statistical cross-validation calculations (R_{free}). The Ramachandran plot for all models show excellent geometry and no outliers for all models. Note that in the H372Y crystal structure, the N terminal end of the peptide forms a symmetry-related contact with the β2 strand of an adjacent PDZ domain in the crystal. The displacement in this part of the peptide is noted but neglected as an artifact of crystallization; we assume this novel contact does not contribute significantly to the structural differences observed. Analysis of all structures was performed in MATLAB (version 6.5.0.180913a (R13), Natick, MA). The MATLAB code written for this analysis is given in appendix B.

References

1. Suel, G.M., et al., *Evolutionarily conserved networks of residues mediate allosteric communication in proteins.* Nat Struct Biol, 2003. **10**(1): p. 59-69.
2. Lockless, S.W. and R. Ranganathan, *Evolutionarily conserved pathways of energetic connectivity in protein families.* Science, 1999. **286**(5438): p. 295-9.
3. 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.
4. Eriksson, A.E., et al., *Response of a protein structure to cavity-creating mutations and its relation to the hydrophobic effect.* Science, 1992. **255**(5041): p. 178-83.
5. Volkman, B.F., et al., *Two-state allosteric behavior in a single-domain signaling protein.* Science, 2001. **291**(5512): p. 2429-33.
6. Eisenmesser, E.Z., et al., *Enzyme dynamics during catalysis.* Science, 2002. **295**(5559): p. 1520-3.
7. Dueber, J.E., et al., *Reprogramming control of an allosteric signaling switch through modular recombination.* Science, 2003. **301**(5641): p. 1904-8.
8. Zarrinpar, A., S.H. Park, and W.A. Lim, *Optimization of specificity in a cellular protein interaction network by negative selection.* Nature, 2003. **426**(6967): p. 676-80.
9. Marles, J.A., et al., *Protein-protein interaction affinity plays a crucial role in controlling the sho1p-mediated signal transduction pathway in yeast.* Mol Cell, 2004. **14**(6): p. 813-23.
10. Sundberg, E.J. and R.A. Mariuzza, *Luxury accommodations: the expanding role of structural plasticity in protein-protein interactions.* Structure Fold Des, 2000. **8**(7): p. R137-42.
11. Miller, D.W. and D.A. Agard, *Enzyme specificity under dynamic control: a normal mode analysis of alpha-lytic protease.* J Mol Biol, 1999. **286**(1): p. 267-78.
12. Russo, A.A., et al., *Structural basis for inhibition of the cyclin-dependent kinase Cdk6 by the tumour suppressor p16INK4a.* Nature, 1998. **395**(6699): p. 237-43.
13. Min, J.H., et al., *Structure of an HIF-1alpha -pVHL complex: hydroxyproline recognition in signaling.* Science, 2002. **296**(5574): p. 1886-9.
14. Clackson, T. and J.A. Wells, *A hot spot of binding energy in a hormone-receptor interface.* Science, 1995. **267**(5196): p. 383-6.
15. Clackson, T., et al., *Structural and functional analysis of the 1:1 growth hormone:receptor complex reveals the molecular basis for receptor affinity.* J Mol Biol, 1998. **277**(5): p. 1111-28.
16. Patten, P.A., et al., *The immunological evolution of catalysis.* Science, 1996. **271**(5252): p. 1086-91.
17. Perutz, M.F., et al., *The stereochemical mechanism of the cooperative effects in hemoglobin revisited.* Annu Rev Biophys Biomol Struct, 1998. **27**: p. 1-34.
18. Kirschner, M. and J. Gerhart, *Evolvability.* Proc Natl Acad Sci U S A, 1998. **95**(15): p. 8420-7.
19. Looger, L.L., et al., *Computational design of receptor and sensor proteins with novel functions.* Nature, 2003. **423**(6936): p. 185-90.
20. Cesareni, G., et al., *Can we infer peptide recognition specificity mediated by SH3 domains?* FEBS Lett, 2002. **513**(1): p. 38-44.

21. Gee, S.H., et al., *Single-amino acid substitutions alter the specificity and affinity of PDZ domains for their ligands.* Biochemistry, 2000. **39**(47): p. 14638-46.
22. Espanel, X. and M. Sudol, *A single point mutation in a group I WW domain shifts its specificity to that of group II WW domains.* J Biol Chem, 1999. **274**(24): p. 17284-9.
23. Sheng, M. and C. Sala, *PDZ domains and the organization of supramolecular complexes.* Annu Rev Neurosci, 2001. **24**: p. 1-29.
24. 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.
25. Stricker, N.L., et al., *PDZ domain of neuronal nitric oxide synthase recognizes novel C-terminal peptide sequences.* Nat Biotechnol, 1997. **15**(4): p. 336-42.
26. Nourry, C., S.G. Grant, and J.P. Borg, *PDZ domain proteins: plug and play!* Sci STKE, 2003. **2003**(179): p. RE7.
27. Carter, P.J., et al., *The use of double mutants to detect structural changes in the active site of the tyrosyl-tRNA synthetase (Bacillus stearothermophilus).* Cell, 1984. **38**(3): p. 835-40.
28. Cohen, N.A., et al., *Binding of the inward rectifier K+ channel Kir 2.3 to PSD-95 is regulated by protein kinase A phosphorylation.* Neuron, 1996. **17**(4): p. 759-67.
29. Hall, R.A., et al., *The beta2-adrenergic receptor interacts with the Na+/H+-exchanger regulatory factor to control Na+/H+ exchange.* Nature, 1998. **392**(6676): p. 626-30.
30. 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.
31. Sandberg, W.S. and T.C. Terwilliger, *Engineering multiple properties of a protein by combinatorial mutagenesis.* Proc Natl Acad Sci U S A, 1993. **90**(18): p. 8367-71.
32. Vaughan, C.K., et al., *A structural double-mutant cycle: estimating the strength of a buried salt bridge in barnase.* Acta Crystallogr D Biol Crystallogr, 2002. **58**(Pt 4): p. 591-600.
33. Pawson, T., *Specificity in signal transduction: from phosphotyrosine-SH2 domain interactions to complex cellular systems.* Cell, 2004. **116**(2): p. 191-203.
34. Pauling L, C.D., Pressman D, Physiology Reviews, 1943. **23**: p. 203-219.
35. Kumar, S. and R. Nussinov, *How do thermophilic proteins deal with heat?* Cell Mol Life Sci, 2001. **58**(9): p. 1216-33.
36. Fersht, A., *Structure and mechanism in protein science : a guide to enzyme catalysis and protein folding*. 1999, New York: W.H. Freeman.
37. Wedemayer, G.J., et al., *Structural insights into the evolution of an antibody combining site.* Science, 1997. **276**(5319): p. 1665-9.
38. Peterson, F.C., et al., *Cdc42 regulates the Par-6 PDZ domain through an allosteric CRIB-PDZ transition.* Mol Cell, 2004. **13**(5): p. 665-76.
39. Garrard, S.M., et al., *Structure of Cdc42 in a complex with the GTPase-binding domain of the cell polarity protein, Par6.* Embo J, 2003. **22**(5): p. 1125-33.
40. Agarwal, P.K., et al., *Network of coupled promoting motions in enzyme catalysis.* Proc Natl Acad Sci U S A, 2002. **99**(5): p. 2794-9.
41. Hammes-Schiffer, S., *Impact of enzyme motion on activity.* Biochemistry, 2002. **41**(45): p. 13335-43.

42. Yun, M., et al., *A structural pathway for activation of the kinesin motor ATPase.* Embo J, 2001. **20**(11): p. 2611-8.
43. Pawlyk, A.C. and D.W. Pettigrew, *Transplanting allosteric control of enzyme activity by protein-protein interactions: coupling a regulatory site to the conserved catalytic core.* Proc Natl Acad Sci U S A, 2002. **99**(17): p. 11115-20.
44. Feher, V.A., et al., *Identification of communication networks in Spo0F: a model for phosphorylation-induced conformational change and implications for activation of multiple domain bacterial response regulators.* FEBS Lett, 1998. **425**(1): p. 1-6.
45. Fuentes, E.J., C.J. Der, and A.L. Lee, *Ligand-dependent dynamics and intramolecular signaling in a PDZ domain.* J Mol Biol, 2004. **335**(4): p. 1105-15.
46. Ota, N., *Personal communication.* 2004.

Chapter 4: The Energetic Topology of Proteins

Introduction

Reductionism has been the central theme of essentially every level of biological research, from the atomic interactions in proteins to the development of tissues. Reductionist approaches have proven invaluable in understanding the behavior of complex systems – both biological and nonbiological – where the behavior of the system is determined by collective interactions among its set of components [1]. For example, the understanding of energy propagation through electrical power grids depends on a detailed map of power lines and relay stations. An explanation of information flow through the internet depends on a map of websites and connecting links. Elucidation of how ligand binding to receptor ultimately results in altered cell behavior requires a detailed description of each intervening step in a signaling pathway. Despite the success of this approach, analyses of such data sets have shown that the output of complex systems depends on a network of interconnected interactions among the components. Regional power grids, local computer networks, and biochemical signaling pathways obviously do not exist in their respective environments in isolation. Rather, they form important interactions with other parts of their networks. Understanding the behavior of a complex system requires a macroscopic description of the architecture of its underlying network of interactions.

Recent research on numerous complex systems has attempted to understand the nature of complex behavior by taking a more global perspective of the individual components. These studies have turned to the emerging mathematical field of graph theory to model the complex interactions. Graph theory began in the 1950s with the work

of Paul Erdos and Alfred Renyi and has recently made a resurgence as applications have been found in numerous complex physical and social systems. In graph theory, any network may be represented as a graph in which vertices represent individual components and interactions between components are represented as links between vertices [1]. This depiction allows the development of parameters that describe the topology and properties of networks.

Similar to the systems described above, proteins display functional behavior that depends on complex interactions among amino acids. Until recently, the map of interactions in a protein were known by mutagenesis studies that generally only revealed contributions of single positions in limited regions of the protein. However, the results of SCA provide a global energetic map based on co-evolutionary interactions between positions in a protein alignment and allow, for the first time, an analysis of the energetic topology in proteins. This chapter presents work I have done that studies the energetic architecture in proteins as revealed by SCA.

Energetic Architecture from Structures

Gross inspection of tertiary structures of proteins generally indicates a compact arrangement of amino acids packed in a relatively ordered network of interactions. For example, figure 4-1A shows a CPK rendering of PDZ3 from PSD95 [2]. A simple interpretation of this and other structures is that the atomic contacts capture the structurally and functionally important energetic interactions. To understand the distribution of contacts and energetic interactions according to this view we can construct a graph representation of this domain as shown in figure 4-1B. Here, residues are

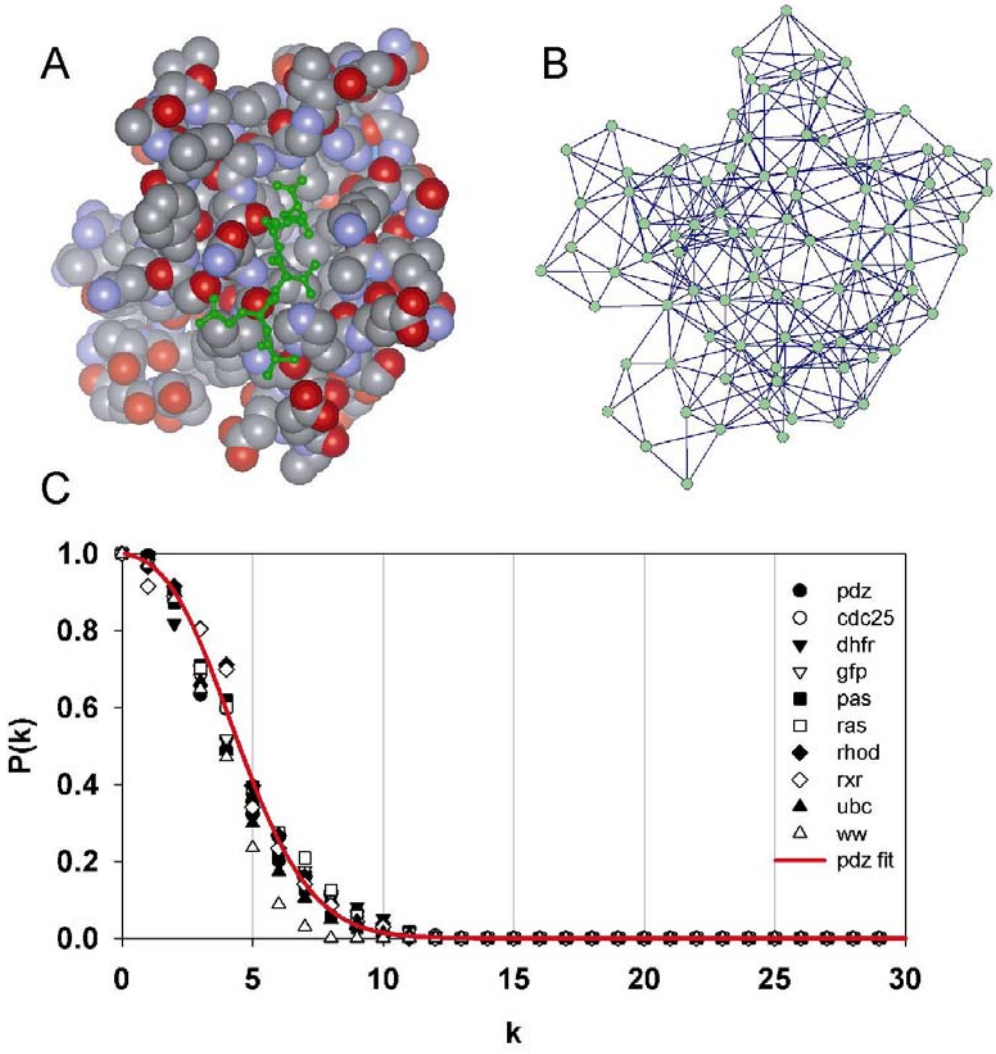


Figure 4-1. Contact network in proteins has a homogeneous network. (A) CPK representation of PDZ3 from PSD95 (co-crystallized peptide shown as stick) shows a tightly packed protein. (B) Network graph representation of PDZ3 where nodes correspond to positions in the protein and edges are drawn between nodes that are contacting (where contact is defined as within the sum of the atomic van der Waals radii plus 20%). Note that the particular position of a node here has no significance. (C) Cumulative histogram of contacts in 10 different proteins shows a very tight, homogeneous distribution. The red curve shows the fit for the PDZ domain data to a cumulative Poisson distribution. Together, the proteins have a mean of 4.93 ± 0.33 contacts.

represented by vertices and contacts between them are represented by edges; the number of links a node has with other nodes is defined as its degree, k. Following a common definition, contact between two residues occurs when any pair of residues between them is within the sum of their van der Waals radii plus 20%. The graph suggests a uniform network of contacts in which each residue makes approximately the same number of contacts with other residues. Indeed, a histogram of the fraction of residues making k or more contacts with other residues in PDZ3 is well fit by a cumulative Poisson distribution (figure 4-1C), the expected distribution of a homogeneous interaction network, showing an average connectivity of approximately five [3]. Indeed, the same distribution of contacts is observed in many structurally and functionally distinct proteins. Taken together, the ten proteins in figure 4-1C show an average connectivity of 4.93 ± 0.31 and clearly demonstrate the established notion that packing density is high within the core of all proteins [4, 5]. The Poisson distribution of vertex degrees, k, is a classic property of homogeneous or 'characteristic scale' networks and indicates that residues homogeneously make about the same number of direct interactions with little deviation at any site [6]. These results are consistent with recent studies that have also applied network analysis to the pattern of amino acid interactions in protein structures [7-9].

Though the characteristic scale network model nicely describes the contact topology of a protein structure, it fails to account for several functional and physical properties of proteins. NMR dynamics experiments show that specific subsets of residues display collective motions that are central to the biological role of the protein. For example, specific binding of an inhibitor to a catalytic antibody [10], the motions of residues during catalysis of peptide bond rotation [11], and stimulus-driven conformational changes in the light-sensor phototropin [12] all involve conformational

dynamics of specific regions of the respective proteins. Furthermore, these dynamic hotspots often occur both near and far from the active sites. Similarly, binding specificity in the human growth hormone receptor [13, 14], catalytic specificity in serine proteases [15], and allosteric communication in signaling proteins [16] all depend on distributed but specific interactions between residues. Thus, proteins demonstrate inhomogeneity in the pattern of energetic interactions between residues and these inhomogeneities are critical for function. Clearly, this heterogeneous energetic architecture is not revealed by the homogeneous, characteristic scale interaction network seen in atomic structures.

Energetic topology of PDZ domain from SCA

Experiments in several systems suggest that the energetic interactions predicted by SCA reflect important physiochemical interactions in proteins. For example, in PDZ domains the results of sequence-based perturbation analysis showed excellent correlation with thermodynamic coupling [17]. Additionally, positions identified in hemoglobin, G protein coupled receptors, and serine proteases showed excellent correlation with a large body of published data [16]. In more recent work, mutagenesis experiments show that SCA successfully maps the functional mechanism in G proteins and ligand binding domains [18, 19]. These results, combined with the more complete SCA formalism presented in chapter 2, set the basis for the description of the energetic topology of proteins. Construction of a network graph representation of the statistical coupling matrix was briefly described in chapter 2 and is reviewed here.

The matrix shown in figure 4-2A contains evolutionary coupling energies between all pairs of positions in the PDZ domain and suggests, as noted previously, that only a

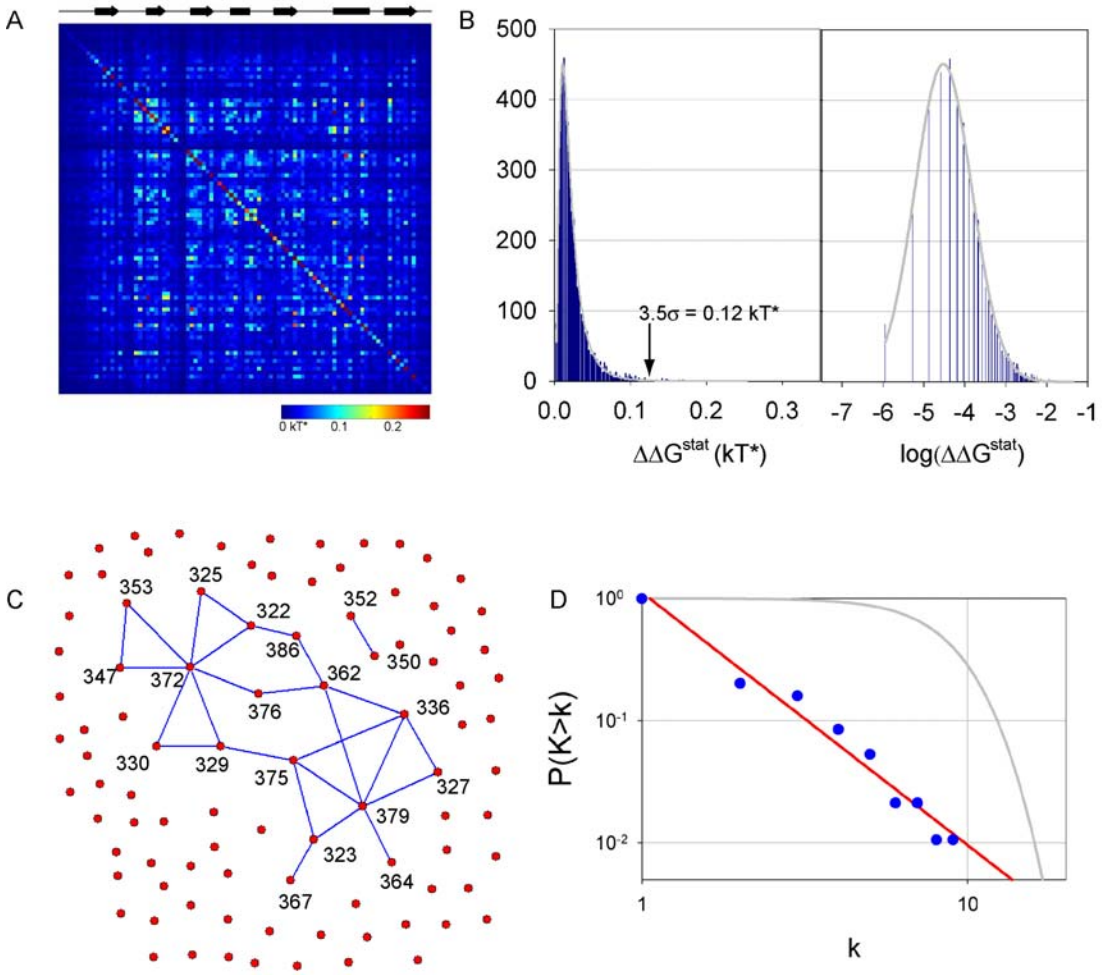


Figure 4-2 Construction of PDZ domain network. A) The 94 x 94 SCA matrix alignment contains global mapping of co-evolutionary interactions in the PDZ domain. B) Histogram in left panel shows that most pairs of positions have very low statistical coupling energies and a small subset are in the tail of the distribution. This distribution is often referred to as heavy tailed and is well fit by a log-normal distribution (grey curve). Log transformation of the x-axis converts this to a normal distribution (grey curve), shown in the right panel. Application of a 3.5σ cutoff identifies a set of 26 highly evolutionarily coupled interactions. C) A graph theoretic representation in which nodes represent PDZ positions and edges represent evolutionary couplings shows that the 26 highly coupled interactions connects a set of 19 positions in a nearly completely connected subgraph. D) Topological analysis of this graph shows that the cumulative degree distribution is well fit by a power law distribution with $\gamma_{PDZ} = 3.1$ (red curve). Note that in this graph, the degrees are shifted by one (so that k=1 represents the fraction of residues with zero or more connections) in order to display the entire range of connectivity degrees on a log-log plot. For comparison, the gray curve represents the cumulative Poisson distribution of contacts plotted in figure 4-1.

small subset of positions is highly coupled. A histogram of the $\Delta\Delta G^{stat}$ values (figure 4-2B, left panel) shows a skewed distribution in which most evolutionary couplings are very low and a small subset with larger values forms a so-called heavy tail in the distribution. These features are well fit by a log-normal distribution function ($r^2 = 1.0$). Taking the logarithm of the x-axis converts the log-normal into the easier to appreciate normal distribution (figure 4-2B, right panel). The fit to the data allows application of an energetic cutoff to determine significant co-evolution. A 3.5σ (0.12 kT*) cutoff to the PDZ domain identifies only 26 of 4371 coupling energies as significant; these significant interactions comprise the so-called heavy tail of the log-normal distribution. To study the topological properties of the strongly co-evolving positions in the PDZ domain, we created a graph representation of the co-evolution map (figure 4-2C). As in figure 4-1C, vertices represent positions in the PDZ domain. However, the edges connect positions with significant co-evolution. The number of links to a vertex defines the degree k of co-evolution for each position.

This view of the amino acid interaction network describes a topology very different from that inferred from the atomic structure (figure 4-1). Specifically, the co-evolution graph shows strong heterogeneity in the pattern of amino acid interactions such that most positions (80%) are not linked to any others, and the remaining 19 positions comprise a highly interconnected network of co-evolving residues. Furthermore, within this group of connected positions, there is significant variation in connectivity; most seem to have a few contacts and a small subset of positions (366, 372, 379) are highly connected. A log-log plot of the fraction of positions evolutionarily connected to k or more positions ($P(K>k)$) shows an unexpected finding: rather than a uniform exponential network observed in the contact graph (figure 4-2D, gray curve), the cumulative

distribution of amino acid interactions follows a power law relationship (figure 4-2D, red curve):

$$P(K > k) \propto k^{-(\gamma-1)} \qquad \text{(Eq 4-1)}$$

with γ_{PDZ} = 3.1. This mathematical relationship is the hallmark of a newly discovered class of heterogeneous networks termed scale-free, since the nature of this network topology is to have no characteristic scale for connections between vertices [6, 20].

If the observations are not simply random, then the interconnectedness and power-law distribution should depend on the arrangement of connectivity between residues. To test this, we scrambled the PDZ statistical coupling matrix and calculated the resulting degree distribution of the randomized network (applying the same energetic cutoff). A representative randomized matrix and network are shown in figure 4-3, A and B respectively. Figure 4-3C shows that randomizing the information in the coupling matrix produces a degree distribution of amino acid co-evolution different than that observed for the natural matrix. The degree distribution of the randomized trials is well described by a Gaussian distribution (figure 4-3C). Note that homogeneous networks in which the number of links is small, as in this case, do not have enough sampling to properly fit a Poisson distribution and instead fit a Gaussian distribution. As expected, a representative randomized network graph shows a homogeneous pattern of interactions that differs from the highly heterogeneous distribution observed in the natural matrix. These results demonstrate that the power law distribution of amino acid interactions is a significant and non-random feature in the evolutionary record of a protein family. It depends on a specific arrangement of mutual evolutionary interactions between residues.

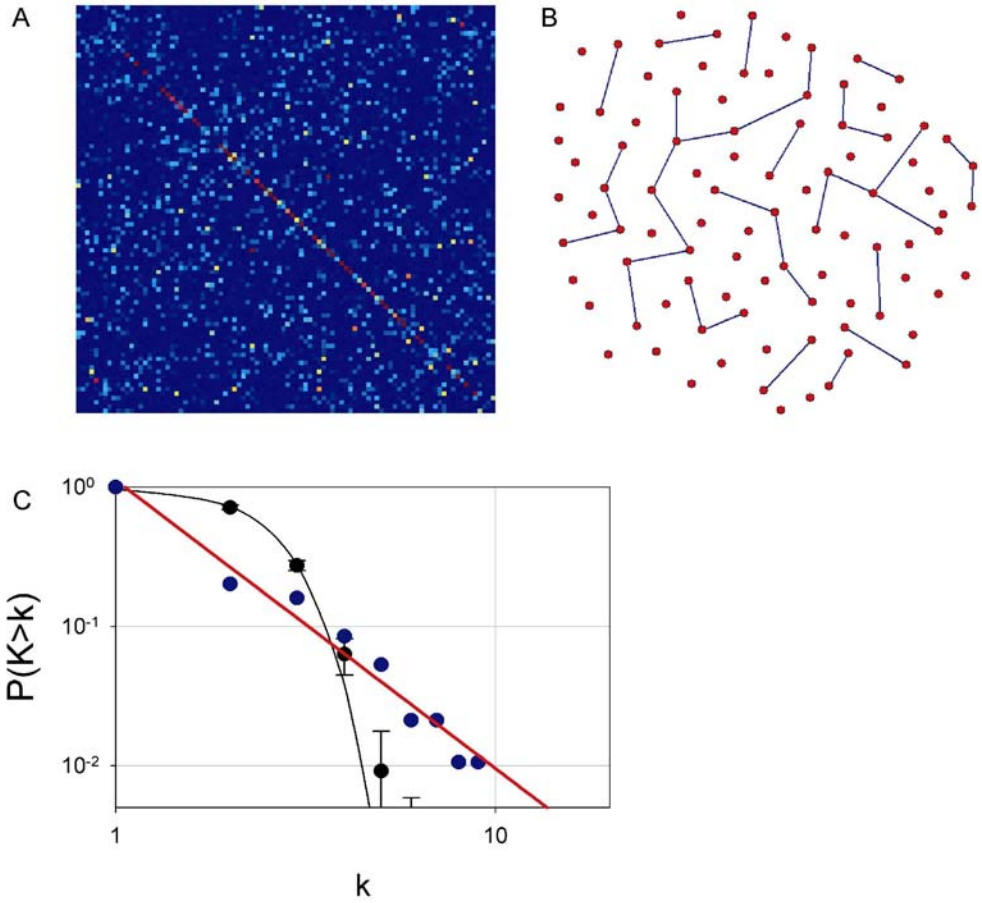


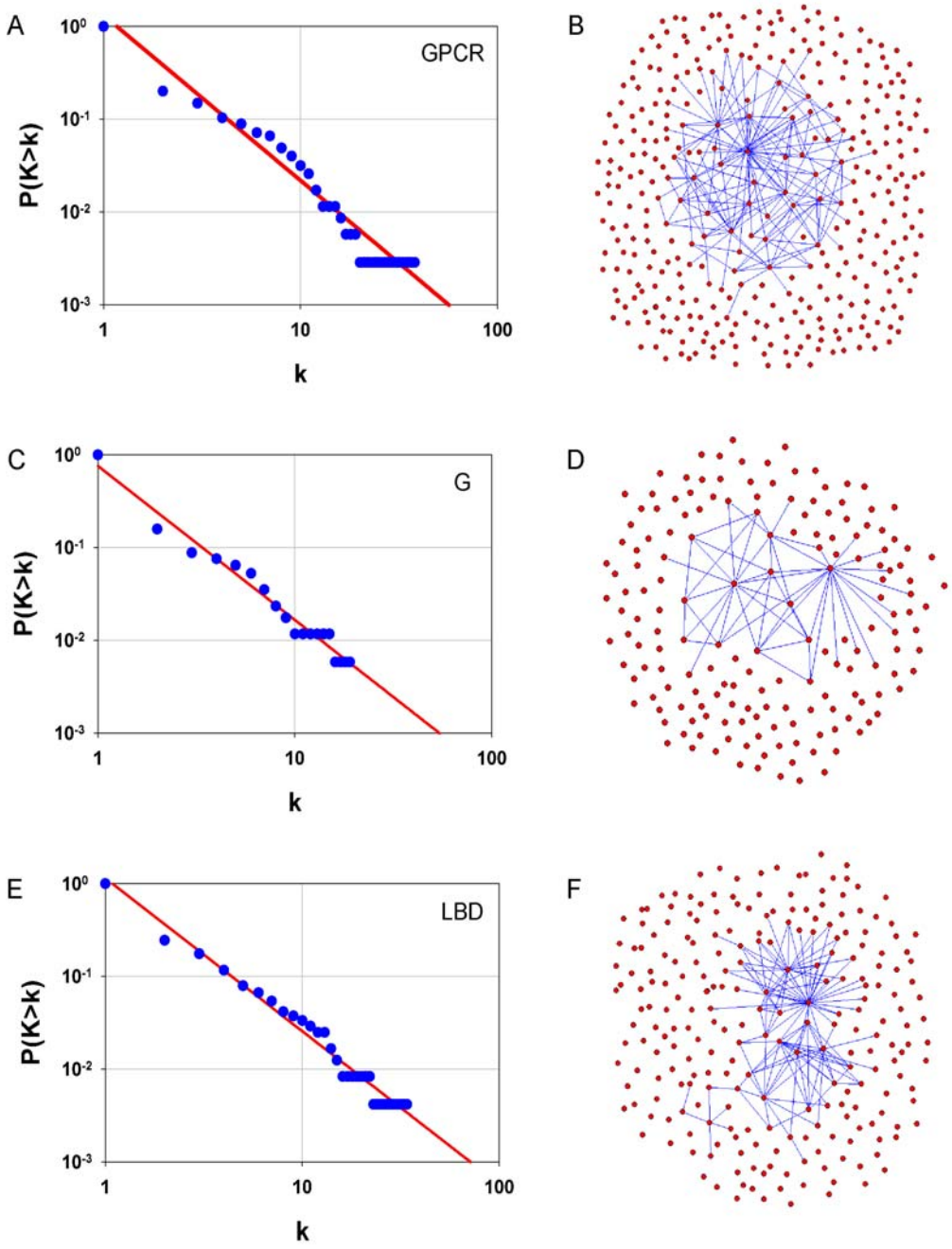
Figure 4-3. Randomization of PDZ domain network. A,B) A representative scrambled PDZ SCA matrix and corresponding graph representation. C) One hundred such trials showed a cumulative degree distribution (black dots) well fit by a Gaussian distribution (black curve), the hallmark of a random network. This shows the heterogeneous, power law distribution represented by the native matrix (figure 4-2, A and C), replotted here in the blue dots and red curve respectively, is a highly nonrandom and significant feature.

The observation of a scale-free energetic architecture provides a conceptual link between the protein structure-function problem and the emerging science of understanding the behaviors of self-organized networks. Scale-free (or power-law distributed) networks occur in many natural and man-made systems including the World-Wide Web [6], the Internet [21], social networks [20], the metabolic network in many organisms [22], and the protein interaction network in yeast [22, 23]. Though different in

nearly every way, all of these networks share three common aspects. First, they share a topology where the pattern of connections between vertices is strongly heterogeneous. Unlike the homogeneously connected Poisson-distributed networks, most vertices in a scale-free network are weakly connected and a few vertices, referred to as hubs, are highly connected and serve as central relay points that connect the whole network. Second, the different networks share the property of being self-organized rather than designed. Indeed, the emergence of scale-free architecture has been proposed to result from a simple generative mechanism in which new vertices tend to connect preferentially to already well-connected nodes, a process that may underlie the evolution of biological networks [6]. Finally, these networks all display small-world character, a phenomenon where vertices are connected by short path lengths despite a high degree of local clustering [24]. With regard to proteins, this type of network displays properties that are strikingly consistent with the empirical observation that free energy interactions between amino acid residues are heterogeneous and sparsely distributed rather than uniform and dense.

Energetic topology in diverse protein families

If the heterogeneous, scale-free topology is a general feature of all proteins, then many diverse protein families should display a distribution of interactions between residues that follows a power law function. We used the SCA to analyze six other protein families (the class A G-protein coupled receptors (GPCR), the guanine nucleotide binding (G) proteins, the ligand-binding domains of the nuclear hormone receptors (LBD), the chymotrypsin class of serine proteases (SP), the WW domains, and the ubiquitin


A
GPCR
P(K>k)
k
B
C
G
P(K>k)
k
D
E
LBD
P(K>k)
k
F

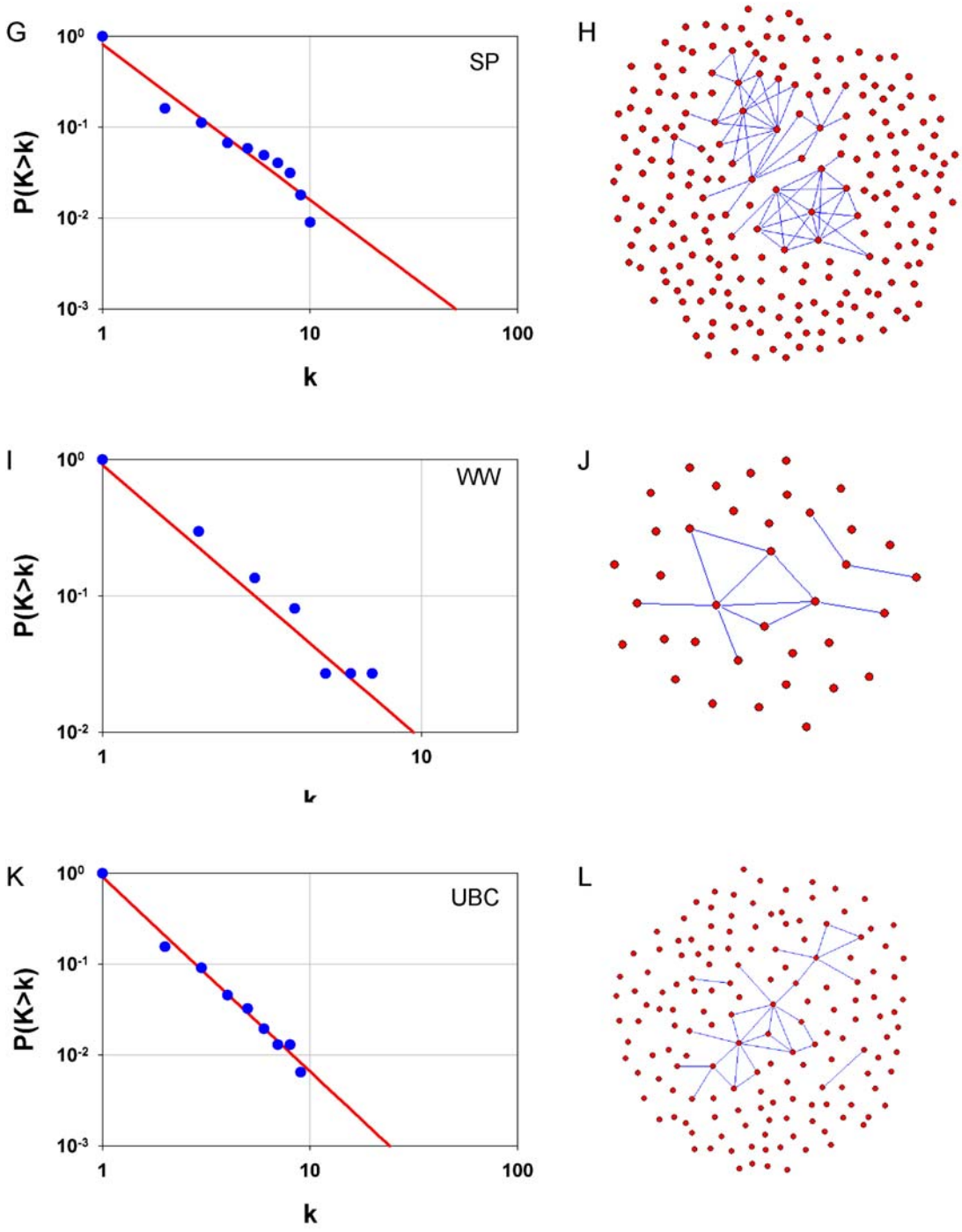


Figure 4-4 Many structurally and functionally diverse proteins display a scale-free topology of co-evolutionary interactions. Graphs show cumulative probability of residues making k or more connections in the SCA matrix (blue circles). Note that axes are log-scaled. As in the PDZ domain (figure 4-2C), fits to these data (red lines) demonstrate the distribution of co-evolutionary interactions in each family is well described by a power-law distribution. Graphs representation of highly co-evolving positions (cutoffs given in methods and materials) form nearly completely connected subgraphs in each protein family. A, B) G protein coupled receptors, $\gamma_{GPCR} = 2.8$. C, D) G proteins, $\gamma_{G} = 2.7$. E,F) Ligand binding domains, $\gamma_{LBD} = 2.7$. G, H) Serine proteases, $\gamma_{SP} = 2.71$. I, J) WW domain, $\gamma_{WW} = 3.0$. K, L) Ubiquitin conjugating enzymes, $\gamma_{UBC} = 3.1$.

conjugating enzymes (UBC)) that together represent a broad spectrum of diversity in protein structure, function, size, and location in either the membrane or cytosolic compartments. Figure 4-4 shows that, regardless of these differences, every one of these protein families shows a pattern of amino-acid interactions that convincingly follows the scale-free network topology. The adjacent network depictions indicate that, in virtually each case, the highly coupled positions form essentially completely connected subgraphs. The one exception occurs in the serine protease family where two separate subgraphs are observed; this is consistent with a previous SCA analysis that also showed the two subgraphs map to two distinct but structural units. The fits to the power law relationships show scaling coefficients that are similar for these protein families (γ_{GPCR} = 2.8, γ_{G} = 2.7, γ_{LBD} = 2.7, γ_{SP} = 2.7, γ_{WW} = 3.0, γ_{UBC} = 3.1). We do not understand this similarity mechanistically, but recognize that this suggests a common evolutionary pressure constraining the fraction of energetically interacting residues relative to weakly interacting ones in all proteins.

Functional importance of hubs: sensitivity to targeted attack

A central property of scale-free networks is that they show significantly higher resistance to random perturbation than do uniform characteristic scale networks. For example, random removal of vertices in a uniformly connected network causes a steady incremental increase in the average distance between vertices, but a similar experiment in a scale-free network causes little change in network connectivity [25]. This makes intuitive sense, since most vertices in a scale-free network are only weakly connected and contribute little to the overall connectivity of the network. However, this performance

advantage comes at a price; targeted removal of the hubs results in a dramatic loss of network connectivity [25]. Thus, scale-free networks are robust to random perturbation but fragile to targeted attack at the hubs. If proteins are energetically scale-free networks, then they should display tolerance to perturbation at weakly connected residues and sensitivity to perturbation of well-connected ones, regardless of where they are situated in the tertiary structure.

As a rigorous test of this hypothesis, we focused on two of the families (the GPCRs and PDZ domains), where comprehensive mutagenesis studies enable evaluation of the correlation between network connectivity and functional importance. The class A GPCRs are integral membrane receptors that transduce ligand binding at an externally accessible site to conformational change at distantly positioned cytoplasmic structural elements that mediate downstream signaling [26, 27]. The hub positions in the GPCR co-evolution graph include functionally crucial residues at the ligand-binding pocket, the cytoplasmic interaction site for G proteins, and known sites of allosteric communication between the two (table 4-1). Residues linked in the co-evolution graph are strongly associated with functional importance; of 76 total network positions, 66 display altered function upon mutagenesis in at least one GPCR family member (table 4-1). Network positions are also associated with sites of clinically relevant mutations. Eighteen of 25 point mutations known to constitutively activate the thyrotropin receptor (TSH-R) and cause hyperthyroidism and 6 of 12 mutations in the luteinizing hormone receptor (LH-R) that cause male precocious puberty are found on the GPCR co-evolution network [28]. Overall, we find that 44% of all residues comprising the GPCR co-evolution network are associated with constitutive activity in at least one GPCR family member. To calculate the statistical significance of these findings, we examined a saturation mutagenesis scan

Structural location	Rhodopsin position (general number)*	k	Role or mutational effect
Extracellular loops	102	1	Mutation in AT1-R reduced ligand binding affinity [31]. Mutation in V2-R altered ligand binding specificity [32].
	103	18	Mutation in NK2-R decreased ligand binding [33]. Mutation in V2-R reduced ligand binding but increased maximum signal transduction and is associated with X-Linked nephrogenic diabetes insipidus [34].
	105	1	Mutation in TRH-R decreased ligand binding [35].
	106	1	Mutation in C5a-R increased ligand binding affinity [36].
	174	1	
	175	6	Mutations in C5a-R [37], CCKB-R [38] and NTR1 [39] decreased ligand binding.
	178	1	Mutation in V2-R causes reduced vasopressin binding and is related to nephrogenic diabetes insipidus [40]
	278	1	Associated with CAM and hyperthyroidism in TSH-R [28]. Mutation in MSH-R decreased ligand binding [41]. Mutation in FSH-R increased ligand binding affinity but decreased signal transduction [42].
Chromophore binding pocket	44 (1.39)	3	Involved in agonist binding and Na^+ allostery in AA2A-R [43]. Mutation in CCKB-R decreases ligand binding affinity [38].
	113 (3.28)	6	Counterion in rhodopsin [44]; associated with CAM [45]. Mutation in CCR5-R reduced potency of ligand [46]. Mutations showed this position affects both agonist binding and potency in CB1-R [47].
	117 (3.32)	3	Associated with CAM in OPRD [29]. Mutation decreased ligand binding in AA2A-R [48], HH4-R [49], NK1-R [50] , D2D-R [51], and GnRH-R [52].
	212 (5.47)	3	Mutations in OT-R [53] and GRP-R [54] affected binding affinity.
	261 (6.44)	4	Associated with CAM and hyperthyroidism in TSH-R [55]. Associated with CAM and male precocious puberty in LSH-R [28]. Associated with CAM in C5a-R [56]. Mutations in A1BA-R gave higher agonist affinity but no signal transduction [57].
	265 (6.48)	7	Involved in chromophore tuning of rhodopsin [58]. Associated with CAM in OPRD [29]. Mutation in GnRH-R reduced ligand binding and signal transduction [59]. Mutation in AA3-R mutant bound agonist normally but was inactive [60].
	268 (6.51)	7	Mutation reduced ligand binding in AchM1-R [61], MSH-R[62], CCKB-R [38], and GnRH-R[63]. Mutation reduced receptor activation in AT1-R [64] and rhodopsin [58].
	269 (6.52)	4	Mutation in AA2A-R weakened binding to agonist and antagonist [65]. Mutation in GnRH-R [63] and D2D-R [66] reduced binding to ligand and signal transduction.
	293 (7.40)	16	Associated with CAM in AchM1-R [67]. Mutation in ETB-R increased signal transduction [68].
	296 (7.43)	2	Schiff base link to chromophore of rhodopsin; mutations associated with CAMs [45]. Associated with CAM in OPRD [29]. Mutations in NK2-R [33] and 5H2A-R [69] decreased ligand binding affinity. Mutants in LSH-R [70] and AT1-R [71] had normal ligand binding but decreased signaling.
Transmembrane domains	48 (1.43)	3	Associated with CAM and hyperthyroidism TSH-R [28].
	51 (1.46)	3	Associated with CAM in LSH-R [72]. Mutation in rhodopsin caused ADRP with normal retinal binding [73].
	54 (1.49)	2	Associated with CAM in TSH-R [74]. Mutation in ET1-R reduced ligand binding [75] .
	58 (1.53)	3	Mutation in rhodopsin caused receptor to accumulate in endoplasmic reticulum and is associated with ADRP [73].
	73 (2.40)	8	Mutation in rhodospin caused small decrease in transducin activation [76]; in TSH-R caused slightly decreased TSH binding and cAMP response [77].
	74 (2.41)	1	
	75 (2.42)	4	
	78 (2.45)	20	
	91 (2.58)	3	Mutations decreased binding affinity in Prostacyclin-R [78], CCR5-R [46], and C5a-R[79].
	92 (2.59)	3	Mutation in Prostacyclin-R affected activation but not ligand binding [78]. Specificity determinant in CCR5-R [46].
	111 (3.26)	1	Mutation in AchM1-R decreases ligand binding affinity [80].
	120 (3.35)	1	Associated with CAM in AT1-R [81].
	124 (3.39)	6	Associated with CAM in OPRD [29]. Rhodopsin mutant had altered activation kinetics and transducin activation was decreased [82]. Mutation in LSH-R decreased potency [83]. D2D-R mutant had decreased agonist affinity but increased antagonist affinity [84].
	125 (3.40)	4	Associated with CAM and hyperthyroidism in TSH-R [28]. Associated with CAM in C5a-R [56]. Mutation in AchM1-R gave increased signaling efficiency with ligand and agonist [61].
	126 (3.41)	2	Involved in photoisomerization in rhodopsin [85].
	129 (3.44)	2	Associated with conformational change in B2Adr-R [86].
	131 (3.46)	2	Mutation in AchM1-R decreased acetylcholine potency [61].
	132 (3.47)	1	AchM1-R mutant had increased ligand binding affinity and is a CAM [87] .
	134 (3.49)	12	Part of DRY motif. In rhodopsin, this position is protonated upon activation and is associated with CAMs [88]. Also associated with CAM in A1BA-R [89]. Mutation in GnRH-R [90] and OT-R [91] abolished ligand binding and signal transduction. Mutation in Mel1A-R impaired activation [92].
	136 (3.51)	7	Part of DRY motif. Mutations in AA3-R [57] and CB2-R [93]reduced potency of respective ligands. Mutation in B2-R reduced signaling [94]
	138 (3.53)	2	Mutations in FSH-R [95] and GnRH-R [96] decreased signal transduction.
	140 (3.55)	1	Critical for coupling IL8-R to G protein [97].
	149 (4.38)	5	
	152 (4.41)	1	Mutation in ACTH-R decreased both ligand affinity and maximal response [98]. Mutation in MSH-R caused decreased potency[99].
	157 (4.46)	1	Associated with CAM in OPRD [29]. Mutation in C5a-R gave normal binding but decreased signaling [79]. Prostacyclin-R mutant had low binding affinity [78].
	164 (4.53)	3	Mutation in AchM1-R reduced agonist binding [67].

	170 (4.59)	6	Associated with CAM in OPRD [29]. Mutants in both AchM1-R and AchM3-R reduced binding to agonist and antagonist [100].
	171 (4.60)	3	Associated with CAMs in AchM1-R [67]. Associated with CAM and hyperthyroidism in TSH-R [101]. Mutation in LSH-R reduced ligand binding and signal transduction [102].
	203 (5.38)	1	Associated with CAM in OPRD [29]. Mutation in V2-R reduced ligand binding affinity [44].
	215 (5.50)	3	Associated with TSH-R CAM and thyroid adenoma [103]. Mutation in ETB-R associated with Hirshsprung's disease and causes reduced surface expression and signal transduction [104].
	219 (5.54)	6	Associated with CAM and hyperthyroidism in TSH-R [28]. Associated with CAM and male precocious puberty in LSH-R [28].
	222 (5.57)	8	
	247 (6.30)	4	Associated with CAM and hyperthyroidism in TSH-R [105]. Associated with CAM and male precocious puberty in LH-R [106]. Also associated with CAMs in FSH-R [72], and HM1-R [107]. Mutation in GnRH-R caused decreased signal transduction [108].
	249 (6.32)	1	Associated with CAM and hyperthyroidism in TSH-R [109]. Also associated with CAM in OPRD [29]. Mutation in muscarinic HM1-R caused decreased binding affinity to agonist [107].
	253 (6.36)	3	Associated with CAMs in OPRD [29] and LSH-R [109]. FSH-R mutant had normal ligand binding but reduced maximal response [110].
	254 (6.37)	2	Associated with CAM and male precocious puberty in LSH-R [111].
	258 (6.41)	1	Associated with CAM and hyperthyroidism in TSH-R [109]. Associated with CAM and male precocious puberty in LSH-R [112].
	259 (6.42)	1	Associated with CAM and hyperthyroidism in TSH-R [109]. Also associated with CAM in LSH-R [28].
	294 (7.41)	2	Associated with CAM and hyperthyroidism in TSH-R[28]. Mutation in ETB-R decreased signal transduction [68].
	295 (7.42)	1	Mutation in AA1-R decreased binding to agonist [113].
	298 (7.45)	12	Associated with CAM and hyperthyroidism in TSH-R [28]. Also associated with CAM in LSH-R [114]. Mutation in AA2A-R decreased ligand binding [115].
	299 (7.46)	4	Mutations in LSH-R [116] and A2a-R [65] show decreased binding to agonists.
	300 (7.47)	2	Associated with CAM and hytperthyroidism in TSH-R [28]. Mutation in AT1-R [117] and [114] decreases potency of bound agonist.
	302 (7.49)	2	Part of NPxxY motif. Associated with CAM in TSH-R [118]. Mutation in CCKB-R prevents G protein activation though binding is normal [119]. Mutation in TRH-R decreased maximal activity [120].
	305 (7.52)	1	Associated with CAM and hyperthyroidism in TSH-R [28]. Mutation in AchM1-R increased ligand binding affinity [67].
Cytoplasmic loops	68	2	Mutation in GnRH-R reduced cAMP production [121].
	69	1	
	141	1	Associated with CAM in AchM5-R and is involved with coupling to G protein [122].
	144	4	Mutation in rhodopsin reduced phosphorylation by rhodopsin kinase. Mutant in MSH-R bound ligand normally but was defective in signal transduction [123].
	230	2	Involved in AT1-R activation [124].
	308	1	Mutation in LSH-R is inactivating and is associated with male pseudohermaphroditism [125].
	313	4	Undergoes conformational change during rhodopsin activation [88].
	317	3	

* Numbers in parentheses follow the general numbering scheme proposed by Ballesteros et al [126]. The number before the decimal represents the helix number (1-7) and the number after the decimal refers to the position relative to the most conserved residue (assigned as 50) in that helix. Loops and the N and C terminal domains are more variable in length and do not have a general numbering. The organization of positions into structural locations is based on the X-ray crystal structure of rhodopsin [127]. Positions for which there are insufficient published data were left blank.

5H2A-R: 5-hydroxytryptamine2A (serotonin2) receptor. A1BA-R: (1B)-adrenergic receptor. AA1-R: adenosine A1 receptor. AA2A-R: adenosine A_{2A} receptor. AA3-R: adenosine A3 receptor. AchM1-R: M1 muscarinic acetylcholine receptor. AchM2-R: M2 muscarinic receptor. AchM3-R: M3 muscarinic receptor. AchM5-R: M5 muscarinic receptor. ACTH-R: adrenocorticotropic hormone receptor. ADRP: autosomal dominant retinitis pigmentosa. AT1-R: type 1 angiotensin II receptor. B2Adr-R: 2 adrenergic receptor. B2-R: Bradykinin receptor. C5a-R: C5a anaphylotoxin receptor. CAM: constitutively active mutant. CB1-[128]R: cannabinoid 1 receptor. CB2-R: CB2 cannabinoid receptor. CCK(A/B)-R: cholecystokinin(A/B) receptor. CCR5-R: chemokine type 5 receptor. CCR2-R: chemokine type 2 receptor. D1D-R: D1 dopamine receptor. D2D-R: D2 dopamine receptor. ET1-R: Endothelin 1 receptor. ETB-R: Endothelin B receptor. FSH-R: follicle stimulating hormone receptor. GnRH-R: gonadotropin releasing hormone receptor. GRP-R: gastrin releasing peptide receptor. HM1-R: muscarinic acetylcholine Hm1 receptors. HH4-R: histamine H4 receptor. LSH-R: luteinizing stimulating hormone receptor. Mel1A-R: Mel1A melatonin receptor. MSH-R: melanocyte stimulating hormone receptor. NK1-R: tachykinin NK_1 receptor. NK2-R: Neurokinin 2 receptor. NTR1: neurotensin receptor 1. OPRD = -opioid receptor. OT-R: oxytocin receptor. TRH-R: thyrotropin releasing hormone receptor. TSH-R: thyroid stimulating hormone receptor. V2-R: V2 vasopressin receptor.

Table 4-1 G protein coupled receptor (GPCR) network positions and reported functional importance.

of the δ-opioid receptor that permits experimental assessment of every residue [29]. The data show that 28 of 342 aligned residues mutated (~8%) cause constitutive activation, of which 40% occur at positions linked by edges in the GPCR co-evolution graph. A statistical evaluation of these data indicates that residues linked in the scale-free network are significantly associated with constitutive activity upon perturbation ($p<0.03$). This is particularly striking given that only a small fraction of residues are linked at all by co-evolution, and constitutive activity is but one measure of altered function.

The selective functional importance of network positions is demonstrated in PDZ domains as well. Skelton and co-workers have reported an alanine scanning mutagenesis of 36 (out of 94 total) aligned residues in the Erbin PDZ domain that comprise the environment of the active site [30]. These authors show that 13 of these mutations display large effects on substrate binding, of which 8 are network positions in the PDZ co-evolution analysis. In contrast, only 1 of the 21 mutations showing no functional effect are network positions ($p < 0.001$). Thus, both the GPCRs and the PDZ domains show resistance to mutation at unconnected sites and sensitivity to mutation at sites linked in the network. Less comprehensive but substantial evidence reinforces this result in all the protein families tested. Co-evolving hubs in the G protein and the LBD families mediate long-range allosteric coupling [18, 19, 129, 130], and in the serine proteases and the WW domain [15, 131, 132], mediate binding specificity. Together, the data demonstrate that the scale-free network architecture is strongly correlated with functional importance of residues in proteins.

Structurally non-intuitive sites are hubs

Where are the hubs in the atomic structure and how is the network as a whole physically organized? In the PDZ domain, the network comprises a physically contiguous group of amino acids (shown as a van der Waals surface) that defines both the peptide binding pocket and a sparse set of interactions within the protein core that connect the active site with distantly positioned sites. Interestingly, proximity to the active site is not well correlated with connection degree; some sites behind and far from the active site (336 and 362 in the PDZ3 numbering) are equally or more connected in the scale-free network topology than active site residues. Despite their distant location, these sites qualify as hubs. For example, residue 362 in the PDZ domain is located far from the peptide ligand (and was therefore not included in the structure-directed strategy for scanning mutagenesis of the Erbin PDZ domain), but makes 4 evolutionary links with other residues (figure 4-5) and physically links to the active site through packing interactions with residue 379. Similarly, residue 336 lies in the core beneath the active site and makes 4 co-evolutionary links. However, not all core residues are evolutionarily linked; residue 359 is equally buried, conserved and distant from ligand as residues 336 and 362, but it makes no evolutionary links at all.

To test the prediction that hubs are functionally important regardless of where they are situated in the tertiary structure, we made mutations in the PDZ3 directed by the network connectivity rather than by structural intuition, and measured the effects on peptide binding (table 4-2 and figure 4-4). Mutation of position 372 (H372A), an active site hub, results in a 30-fold destabilization of binding, consistent with the established

role of this position in mediating substrate specificity. However, mutation of position 336 or 362, non-active site hubs, also results in significant loss of binding (250 and 6.3-fold, respectively), while mutation of positions 359 and 328, evolutionarily unlinked residues, shows no significant change in binding despite proximity to network positions. Network residues with intermediate connectivity (322, 371, and 386) show intermediate effects. These data support the model that hubs in the scale-free co-evolution graph mediate

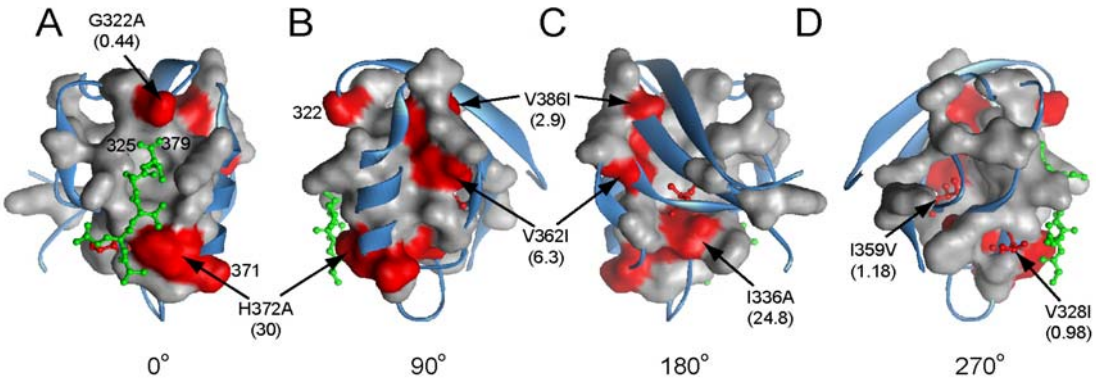


Figure 4-5 Functional architecture of the scale-free amino acid network in a PDZ domain atomic structure. A-D) Four successive views of 90^{o} rotations of PDZ3 from PSD95 with bound substrate peptide (green). The 19 residues in the co-evolution network of the PDZ domain are shown as a van der Waals surface (grey), with those residues discussed in the text numbered. The network positions comprise a physically contiguous sub-structure within the PDZ domain that defines the peptide binding pocket and a specific set of long-range sites that are linked through a sparse network of core contacts. The sites included in the mutagenesis study (table 4-1) are colored with fold changes on peptide binding relative to wild-type shown in parentheses. Dissociation constants were measured by isothermal titration calorimetry. The data show that hubs in the scale free co-evolution network make important energetic contributions even if distantly positioned and not predictable from the atomic structure.

protein function, either directly by acting at the active site, or indirectly through other network linkages that act as pathways of energetic connectivity. It is important to note that connection degree is unlikely to be the sole quantitative predictor of functional

importance [133]; some sites, such as the catalytic triad residues of serine proteases are nearly invariant and therefore exhibit little co-evolution with other sites, but are clearly critical for function [134]. In addition, the strength of connections is not considered in the current analysis, and will certainly influence the magnitude of functional effect upon perturbation. Nevertheless, the data demonstrate a core principle of scale-free networks in proteins: hubs play a dominant functional role.

Location of mutation	Protein	Connection Degree (k)	K_d (μM)	Fold Effect (mut/WT)
	WT		0.87 ± 0.13	--
Active site, on network	H372A	6	26.12 ± 2.82	30.0
	G322A	3	0.38 ± 0.02	2.3
Peripheral to or distant from active site and on network	G329A	6	55.31 ± 9.72	63.6
	I336A	7	21.55 ± 0.53	24.8
	V362I	7	5.50 ± 0.71	6.3
	S371A	1	2.01 ± 0.76	2.3
	V386I	4	2.50 ± 0.23	2.9
Off network	I359V	0	1.03 ± 0.18	1.2
	V328I	0	0.85	1.0

Table 4-2 Dissociation constants of PSD95-PDZ3 domain mutants. Measurements were made using isothermal titration calorimetry. Each K_d reports mean and standard deviation from three trials; the one exception was V328I which was only measured once.

Conclusions

In summary, the analysis shows that the topology of the network of energetic interactions in proteins has several recurring and surprising features:

1) Sparseness, such that most positions show evolutionary independence and a small subset have significant co-evolutionary interaction.
2) Organization, such that highly co-evolving residues form nearly completely connected subgraphs.

3) Heterogeneity, such that most positions have very few links and a few positions are hubs having many strong links. In all proteins, the arrangement follows a power-law distribution, the characteristic pattern of the scale-free class of the networks.
4) Functional correlation of network positions regardless of structural location. This is in agreement with the established property of scale-free networks to be robust to random perturbation yet sensitive to targeted attack.

This architecture is apparently predictable from a simple analysis of the evolutionary record of a protein family if sufficient and diverse sequence data are available. Given the wide spectrum of protein structures and function examined in this study, we suggest that this topology is a fundamental energetic feature in all natural proteins. The properties of this topology listed above raise potential applications, insights, and further questions, some of which I will discuss below.

Identification of critical functional sites in proteins

The limitations of mutagenesis and structural studies often make it difficult to interpret the complex arrangement of atoms in protein structures. The correlation between connectivity and functional importance suggests that SCA results could be used to focus attention on specific amino acids and regions of proteins. Previous studies as well as the PDZ domain structural studies described in chapter 3 demonstrate that SCA results indeed capture functionally important energetic interactions. Provided sufficient sequence information for a protein family is available, SCA results could be used to focus mutagenesis, structural, and dynamics experiments to more completely understand the

physical underpinnings of function. In addition, the observation of clinically relevant mutations in numerous GPCRs suggests SCA results may be of significant help, even in the absence of structure, in prediction of clinically important sites in other proteins.

Potential insights into physical mechanisms

The power law organization in the energetic architectures of proteins suggests behavioral features observed in other systems with this topology may also be found in proteins. The first observation of a physical system displaying power law distributed features came from studies of phase transitions. Here, phase transition is used in its broadest sense to include any phenomenon in which a disorder to order transition occurs. In this sense, both freezing of a liquid and the emergence of magnetization in a metal with decreasing temperature are considered phase transitions. Each system exhibits a unique critical point at which "the system is poised to choose between two phases [1]." Interestingly, measurements of numerous systems revealed that when the system is brought close to its critical point several key features, usually normally distributed, begin follow power law distributions. For example, at the liquid-gas critical point the distribution of droplet sizes follows a power law: many are very small and a few are very large. In another example, the correlation length of metals measures the length of aligned magnetic spins in atoms and refers to the "distance over which atoms communicate"; at high temperatures correlation lengths are randomly distributed and the metal has no magnetization [1]. However, at the particular critical temperature of a metal, correlation lengths follow a power law distribution and the metal becomes magnetized. Thus, the

observation of power-law distributed features is regarded as a sign that order is emerging from disorder.

Given the observations of disorder to order transitions in proteins and, now, the finding of power-law distributed interactions, an interesting concept emerges. The two observations suggest that proteins are built close to a transition point, energetically close to a phase transition. Functionality, then, arises from molecular interactions specifically designed to push a protein over its transition point and to trigger a built-in disorder to order transition. The SCA network may identify the core structural elements necessary for a liquid to solid transition. This model is consistent with crystallographic and mutagenesis data that show proteins in nature are not optimally stable and functionality requires a degree of disorder [135]. If true, such phase transitions in proteins may be reflected by changes in the distribution of 'correlation lengths' in a protein, the length over which atoms in proteins propagate energy. For example, the distribution of correlation lengths in the ground state of a signaling protein such as a GPCR would be normally distributed with a small mean and standard deviation. Upon activation, the distribution of correlation lengths may become power law distributed allowing a small subset of atoms to propagate energy over a significant distance and trigger a conformational change. Such changes in correlation lengths should be reflected in coupled motions of atoms; future studies of the dynamic state of proteins may reveal these features.

A generative model for the energetic architecture

The energetic architectures in proteins as revealed by SCA display several remarkable and unexpected features. The analysis described in this chapter suggests not only that co-evolutionary interactions are sparsely distributed, but also that they are organized in an inter-connected and heterogeneous manner – an organization that fits a scale-free distribution. Clearly not the product of a random process, this surprising arrangement of energetic interactions demands explanation. Why should energetic interactions in proteins be scale-free? The highly non-random nature of the energetic architecture suggests this common topology is the solution to specific evolutionary pressures.

As a hypothesis, the heterogeneous energetic topology may represent a design well suited for the evolvability of proteins, that is, the capacity of proteins to evolve novel functionality. In general, there are two tendencies that underlie the evolvability of any system: 1) to minimize the detrimental effect of a change (mutation) to the system, and 2) to decrease the number of changes needed to create a new function [136]. How do proteins achieve this capacity? As descriptions of every level of biological systems – from proteins to whole organism development – become more complete, common design features that contribute to the adaptability of any system have begun to emerge. A recent review by Kirschner and Gerhart [136] discussed several such features and illustrated how these properties contribute to the adaptability of a system to change. One critical feature of an evolvable system is the presence of weak coupling among its components; this reduces the dependence of one process on another and allows local changes without propagated effects. The sparseness of the SCA matrix is consistent with this design feature: most of the protein is energetically independent and only a small subset of

positions shows significant coupling. Indeed, proteins are empirically known to be tolerant to mutagenesis at most sites and very sensitive to mutation at targeted sites. At the same time, the scale-free energetic topology provides a set of complex distributed energetic interactions evidently necessary for function. Included in these functionally important interactions are the hotspot residues that are sensitive to mutation. As discussed in chapter 3, mutations at such energetic hotspots may allow rapid change in functional properties. This suggests that the scale-free heterogeneous topology in proteins represents a solution to two opposing evolutionary pressures: 1) maintaining weak coupling to minimize detrimental effects of mutations and 2) construction of distributed coupled interactions that endow a protein with stability and functionality.

How did this scale-free architecture come to be? While we have not yet developed a generative model for the energetic topology, several properties of this model emerge from the discussion above and previous research on scale-free systems. First, the protein tends toward minimal coupling. Minimal coupling should be entropically favored and, as discussed, improves the evolvability of the protein. Coupled interactions, then, should only develop to the extent necessary for stability and function. Secondly, formation of a new coupled interaction tends to involve pre-existing coupled interactions since this should maximize the entropic state of the protein as a whole. This concept is consistent with modeling studies that show scale-free topologies are generated by a so-called “rich-get-richer” principle [6]. In other words, systems in which new nodes tend to connect to already existing nodes develop a scale-free topology. Future studies on modeling the forces necessary and sufficient to account for the heterogeneous energetic topology should yield insight into the evolutionary process that created proteins.

Conclusions of thesis work

The results I have discussed emphasize two features of proteins: tuning and heterogeneity. A principle that has guided this work is the idea that important energetic interactions among residues in proteins are conserved through evolution. Measuring and understanding these interactions is central to the sequence-structure-function problem. The new SCA methodology builds on the core formalism of the original version and produced consistent results but is now completely global and symmetric. In general, SCA results suggest that proteins are far simpler than might be anticipated by simply looking at structures. Only a small fraction of the residues in proteins have significant co-evolutionary interactions. These interactions have been corroborated by a large body of mutagenesis work in other systems.

Together, these observations motivated a detailed dissection of one set of interactions in the PDZ domain to understand their role in ligand binding. SCA results suggest that position 322 in the carboxylate binding loop influences the energetic interactions of position 372, a known specificity-determinant on the opposite end of the binding pocket. Thermodynamic measurements showed that, indeed, the strength of the interaction between 372 and its ligand contact, position P_{-2}, is modulated by mutations at 322. These experiments exposed a logic behind the co-evolution of these positions. Position 322 is tuned to optimize the strength of the specificity determining contact. Seen another way, the interaction between 372 and 322 is tuned to maximize the evolvability of the domain. Mutation at 372 can significantly shift the class-specificity of the domain. However, this evolutionary capacity at 372 is reduced in the presence of a mutation at 322. Structures of mutants revealed the physical basis of this tuning. The flexibility at position

322 is tuned to make conformational change in the carboxylate binding loop sensitive to the interaction at the specificity-determining contact. The structural inhomogeneity in the loop is a critical feature of the PDZ domain energetic architecture. This built-in mechanism sacrifices affinity but endows the domain with AND gate-like behavior to select for specific binding interactions. The combination of the logical and structural understanding lends further support to the usefulness of SCA in identifying important physical interactions in proteins.

A network graph analysis of the energetic topologies of several structurally and functionally diverse proteins revealed several recurring features. Consistent with the empirical demonstration of energetic heterogeneity, the SCA map shows a highly heterogeneous distribution of co-evolutionary interactions. Most positions are evolutionarily independent and only a small fraction of the residues have significantly co-evolved. Interestingly, depiction of co-evolutionary links in a graph representation revealed two interesting features regarding their organization. First, a small set of co-evolving positions in a protein show a high degree of mutual co-evolution such that they form a nearly connected subgraph. Mutagenesis data from several systems shows that mutations at positions in this subgraph have significant effects on function, regardless of their three-dimensional location in the structure. Secondly, there is heterogeneity in the co-evolutionary links among the residues: most positions have very few co-evolutionary links and a few positions form many co-evolutionary links with other positions. This distribution is well fit by a power law, the signature of a class of networks termed scale-free. Qualitatively, these networks display behaviors consistent with known features of proteins. Most notably, they are known to be robust to mutation at most positions but highly sensitive to mutation at a select few positions. The conceptual link to power law

networks may provide further insights in to the mechanism and generative model underlying the energetic architecture of proteins.

Methods and Materials

Sequence Alignments. Most alignments used were provided by members of the Ranganathan lab; UBC alignment was a provided by E. Ozkan. Briefly, all sequences (except the GPCR family) were collected from the non-redundant database using PSI-BLAST (e-score < 0.001) and aligned using Clustal W and manually adjusted using standard structure-based sequence alignment techniques. Class A GPCR sequences were collected as an alignment from the GPCRdb and TinyGRAP database.

Contact map calculation. In the network representation of atomic structures, nodes represent PSD95PDZ3 residues in no particular spatial orientation. Edges are drawn between nodes if the residues are contacting in the crystal structure (1BE9). We define contact between two residues if at least one pair of atoms from these residues is separated by less than the sum of their van de Waals radii plus 20%. PDB accession numbers for the structures used to generate figure 4-1C are: PDZ (1BE9), cdc25 (1C25), DHFR (1RX2), GFP (1EMB), PAS (1BYW), Ras (5Q21), Rhodopsin (1F88), RXR (1FM9), UBC (1C4Z). These calculations were performed with MATLAB code provided in the appendix A.

Statistical Coupling Analysis. All analyses followed the single sequence elimination method described in chapter 2. As before, the MATLAB was used to perform all calculations; the code is provided in appendix A.

Network graph construction and analysis. In the network graphs of the co-evolution matrices, the nodes represent residues and links are drawn between two nodes in the associated ddGstat value in the matrix is more than a cutoff value. The cutoff values were determined by fitting a histogram of the $\Delta\Delta G^{stat}$ values to a log-normal distribution. To avoid redundancy and trivial self coupling, only the values in the upper triangle of the $\Delta\Delta G^{stat}$ matrix were used in making the histogram. Cutoff values for determining links in the different protein families ranged between 2.5 and 3.6σ above the mean log-transformed $\Delta\Delta G^{stat}$ values in each matrix: give specific values for each family. Differences in the cutoff values for protein families are due to differences in the size and diversity of the respective multiple sequence alignments. These cutoffs were used to determine the number of significant links to each position which were in turn used to generate the log-log plot of k vs cumulative probability. Fitting these plots to equation 4-1 gave values for γ that ranged from 2.7 to 3.1. All fitting was done using MATLAB and the M-files given in the appendix. The networks graphs were generated using pajek (http://vlado.fmf.uni-lj.si/pub/networks/pajek/).

Mutagenesis and protein expression. Site-directed mutagenesis was carried out on PDZ3 of rat PSD-95 (residues 294-402) using standard polymerase chain reaction-based techniques. The domains were expressed as N-terminal glutathione S-transferase (GST) or His6 fusions using the pGEX-4T-1 vector (Amersham) in Escherichia coli [strain BL21(RP), Stratagene]. Cultures (1L) were grown in Terrific Broth to an optical density (600nm) of 1.6 at 37C, induced for 4 hours at 25^{o}C with 500 μM isopropyl-b-D-thiogalactopyranoside and then harvested by centrifugation. Cells were resuspended in buffer A (140 mM NaCl, 2.7 mM KCl, 10mM Na_2HPO_4, 1.8 mM KH_2PO_4 (pH 7.3), 1.0

mM dithiothreitol (DTT)) with protease inhibitors (10 μg/ml pepstatin, 10 μg/ml leupeptin, 0.1 mM phenyl methyl sulphonylfluoride), lysed by sonication, and the fusion protein batch purified from supernatant through GST or Ni-NTA affinity chromatography. The PDZ domains were cleaved off the resin through thrombin proteolysis (Sigma, 100U per 6 ml resin, 4 hr at room temperature) and purified to homogeneity using a Mono Q HR5/5 (Amersham) column run with a linear gradient from low salt (1.0 mM DTT, 20 mM Tris-HCL (pH 7.5)) to high salt (1.0M NaCl, 1.0mM DTT, 20mM Tris-HCl (pH 7.5)). The protein was dialyzed into 10 mM NaCl, 10 mM HEPES (pH 7.2), 1.0 mM DTT and concentrated as necessary.

Binding measurements. Isothermal titration calorimetry (ITC) measurement were conducted at 25C using the VP-ITC microcalorimeter (MicroCal Inc) by making 38 injections (8μl each) of peptide ligand into PDZ protein. The peptide (N-TKNYKQTSV-C) was dissolved in 10 mM NaCl, 10 mM HEPES (pH 7.2), 1.0 mM DTT. Concentrations of peptide (0.5 mM to 2.8 mM) and protein (0.05 mM to 0.15 mM) in each run were determined from absorption at 280 nm. The ratio of peptide to protein concentrations was adjusted between 10:1 and 30:1 (depending on the dissociation constant) in order to reach saturation in the binding reaction. In all titrations, the reference power was 12.9 μcal/s and equilibration time was 180s between peptide injections. Peaks were integrated and the titration curve was fit in Origin (MicroCal) assuming a 1:1 stoichiometry. The values given in Table 4-2 are averages and standard deviations from three measurements for each protein. The exception was V328I, for which only one measurement was made.

References

1. Barabasi, A.L., *Linked*. 2003, New York: Penguin.
2. 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.
3. Erdos, P., and Renyi, A., *On the evolution of random graphs.* Publ Math Inst Hung Acad Sci, 1960. **5**: p. 17-61.
4. Richards, F.M. and W.A. Lim, *An analysis of packing in the protein folding problem.* Q Rev Biophys, 1993. **26**(4): p. 423-98.
5. Tsai, J., et al., *The packing density in proteins: standard radii and volumes.* J Mol Biol, 1999. **290**(1): p. 253-66.
6. Barabasi, A.L. and R. Albert, *Emergence of scaling in random networks.* Science, 1999. **286**(5439): p. 509-12.
7. Atilgan, A.R., P. Akan, and C. Baysal, *Small-world communication of residues and significance for protein dynamics.* Biophys J, 2004. **86**(1 Pt 1): p. 85-91.
8. Greene, L.H. and V.A. Higman, *Uncovering network systems within protein structures.* J Mol Biol, 2003. **334**(4): p. 781-91.
9. Vendruscolo, M., et al., *Small-world view of the amino acids that play a key role in protein folding.* Phys Rev E Stat Nonlin Soft Matter Phys, 2002. **65**(6 Pt 1): p. 061910.
10. Kroon, G.J., et al., *Changes in structure and dynamics of the Fv fragment of a catalytic antibody upon binding of inhibitor.* Protein Sci, 2003. **12**(7): p. 1386-94.
11. Eisenmesser, E.Z., et al., *Enzyme dynamics during catalysis.* Science, 2002. **295**(5559): p. 1520-3.
12. Harper, S.M., L.C. Neil, and K.H. Gardner, *Structural basis of a phototropin light switch.* Science, 2003. **301**(5639): p. 1541-4.
13. Clackson, T. and J.A. Wells, *A hot spot of binding energy in a hormone-receptor interface.* Science, 1995. **267**(5196): p. 383-6.
14. Atwell, S., et al., *Structural plasticity in a remodeled protein-protein interface.* Science, 1997. **278**(5340): p. 1125-8.
15. Perona, J.J., et al., *Structural origins of substrate discrimination in trypsin and chymotrypsin.* Biochemistry, 1995. **34**(5): p. 1489-99.
16. Suel, G.M., et al., *Evolutionarily conserved networks of residues mediate allosteric communication in proteins.* Nat Struct Biol, 2003. **10**(1): p. 59-69.
17. Lockless, S.W. and R. Ranganathan, *Evolutionarily conserved pathways of energetic connectivity in protein families.* Science, 1999. **286**(5438): p. 295-9.
18. 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.
19. Shulman, A.I., et al., *Structural determinants of allosteric ligand activation in RXR heterodimers.* Cell, 2004. **116**(3): p. 417-29.
20. Amaral, L.A., et al., *Classes of small-world networks.* Proc Natl Acad Sci U S A, 2000. **97**(21): p. 11149-52.
21. Faloutsos, M., Faloutsos, P., Faloutsos, C.,, *On Power-law Relationships of the Internet Topology.* Proc ACM SIGCOMM, Comput Commun Rev, 1999. **29**: p. 251-262.

22. Jeong, H., et al., *The large-scale organization of metabolic networks.* Nature, 2000. **407**(6804): p. 651-4.
23. Park, J., M. Lappe, and S.A. Teichmann, *Mapping protein family interactions: intramolecular and intermolecular protein family interaction repertoires in the PDB and yeast.* J Mol Biol, 2001. **307**(3): p. 929-38.
24. Watts, D.J. and S.H. Strogatz, *Collective dynamics of 'small-world' networks.* Nature, 1998. **393**(6684): p. 440-2.
25. Albert, R., H. Jeong, and A.L. Barabasi, *Error and attack tolerance of complex networks.* Nature, 2000. **406**(6794): p. 378-82.
26. Gether, U., *Uncovering molecular mechanisms involved in activation of G protein-coupled receptors.* Endocr Rev, 2000. **21**(1): p. 90-113.
27. Sakmar, T.P., et al., *Rhodopsin: insights from recent structural studies.* Annu Rev Biophys Biomol Struct, 2002. **31**: p. 443-84.
28. Parnot, C., et al., *Lessons from constitutively active mutants of G protein-coupled receptors.* Trends Endocrinol Metab, 2002. **13**(8): p. 336-43.
29. Decaillot, F.M., et al., *Opioid receptor random mutagenesis reveals a mechanism for G protein-coupled receptor activation.* Nat Struct Biol, 2003. **10**(8): p. 629-36.
30. Skelton, N.J., et al., *Origins of PDZ domain ligand specificity. Structure determination and mutagenesis of the Erbin PDZ domain.* J Biol Chem, 2003. **278**(9): p. 7645-54.
31. Costa-Neto, C.M., et al., *Mutational analysis of the interaction of the N- and C-terminal ends of angiotensin II with the rat AT(1A) receptor.* Br J Pharmacol, 2000. **130**(6): p. 1263-8.
32. Phalipou, S., et al., *Mapping peptide-binding domains of the human V1a vasopressin receptor with a photoactivatable linear peptide antagonist.* J Biol Chem, 1997. **272**(42): p. 26536-44.
33. Labrou, N.E., et al., *Interaction of Met297 in the seventh transmembrane segment of the tachykinin NK2 receptor with neurokinin A.* J Biol Chem, 2001. **276**(41): p. 37944-9.
34. Pasel, K., et al., *Functional characterization of the molecular defects causing nephrogenic diabetes insipidus in eight families.* J Clin Endocrinol Metab, 2000. **85**(4): p. 1703-10.
35. Han, B. and A.H. Tashjian, Jr., *Importance of extracellular domains for ligand binding in the thyrotropin-releasing hormone receptor.* Mol Endocrinol, 1995. **9**(12): p. 1708-19.
36. Cain, S.A., et al., *Modulation of ligand selectivity by mutation of the first extracellular loop of the human C5a receptor.* Biochem Pharmacol, 2001. **61**(12): p. 1571-9.
37. Raffetseder, U., et al., *Site-directed mutagenesis of conserved charged residues in the helical region of the human C5a receptor. Arg2O6 determines high-affinity binding sites of C5a receptor.* Eur J Biochem, 1996. **235**(1-2): p. 82-90.
38. Blaker, M., et al., *CCK-B/Gastrin receptor transmembrane domain mutations selectively alter synthetic agonist efficacy without affecting the activity of endogenous peptides.* Mol Pharmacol, 2000. **58**(2): p. 399-406.
39. Labbe-Jullie, C., et al., *Mutagenesis and modeling of the neurotensin receptor NTR1. Identification of residues that are critical for binding SR 48692, a nonpeptide neurotensin antagonist.* J Biol Chem, 1998. **273**(26): p. 16351-7.

40. Pan, Y., P. Wilson, and J. Gitschier, *The effect of eight V2 vasopressin receptor mutations on stimulation of adenylyl cyclase and binding to vasopressin.* J Biol Chem, 1994. **269**(50): p. 31933-7.
41. Frandberg, P.A., et al., *Cysteine residues are involved in structure and function of melanocortin 1 receptor: Substitution of a cysteine residue in transmembrane segment two converts an agonist to antagonist.* Biochem Biophys Res Commun, 2001. **281**(4): p. 851-7.
42. Ryu, K., et al., *High affinity hormone binding to the extracellular N-terminal exodomain of the follicle-stimulating hormone receptor is critically modulated by exoloop 3.* J Biol Chem, 1998. **273**(44): p. 28953-8.
43. Gao, Z.G., et al., *Site-directed mutagenesis studies of human A(2A) adenosine receptors: involvement of glu(13) and his(278) in ligand binding and sodium modulation.* Biochem Pharmacol, 2000. **60**(5): p. 661-8.
44. Birnbaumer, M., *Vasopressin receptor mutations and nephrogenic diabetes insipidus.* Arch Med Res, 1999. **30**(6): p. 465-74.
45. Rao, V.R. and D.D. Oprian, *Activating mutations of rhodopsin and other G protein-coupled receptors.* Annu Rev Biophys Biomol Struct, 1996. **25**: p. 287-314.
46. Blanpain, C., et al., *The core domain of chemokines binds CCR5 extracellular domains while their amino terminus interacts with the transmembrane helix bundle.* J Biol Chem, 2003. **278**(7): p. 5179-87.
47. Chin, C.N., et al., *Ligand binding and modulation of cyclic AMP levels depend on the chemical nature of residue 192 of the human cannabinoid receptor 1.* J Neurochem, 1998. **70**(1): p. 366-73.
48. Jiang, Q., et al., *Hydrophilic side chains in the third and seventh transmembrane helical domains of human A2A adenosine receptors are required for ligand recognition.* Mol Pharmacol, 1996. **50**(3): p. 512-21.
49. Shin, N., et al., *Molecular modeling and site-specific mutagenesis of the histamine-binding site of the histamine H4 receptor.* Mol Pharmacol, 2002. **62**(1): p. 38-47.
50. Holst, B., et al., *Steric hindrance mutagenesis versus alanine scan in mapping of ligand binding sites in the tachykinin NK1 receptor.* Mol Pharmacol, 1998. **53**(1): p. 166-75.
51. Mansour, A., et al., *Site-directed mutagenesis of the human dopamine D2 receptor.* Eur J Pharmacol, 1992. **227**(2): p. 205-14.
52. Zhou, W., et al., *A locus of the gonadotropin-releasing hormone receptor that differentiates agonist and antagonist binding sites.* J Biol Chem, 1995. **270**(32): p. 18853-7.
53. Gimpl, G. and F. Fahrenholz, *The oxytocin receptor system: structure, function, and regulation.* Physiol Rev, 2001. **81**(2): p. 629-83.
54. Tokita, K., et al., *Tyrosine 220 in the 5th transmembrane domain of the neuromedin B receptor is critical for the high selectivity of the peptoid antagonist PD168368.* J Biol Chem, 2001. **276**(1): p. 495-504.
55. Kosugi, S., A. Shenker, and T. Mori, *Constitutive activation of cyclic AMP but not phosphatidylinositol signaling caused by four mutations in the 6th transmembrane helix of the human thyrotropin receptor.* FEBS Lett, 1994. **356**(2-3): p. 291-4.

56. Baranski, T.J., et al., *C5a receptor activation. Genetic identification of critical residues in four transmembrane helices.* J Biol Chem, 1999. **274**(22): p. 15757-65.
57. Chen, S., et al., *Dominant-negative activity of an alpha(1B)-adrenergic receptor signal-inactivating point mutation.* Embo J, 2000. **19**(16): p. 4265-71.
58. Nakayama, T.A. and H.G. Khorana, *Mapping of the amino acids in membrane-embedded helices that interact with the retinal chromophore in bovine rhodopsin.* J Biol Chem, 1991. **266**(7): p. 4269-75.
59. Chauvin, S., et al., *Functional importance of transmembrane helix 6 Trp(279) and exoloop 3 Val(299) of rat gonadotropin-releasing hormone receptor.* Mol Pharmacol, 2000. **57**(3): p. 625-33.
60. Gao, Z.G., et al., *Identification by site-directed mutagenesis of residues involved in ligand recognition and activation of the human A3 adenosine receptor.* J Biol Chem, 2002. **277**(21): p. 19056-63.
61. Hulme, E.C., et al., *The conformational switch in 7-transmembrane receptors: the muscarinic receptor paradigm.* Eur J Pharmacol, 1999. **375**(1-3): p. 247-60.
62. Yang, Y., et al., *Molecular basis for the interaction of [Nle4,D-Phe7]melanocyte stimulating hormone with the human melanocortin-1 receptor.* J Biol Chem, 1997. **272**(37): p. 23000-10.
63. Hovelmann, S., et al., *Impact of aromatic residues within transmembrane helix 6 of the human gonadotropin-releasing hormone receptor upon agonist and antagonist binding.* Biochemistry, 2002. **41**(4): p. 1129-36.
64. Miura, S., et al., *Role of aromaticity of agonist switches of angiotensin II in the activation of the AT1 receptor.* J Biol Chem, 1999. **274**(11): p. 7103-10.
65. Kim, J., et al., *Site-directed mutagenesis identifies residues involved in ligand recognition in the human A2a adenosine receptor.* J Biol Chem, 1995. **270**(23): p. 13987-97.
66. Cho, W., et al., *Hydrophobic residues of the D2 dopamine receptor are important for binding and signal transduction.* J Neurochem, 1995. **65**(5): p. 2105-15.
67. Lu, Z.L., J.W. Saldanha, and E.C. Hulme, *Transmembrane domains 4 and 7 of the M(1) muscarinic acetylcholine receptor are critical for ligand binding and the receptor activation switch.* J Biol Chem, 2001. **276**(36): p. 34098-104.
68. Vichi, P., A. Whelchel, and J. Posada, *Transmembrane helix 7 of the endothelin B receptor regulates downstream signaling.* J Biol Chem, 1999. **274**(15): p. 10331-8.
69. Roth, B.L., et al., *5-Hydroxytryptamine2-family receptors (5-hydroxytryptamine2A, 5-hydroxytryptamine2B, 5-hydroxytryptamine2C): where structure meets function.* Pharmacol Ther, 1998. **79**(3): p. 231-57.
70. Fernandez, L.M. and D. Puett, *Identification of amino acid residues in transmembrane helices VI and VII of the lutropin/choriogonadotropin receptor involved in signaling.* Biochemistry, 1996. **35**(13): p. 3986-93.
71. Marie, J., et al., *Tyr292 in the seventh transmembrane domain of the AT1A angiotensin II receptor is essential for its coupling to phospholipase C.* J Biol Chem, 1994. **269**(33): p. 20815-8.
72. Gromoll, J., et al., *A mutation in the first transmembrane domain of the lutropin receptor causes male precocious puberty.* J Clin Endocrinol Metab, 1998. **83**(2): p. 476-80.

73. Sung, C.H., C.M. Davenport, and J. Nathans, *Rhodopsin mutations responsible for autosomal dominant retinitis pigmentosa. Clustering of functional classes along the polypeptide chain.* J Biol Chem, 1993. **268**(35): p. 26645-9.
74. Biebermann, H., et al., *The first activating TSH receptor mutation in transmembrane domain 1 identified in a family with nonautoimmune hyperthyroidism.* J Clin Endocrinol Metab, 2001. **86**(9): p. 4429-33.
75. Breu, V., et al., *Separable binding sites for the natural agonist endothelin-1 and the non-peptide antagonist bosentan on human endothelin-A receptors.* Eur J Biochem, 1995. **231**(1): p. 266-70.
76. Shi, W., et al., *Rhodopsin mutants discriminate sites important for the activation of rhodopsin kinase and Gt.* J Biol Chem, 1995. **270**(5): p. 2112-9.
77. Nagashima, T., et al., *Novel inactivating missense mutations in the thyrotropin receptor gene in Japanese children with resistance to thyrotropin.* Thyroid, 2001. **11**(6): p. 551-9.
78. Stitham, J., et al., *The unique ligand-binding pocket for the human prostacyclin receptor. Site-directed mutagenesis and molecular modeling.* J Biol Chem, 2003. **278**(6): p. 4250-7.
79. Geva, A., et al., *Genetic mapping of the human C5a receptor. Identification of transmembrane amino acids critical for receptor function.* J Biol Chem, 2000. **275**(45): p. 35393-401.
80. Fraser, C.M., et al., *Site-directed mutagenesis of m1 muscarinic acetylcholine receptors: conserved aspartic acids play important roles in receptor function.* Mol Pharmacol, 1989. **36**(6): p. 840-7.
81. Le, M.T., et al., *Angiotensin IV is a potent agonist for constitutive active human AT1 receptors. Distinct roles of the N-and C-terminal residues of angiotensin II during AT1 receptor activation.* J Biol Chem, 2002. **277**(26): p. 23107-10.
82. Garriga, P., X. Liu, and H.G. Khorana, *Structure and function in rhodopsin: correct folding and misfolding in point mutants at and in proximity to the site of the retinitis pigmentosa mutation Leu-125-->Arg in the transmembrane helix C.* Proc Natl Acad Sci U S A, 1996. **93**(10): p. 4560-4.
83. Munshi, U.M., I.D. Pogozheva, and K.M. Menon, *Highly conserved serine in the third transmembrane helix of the luteinizing hormone/human chorionic gonadotropin receptor regulates receptor activation.* Biochemistry, 2003. **42**(13): p. 3708-15.
84. Neve, K.A., et al., *Modeling and mutational analysis of a putative sodium-binding pocket on the dopamine D2 receptor.* Mol Pharmacol, 2001. **60**(2): p. 373-81.
85. Lin, S.W. and T.P. Sakmar, *Specific tryptophan UV-absorbance changes are probes of the transition of rhodopsin to its active state.* Biochemistry, 1996. **35**(34): p. 11149-59.
86. Gether, U., et al., *Agonists induce conformational changes in transmembrane domains III and VI of the beta2 adrenoceptor.* Embo J, 1997. **16**(22): p. 6737-47.
87. Lu, Z.L. and E.C. Hulme, *The functional topography of transmembrane domain 3 of the M1 muscarinic acetylcholine receptor, revealed by scanning mutagenesis.* J Biol Chem, 1999. **274**(11): p. 7309-15.
88. Menon, S.T., M. Han, and T.P. Sakmar, *Rhodopsin: structural basis of molecular physiology.* Physiol Rev, 2001. **81**(4): p. 1659-88.

89. Scheer, A., et al., *Constitutively active mutants of the alpha 1B-adrenergic receptor: role of highly conserved polar amino acids in receptor activation.* Embo J, 1996. **15**(14): p. 3566-78.
90. Ballesteros, J., et al., *Functional microdomains in G-protein-coupled receptors. The conserved arginine-cage motif in the gonadotropin-releasing hormone receptor.* J Biol Chem, 1998. **273**(17): p. 10445-53.
91. Fanelli, F., et al., *Activation mechanism of human oxytocin receptor: a combined study of experimental and computer-simulated mutagenesis.* Mol Pharmacol, 1999. **56**(1): p. 214-25.
92. Kokkola, T., et al., *Mutagenesis of human Mel1a melatonin receptor expressed in yeast reveals domains important for receptor function.* Biochem Biophys Res Commun, 1998. **249**(2): p. 531-6.
93. Rhee, M.H., et al., *Role of the highly conserved Asp-Arg-Tyr motif in signal transduction of the CB2 cannabinoid receptor.* FEBS Lett, 2000. **466**(2-3): p. 300-4.
94. Prado, G.N., L. Taylor, and P. Polgar, *Effects of intracellular tyrosine residue mutation and carboxyl terminus truncation on signal transduction and internalization of the rat bradykinin B2 receptor.* J Biol Chem, 1997. **272**(23): p. 14638-42.
95. Timossi, C., et al., *Structural determinants in the second intracellular loop of the human follicle-stimulating hormone receptor are involved in G(s) protein activation.* Mol Cell Endocrinol, 2002. **189**(1-2): p. 157-68.
96. Miura, S., J. Zhang, and S.S. Karnik, *Angiotensin II type 1 receptor-function affected by mutations in cytoplasmic loop CD.* FEBS Lett, 2000. **470**(3): p. 331-5.
97. Damaj, B.B., et al., *Identification of G-protein binding sites of the human interleukin-8 receptors by functional mapping of the intracellular loops.* Faseb J, 1996. **10**(12): p. 1426-34.
98. Elias, L.L., et al., *Functional characterization of naturally occurring mutations of the human adrenocorticotropin receptor: poor correlation of phenotype and genotype.* J Clin Endocrinol Metab, 1999. **84**(8): p. 2766-70.
99. Schioth, H.B., et al., *Loss of function mutations of the human melanocortin 1 receptor are common and are associated with red hair.* Biochem Biophys Res Commun, 1999. **260**(2): p. 488-91.
100. Wess, J., et al., *Functional role of proline and tryptophan residues highly conserved among G protein-coupled receptors studied by mutational analysis of the m3 muscarinic receptor.* Embo J, 1993. **12**(1): p. 331-8.
101. Abramowicz, M.J., et al., *Familial congenital hypothyroidism due to inactivating mutation of the thyrotropin receptor causing profound hypoplasia of the thyroid gland.* J Clin Invest, 1997. **99**(12): p. 3018-24.
102. Lee, C., et al., *Two defective heterozygous luteinizing hormone receptors can rescue hormone action.* J Biol Chem, 2002. **277**(18): p. 15795-800.
103. Sykiotis, G.P., et al., *Functional significance of the thyrotropin receptor germline polymorphism D727E.* Biochem Biophys Res Commun, 2003. **301**(4): p. 1051-6.
104. Abe, Y., et al., *Functional analysis of five endothelin-B receptor mutations found in human Hirschsprung disease patients.* Biochem Biophys Res Commun, 2000. **275**(2): p. 524-31.

105. Parma, J., et al., *Somatic mutations in the thyrotropin receptor gene cause hyperfunctioning thyroid adenomas.* Nature, 1993. **365**(6447): p. 649-51.
106. Kosugi, S., T. Mori, and A. Shenker, *An anionic residue at position 564 is important for maintaining the inactive conformation of the human lutropin/choriogonadotropin receptor.* Mol Pharmacol, 1998. **53**(5): p. 894-901.
107. Hogger, P., et al., *Activating and inactivating mutations in N- and C-terminal i3 loop junctions of muscarinic acetylcholine Hm1 receptors.* J Biol Chem, 1995. **270**(13): p. 7405-10.
108. de Roux, N., et al., *The same molecular defects of the gonadotropin-releasing hormone receptor determine a variable degree of hypogonadism in affected kindred.* J Clin Endocrinol Metab, 1999. **84**(2): p. 567-72.
109. Tonacchera, M., et al., *Hyperfunctioning thyroid nodules in toxic multinodular goiter share activating thyrotropin receptor mutations with solitary toxic adenoma.* J Clin Endocrinol Metab, 1998. **83**(2): p. 492-8.
110. Beau, I., et al., *A novel phenotype related to partial loss of function mutations of the follicle stimulating hormone receptor.* J Clin Invest, 1998. **102**(7): p. 1352-9.
111. Themmen, A.P.N. and I.T. Huhtaniemi, *Mutations of gonadotropins and gonadotropin receptors: elucidating the physiology and pathophysiology of pituitary-gonadal function.* Endocr Rev, 2000. **21**(5): p. 551-83.
112. Laue, L., et al., *Heterogeneity of activating mutations of the human luteinizing hormone receptor in male-limited precocious puberty.* Biochem Mol Med, 1996. **58**(2): p. 192-8.
113. Townsend-Nicholson, A. and P.R. Schofield, *A threonine residue in the seventh transmembrane domain of the human A1 adenosine receptor mediates specific agonist binding.* J Biol Chem, 1994. **269**(4): p. 2373-6.
114. Angelova, K., et al., *Functional role of transmembrane helix 7 in the activation of the heptahelical lutropin receptor.* Mol Endocrinol, 2000. **14**(4): p. 459-71.
115. Parent, J.L., et al., *Identification of transmembrane domain residues determinant in the structure-function relationship of the human platelet-activating factor receptor by site-directed mutagenesis.* J Biol Chem, 1996. **271**(38): p. 23298-303.
116. Tsigos, C., C. Latronico, and G.P. Chrousos, *Luteinizing hormone resistance syndromes.* Ann N Y Acad Sci, 1997. **816**: p. 263-73.
117. Miura, S., et al., *TM2-TM7 interaction in coupling movement of transmembrane helices to activation of the angiotensin II type-1 receptor.* J Biol Chem, 2003. **278**(6): p. 3720-5.
118. Govaerts, C., et al., *A conserved Asn in transmembrane helix 7 is an on/off switch in the activation of the thyrotropin receptor.* J Biol Chem, 2001. **276**(25): p. 22991-9.
119. Gales, C., et al., *Mutation of Asn-391 within the conserved NPXXY motif of the cholecystokinin B receptor abolishes Gq protein activation without affecting its association with the receptor.* J Biol Chem, 2000. **275**(23): p. 17321-7.
120. Perlman, J.H., et al., *Interactions between conserved residues in transmembrane helices 1, 2, and 7 of the thyrotropin-releasing hormone receptor.* J Biol Chem, 1997. **272**(18): p. 11937-42.
121. Arora, K.K., et al., *Mediation of cyclic AMP signaling by the first intracellular loop of the gonadotropin-releasing hormone receptor.* J Biol Chem, 1998. **273**(40): p. 25581-6.

122. Burstein, E.S., T.A. Spalding, and M.R. Brann, *The second intracellular loop of the m5 muscarinic receptor is the switch which enables G-protein coupling.* J Biol Chem, 1998. **273**(38): p. 24322-7.
123. Frandberg, P.A., et al., *Human pigmentation phenotype: a point mutation generates nonfunctional MSH receptor.* Biochem Biophys Res Commun, 1998. **245**(2): p. 490-2.
124. Hunyady, L., et al., *Dependence of agonist activation on a conserved apolar residue in the third intracellular loop of the AT1 angiotensin receptor.* Proc Natl Acad Sci U S A, 1996. **93**(19): p. 10040-5.
125. Wu, S.M. and W.Y. Chan, *Male pseudohermaphroditism due to inactivating luteinizing hormone receptor mutations.* Arch Med Res, 1999. **30**(6): p. 495-500.
126. Ballesteros, J.A., L. Shi, and J.A. Javitch, *Structural mimicry in G protein-coupled receptors: implications of the high-resolution structure of rhodopsin for structure-function analysis of rhodopsin-like receptors.* Mol Pharmacol, 2001. **60**(1): p. 1-19.
127. Palczewski, K., et al., *Crystal structure of rhodopsin: A G protein-coupled receptor.* Science, 2000. **289**(5480): p. 739-45.
128. Arora, K.K., A. Sakai, and K.J. Catt, *Effects of second intracellular loop mutations on signal transduction and internalization of the gonadotropin-releasing hormone receptor.* J Biol Chem, 1995. **270**(39): p. 22820-6.
129. Schulman, I.G., et al., *The phantom ligand effect: allosteric control of transcription by the retinoid X receptor.* Genes Dev, 1997. **11**(3): p. 299-308.
130. Sprang, S.R., *G protein mechanisms: insights from structural analysis.* Annu Rev Biochem, 1997. **66**: p. 639-78.
131. Espanel, X. and M. Sudol, *A single point mutation in a group I WW domain shifts its specificity to that of group II WW domains.* J Biol Chem, 1999. **274**(24): p. 17284-9.
132. Kasanov, J., et al., *Characterizing Class I WW domains defines key specificity determinants and generates mutant domains with novel specificities.* Chem Biol, 2001. **8**(3): p. 231-41.
133. Goh, K.I., et al., *Classification of scale-free networks.* Proc Natl Acad Sci U S A, 2002. **99**(20): p. 12583-8.
134. Krem, M.M. and E. Di Cera, *Molecular markers of serine protease evolution.* Embo J, 2001. **20**(12): p. 3036-45.
135. Shoichet, B.K., et al., *A relationship between protein stability and protein function.* Proc Natl Acad Sci U S A, 1995. **92**(2): p. 452-6.
136. Kirschner, M. and J. Gerhart, *Evolvability.* Proc Natl Acad Sci U S A, 1998. **95**(15): p. 8420-7.

Appendix A: MATLAB code for Statistical Coupling Analysis

Single Sequence Elimination Code:

```
function [out_mat,out_vect, lnp]=stat_fluc3(A);
%usage: [out_matrix, out_vectors, lnp]=stat_fluc3(alignment)

% initalize variables
aminoacids=('ACDEFGHIKLMNPQRSTVWY');
x=100;  % used to normalize to 100 sequences
rr_factx = rr_fact(x);
random=[0.072658 0.024692 0.050007 0.061087 0.041774 0.071589 0.023392...
        0.052691 0.063923 0.089093 0.023150 0.042931 0.052228 0.039871...
        0.052012 0.073087 0.055606 0.063321 0.012720 0.032955];
[numseqs,numpos]=size(A);
numseqs_sub = numseqs-1;

site_parent=zeros(20, numpos);
lnp = zeros(20,numpos);

% determine amino acid frequency (actually, the number of seqs with each aa normalized to an alignment of 100 total seqs)
% in parent alignment and calculate ln(prob)
for aa = 1:20
    site_parent(aa,1:numpos) = sum(A == aminoacids(aa)).*x/numseqs;
    lnp(aa,:) = rr_factx - gammaln(site_parent(aa,:)+1) - gammaln(100-site_parent(aa,:)+1)...
        + site_parent(aa,:)*log(random(aa)) + (x-site_parent(aa,:))*log(1-random(aa));
end;

% initalize variables
out_mat = zeros(20, numpos, numseqs);
out_vect = zeros(numpos, numseqs);

for n = 1:numseqs

    % initialize variables
    DDG=zeros(1,numpos);
    DDGmat=zeros(20,numpos);
    site_sub=zeros(20,numpos);
    lnp_sub = zeros(20,numpos);

    % determine amino acid frequency in subalignment (actually, number of
    % seqs with each amino acid normalized to 100 total seqs)
    site_sub = site_parent*numseqs/x;
    for j = 1:numpos
        site_sub(:,j) = site_sub(:,j) - (aminoacids==A(n,j))';
    end
    site_sub = site_sub.*x/(numseqs_sub);

    % calculate ln(prob) for sub alignment
    for aa = 1:20
        lnp_sub(aa,:) = rr_factx - gammaln(site_sub(aa,:)+1) - gammaln(100-site_sub(aa,:)+1)...
            + site_sub(aa,:)*log(random(aa)) + (100-site_sub(aa,:))*log(1-random(aa));
    end;

    diff_lnp=lnp-lnp_sub;
    DDGmat=diff_lnp;
    DDG = sqrt(sum(diff_lnp.^2));

    out_vect(:,n) = DDG';
    out_mat(:,:,n)= DDGmat;
end
% out_vect=mean(out_vect');


%the ln(factorial) function, two ways: by gamma function if <170, and
%by stirlings approximation if >170.
function [m]=rr_fact(n)
%This checks for size and calculates the ln(factorial)
if n<=170
   m=gammaln(n+1);
else
   m=n*log(n)-n;
end
```

```
function[coupling_matrix_aa,coupling_matrix_res]=global_sca2(randpert_mat);
%usage: [coupling_matrix_aminoacid, coupling_matrix_residues]=global_sca(randpert_mat)
%
% Modified from Rama's code

[numaa, numpos, numtrials]=size(randpert_mat);

% amino_acids=('ACDEFGHIKLMNPQRSTVWY');
coupling_matrix_aa = zeros(numpos, numpos, 20, 20);
coupling_matrix_res = zeros(numpos,numpos);
for m = 1:numpos
    site1 = squeeze(randpert_mat(:,m,:));
    for n = m:numpos
```

```
        site2 = squeeze(randpert_mat(:,n,:));
        coupling_matrix_aa(m,n,:,:) = site1*site2';
        coupling_matrix_aa(n,m,:,:) = squeeze(coupling_matrix_aa(m,n,:,:))';
    end
end

coupling_matrix_aa=coupling_matrix_aa./numtrials;

for m=1:numpos
    for n=1:numpos
%         coupling_matrix_res(m,n)=norm(reshape(squeeze(coupling_matrix_aa(m,n,:,:)),1,400));
        coupling_matrix_res(m,n) = sqrt(sum(sum(coupling_matrix_aa(m,n,:,:).^2)));
    end
end
```

```
function [pdb] = read_pdb2(filename);
% This function reads in the fields of a pdb into a structure array.

numofatoms = 0;
pdb = [];

[fid1,message] = fopen(filename, 'r');
atoms = 0;
if fid1 == -1
    disp(message)
end


while 1
    line = fgetl(fid1);
    if ~ischar(line), break, end
    sz = size(line);
    line = [line blanks(80 - sz(2))];

    if (line(1:4)=='ATOM') & (~strcmp(char(line(18:20)),'WAT')) & (~strcmp(char(line(18:20)),'HOH'))
        numofatoms = numofatoms + 1;
        pdb.atomnum(numofatoms,1) = str2int(line(7:11));
        pdb.atomid(numofatoms,1) = cellstr(removeblanks(line(13:16)));
        pdb.ac(numofatoms,1) = cellstr(line(17));
        pdb.res(numofatoms,1) = cellstr(line(18:20));
        pdb.chainid(numofatoms,1) = cellstr(line(22));
        pdb.resnum(numofatoms,1) = str2int(line(23:26));
        % need to have a resnum2 for pdbs that have redundant resnum.  For
        % example, in ser prot there is 184A and 184.
        pdb.resnum2(numofatoms,1) = cellstr(removeblanks(line(23:27)));
        pdb.x(numofatoms,1) = str2double(line(31:38));
        pdb.y(numofatoms,1) = str2double(line(39:46));
        pdb.z(numofatoms,1) = str2double(line(47:54));
        pdb.occ(numofatoms,1) = str2double(line(55:60));
        pdb.bfactor(numofatoms,1) = str2double(line(61:66));
        pdb.segid(numofatoms,1) = cellstr(line(73:76));
        pdb.element(numofatoms,1) = cellstr(line(77:78));
%         pdb.charge(numofatoms) = str2num(char(line(79:80)));
    end
end

fclose(fid1);

[pdb.strseq3, pdb.strseq1, pdb.resnumlist] = make_pdb_seq(pdb);

function val = str2int(str)
val = sscanf(str,'%d');

function out = removeblanks(in)
[r,c] = find(~isspace(in));
if isempty(c),
    out = in([]);
else
    out = in(:,c(1):c(end));
end
```

```
function [alignment_trunc, align_to_strseq, best_align, strseqnum, startat, topscore] = find_seq_in_alignment(strseq1,
alignment, resnumlist);
% This function makes pairwise alignments of a certain sequence and each
% sequence in an alignment.  The alignment is truncated according to the tophit.
% The alignment and its score between the tophit and  the input sequence (strseq1)
% are determined and passed back.  Lastly, the function makes a lookup
% table between alignment position number and strseq number.

% score each alignment sequence (removed of gaps) to the pdb sequence.
scores = zeros(1,size(alignment,1));    %initialize scores
for rownum = 1:size(alignment,1)
    [scores(rownum),junk] = swalign(alignment(rownum,find(isletter(alignment(rownum,:)))), strseq1);
end

% the highest scoring sequence is assumed to be the same as the pdb
% sequence and the alignment is truncated according to its gaps
strseqnum = find(scores == max(scores));
alignment_trunc = alignment(:,find(isletter(alignment(strseqnum,:))));
disp(['     truncated alignment using sequence #' num2str(strseqnum) '  (score: ' num2str(max(scores)) ')']);
[topscore, best_align, startat] = swalign(alignment_trunc(strseqnum,:), strseq1);
```

```
% note that the top sequence in pdb.best_align is the whole topscore
% seqeuence from the truncated alignment and the bottom is the
% complete structure sequence.

% use swalign and a loop to make a lookup table to convert alignment # to
% str seq number (pdb.align_to_strseq)
alignment_pos = startat(1); strseq_pos = startat(2);
for best_align_pos = 1:size(best_align,2)
    if best_align(2,best_align_pos)=='|' | best_align(2,best_align_pos)==':'
        align_to_strseq(alignment_pos) = resnumlist(strseq_pos);
    end

    % only advance the alignment/strseq_pos position if the next element in
    % the top/bottom row of best_align is a letter; but, can only do this
    % if not at the end of the line
    if best_align_pos ~= size(best_align,2)
        if isletter(best_align(1, best_align_pos+1))
            alignment_pos = alignment_pos + 1;
        end
        if isletter(best_align(3,best_align_pos+1))
            strseq_pos = strseq_pos + 1;
        end
    end
end
```

```
function x = fitcoupling(coupmatrix)
% takes in an n x n coupling matrix and plots the normal, lognormal, and
% imshow figures.  Also fits normal and lognormal histograms and returns
% fit to normal distribution (results from log-transformation of x axis).

numbins = 100;
ndiagonal = 1;  % set to 0 to include self-coupling and 1 to exclude.


matrix_uppertri_reshape = coupmatrix(find(triu(coupmatrix,ndiagonal)));
[yhist,xhist] = hist(matrix_uppertri_reshape, numbins);

scrsz = get(0,'ScreenSize');
figure('Position',[scrsz(3)*3.5/10 (scrsz(4)*2.25)/4 (scrsz(3)*6)/10 (scrsz(4)*1.3)/4]);

%_____LOG NORMAL GRAPH_____
subplot(1,2,1);
bar(xhist,yhist);
axis([0 max(matrix_uppertri_reshape) 0 max(yhist)]);
title('log normal distribution');
ylabel('number of pixels');
xlabel('\Delta\DeltaG (kT^*)');
hold on;

options=optimset('display','final','MaxIter',[500],'MaxFunEvals',[5000],'TolFun',1e-2);     % set options
xmax = xhist(find(yhist == max(yhist))); ymax = max(yhist);                                 % set initial values
x0 = [ymax xmax 0.5];
% xlb = [ymax-100 0.5*xmax]; xub = [ymax+100 1.5*xmax];
xlb = []; xub = [];                                                                         % set lower and upper bounds
x = lsqcurvefit(@lognormaldist, x0, xhist, yhist, xlb, xub, options);                       % do fitting and plot
plot(xhist, lognormaldist(x,xhist), '-r','LineWidth',1.5);
line(1) = {['f = ae^{-0.5(ln(x/x_o))/\sigma)^2}']};
line(2) = {['   a = ' num2str(x(1))]};
line(3) = {['   x_o = ' num2str(x(2))]};
line(4) = {['   \sigma = ' num2str(x(3))]};
text(max(xhist)*2/5, max(yhist)*3/4, line, 'Fontsize',8, 'Color','r');
hold off;

%____NORMAL GRAPH____
subplot(1,2,2);
logxhist = log(xhist);
bar(logxhist, yhist);
title ('normal distribution');
xlabel('log(\Delta\DeltaG)');
hold on;

options=optimset('display','final','MaxIter',[500],'MaxFunEvals',[5000],'TolFun',1e-2);     % set optioons
xmax = log(xhist(find(yhist == max(yhist)))); ymax = max(yhist);                            % set initial values
x0 = [ymax xmax 1];
xlb = []; xub = [];                                                                         % set lower and upper bounds
x = lsqcurvefit(@logdist, x0, logxhist, yhist, xlb, xub, options);                          % do fitting, then plot
plot([min(logxhist):0.1:max(logxhist)], logdist(x,[min(logxhist):0.1:max(logxhist)]), '-r', 'LineWidth', 1.5);
line(1) = {['f = ae^{-0.5((x-x_o)/\sigma)^2}']};
line(2) = {['   a = ' num2str(x(1))]};
line(3) = {['   x_o = ' num2str(x(2))]};
line(4) = {['   \sigma = ' num2str(x(3))]};
text((min(logxhist)+max(logxhist))*2/4, max(yhist)*3/4, line(1:4),'Fontsize',8,'Color','r');
sigma3 = x(2) + 3*x(3);
line(5) = {['x_o+3\sigma = ' num2str(sigma3)]};
line(6) = {['e^{x_o+3\sigma} = ' num2str(exp(sigma3)) ' kT^*']};
text(sigma3-0.2, max(yhist)*1/6, line(5:6),'Fontsize',8,'Color',[0.5 0.5 0.5]);
text(sigma3, logdist(x,sigma3)+max(yhist)/20, '\downarrow','Color',[0.5 0.5 0.5]);
hold off;


%____IMSHOW FIGURE____
figure ('Position', [scrsz(3)/20 (scrsz(4)*1.25)/4 (scrsz(3)*2.5)/10 (scrsz(4)*4)/10]);
imshow(coupmatrix, [min(matrix_uppertri_reshape) exp(x(2)+4.5*x(3))],'notruesize');
% imshow(coupmatrix, [min(matrix_uppertri_reshape) max(matrix_uppertri_reshape)],'notruesize');
color_map = jet(256); color_map = color_map(1:255,:);
colormap(color_map); colorbar;
```

```
function F = logdist(x,xdata)
F = x(1)*exp(-0.5*((xdata-x(2))/x(3)).^2);

function F = lognormaldist(x,xdata)
F = x(1)*exp(-0.5*(log(xdata/x(2))/x(3)).^2);
```

```
function [links, cumpk] = analyze_graph (matrix, threshold);

removeself = 1;                      % set to 1 to remove self coupling

[rows,cols] = size(matrix);

% calculate number of links
matrix_bin = matrix >= threshold;
links = sum(matrix_bin);        % note: this counts self coupling!

% remove self coupling if removeself set to 1 above.
if removeself
    for n = 1:rows
        if matrix(n,n) >= threshold
            links(n) = links(n)-1;
        end
    end
end


% calculate cumulative distribution.  Values are stored such that cumpk(n)
% contains number of nodes with n-1 more links.
for n = 1:max(links)+1
    cumpk(n) = size(find(links >= (n-1)),2);
end;

% switch to other way of counting: not shifted
% for n = 1:max(links)
%     cumpk(n) = size(find(links >= n),2);
% end;

cumpk = cumpk/rows;

% graph cumpk
scrsz = get(0,'ScreenSize');
figure('Position',[scrsz(3)*3.5/10 (scrsz(4))/10 (scrsz(3)*3)/10 (scrsz(4)*1.3)/4]);
k = 1:max(links)+1;      % to shift by 1
% k = 1:max(links);      % not shifted
plot(log10(k), log10(cumpk), 'o','MarkerFaceColor','b','MarkerEdgeColor','k');
axis([0 log10(max(links)+5) min(log10(cumpk/2)) 0]);
title(['connections P(K>k) (threshold = ' num2str(threshold) ')'],'Color',[0 0 1.0]);
xlabel('log_{10}(k)'); ylabel('log_{10}(P(K>k))');
hold on;

% FIT POWER LAW and plot
x0 = [0 -2];
x = lsqcurvefit(@powerlawfit, x0, log10(k), log10(cumpk));
plot(log10(1:0.1:max(k)+5), powerlawfit(x, log10(1:0.1:max(k)+5)),'-', 'Color',[0.5 0.5 0.5]);
line(1) = {['f = mx + b']};
line(2) = {['     m = ' num2str(x(2))]};
line(3) = {['     b = ' num2str(x(1))]};
line(4) = {['P(K>k) = ak^{-\gamma+1}']};
line(5) = {['     \gamma = ' num2str(-1*(x(2)-1))]};
text (log10(max(k)+5)*2/3, log10(min(cumpk(cumpk~=0)))/3, line, 'Fontsize',8,'Color',[0.5 0.5 0.5]);
hold off;

function F = powerlawfit(x, xdata);
F = x(1) + x(2).*xdata;
```

```
function solution = fit_cumpoisson (xdata, ydata)
l = 10;

plot (xdata, ydata, 'o','Markerfacecolor','b');
hold on;

solution = lsqcurvefit(@cumpoisson, l, xdata, ydata);

ysolution = cumpoisson(solution, xdata);
plot(xdata, ysolution, 'o', 'Markerfacecolor','r');


function F = cumpoisson (lambda, x)
F = 1-poisscdf(x,lambda);
```

```
function [pdb] = contacts (pdb_file, cutoff, write_flag, file_out);

% Reads in a pdb file and calculates the contact matrices for residues
% (#res x #res) and for atoms (not working yet)(#atoms x #atoms).  The input must include:
% the filename (.pdb), the cutoff, a write_flag, and a name for the output
% file.  No waters will be read in from the pdb file.  If cutoff is zero
% then the contact is based on the default calculation (20% more than the
% sum of the van der Waals radii).

% Example:
% >> ww2 = contact_test('ww2_12.pdb',0,1,'pdzpajek.net')
```

```
[atom_type] = ['C', 'O', 'N', 'S'];
[atom_radius] = [1.9, 1.4, 1.5, 1.85];
cushion = 0.2;

% First, read the pdb into a cell structure array
pdb = read_pdb2(pdb_file);
disp('read pdb file')

% Make list of residues (in case there is a gap at some point) and
% initialize pdb.contacts
% num_atoms = size(pdb.atomnum,1);
% pdb.resnumlist(1) = pdb.resnum(1);
% pdb.sequence(1) = pdb.res(1);
% count = 1;
% for n = 2:num_atoms
%     if pdb.resnum(n) ~= pdb.resnumlist(count)
%         count = count + 1;
%         pdb.resnumlist(count) = pdb.resnum(n);
%         pdb.sequence(count) = pdb.res(n);
%     end
% end
% pdb.resnumlist = pdb.resnumlist';
num_res = size(pdb.resnumlist,1);
pdb.contacts = zeros(num_res, num_res);


% Calculate contact matrix
disp('Calculating contact matrix')
for i = 1:num_res
    for j = (i+1):num_res
        % if the residues are adjacent then they are contacting
        if j == (i+1)
            pdb.contacts(i,j) = 1;
            pdb.contacts(j,i) = 1;
        else
            % otherwise, find the indices of atoms in both aa find
            % distances between all atom pairs from the two aa.
            ind_i = find(pdb.resnum==str2num(char(pdb.resnumlist(i))));
            ind_j = find(pdb.resnum==str2num(char(pdb.resnumlist(j))));
            for i2 = 1:size(ind_i,1)
                for j2 = 1:size(ind_j,1)
                    %only do the calculation for contact if a contact has
                    %not already been found
                    if pdb.contacts(i,j) == 0
                        dist = ((pdb.x(ind_i(i2))-pdb.x(ind_j(j2)))^2 + (pdb.y(ind_i(i2))-pdb.y(ind_j(j2)))^2 +
(pdb.z(ind_i(i2))-pdb.z(ind_j(j2)))^2)^0.5;
                        % if the cutoff is specified then see if the two atoms
                        % are in contact, if not use the default cutoff
                        % (sum of vdw radii plus some cushion - usually 20%).
                        if cutoff > 0
                            if dist <= cutoff
                                pdb.contacts(i,j) = 1;
                                pdb.contacts(j,i) = 1;
                            end
                        else
                            atom1 = char(pdb.atomid(ind_i(i2))); atom1 = atom1(1);
                            atom2 = char(pdb.atomid(ind_j(j2))); atom2 = atom2(1);
                            contact_cutoff = (atom_radius(findstr(atom1,atom_type)) +
atom_radius(findstr(atom2,atom_type)))*(1+cushion);
                            if dist <= contact_cutoff
                                pdb.contacts(i,j) = 1;
                                pdb.contacts(j,i) = 1;
                            end
                        end
                    end
                end
            end
        end
    end
end


if write_flag
    disp('writing pajek file')
    if exist (file_out,'file')
        delete (file_out);
        disp ([file_out ' overwritten.']);
    end
    fid = fopen(file_out,'w');

    fprintf(fid, '*Vertices %1.0f\n', num_res);
    for p = 1:num_res
        ind_Ca = find(pdb.resnum==pdb.resnumlist(p) & strcmp(pdb.atomid,'CA'));
        fprintf(fid, '%1.0f "%1.0f"\n', p, pdb.resnumlist(p));
%         fprintf(fid, '%1.0f "%1.0f" %3.1f %3.1f %3.1f\n', p, pdb.resnumlist(p),pdb.x(ind_Ca), pdb.y(ind_Ca),
pdb.z(ind_Ca));
    end

    fprintf (fid, '*Edges\n');
    for p1 = 1:num_res
        for p2 = (p1+1):num_res
            if pdb.contacts(p1,p2)==1
                fprintf(fid, '%1.0f %1.0f 1\n',p1,p2);
            end
        end
    end
    fclose(fid);
```

```
end

pdb.num_contacts = sum(pdb.contacts);
for m = min(pdb.num_contacts):(max(pdb.num_contacts)+5)
    pdb.pk(m) = size(find(pdb.num_contacts==m),2);
end
pdb.pk = pdb.pk./num_res;

for m2 = 1:(max(pdb.num_contacts)+5)
    pdb.cum_pk(m2) = sum(pdb.pk(m2:end));
end

xdata = 1:(max(pdb.num_contacts)+5);
figure;
[AX, H1, H2] = plotyy(xdata, pdb.pk, xdata, pdb.cum_pk);
set(get(AX(1),'Ylabel'),'String','Probability');
set(get(AX(2),'Ylabel'),'String','Cumulative Probability');

set(H1,'Marker','o','MarkerFaceColor','b');
set(H2,'Marker','o','MarkerFaceColor','r');
```

```
function [distmatrix, total_coupedges, coupledandconnected] = couplingcontacts(coupmatrix, contactmatrix);
num = 50;
maxcoupvalue = max(reshape(triu(coupmatrix,1), 1, size(coupmatrix,1)^2))

for m = 1:num
    % first binarize the coupling matrix around maxcoupvalue*m/20
    bin_coupmatrix = zeros(size(coupmatrix));
    bin_coupmatrix = double(coupmatrix > maxcoupvalue*m/num);

    % only consider the contacts that also have high coupling values
    coupcontactmatrix = contactmatrix.*bin_coupmatrix;

    % determine how many steps it takes to get from any node to any other
    % node and binarize this matrix
    distmatrix = dist_from_contacts(coupcontactmatrix);
    bin_distmatrix = distmatrix > 0;

    % count fraction of binarized coupling matrix edges (not including self
    % coupling.
    total_coupedges(m) = sum(sum(triu(bin_coupmatrix,1)));

    % count number of binarized coupling matrix edges that have edges in
    % bin_distmatrix
    coupledandconnected(m) = 0;
    for i = 1:size(bin_distmatrix,1)
        for j = i+1:size(bin_distmatrix,1)
            if bin_coupmatrix(i,j)==1 & bin_distmatrix(i,j)==1
                coupledandconnected(m) = coupledandconnected(m) + 1;
            end
        end
    end
end
```

```
function scrambled = scrambler(matrix);
% scrambles an input matrix while preserving symmetry and the diagonal
% elements

[rows,cols] = size(matrix);

scrambled = matrix;

for n = 1:(100*rows^2)
    % randomly pick two elements in matrix, making sure to not pick a
    % diagonal element.
    x1 = round(1 + (rows-1)*rand);
    y1 = x1;
    while y1==x1
        y1 = round(1 + (rows-1)*rand);
    end
    x2 = round(1 + (rows-1)*rand);
    y2 = x2;
    while y2==x2
        y2 = round(1 + (rows-1)*rand);
    end

    % swap two elements
    temp = scrambled(x1,y1);
    scrambled(x1,y1) = scrambled(x2,y2);
    scrambled(x2,y2) = temp;

    % retain symmetry
    scrambled(y1,x1) = scrambled(x1,y1);
    scrambled(y2,x2) = scrambled(x2,y2);
end
```

```
function [ident, trimalign] = trimalignment(A, cutoff);
% makes an alignment that consists only of sequences with maximum sequence
% identity.  The sequence identity threshold is passed as cutoff.

[numseqs, numpos] = size(A);

ident = zeros(numseqs,numseqs);
```

```
for r = 1:numseqs
    for s = (r):numseqs
        % determine percent identity
        commonres = isletter(A(r,:)) & isletter(A(s,:));
        ident(r,s) = sum(A(r,commonres) == A(s,commonres))/sum(commonres);
        ident(s,r) = ident(r,s);
    end
end

count = 1;
for n = 1:numseqs;
    seqsabovecutoff = find(ident(n,:) > cutoff);
    if size(seqsabovecutoff,2)==1 | seqsabovecutoff(2) > n
        trimalign(count,:) = A(n,:);
        count = count + 1;
    end;
end;
```

```
function [weights, freq, uniqueaa] = weightseqs (alignment);
% Takes in an alignment and determines a weight for each sequence.  Weight
% is calculated as described in Henikoff and Henikoff (JMB, 1994).

aa = ('ACDEFGHIKLMNPQRSTVWY');
[numseqs, numpos] = size(alignment);

% determine aa frequency at all positions
freq = zeros(20,numpos);
for n = 1:20
    freq(n,:) = sum(alignment == aa(n));
end;

% determine number of unique aa at all sites
uniqueaa = zeros(1,numpos);
% for n = 1:numpos
%     uniqueaa(n) = size(unique(alignment(isletter(alignment(:,n)),n)),1);
% end;
uniqueaa = sum(freq>0);

weights = zeros(1,numseqs);
for seq = 1:numseqs
    for pos = 1:numpos
        if findstr(alignment(seq,pos), aa)
            weights(seq) = weights(seq) + 1/(uniqueaa(pos) * freq(aa==alignment(seq,pos), pos));
        end;
    end;
end;

weights = weights/(sum(weights));
```

```
function writepajekfile(matrix, threshold, align_to_strseq, filename);
% This function will binarize a raw matrix about a specified threshold and
% write out the corresponding network to a pajek file with specified name.

filename = [filename '.net'];
fid = fopen(filename, 'w');

binmat = matrix >= threshold;

% write list of vertices
fprintf (fid, '*Vertices %1.0f\r',size(binmat,1));
for n = 1:size(binmat,1)
    if iscellstr(align_to_strseq(n))
        fprintf (fid, '%1.0f "%s" ic Red\t bc Black\r', n, char(align_to_strseq(n)));
    else
        fprintf (fid, '%1.0f "%s" ic Red\t bc Black\r', n, [num2str(n) 'na']);
    end
end

% write list of arcs
matrixmax = max(reshape(matrix, 1, size(matrix,1)*size(matrix,2)));
fprintf (fid, '*Edges\r');
for row = 1:size(binmat,1)
    for col = row:size(binmat,1)
        if binmat(row, col) == 1
            weight = (matrix(row,col)/matrixmax) * 20;
            fprintf(fid, '%1.0f %1.0f %3.1f\r',row, col, weight);
        end
    end
end
fclose(fid);
```

```
function writepyfile(raw, threshold, pertnum, align_to_strseq, pdb, filename_prefix)
% writes out a .py file that draws lines between highly coupled positions
% (depending on threshold) in a structure.  Lines are colored according to whether they
% are contacting.

color_by_contact = 0;
color_by_ddg = 1;

% atom radii and contact cushion and distance calculation
[atom_type] = ['C', 'O', 'N', 'S'];
[atom_radius] = [1.9, 1.4, 1.5, 1.85];
cushion = 0.2;
distances = squareform(pdist([pdb.x pdb.y pdb.z]));
```

```
% arrow parameters
theta = pi/10;
phi = 0;
maxarrow = 1.0; minarrow = 0.1;
arr_red = 0.0; arr_green = 0.0; arr_blue = 1.0;     % colors range from 0 to 1

% binarize matrix and count number of incoming arcs
binmat = raw > threshold;
arcsin = sum(binmat,2);

% open file and begin writing python script
filename = [filename_prefix '.py'];
if exist(filename,'file')
    delete (filename); disp ([filename ' deleted'])
end
fid = fopen(filename,'w');
% Begin writing python script
fprintf (fid, 'from pymol.cgo import *\nfrom pymol import cmd\nimport math\nimport whrandom\n\n');
fprintf (fid, 'obj = []\n\n');

% write script to put sphere at all Ca of all network positions
for n = 1:size(binmat,1)
    if arcsin(n)
        ind = find(pdb.resnum==align_to_strseq(n) & strcmp(pdb.atomid,'CA'));
        fprintf(fid, 'obj.extend([ COLOR, 0.0, 0.0, 1.0 ])\n');
        fprintf(fid, 'obj.extend([ SPHERE, %3.1f, %3.1f, %3.1f, 0.5 ])\n',pdb.x(ind), pdb.y(ind), pdb.z(ind));
    end
end

% write script to draw arrows as indicated in binmat
for row = 1:size(binmat,1)
    for col = 1:size(binmat,2)
        if (binmat(row,col)==1) & (row ~= pertnum(col))
            % find indices of perturbed and coupled positions
            ind1 = find(pdb.resnum == align_to_strseq(pertnum(col)) & strcmp(pdb.atomid,'CA'));
            ind2 = find(pdb.resnum == align_to_strseq(row) & strcmp(pdb.atomid,'CA'));

            % determine coordinates for ends of arrowhead lines
            % first find theta and phi for spherical representation of second
            % point relative to first point.
            [th,ph,length] = cart2sph(pdb.x(ind2)-pdb.x(ind1), pdb.y(ind2)-pdb.y(ind1), pdb.z(ind2)-pdb.z(ind1));
            % length of arrow determined by sigmoidal formula (min and max defined above).
            arrow_length = minarrow + (maxarrow-minarrow)/(1+10^(log10(500000)-length)^0.3);
            % find coordinates of arrows based on length, theta, and phi (above)
            [rx, ry, rz] = sph2cart(theta+th, phi+ph, arrow_length);
            arrow1x = pdb.x(ind2) - rx; arrow1y = pdb.y(ind2) - ry; arrow1z = pdb.z(ind2) - rz;
            [rx, ry, rz] = sph2cart((-1)*theta+th, phi+ph, arrow_length);
            arrow2x = pdb.x(ind2) - rx; arrow2y = pdb.y(ind2) - ry; arrow2z = pdb.z(ind2) - rz;


            % are positions contacting? if yes, color green.  if not, color blue
            if color_by_contact
                contact = 0;
                atoms_in_1 = find(pdb.resnum==align_to_strseq(row));
                atoms_in_2 = find(pdb.resnum==align_to_strseq(pertnum(col)));
                for r = 1:size(atoms_in_1,1)
                    for c = 1:size(atoms_in_2,1)
                        atom1 = char(pdb.atomid(atoms_in_1(r))); atom2 = char(pdb.atomid(atoms_in_2(c)));
                        sumofradii = atom_radius(findstr(atom_type,atom1(1))) + atom_radius(findstr(atom_type,atom2(1)));
                        if distances(atoms_in_1(r), atoms_in_2(c)) <= sumofradii*(1+cushion);
                            contact = 1;
                        end
                    end
                end
                if contact
                    arr_red = 1.0; arr_green = 1.0; arr_blue = 0.0;
                else
                    arr_red = 0.0; arr_green = 0.0; arr_blue = 1.0;
                end
            end

            if color_by_ddg
                for n = 1:size(raw,2); raw(pertnum(n),n) = 0; end;
                maxddg = max(reshape(raw,1,size(raw,1)*size(raw,2)));
                arr_red = 0.0; arr_green = 0.0;
                arr_blue = raw(row,col)/maxddg;
            end

            fprintf (fid, 'obj.extend([ COLOR, %2.1f, %2.1f, %2.1f])\n', arr_red, arr_green, arr_blue);

            % main line
            fprintf (fid, 'obj.extend([ BEGIN, LINE_STRIP])\n');
            fprintf (fid, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', pdb.x(ind1), pdb.y(ind1), pdb.z(ind1));
            fprintf (fid, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', pdb.x(ind2), pdb.y(ind2), pdb.z(ind2));
            fprintf (fid, 'obj.append (END)\n');
            % arrowhead 1
            fprintf (fid, 'obj.extend([ BEGIN, LINE_STRIP])\n');
            fprintf (fid, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', pdb.x(ind2), pdb.y(ind2), pdb.z(ind2));
            fprintf (fid, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow1x, arrow1y, arrow1z);
            fprintf (fid, 'obj.append (END)\n');
            % arrowhead 2
            fprintf (fid, 'obj.extend([ BEGIN, LINE_STRIP])\n');
            fprintf (fid, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', pdb.x(ind2), pdb.y(ind2), pdb.z(ind2));
            fprintf (fid, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow2x, arrow2y, arrow2z);
            fprintf (fid, 'obj.append (END)\n');
        end
```

```
    end
end

obj_name = ['temppy'];
fprintf (fid, 'cmd.load_cgo(obj, ''%s'')',obj_name);
fclose(fid);
```

```
function writewvcfile(matrix, threshold, align_to_strseq, filename);
% This function writes outs .wvc script for viewer pro.  The script creates
% groups of residues according to their number of incoming links.  All
% residues with k=1 will be in group k1, etc.

% Example: writewvcpfile(matrix, 0.08, pdz.align_to_strseq, 'pdztest');
%_____________________________________________________________________


filename = [filename '.wvc'];
fid = fopen(filename, 'w');

binmat = matrix >= threshold;
links = sum(binmat);

for n = max(links):-1:2
    list = find(links == n);
    fprintf (fid, 'UnSelectAll\r');
    fprintf (fid, 'SetProperty Residue ');
    if ~isempty(list)
        for p = 1:size(list,2)
            fprintf (fid, 'id = %s', char(align_to_strseq(list(p))));
            if p < size(list,2)
                fprintf (fid, ', ');
            end
        end
    end
    fprintf (fid, ': select = on\r');
    fprintf (fid, 'Group k%s\r',num2str(n));
end
fprintf (fid, 'UnSelectAll');
fclose(fid);
```

```
function scrambled = scrambler(matrix);
% scrambles an input matrix while preserving symmetry and the diagonal
% elements

[rows,cols] = size(matrix);

count = 1;
for r = 1:rows
    for c = r+1:cols
        uppertri_vals(count) = matrix(r,c);
        count = count + 1;
    end
end

uppertri_vals_scr = uppertri_vals(randperm((rows^2-rows)/2));


count = 1;
for r = 1:rows
    scrambled(r,r) = matrix(r,r);
    for c = r+1:cols
        scrambled(r,c) = uppertri_vals_scr(count);
        scrambled(c,r) = uppertri_vals_scr(count);
        count = count + 1;
    end
end
```

```
function [links, cum_links] = scrambler2(matrix,threshold);
% this program makes ten scrambled matrices using scrambler.  these are
% used to determine a mean and standard deviation.  The data are fit with a
% cumulative poisson distribution.

% input:    matrix      N x N matrix
%           threshold   cutoff for binarization

% returns: links        matrix that is numtrials x N; contains number of links
%                       for each position in the matrix
%          cum_links    matrix that is numtrials x (max# of links+1) that
%                       contains the number of positions with k or more links.


% options
removeself = 0;                % if 1 then set diagonal to zero, if 0 then leave as is.
scrambler_method = 'symmetric';     % 'symmetric', 'rows' only, or 'whole'

numtrials = 100;

sizematrix = size(matrix,1);

% if not counting self coupling then set diagonal elements to zero
if removeself == 1
    for r = 1:sizematrix
        matrix(r,r) = 0;
    end
end
```

```
% cum_links = zeros(numtrials, 20);
links = zeros(numtrials, sizematrix);

for n = 1:numtrials

    % make scrambled matrix and binarize.
    switch scrambler_method
        case {'symmetric'}
            scrambled = scrambler(matrix);
        case {'rows'}
            scrambled = zeros(size(matrix));
            for row = 1:sizematrix
                scrambled(row,:) = matrix(row, randperm(sizematrix));
            end
        case {'whole'}
            scrambled = reshape(matrix(randperm(sizematrix*sizematrix)),sizematrix,sizematrix);
    end

    % binarize scrambled matrix
    scrambled_bin = scrambled >= threshold;

    % find number of links for each node
    links(n,:) = sum(scrambled_bin);

    % calculate cumulative number of links
    % note: number of links shifted by 1.  i.e. cum_links(1) has number of
    % nodes with 0 or more links.
    for p = 1:max(links(n,:)+1)
        cum_links(n,p) = sum(links(n,:) >= (p-1));
    end
    cum_links(n,:) = cum_links(n,:)/size(matrix,1);
end
```

```
function solution = fit_cumpoisson (xdata, ydata)
l = 4;

loglog (xdata, ydata, 'o','Markerfacecolor','b');
hold on;

solution = lsqcurvefit(@cumpoisson, l, xdata, ydata);

ysolution = cumpoisson(solution, xdata);
loglog (xdata, ysolution, '-g', 'Linewidth',1.5);


function F = cumpoisson (lambda, x)
F = 1-poisscdf(x,lambda);
```

```
function [out] = test_normcdf(x,xdata);

out = 1.*(1-normcdf(xdata,x(1), x(2)));
```

```
function [fitted_curve, y, resnorm] = scmatrix_norm(xdata, ydata);
% from Rama

lb = [0 0];
ub = [10 10];
options = optimset('display', 'final', 'TolFun', 1e-3);

plot (log10(xdata), log10(ydata), 'bo', 'MarkerFaceColor', 'blue');

[y, resnorm, residual] = lsqcurvefit('test_normcdf', [1 1], xdata, ydata, lb, ub, options);

xplot = [1:max(size(xdata))];
yplot = test_normcdf(y, xplot);
hold on;
plot (log10(xplot), log10(yplot), '-g', 'LineWidth', 1.5);
fitted_curve = [xplot', yplot'];
```

Appendix B: MATLAB code for structural analysis

```
function [pdb] = loadmodel(filename, name)
% Loads model in proper pdb format into structure array, calculates
% positional errors, gets data/model parameters from datasets.txt,
% and normalizes occupancies.

% Input:  .pdb file must be in proper format (i.e. ac identifier in col 17
%           with alt conf lines together).  This is the output of the pdb_write
%           command in O, version8 (note that the pdb file output from this
%           command differs from proper pdb format in one way: atom numbers
%           for atoms with alternate conformations are the same).
%         name must agree with one of the setnames in the datasets.txt
%           file.
% Output:  structure array with setname as prefix.  structure contains the
%         following fields:
%           atomnum         atom number in structure
%           atomid          atom identifier
%           ac              alternate conformation identifier
%           res             residue name
%           chainid         chain identifier
%           resnum          residue number
%           x               x coord
%           y               y coord
%           z               z coord
%           occ             occupancy (normalized)
%           bfactor         B factor
%           segid           segment identifier
%           label           res name,res#,atomid    (eg: GLY322N)
%           label2                  res#,atomid    (eg: ALA322N -- at mutated pos)
%           set             setname
%           protein         name of protein
%           synch           synchrotron where collected
%           refl            number of unique reflections
%           resolution      resolution of data set
%           rfact           R factor of data set
%           mos             mosaicity
%           length          length of unit cell
%           rfr             R free of latest refinement
%           r               R of latest refinement
%           totalatoms1     total number of atoms in model (including all
%                               acs and those with occ = 0)
%           totalatoms2     total number of atoms in model not including
%                               occ = 0 and counting alt conf atoms only once
%           poserr          positional error - calculated by stroud formula


prefix = name;
[fid1,message] = fopen(filename, 'r');
atoms = 0;
if fid1 == -1
    disp(message)
else
    fid2 = fopen('temp.pdb','w+');
    while feof(fid1) == 0
        line = fgets(fid1);
        num = findstr('ATOM ',line);
        if size(num) > 0                                        % if the line has "ATOM" then read it and write
to temp.pdb
            fprintf(fid2,line);
            atoms = atoms + 1;
        end
    end
end
status1 = fclose(fid1);
frewind(fid2);
last = 0;

for n = 1:atoms                                                 % read from all lines:
    fseek(fid2,6,'cof');
    name.atomnum(n) = str2num(char(fread(fid2,5,'char')'));     % atom number
    fseek(fid2,2,'cof');
    name.atomid(n) = cellstr(char(fread(fid2,3,'char')'));      % atom identifier
    name.ac(n) = cellstr(char(fread(fid2,1,'char')'));          % alternate conformation identifier
    name.res(n) = cellstr(char(fread(fid2,3,'char')'));         % residue
    fseek(fid2,1,'cof');
    name.chainid(n) = cellstr(char(fread(fid2,1,'char')'));     % chain identifier
    name.resnum(n) = str2num(char(fread(fid2,4,'char')'));      % residue number
    fseek(fid2,4,'cof');
    name.x(n) = str2num(char(fread(fid2,8,'char')'));           % x coord
    name.y(n) = str2num(char(fread(fid2,8,'char')'));           % y coord
    name.z(n) = str2num(char(fread(fid2,8,'char')'));           % z coord
    name.occ(n) = str2num(char(fread(fid2,6,'char')'));        % occupancy

    if strcmp(name.ac(n),'B')                                      % normalize occupancies if there are
alternate confs only
        name.occ(n) = name.occ(n)/(name.occ(n)+name.occ(n-1));  %   do calculation only after second
conformation is read
        name.occ(n-1) = 1-name.occ(n);                          %   note: this requires the pdb format is proper!
    end
```

```
    name.bfactor(n) = str2num(char(fread(fid2,6,'char')'));             % bfactor
    fseek(fid2,6,'cof');
    name.segid(n) = cellstr(char(fread(fid2,4,'char')'));               % segment identifier
    fseek(fid2,1,'cof');                                                % this goes to the beginning of the next line
end

status2 = fclose(fid2);
delete('temp.pdb');

name.atomnum = name.atomnum'; name.atomid = name.atomid'; name.ac = name.ac'; name.res = name.res';
name.chainid = name.chainid'; name.resnum = name.resnum'; name.x = name.x'; name.y = name.y'; name.z = name.z';
name.occ = name.occ'; name.bfactor = name.bfactor'; name.segid = name.segid';

name.label = cellstr([char(name.res) num2str(name.resnum) char(name.atomid)]);
name.label2 = cellstr([num2str(name.resnum) char(name.atomid)]);


                                                                        % get information about data set from
datasets.txt
[setname protein synch res comp rfact lastr mos length refl atomnum rfr r]=textread('datasets.txt',...
    '%s  %s      %s    %f  %f   %f    %f    %f  %f     %d   %d      %f  %f','headerlines',1);
[numsets,blah]=size(setname);
for m = 1:numsets
    if strcmp(setname(m),prefix)                                        % if prefix matches assign:
        name.set = setname(m);                                          %   setname, #refl, res, rfactor, mos, length
        name.protein = protein(m);
        name.synch = synch(m);                                          %   synchrotron, r, rfree
        name.refl = refl(m);
        name.resolution = res(m);
        name.rfact = rfact(m);
        name.mos = mos(m);
        name.length = length(m);
        name.rfr = rfr(m);
        name.r = r(m);
    end
end

name.totalatoms1 = atoms;                                               % total number of atoms in model
[num_occ0,blah] = size(find(name.occ==0));                              % number of atoms with occ = 0
[num_ac,blah] = size(find(strcmp(name.ac,'A')));                        % number of atoms with alternate conformations
name.totalatoms2 = name.totalatoms1 - num_occ0 - num_ac;                % number of atoms in model excluding those occ =
o and
                                                                        %       counting those with alt confs only once.

for n = 1:name.totalatoms1                                              % Calculate positional errors for all atoms
    name.poserr(n)=stroud(name.bfactor(n), name.totalatoms2, name.refl);
end
name.poserr = name.poserr';

pdb = name;
```

```
function [str1,str2] = dr(str1,str2,alt)
% This function accepts two structures and finds the difference of the two:
%           str1 --> str2
% It returns the x-displacement (.dx), y-displacement (.dy), z-displacement (.dz),
% displacement (.dr), normalized displacement(.drnorm).  Using the
% direction of the vector (calculated in spherical coordinates) and the
% magnitude of the normalized displacement, the function also calculates
% the normalized x, y, and z vectors.

% If alt = 1, atoms that have alternate conformations the difference is calculated
% using a weighted average of its position and propagated positional
% errors. If alt = 0 then the second conformation is not used.

% Example usage: [ww2,aw27] = dr(ww2,aw27,0);

% Input: two models in structure arrays.
% Output:  additional fields in each structure:
%       1) .dr      raw disp
%       2) .drn     normalized displacement
%       3) .dx      raw displacement in x
%       4) .dy      raw displacement in y
%       5) .dz      raw displacement in z
%       6) .dxn     normalized displacement in x
%       7) .dyn     normalized displacement in y
%       8) .dzn     normalized displacement in z
%       9) .drsets  list of sets used for calculation
%       10).drmsg   'alternate conformations used/not used'


[numchainA,blah] = size(find(strcmp(str1.chainid,'A')));        % number of atoms with chainid A
[numchainP,blah] = size(find(strcmp(str1.chainid,'P')));        % number of atoms with chainid P
total = numchainA + numchainP;                                  % number of atoms in model excluding waters

for n = 1:total                                                 % Go through all protein and peptide atoms in model 1

                                                                % find indices and number of occurrences for label1
(res,#,atomid) and label2(#,atomid)
     ind1_1 = find(strcmp(str1.label,str1.label(n)));           % indices of label1 (res,#,atomid) in str1
     [num1_1,blah] = size(ind1_1);                              % number of occurrences of label1 in str1 (could be 1,2)
     mc = strcmp(str1.atomid(n),'N')|strcmp(str1.atomid(n),'CA')|strcmp(str1.atomid(n),'C')|strcmp(str1.atomid(n),'O');
% is it mainchain atom?

     ind1_2 = find(strcmp(str2.label,str1.label(n)));           % indices of label1 in str2
```

```
    [num1_2,blah] = size(ind1_2);                               % number of occurrences of label1 in str2 (could be
0,1,2)
    ind2_2 = find(strcmp(str2.label2,str1.label2(n)));          % indices of label2 in str2  (label2 only has #,atomid -
can be used to check for mutation)
    [num2_2,blah] = size(ind2_2);                               % number of occurrences of label2 (#,atomid) in str2
(could be 0,1,2)

    if (alt==0) & strcmp(str1.ac(n),'B')
        occupiedin1 = 0;
    else
        occupiedin1 = str1.occ(n);
    end

    occupiedin2=0;
    switch num1_2                                               % is the atom found in structure 2?
        case 0
            if (num2_2==0)                                      % if label1 is not found and label2 is not found then it
must not be modeled in str2
                occupiedin2 = 0;
            elseif ((num2_2==1)&mc)                             % if label1 is not found but label2 is found it must be
a mutated position (only residue names don't match).
                occupiedin2 = str2.occ(ind2_2(1));              % Only count as occupied if mainchain atom; let
occupiedin2 be occupancy of that mc atom (should be 1).
            end
        case 1
            occupiedin2 = str2.occ(ind1_2(1));                  % if label1 is found once then let occupiedin2 be
whatever the occupancy of that atom.
        case 2
            occupiedin2 = 1;                                    % if label2 is found twice then it must be modeled as
alternate confs in str2.
    end

    occupiedinboth = occupiedin1 & occupiedin2;                 % is it occupied in both structures?

    if occupiedinboth                                           % if the atom is in both structures...

                                                                % str1 atom will be assigned coordinates,pe.
        if ((num1_1==2) & alt)                                  % If atom is found twice and alt conf = 'on' = 1, then
weight.
            x1 = str1.x(ind1_1(1))*str1.occ(ind1_1(1)) +  str1.x(ind1_1(2))*str1.occ(ind1_1(2));
            y1 = str1.y(ind1_1(1))*str1.occ(ind1_1(1)) +  str1.y(ind1_1(2))*str1.occ(ind1_1(2));
            z1 = str1.z(ind1_1(1))*str1.occ(ind1_1(1)) +  str1.z(ind1_1(2))*str1.occ(ind1_1(2));
            pe1 = ((str1.occ(ind1_1(1))*str1.poserr(ind1_1(1)))^2 + (str1.occ(ind1_1(2))*str1.poserr(ind1_1(2)))^2)^0.5;
        else                                                    % Otherwise, x1,y1,z1,pe1 are simply its corresponding
values.
            x1 = str1.x(ind1_1(1));
            y1 = str1.y(ind1_1(1));
            z1 = str1.z(ind1_1(1));
            pe1 = str1.poserr(ind1_1(1));
        end
                                                                % str2 atom will be assigned coordinates,pe, and index.
        if ((num1_2(1)==2) & alt)                               % If the atom is found twice and alt confs are 'on' then
take weighted ave
            x2 = str2.x(ind1_2(1))*str2.occ(ind1_2(1)) +  str2.x(ind1_2(2))*str2.occ(ind1_2(2));
            y2 = str2.y(ind1_2(1))*str2.occ(ind1_2(1)) +  str2.y(ind1_2(2))*str2.occ(ind1_2(2));
            z2 = str2.z(ind1_2(1))*str2.occ(ind1_2(1)) +  str2.z(ind1_2(2))*str2.occ(ind1_2(2));
            pe2 = ((str2.occ(ind1_2(1))*str2.poserr(ind1_2(1)))^2 + (str2.occ(ind1_2(2))*str2.poserr(ind1_2(2)))^2)^0.5;
            i2 = ind1_2(1);
        else                                                    % Otherwise, coords and error are corresponding values.
            x2 = str2.x(ind2_2(1));
            y2 = str2.y(ind2_2(1));
            z2 = str2.z(ind2_2(1));
            pe2 = str2.poserr(ind2_2(1));
            i2 = ind2_2(1);
        end
                                                                % Calculate displacements
        str1.dx(n) = x2 - x1;
        str1.dy(n) = y2 - y1;
        str1.dz(n) = z2 - z1;
        str1.dr(n) = (str1.dx(n)^2 + str1.dy(n)^2 + str1.dz(n)^2)^0.5;

        str1.drn(n) = str1.dr(n)/(pe1^2 + pe2^2)^0.5;
        [theta, phi, r] = cart2sph (str1.dx(n),str1.dy(n),str1.dz(n));
        [str1.dxn(n), str1.dyn(n), str1.dzn(n)] = sph2cart (theta, phi, str1.drn(n));

        str2.dx(i2) = str1.dx(n); str2.dy(i2) = str1.dy(n); str2.dz(i2) = str1.dz(n); str2.dr(i2) = str1.dr(n);
        str2.dxn(i2) = str1.dxn(n); str2.dyn(i2) = str1.dyn(n); str2.dzn(i2) = str1.dzn(n); str2.drn(i2) = str1.drn(n);

    else                                                        % if atom is not found in str1 and str2 in above
conditions then set values to 0.
        str1.dx(n) = 0; str1.dy(n) = 0; str1.dz(n) = 0; str1.dr(n) = 0; str1.dxn(n) = 0; str1.dyn(n) = 0; str1.dzn(n) =
0; str1.drn(n) = 0;
    end
end

str1.dx = str1.dx';
str1.dy = str1.dy';
str1.dz = str1.dz';
str1.dr = str1.dr';
str1.dxn = str1.dxn';
str1.dyn = str1.dyn';
str1.dzn = str1.dzn';
str1.drn = str1.drn';

str2.dx = str2.dx';
str2.dy = str2.dy';
str2.dz = str2.dz';
```

```
str2.dr = str2.dr';
str2.dxn = str2.dxn';
str2.dyn = str2.dyn';
str2.dzn = str2.dzn';
str2.drn = str2.drn';

str1.dr_sets = [str1.set str2.set];
str2.dr_sets = [str1.set str2.set];

if alt == 1
    str1.drmsg = 'alternate conformations used';
    str2.drmsg = 'alternate conformations used';
else
    str1.drmsg = 'alternate conformations not used';
    str2.drmsg = 'alternate conformations not used';
end
```

```
function [str1, str2] = dr_outfiles(model_list, arrow_zone, raw_or_norm)

% Function that automates analysis and outputs .txt and .py files.  The
% .txt file has all the dr and drn values listed and can be read into
% sigmaplot or excel.  The .py file has the python commands to draw arrows.
% The user can specify one of three options defining which set of arrows to
% be drawn:
%       1) all
%       2) Calpha atoms in peptide binding pocket
%       3) Calpha atoms in carboxylate binding loop
% The length of the arrows can be either:
%       1) raw (dr)
%       2) normalized (drn).
% If the normalized length is chosen they will automatically be scaled by a
% factor of 6.

% Example: models = strvcat('ww2','aw27');
%          [ww2,aw27] = ddr_outfiles (models, 2, 'dr')

%-------------------------------------------------------------------------

% regions of PDZ domain
backbone = ['N' 'CA' 'O'];
pep_bind_pocket_res = [3:7,315:330,371:385];
carb_bind_loop_res = [315:325];

% arrow parameters
theta = pi/10;
phi = 0;
maxarrow = 1.0; minarrow = 0.1;
arr_red = 0.3; arr_green = 0.3; arr_blue = 1.0;     % colors range from 0 to 1

if strcmp(raw_or_norm,'dr')
    scale = 1;
elseif strcmp(raw_or_norm,'drn')
    scale = 6;
end

% obtain structures in structure arrays using loadmodel function.  Always
% read in ww2 structure as temp since .txt file is printed out using these labels.
ww2temp = loadmodel('models/ww2_12.pdb','ww2'); disp('ww2temp loaded')

% model1_file = ['models/' file1];
% str1 = loadmodel(model1_file,model1); disp([model1 ' loaded']);
%
% model2_file = ['models/' file2];
% str2 = loadmodel(model2_file,model2); disp([model2 ' loaded']);

model_list = cellstr(model_list);
files_in_models_folder = dir('models');

for modelnum = 1:size(model_list,1)
    for filenum = 1:size(files_in_models_folder,1)
        if findstr(char(model_list(modelnum)), files_in_models_folder(filenum).name)
            full_filename = ['models/' files_in_models_folder(filenum).name]
            switch modelnum
                case 1
                    str1 = loadmodel(full_filename, model_list(1)); disp([char(model_list(1)) ' loaded']);
                case 2
                    str2 = loadmodel(full_filename, model_list(2)); disp([char(model_list(2)) ' loaded']);
                case 3
                    str3 = loadmodel(full_filename, model_list(3)); disp([char(model_list(3)) ' loaded']);
                case 4
                    str4 = loadmodel(full_filename, model_list(4)); disp([char(model_list(4)) ' loaded']);
            end %switch
        end %if
    end %for filenum
end %for modelnum

% Calculate difference with dr.m function: str1 --> str2
[str1,str2] = dr(str1,str2,0);
disp(['calculated dr of ' char(model_list(1)) ' and ' char(model_list(2))])

% open .txt and .py files as txt_file and py_file, respectively
txt_file = [char(model_list(1)) '_' char(model_list(2)) '_' raw_or_norm '.txt'];
if exist (txt_file,'file')
```

```
    delete (txt_file); disp (['old ' txt_file ' overwritten'])
end
fid = fopen(txt_file,'w');

switch arrow_zone
    case 1
        py_arrows = 'all';
    case 2
        py_arrows = 'pbp';
    case 3
        py_arrows = 'cbl';
end

py_file = [char(model_list(1)) '_' char(model_list(2)) '_' py_arrows '_' raw_or_norm '.py'];
if exist (py_file,'file')
    delete (py_file); disp (['old ' py_file ' overwritten'])
end
fid2 = fopen(py_file,'w');

% Begin writing python script
fprintf (fid2, 'from pymol.cgo import *\nfrom pymol import cmd\nimport math\nimport whrandom\n\n');
fprintf (fid2, 'obj = []\n\n');
fprintf (fid2, 'obj.extend([ LINEWIDTH, 0.5])\n');

total_pro_pep = size(find(strcmp(ww2temp.chainid,'A')),1) + size(find(strcmp(ww2temp.chainid,'P')),1);

for n = 1:total_pro_pep
    % to the text file, write: ww2temp.label and corresponding str1.dr and str1.drn.
    index = find(strcmp(ww2temp.label(n),str1.label));
    if index
        fprintf (fid, '%s \t %f \t %f\n', char(ww2temp.label(n)), str1.dr(index), str1.drn(index));
    else
        fprintf (fid, '%s \t 0 \t 0\n', char(ww2temp.label(n)));
    end

    % if the atom is in both ww2temp and str1 and the displacement is
    % nonzero then calculate coordinates of the arrow lines

    if index & (str1.dr(index)~=0)
        % determine coordinates for end of arrow.
        if strcmp(raw_or_norm,'dr')
            x2 = str1.x(index) + str1.dx(index)/scale;
            y2 = str1.y(index) + str1.dy(index)/scale;
            z2 = str1.z(index) + str1.dz(index)/scale;
        else
            x2 = str1.x(index) + str1.dxn(index)/scale;
            y2 = str1.y(index) + str1.dyn(index)/scale;
            z2 = str1.z(index) + str1.dzn(index)/scale;
        end

        % determines coordinates for ends of arrowhead lines
            % first find theta and phi for spherical representation of second
            % point relative to first point.
        [th,ph,length] = cart2sph(x2-str1.x(index), y2-str1.y(index), z2-str1.z(index));
            % length of arrow determined by sigmoidal formula (min and max
            % defined above).
        arrow_length = minarrow + (maxarrow-minarrow)/(1+10^(log10(500000)-str1.drn(index))^0.3);
            % find coordinates of arrows based on length, theta, and phi (above)
        [rx, ry, rz] = sph2cart(theta+th, phi+ph, arrow_length);
        arrow1x = x2 - rx; arrow1y = y2 - ry; arrow1z = z2 - rz;
        [rx, ry, rz] = sph2cart((-1)*theta+th, phi+ph, arrow_length);
        arrow2x = x2 - rx; arrow2y = y2 - ry; arrow2z = z2 - rz;

        % write the pymol line commands to file in appropriate conditions
        switch arrow_zone
            case 1
                % if all is selected then write lines for every atom with nonzero
                % displacement
                if str1.dr(index)~=0
                    % main line
                    fprintf (fid2, 'obj.extend([ COLOR, %2.1f, %2.1f, %2.1f])\n', arr_red, arr_green, arr_blue);
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', str1.x(index), str1.y(index),
str1.z(index));
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.append (END)\n');
                    % arrowhead 1
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow1x, arrow1y, arrow1z);
                    fprintf (fid2, 'obj.append (END)\n');
                    % arrowhead 2
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow2x, arrow2y, arrow2z);
                    fprintf (fid2, 'obj.append (END)\n');
                end
            case 2
                % if only Calpha of pep_bind_pocket is selected then only write
                % lines for backbone atoms in this region
                if str1.dr(index)~=0 & findstr(backbone,char(ww2temp.atomid(n))) &
find(pep_bind_pocket_res==str1.resnum(index))
%                 if str1.dr(index)~=0 & find(pep_bind_pocket_res==str1.resnum(index))
                    fprintf (fid2, 'obj.extend([ COLOR, %2.1f, %2.1f, %2.1f])\n', arr_red, arr_green, arr_blue);
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', str1.x(index), str1.y(index),
str1.z(index));
```

```
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.append (END)\n');

                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow1x, arrow1y, arrow1z);
                    fprintf (fid2, 'obj.append (END)\n');

                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow2x, arrow2y, arrow2z);
                    fprintf (fid2, 'obj.append (END)\n');
                end
            case 3
                % if only Calpha of carb_bind_loop is selected then only write
                % lines for backbone atoms in this region
                if str1.dr(index)~=0 & findstr(backbone,char(ww2temp.atomid(n))) &
find(carb_bind_loop_res==str1.resnum(index))
                    fprintf (fid2, 'obj.extend([ COLOR, %2.1f, %2.1f, %2.1f])\n', arr_red, arr_green, arr_blue);
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', ww2.x(n), ww2.y(n), ww2.z(n));
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.append (END)\n');

                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow1x, arrow1y, arrow1z);
                    fprintf (fid2, 'obj.append (END)\n');

                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow2x, arrow2y, arrow2z);
                    fprintf (fid2, 'obj.append (END)\n');
                end
        end     % of switch
    end     % of if
end

obj_name = [char(model_list(1)) '_' char(model_list(2)) '_' py_arrows '_' raw_or_norm];
fprintf (fid2, 'cmd.load_cgo(obj,''%s'')',obj_name);

fclose(fid); disp([txt_file ' written'])
fclose(fid2); disp ([py_file ' written'])
```

```
function [str1, str2, str3, str4] = ddr(str1, str2, str3, str4, alt)
% ddr calculates the structural coupling vector for 4 structures in a
% structure cycle with format:
%       str1 -------> str2
%         |             |
%         |             |
%       str3 -------> str4
% If alternate conformations exist they will be treated as weighted
% averages if alt=1; if alt=0 then second conformation is not used.

% Input: 4 models in structure arrays; alternate conformation flag

% Output: 4 structures with additional fields for ddr and ddrnorm:
%   .ddr            raw structural coupling
%   .ddrn           normalized coupling
%   .ddx            x component of ddr
%   .ddy            y component of ddr
%   .ddz            z component of ddr
%   .ddx            x component of ddrn
%   .ddy            y component of ddrn
%   .ddz            z component of ddrn
%   .ddr_sets       list of sets used for calculations
%   .ddrmsg         'alternate conformations used/not used'

[numchainA,blah] = size(find(strcmp(str1.chainid,'A')));            % number of atoms with chainid A
[numchainP,blah] = size(find(strcmp(str1.chainid,'P')));            % number of atoms with chainid P
total = numchainA + numchainP;                                      % number of atoms in model excluding waters


for n = 1:total
    ind1_1 = find(strcmp(str1.label(n),str1.label));                % where atom label1 (res, #, atomid) occurs in str1
(1,2)
    [num1_1,blah] = size(ind1_1);                                   % how many times label1 occurs in str1 (1,2)
    ind2_1 = find(strcmp(str1.label2(n),str1.label2));              % where atom label2 (#,atomid) occurs in str1 (1,2)
    mc = strcmp(str1.atomid(n),'N')|strcmp(str1.atomid(n),'CA')|strcmp(str1.atomid(n),'C')|strcmp(str1.atomid(n),'O');
% is it mainchain atom?
    if (alt==0)&(strcmp(str1.ac(n),'B'))
        occupiedin1 = 0;
    else
        occupiedin1 = str1.occ(n);
    end

    ind1_2 = find(strcmp(str1.label(n),str2.label));                % where atom label1 occurs in str2 (0,1,2)
    [num1_2,blah] = size(ind1_2);                                   % how many times label1 occurs in str2
    ind2_2 = find(strcmp(str1.label2(n),str2.label2));              % where atom label2 occurs in str2 (0,1,2) - should
be superset of ind1_x
    [num2_2,blah] = size(ind2_2);                                   % how many times label2 occurs in str2
    occupiedin2=0;
    switch num1_2                                                   % is the atom found in structure 2?
        case 0
```

```
            if (num2_2==0)                                          % if label1 is not found and label2 is not found
then it must not be modeled in str2
                occupiedin2 = 0;
            elseif ((num2_2==1)&mc)                                 % if label1 is not found but label2 is found it must
be a mutated position (only residue names don't match).
                occupiedin2 = str2.occ(ind2_2(1));                  %     Only count as occupied if mainchain atom; let
occupiedin2 be occupancy of that mc atom (should be 1).
            end
        case 1
            occupiedin2 = str2.occ(ind1_2(1));                      % if label1 is found once then let occupiedin2 be
whatever the occupancy of that atom.
        case 2
            occupiedin2 = 1;                                        % if label2 is found twice then it must be modeled
as alternate confs in str2.
    end

    ind1_3 = find(strcmp(str1.label(n),str3.label));                % where atom label1 occurs in str3 (0,1,2)
    [num1_3,blah] = size(ind1_3);                                   % how many times label1 occurs in str3
    ind2_3 = find(strcmp(str1.label2(n),str3.label2));              % where atom label2 occurs in str3 (0,1,2)
    [num2_3,blah] = size(ind2_3);                                   % how many times labels occurs in str3
    switch num1_3                                                   % is the atom found in structure 2?
        case 0
            if (num2_3==0)                                          % if label1 is not found and label2 is not found
then it must not be modeled in str2
                occupiedin3 = 0;
            elseif ((num2_3==1)&mc)                                 % if label1 is not found but label2 is found it must
be a mutated position (only residue names don't match).
                occupiedin3 = str3.occ(ind2_3(1));                  %     Only count as occupied if mainchain atom; let
occupiedin2 be occupancy of that mc atom (should be 1).
            end
        case 1
            occupiedin3 = str3.occ(ind1_3(1));                      % if label1 is found once then let occupiedin2 be
whatever the occupancy of that atom.
        case 2
            occupiedin3 = 1;                                        % if label2 is found twice then it must be modeled
as alternate confs in str2.
    end

    ind1_4 = find(strcmp(str1.label(n),str4.label));                % where atom label1 occurs in str4 (0,1,2)
    [num1_4,blah] = size(ind1_4);                                   % how many times label1 occurs in str4
    ind2_4 = find(strcmp(str1.label2(n),str4.label2));              % where atom label2 occurs in str4 (0,1,2)
    [num2_4,blah] = size(ind2_4);                                   % how many times label2 occurs in str4
    switch num1_4                                                   % is the atom found in structure 2?
        case 0
            if (num2_4==0)                                          % if label1 is not found and label2 is not found
then it must not be modeled in str2
                occupiedin4 = 0;
            elseif ((num2_4==1)&mc)                                 % if label1 is not found but label2 is found it must
be a mutated position (only residue names don't match).
                occupiedin4 = str4.occ(ind2_4(1));                  %     Only count as occupied if mainchain atom; let
occupiedin2 be occupancy of that mc atom (should be 1).
            end
        case 1
            occupiedin4 = str4.occ(ind1_4(1));                      % if label1 is found once then let occupiedin2 be
whatever the occupancy of that atom.
        case 2
            occupiedin4 = 1;                                        % if label2 is found twice then it must be modeled
as alternate confs in str2.
    end

    if ((num1_2>=1) & (num1_3>=1) & (num1_4>=1))                    % if label1 occurs at least once in each then it is
common
        common = 1;
    elseif (mc & (num2_2>=1) & (num2_3>=1) & (num2_4>=1))           % otherwise, only if it is a mainchain atom common
to all
        common = 1;                                                  % will it be considered common. (ie mutated position)
    else
        common = 0;
    end

    occupiedinall = occupiedin1 & occupiedin2 & occupiedin3 & occupiedin4;  % is the atom found in all 4 structures?

    if occupiedinall                                                % if atom is in all or if mainchain common to all...
                                                                    % assign str1 coordinates and pe
        if ((num1_1 == 2) & alt)                                    % if atom is found twice in str1 and alt is on,
coords and pe are weighted
            x1 = str1.x(ind1_1(1))*str1.occ(ind1_1(1)) + str1.x(ind1_1(2))*str1.occ(ind1_1(2));
            y1 = str1.y(ind1_1(1))*str1.occ(ind1_1(1)) + str1.y(ind1_1(2))*str1.occ(ind1_1(2));
            z1 = str1.z(ind1_1(1))*str1.occ(ind1_1(1)) + str1.z(ind1_1(2))*str1.occ(ind1_1(2));
            pe1 = ((str1.occ(ind1_1(1))*str1.poserr(ind1_1(1)))^2 + (str1.occ(ind1_1(2))*str1.poserr(ind1_1(2)))^2)^0.5;
        else                                                        % if atom found once, then coords and pe are same
            x1 = str1.x(ind1_1(1));
            y1 = str1.y(ind1_1(1));
            z1 = str1.z(ind1_1(1));
            pe1 = str1.poserr(ind1_1(1));
        end
                                                                    % assign str2 coordinates and pe
        if ((num1_2 == 2) & alt)                                    % if atom is found twice and alt is on then weight
            x2 = str2.x(ind1_2(1))*str2.occ(ind1_2(1)) + str2.x(ind1_2(2))*str2.occ(ind1_2(2));
            y2 = str2.y(ind1_2(1))*str2.occ(ind1_2(1)) + str2.y(ind1_2(2))*str2.occ(ind1_2(2));
            z2 = str2.z(ind1_2(1))*str2.occ(ind1_2(1)) + str2.z(ind1_2(2))*str2.occ(ind1_2(2));
            pe2 = ((str2.occ(ind1_2(1))*str2.poserr(ind1_2(1)))^2 + (str2.occ(ind1_2(2))*str2.poserr(ind1_2(2)))^2)^0.5;
        else                                                        % otherwise, assign to values for index of label2,
first element
            x2 = str2.x(ind2_2(1));
            y2 = str2.y(ind2_2(1));
            z2 = str2.z(ind2_2(1));
```

```
            pe2 = str2.poserr(ind2_2(1));
        end
        i2 = ind2_2(1);
                                                        % assign str3 coordinate and pe
        if ((num1_3 == 2) & alt)                        % if atom is found twice and alt is on then weight
            x3 = str3.x(ind1_3(1))*str3.occ(ind1_3(1)) + str3.x(ind1_3(2))*str3.occ(ind1_3(2));
            y3 = str3.y(ind1_3(1))*str3.occ(ind1_3(1)) + str3.y(ind1_3(2))*str3.occ(ind1_3(2));
            z3 = str3.z(ind1_3(1))*str3.occ(ind1_3(1)) + str3.z(ind1_3(2))*str3.occ(ind1_3(2));
            pe3 = ((str3.occ(ind1_3(1))*str3.poserr(ind1_3(1)))^2 + (str3.occ(ind1_3(2))*str3.poserr(ind1_3(2)))^2)^0.5;
        else                                            % otherwise, assign to values for index of label2,
first element
            x3 = str3.x(ind2_3(1));
            y3 = str3.y(ind2_3(1));
            z3 = str3.z(ind2_3(1));
            pe3 = str3.poserr(ind2_3(1));
        end
        i3 = ind2_3(1);
                                                        % assign str4 coordinate and pe
        if ((num1_4 == 2) & alt)                        % if atom is found twice and alt is on then weight
            x4 = str4.x(ind1_4(1))*str4.occ(ind1_4(1)) + str4.x(ind1_4(2))*str4.occ(ind1_4(2));
            y4 = str4.y(ind1_4(1))*str4.occ(ind1_4(1)) + str4.y(ind1_4(2))*str4.occ(ind1_4(2));
            z4 = str4.z(ind1_4(1))*str4.occ(ind1_4(1)) + str4.z(ind1_4(2))*str4.occ(ind1_4(2));
            pe4 = ((str4.occ(ind1_4(1))*str4.poserr(ind1_4(1)))^2 + (str4.occ(ind1_4(2))*str4.poserr(ind1_4(2)))^2)^0.5;
        else                                            % otherwise, assign to values for index of label2,
first element
            x4 = str4.x(ind2_4(1));
            y4 = str4.y(ind2_4(1));
            z4 = str4.z(ind2_4(1));
            pe4 = str4.poserr(ind2_4(1));
        end
        i4 = ind2_4(1);
                                                        % calculate ddr values
        str1.ddx(n) = (x2 - x1) - (x4 - x3);
        str1.ddy(n) = (y2 - y1) - (y4 - y3);
        str1.ddz(n) = (z2 - z1) - (z4 - z3);
        str1.ddr(n) = (str1.ddx(n)^2 + str1.ddy(n)^2 + str1.ddz(n)^2)^0.5;
        prop_pe = (pe1^2 + pe2^2 + pe3^2 + pe4^2)^0.5;
        str1.ddrn(n) = str1.ddr(n)/prop_pe;
%         str1.ddxn(n) = str1.ddx(n)/prop_pe;
%         str1.ddyn(n) = str1.ddy(n)/prop_pe;
%         str1.ddzn(n) = str1.ddz(n)/prop_pe;

        [theta, phi, r] = cart2sph (str1.ddx(n),str1.ddy(n),str1.ddz(n));
        [str1.ddxn(n), str1.ddyn(n), str1.ddzn(n)] = sph2cart (theta, phi, str1.ddrn(n));
                                                        % assign to appropriate indices in other structures
        str2.ddx(i2) = str1.ddx(n); str3.ddx(i3) = str1.ddx(n); str4.ddx(i4) = str1.ddx(n);
        str2.ddxn(i2) = str1.ddxn(n); str3.ddxn(i3) = str1.ddxn(n); str4.ddxn(i4) = str1.ddxn(n);
        str2.ddy(i2) = str1.ddy(n); str3.ddy(i3) = str1.ddy(n); str4.ddy(i4) = str1.ddy(n);
        str2.ddyn(i2) = str1.ddyn(n); str3.ddyn(i3) = str1.ddyn(n); str4.ddyn(i4) = str1.ddyn(n);
        str2.ddz(i2) = str1.ddz(n); str3.ddz(i3) = str1.ddz(n); str4.ddz(i4) = str1.ddz(n);
        str2.ddzn(i2) = str1.ddzn(n); str3.ddzn(i3) = str1.ddzn(n); str4.ddzn(i4) = str1.ddzn(n);
        str2.ddr(i2) = str1.ddr(n); str3.ddr(i3) = str1.ddr(n); str4.ddr(i4) = str1.ddr(n);
        str2.ddrn(i2) = str1.ddrn(n); str3.ddrn(i3) = str1.ddrn(n); str4.ddrn(i4) = str1.ddrn(n);
    else                                                % otherwise set to zero
        str1.ddx(n) = 0; str1.ddy(n) = 0; str1.ddz(n) = 0; str1.ddr(n) = 0;
        str1.ddrn(n) = 0; str1.ddxn(n) = 0; str1.ddyn(n) = 0; str1.ddzn(n) = 0;
    end
end

str1.ddx = str1.ddx'; str1.ddy = str1.ddy'; str1.ddz = str1.ddz';
str1.ddxn = str1.ddxn'; str1.ddyn = str1.ddyn'; str1.ddzn = str1.ddzn';
str1.ddr = str1.ddr'; str1.ddrn = str1.ddrn';

str2.ddx = str2.ddx'; str2.ddy = str2.ddy'; str2.ddz = str2.ddz';
str2.ddxn = str2.ddxn'; str2.ddyn = str2.ddyn'; str2.ddzn = str2.ddzn';
str2.ddr = str2.ddr'; str2.ddrn = str2.ddrn';

% str3.ddx = str3.ddx';
% str3.ddy = str3.ddy';
% str3.ddz = str3.ddz';
% str3.ddxn = str3.ddxn';
% str3.ddyn = str3.ddyn';
% str3.ddzn = str3.ddzn';
% str3.ddr = str3.ddr';
% str3.ddrn = str3.ddrn';
%
% str3.ddx = str3.ddx';
% str3.ddy = str3.ddy';
% str3.ddz = str3.ddz';
% str3.ddxn = str3.ddxn';
% str3.ddyn = str3.ddyn';
% str3.ddzn = str3.ddzn';
% str3.ddr = str3.ddr';
% str3.ddrn = str3.ddrn';

str1.ddr_sets = [str1.set str2.set str3.set str4.set];
str2.ddr_sets = [str1.set str2.set str3.set str4.set];
str3.ddr_sets = [str1.set str2.set str3.set str4.set];
str4.ddr_sets = [str1.set str2.set str3.set str4.set];

if alt == 1
    str1.ddrmsg = 'alternate conformations used'; str2.ddrmsg = 'alternate conformations used';
    str3.ddrmsg = 'alternate conformations used'; str4.ddrmsg = 'alternate conformations used';
else
    str1.ddrmsg = 'alternate conformations not used'; str2.ddrmsg = 'alternate conformations not used';
    str3.ddrmsg = 'alternate conformations not used'; str4.ddrmsg = 'alternate conformations not used';
end
```

```
function [str1, str2, str3, str4] = ddr_outfiles(model_list, arrow_zone, raw_or_norm)

% Function that automates analysis and outputs .txt and .py files.  The
% .txt file has all the dr and drn values listed and can be read into
% sigmaplot or excel.  The .py file has the python commands to draw arrows.
% The user can specify one of three options defining which set of arrows to
% be drawn:
%       1) all
%       2) Calpha atoms in peptide binding pocket
%       3) Calpha atoms in carboxylate binding loop
% The length of the arrows can be either:
%       1) raw (ddr)
%       2) normalized (ddrn).
% If the normalized length is chosen they will automatically be scaled by a
% factor of 6.

% Example: models = strvcat('ww2',aw27','bw1','aw27');
%          [ww2,aw27,bw1,aw27] = ddr_outfiles (models, 2, 'ddr')

%--------------------------------------------------------------------------

% regions of PDZ domain
backbone = ['N' 'CA' 'O'];
pep_bind_pocket_res = [3:7,315:330,371:385];
carb_bind_loop_res = [315:325];

% arrow parameters
theta = pi/10;
phi = 0;
maxarrow = 0.6; minarrow = 0.05;
arr_red = 0.0; arr_green = 0.0; arr_blue = 1.0;     % colors range from 0 to 1

if strcmp(raw_or_norm,'ddr')
    scale = 1;
elseif strcmp(raw_or_norm,'ddrn')
    scale = 6;
end

% obtain structures in structure arrays using loadmodel function.  Always
% read in ww2 structure since .txt file is printed out using these labels.
ww2temp = loadmodel('models/ww2_12.pdb','ww2'); disp('ww2temp loaded')

model_list = cellstr(model_list);
files_in_models_folder = dir('models');

for modelnum = 1:size(model_list,1)
    for filenum = 1:size(files_in_models_folder,1)
        if findstr(char(model_list(modelnum)), files_in_models_folder(filenum).name)
            full_filename = ['models/' files_in_models_folder(filenum).name];
            switch modelnum
                case 1
                    str1 = loadmodel(full_filename, model_list(1)); disp([char(model_list(1)) ' loaded']);
                case 2
                    str2 = loadmodel(full_filename, model_list(2)); disp([char(model_list(2)) ' loaded']);
                case 3
                    str3 = loadmodel(full_filename, model_list(3)); disp([char(model_list(3)) ' loaded']);
                case 4
                    str4 = loadmodel(full_filename, model_list(4)); disp([char(model_list(4)) ' loaded']);
            end %switch
        end %if
    end %for filenum
end %for modelnum


% Calculate difference with dr.m function: str1 --> str2
[str1,str2,str3,str4] = ddr(str1,str2,str3,str4,0);
disp(['calculated ddr of: ' char(model_list(1)) ', ' char(model_list(2)),', ' char(model_list(3)),' and
',char(model_list(4))])

% open .txt and .py files as txt_file and py_file, respectively
txt_file = [char(model_list(1)) '_' char(model_list(2)) '_' char(model_list(3)) '_' char(model_list(4)) '.txt'];
if exist (txt_file,'file')
    delete (txt_file); disp ([txt_file ' overwritten'])
end
fid = fopen(txt_file,'w');

switch arrow_zone
    case 1
        py_arrows = 'all';
    case 2
        py_arrows = 'pbp';
    case 3
        py_arrows = 'cbl';
end
py_file = [char(model_list(1)) '_' char(model_list(4)) '_' py_arrows '_' raw_or_norm '.py'];
if exist (py_file,'file')
    delete (py_file); disp ([py_file ' overwritten'])
end
fid2 = fopen(py_file,'w');

% Begin writing python script
fprintf (fid2, 'from pymol.cgo import *\nfrom pymol import cmd\nimport math\nimport whrandom\n\n');
fprintf (fid2, 'obj = []\n\n');
fprintf (fid2, 'obj.extend([ LINEWIDTH, 0.5])\n');
```

```
total_pro_pep = size(find(strcmp(ww2temp.chainid,'A')),1) + size(find(strcmp(ww2temp.chainid,'P')),1);

for n = 1:total_pro_pep
    % to the text file, write: ww2temp.label and corresponding str1.dr and str1.drn.
    index = find(strcmp(ww2temp.label(n),str1.label));
    if index
        fprintf (fid, '%s \t %f \t %f\n', char(ww2temp.label(n)), str1.ddr(index), str1.ddrn(index));
    else
        fprintf (fid, '%s \t 0 \n', char(ww2temp.label(n)));
    end

    % if the atom is in both ww2temp and str1 and the displacement is
    % nonzero then calculate coordinates of the arrow lines

    if index & (str1.ddr(index)~=0)
        % determine coordinates for end of arrow.
        x2 = str1.x(index) + str1.(raw_or_norm)(index)/scale;
        y2 = str1.y(index) + str1.(raw_or_norm)(index)/scale;
        z2 = str1.z(index) + str1.(raw_or_norm)(index)/scale;

        % determines coordinates for ends of arrowhead lines
            % first find theta and phi for spherical representation of second
            % point relative to first point.
        [th,ph,length] = cart2sph(x2-str1.x(index), y2-str1.y(index), z2-str1.z(index));
            % length of arrow determined by sigmoidal formula (min and max
            % defined above).
        arrow_length = minarrow + (maxarrow-minarrow)/(1+10^(log10(500000)-str1.ddr(index))^0.3);
            % find coordinates of arrows based on length, theta, and phi (above)
        [rx, ry, rz] = sph2cart(theta+th, phi+ph, arrow_length);
        arrow1x = x2 - rx; arrow1y = y2 - ry; arrow1z = z2 - rz;
        [rx, ry, rz] = sph2cart((-1)*theta+th, phi+ph, arrow_length);
        arrow2x = x2 - rx; arrow2y = y2 - ry; arrow2z = z2 - rz;

        % write the pymol line commands to file in appropriate conditions
        switch arrow_zone
            case 1
                % if all is selected then write lines for every atom with nonzero
                % displacement
                if str1.ddr(index)~=0
                    % main line
                    fprintf (fid2, 'obj.extend([ COLOR, %2.1f, %2.1f, %2.1f])\n', arr_red, arr_green, arr_blue);
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', str1.x(index), str1.y(index),
str1.z(index));
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.append (END)\n');
                    % arrowhead 1
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow1x, arrow1y, arrow1z);
                    fprintf (fid2, 'obj.append (END)\n');
                    % arrowhead 2
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow2x, arrow2y, arrow2z);
                    fprintf (fid2, 'obj.append (END)\n');
                end
            case 2
                % if only Calpha of pep_bind_pocket is selected then only write
                % lines for backbone atoms in this region
                if str1.ddr(index)~=0 & findstr(backbone,char(ww2temp.atomid(n))) &
find(pep_bind_pocket_res==str1.resnum(index))
                    fprintf (fid2, 'obj.extend([ COLOR, %2.1f, %2.1f, %2.1f])\n', arr_red, arr_green, arr_blue);
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', str1.x(index), str1.y(index),
str1.z(index));
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.append (END)\n');

                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow1x, arrow1y, arrow1z);
                    fprintf (fid2, 'obj.append (END)\n');

                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow2x, arrow2y, arrow2z);
                    fprintf (fid2, 'obj.append (END)\n');
                end
            case 3
                % if only Calpha of carb_bind_loop is selected then only write
                % lines for backbone atoms in this region
                if str1.ddr(index)~=0 & findstr(backbone,char(ww2temp.atomid(n))) &
find(carb_bind_loop_res==str1.resnum(index))
                    fprintf (fid2, 'obj.extend([ COLOR, %2.1f, %2.1f, %2.1f])\n', arr_red, arr_green, arr_blue);
                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', ww2.x(n), ww2.y(n), ww2.z(n));
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.append (END)\n');

                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow1x, arrow1y, arrow1z);
                    fprintf (fid2, 'obj.append (END)\n');

                    fprintf (fid2, 'obj.extend([ BEGIN, LINE_STRIP])\n');
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', x2, y2, z2);
```

```
                    fprintf (fid2, 'obj.extend([ VERTEX, %3.2f, %3.2f, %3.2f])\n', arrow2x, arrow2y, arrow2z);
                    fprintf (fid2, 'obj.append (END)\n');
                end
        end     % of switch
    end     % of if
end

obj_name = [char(model_list(1)) '_' char(model_list(4)) '_' py_arrows '_' raw_or_norm];
fprintf (fid2, 'cmd.load_cgo(obj,''%s'')',obj_name);

fclose(fid); fclose(fid2);
```

```
function bargraph (model,field,title_on,cutoff)
% Makes a bargraph of a specified field in a model.  The x-axis labels are
% the atom labels.  The zoom is left so only x axis zooms.

% Options:  If title_on is 1 then the title, y axis label, and data sets
%               used in the figure are displayed.
%           If a non-zero cutoff is passed then the non-zero elements (in a
%               sorted list) in the field are tested by the lillietest.  The
%               list that passes the test is used to determine a mean and std dev.
%               If the list does not pass the test then a mean and std dev is
%               calculated for the entire non-zero list.

fig1_pos = [50 150 800 350];                                        % 'Position' of bargraph
fig2_pos = [100 100 250 250];                                        % 'Position' of histogram

figure('Position',fig1_pos);                                         % makes figure
b = bar(model.(field));                                              % makes bar graph
set(b,'FaceColor',[0 0 0.5],'EdgeColor',[0.0 0.0 0.5]);

y=max(model.(field))/(30);                                           % Defines normalized unit that will be used to place
text in bargraph

% set(gca,'XTick',[1:max(model.atomnum)]);                           % set ticks and ticklabels
set (gca,'XTick',[]);
set(gca,'XTickLabel','');

[chainidA,blah] = size(find(strcmp(model.chainid,'A')));
[chainidP,blah] = size(find(strcmp(model.chainid,'P')));             % Calculated the total number of atoms to be plotted
(includes
atoms = chainidA + chainidP;                                         %   alternate conformations and atoms with occ = 0).

text((atoms-100),(y*(31)),datestr(now),'Fontsize',6);                % displays the DATE

for n = 2:atoms
    text(n,(-1)*y,model.atomid(n),'FontSize',5.5);                   % displays the ATOMID of all atoms
    if model.resnum(n)~=model.resnum(n-1)                            % displays the residue NUMBER only if a new one
        text(n,(-2)*y,strcat(model.res(n),num2str(model.resnum(n))),'FontSize',8)
    end
end

if chainidP>0                                                        % if there are peptide atoms then label the xaxis
with protein and peptide
    text (1,(-3*y),'Protein','Fontsize',7);
    firstpep = min(find(strcmp(model.chainid,'P')));
    text (firstpep,(-3*y),'Peptide','Fontsize',7);
end

%----------if the title_on flag is 1 then show: title, ylabel, dataset info
if title_on

    if strcmp(field,'dr')|strcmp(field,'drn')                        % figure out if analysis being show involves 2 or 4
structures
        fieldsets = 'dr_sets';
        numsets = 2;
    elseif strcmp(field,'ddr')|strcmp(field,'ddrn')
        fieldsets = 'ddr_sets';
        numsets = 4;
    end

    [setname protein synch res comp rfact lastr mos length refl atomnum rfr r]=textread('datasets.txt',...
        '%s  %s      %s    %f  %f   %f    %f    %f  %f     %d   %d      %f  %f','headerlines',1);

    for setnum = 1:numsets                                           % for each data set determine the corresponding
protein structure
        ind = find(strcmp(setname,model.(fieldsets)(setnum)));
        name(setnum) = setname(ind);                                 %   setname
        pro(setnum) = protein(ind);                                  %   protein name
        synchrotron(setnum) = synch(ind);                            %   synchrotron
        reflections(setnum) = refl(ind);                             %   # reflections
        resolution(setnum) = res(ind);                               %   resolution
        rfactor(setnum) = rfact(ind);                                %   r factor
        mosaicity(setnum) = mos(ind);                                %   mosaicity
        celllength(setnum) = length(ind);                            %   unit cell length
        rfree(setnum) = rfr(ind);                                    %   R free
        rother(setnum) = r(ind);                                     %   R
    end

    switch numsets                                                   % display information about DATASETS (with arrows)
        case 2
            text (20,(y*28),model.(fieldsets)(1), 'Fontsize',8,'Fontweight','bold');       % structure 1
            text (23,(y*27),[num2str(model.resolution) 'A'], 'Fontsize',7);
            text (23,(y*26),[num2str(model.rfr) ' / ' num2str(model.r)],'Fontsize',7);
```

```
            text (70,(y*(28.2)),'\rightarrow','Fontweight','bold');                         % -->

            text (110,(y*28),model.(fieldsets)(2),'Fontsize',8,'Fontweight','bold');        % structure 2
            text (113,(y*27),[num2str(resolution(2)) 'A'], 'Fontsize',7);
            text (113,(y*26),[num2str(rfree(2)) ' / ' num2str(rother(2))],'Fontsize',7);

            text (30,(y*24),model.drmsg, 'Fontsize',6.5,'Fontangle','italic');              % alt conf message

        case 4
            text (20,(y*28),model.(fieldsets)(1), 'Fontsize',8,'Fontweight','bold');        % structure 1
            text (23,(y*27),[num2str(model.resolution) 'A'],'Fontsize',7);
            text (23,(y*26),[num2str(model.rfr) ' / ' num2str(model.r)],'Fontsize',7);

            text (110,(y*28),model.(fieldsets)(2),'Fontsize',8,'Fontweight','bold');        % structure 2
            text (113,(y*27),[num2str(resolution(2)) 'A'],'Fontsize',7);
            text (113,(y*26),[num2str(rfree(2)) ' / ' num2str(rother(2))],'Fontsize',7);

            text (20,(y*23),model.(fieldsets)(3),'Fontsize',8,'Fontweight','bold');         % structure 3
            text (23,(y*22),[num2str(resolution(3)) 'A'],'Fontsize',7);
            text (23,(y*21),[num2str(rfree(3)) ' / ' num2str(rother(3))],'Fontsize',7);

            text (110,(y*23),model.(fieldsets)(4),'Fontsize',8,'Fontweight','bold');        % structure 4
            text (113,(y*22),[num2str(resolution(4)) 'A'],'Fontsize',7);
            text (113,(y*21),[num2str(rfree(4)) ' / ' num2str(rother(4))],'Fontsize',7);

            text (70,(y*(28.2)),'\rightarrow'); text (70,(y*(23.2)),'\rightarrow');         % four arrows
            text (30,(y*(24.5)),'\downarrow'); text(120,(y*(24.5)),'\downarrow');

            text (30,(y*19),model.ddrmsg, 'Fontsize',6.5,'Fontangle','italic');             % alt conf message
    end


    switch field                                                % show appropriate title
        case 'dr'
            text(10,(y*(31)),[char(model.protein) '  \rightarrow  ' char(pro(2)) ':
\Deltar_{raw}'],'Fontsize',10,'Fontweight','bold');
            ylabel('raw displacement (A)');
        case 'drn'
            text(10,(y*(31)),[char(model.protein) '   \rightarrow   ' char(pro(2)) ':   \Deltar_{norm}'],
'Fontsize',10,'Fontweight','bold');
            ylabel('normalized displacement (\sigma)');
        case 'ddr'
            text(10,(y*(31)),[char(model.protein) ', ' char(pro(2)) ', ' char(pro(3)) ', ' char(pro(4)) ':
\Delta\Deltar_{raw}'],...
                'Fontsize',10,'Fontweight','bold');
            ylabel('raw coupling (A)')
        case 'ddrn'
            text(10,(y*(31)),[char(pro(1)) ', ' char(pro(2)) ', ' char(pro(3)) ', ' char(pro(4)) ':
\Delta\Deltar_{norm}'],...
                'Fontsize',10, 'Fontweight','bold', 'Fontname','Helvetica');
            ylabel('normalized coupling (\sigma)');
    end
end
axis tight;


%---------- do lillietest calculation and display peaks above specified cutoff
if cutoff > 0
    sorted_nonzero = sort(model.(field)(find(model.(field)~=0)));
    [rows cols] = size(sorted_nonzero);
    for x = rows:-1:1
        if lillietest(sorted_nonzero(1:x),0.01) == 0
            x;
            mean_field_nonzero = mean(sorted_nonzero(1:x));
            std_field_nonzero = std(sorted_nonzero(1:x));
            nrml_dist = 1;
            break
        end
        if x == 1
            nrml_dist = 0;
            mean_field_nonzero = mean(sorted_nonzero);
            std_field_nonzero = std(sorted_nonzero);
        end
    end

    cutoff_2 = mean_field_nonzero + std_field_nonzero*cutoff;
    cutoff_line(1:atoms) = cutoff_2;
    mean_line(1:atoms) = mean_field_nonzero;
    hold;
    plot(mean_line,'-','Color',[0.5 0.5 0.5]);
    plot(cutoff_line,':','Color',[0.5 0.5 0.5]);hold off;

    peaks = find(model.(field)>(cutoff_2));
    [numpeaks blah] = size(peaks);
    listlimit = atoms - ceil((numpeaks+1)/15) * 100 - 15;

    if ~(nrml_dist)
        text((listlimit-110),(y*25),'not normal','Fontsize',6.5,'Fontangle','italic')
    end
    text ((listlimit-110),(y*(28)),['mean: ' num2str(mean_field_nonzero,3)],'Fontsize',6.5);
    text ((listlimit-110),(y*(27)),['std dev: ' num2str(std_field_nonzero,3)],'Fontsize',6.5);
    text ((listlimit-110),(y*(26)),['cutoff: ' num2str(cutoff) '\sigma'],'Fontsize',6.5);
    text (listlimit,(y*(28)),['Peaks > ' num2str(cutoff_2,3)],...
        'Fontsize',6.5,'Fontangle','italic');

    for p = 1:numpeaks
```

```
        if (model.(field)(peaks(p)) + y) < max(model.(field))
            text (peaks(p),(model.(field)(peaks(p))+y),num2str(p),'Fontsize',5.5);
        else
            text (peaks(p)+2,(model.(field)(peaks(p))-y),num2str(p),'Fontsize',5.5);
        end
        text ((listlimit +(100*fix(p/15))),(y*(28 - mod(p,15))),[num2str(p) '. ' char(model.label(peaks(p))) ', ' ...
                num2str(model.(field)(peaks(p)),3)],'Fontsize',5.5, 'Color',[0.4 0.4 0.4]);
    end

    %---------- MAKE HISTOGRAM in second figure window

    figure('Position',fig2_pos);
    hist(model.(field)(find(model.(field)~=0)),50);
    hold on;
    bars = findobj(gca,'Type','patch');
    set(bars,'FaceColor',[0.0 0.0 0.5],'EdgeColor',[0 0 0.5]);
    axis tight;

    ylabel ('# atoms');
    switch field                                                    % show appropriate title and x label
        case 'dr'
            title([char(model.protein) '  \rightarrow  ' char(pro(2)) ':
\Deltar_{raw}'],'Fontsize',10,'Fontweight','bold');
            xlabel ('bins (A)');
        case 'drn'
            title([char(model.protein) '   \rightarrow   ' char(pro(2)) ':   \Deltar_{norm}'],
'Fontsize',10,'Fontweight','bold');
            xlabel ('bins (\sigma)');
        case 'ddr'
            title([char(model.protein) ', ' char(pro(2)) ', ' char(pro(3)) ', ' char(pro(4)) ':
\Delta\Deltar_{raw}'],...
                'Fontsize',10,'Fontweight','bold');
            xlabel ('bins (A)');
        case 'ddrn'
            title([char(protein(1)) ', ' char(pro(2)) ', ' char(pro(3)) ', ' char(pro(4)) ':   \Delta\Deltar_{norm}'],...
                'Fontsize',10,'Fontweight','bold');
            xlabel ('bins (\sigma)');
    end

    [numpeaks, barcenters] = hist(model.(field)(find(model.(field)~=0)),50);    % add lines and text for mean, cutoff_2
    plot (mean_field_nonzero,1:max(numpeaks),'-','Color',[0.6 0.6 0.6]);
    plot (cutoff_2,1:max(numpeaks),':','Color',[0.3 0.3 0.3]);

    unity2 = max(numpeaks)/30;
    unitx2 = max(barcenters)/30;                                    % define units of figure 2.

    text (unitx2*15, unity2*28, 'based on non-zero peaks', 'Fontsize', 6.5, 'Fontangle', 'italic');
    if numsets == 2
        text (unitx2*15, unity2*26, model.drmsg, 'Fontsize', 6.5, 'Fontangle', 'italic');
    elseif numsets == 4
        text (unitx2*15, unity2*26, model.ddrmsg, 'Fontsize', 6.5, 'Fontangle', 'italic');
    end
    text (mean_field_nonzero + unitx2/2, unity2*27, ['mean:  ' num2str(mean_field_nonzero,3)], 'Fontsize', 8);
    text (mean_field_nonzero + unitx2/2, unity2*25.5, ['std dev:  ' num2str(std_field_nonzero,3)], 'Fontsize', 8);
    text (cutoff_2 + unitx2/2, unity2*23,['cutoff:  ' num2str(cutoff) '\sigma =  ' num2str(cutoff_2,3)],'Fontsize',8);

    low = min(model.(field)(find(model.(field)~=0)));                       % label bars over cutoff
    high = max(model.(field));
    [numpeaks, barcenters] = hist(model.(field)(find(model.(field)~=0)),50);    % add lines and text for mean, cutoff_2

    barwidth = (high-low)/50;
    barsovercutoff = find(barcenters > cutoff_2);
    count = 1;

    for w = min(barsovercutoff):max(barsovercutoff)                         % label all other bars
        count = count + 0.8;
        for z = 1:numpeaks(w)
            bar_low = barcenters(w) - barwidth/2;
            bar_high = barcenters(w) + barwidth/2;
            if w==max(barsovercutoff)
                barlabels = model.label2(find((model.(field)>bar_low) & (model.(field)<=max(model.(field)))));
            else
                barlabels = model.label2(find((model.(field)>bar_low) & (model.(field)<=bar_high)));
            end
            [num_barlabels blah] = size(barlabels);
            for u = 1:num_barlabels
                ylabel_loc = unity2*6*(mod(count,3)+1);
                plot((barcenters(w)-barwidth/2), numpeaks(w):(ylabel_loc-unity2),':','Color',[0.8 0.8 0.8]);
                if w == max(barsovercutoff)
                    text(barcenters(w)-unitx2*2, ylabel_loc-unity2*u, barlabels(u),'Fontsize',5, 'Color', [0.4 0.4 0.4]);
                else
                    text(barcenters(w), ylabel_loc-unity2*u, barlabels(u),'Fontsize',5, 'Color', [0.5 0.5 0.5]);
                end
            end
        end
    end
end

hold off;

vectorplot (model,field,0);                                 % Make vector plot with only str1 Calpha trace shown

figure(1);
zoom xon;
axis tight;
```

```
function vectorplot (model,field,scale)
% Draws a C-alpha trace of the model and the vectors for the specified field.
% Input:  model and a field.  The field is one of four:
%           1) dr
%           2) drn
%           3) ddr
%           4) ddrn
% Ouput: a new figure window with a C-alpha trace of the molecule and a
%           a quiver3 plot.

figure ('Position',[50 100 800 750]); hold on;
set(gca,'XTick',[]);                            % set ticks and ticklabels
set(gca,'XTickLabel',[]);
set(gca,'YTick',[]);
set(gca,'YTickLabel',[]);
set(gca,'ZTick',[]);
set(gca,'ZTickLabel',[]);


prot_ca_ind = find(strcmp(model.atomid,'CA')&(strcmp(model.chainid,'A')));
pep_ca_ind = find(strcmp(model.atomid,'CA')&(strcmp(model.chainid,'P')));

% plot vectors only for protein or peptide atoms that have nonzero value for field
% ind = find(strcmp(model.chainid,'A')|strcmp(model.chainid,'P'));
ind = find(model.(field)~=0);

plot3 (model.x(prot_ca_ind), model.y(prot_ca_ind), model.z(prot_ca_ind),'Linewidth',3,'Color',[0.7 0.7 0.7]);
plot3 (model.x(pep_ca_ind), model.y(pep_ca_ind), model.z(pep_ca_ind),'Linewidth',3,'Color',[0.8 0.8 0.8]);

switch field
    case 'dr'
        quiver3 (model.x(ind), model.y(ind), model.z(ind), model.dx(ind), model.dy(ind), model.dz(ind),scale);
    case 'drn'
        sc = 10/80;
%         sc = 75.19/max(model.drn);
        model.dxnsc = model.dxn;
        model.dxnsc = model.dxn*sc; model.dynsc = model.dyn*sc; model.dznsc = model.dzn*sc;
        quiver3 (model.x(ind), model.y(ind), model.z(ind), model.dxnsc(ind) , model.dynsc(ind), model.dznsc(ind),scale);
    case 'ddr'
        quiver3 (model.x(ind), model.y(ind), model.z(ind), model.ddx(ind), model.ddy(ind), model.ddz(ind),scale);
    case 'ddrn'
        sc = 5/24;
        model.ddxnsc = model.ddxn*sc; model.ddynsc = model.ddyn*sc; model.ddznsc = model.ddzn*sc;
        quiver3 (model.x(ind), model.y(ind), model.z(ind), model.ddxnsc(ind), model.ddynsc(ind),
model.ddznsc(ind),scale);
end

if strcmp(field,'dr')|strcmp(field,'drn')                       % figure out if analysis being show involves 2 or 4
structures
    fieldsets = 'dr_sets';
    numsets = 2;
elseif strcmp(field,'ddr')|strcmp(field,'ddrn')
    fieldsets = 'ddr_sets';
    numsets = 4;
end

[setname protein synch res comp rfact lastr mos length refl atomnum rfr r]=textread('datasets.txt',...
    '%s  %s      %s    %f  %f   %f    %f    %f  %f     %d   %d      %f  %f','headerlines',1);

for setnum = 1:numsets                                          % for each data set determine the corresponding protein
structure
    ind = find(strcmp(setname,model.(fieldsets)(setnum)));
    name(setnum) = setname(ind);                                %   setname
    pro(setnum) = protein(ind);                                 %   protein name
end

switch field                                                        % show appropriate title
    case 'dr'
        title([char(model.protein) '  \rightarrow  ' char(pro(2)) ':
\Deltar_{raw}'],'Fontsize',10,'Fontweight','bold');
    case 'drn'
        title([char(model.protein) '   \rightarrow   ' char(pro(2)) ':   \Deltar_{norm}'],
'Fontsize',10,'Fontweight','bold');
    case 'ddr'
        title([char(model.protein) ', ' char(pro(2)) ', ' char(pro(3)) ', ' char(pro(4)) ':   \Delta\Deltar_{raw}'],...
            'Fontsize',10,'Fontweight','bold');
    case 'ddrn'
        title([char(protein(1)) ', ' char(pro(2)) ', ' char(pro(3)) ', ' char(pro(4)) ':   \Delta\Deltar_{norm}']);
end
view(-1,-90)
hold off;
```

```
function makepdb (str, field, filename)
% This data accepts 1) the data for a model in the form of a structure, 2)
% a field, and 3) the output filename.  It writes out a filename.pdb with
% the B factor column replaced by the data in the specified field.

if exist (filename,'file')
    delete (filename);
    disp ([filename ' overwritten.'])
end

fid = fopen(filename,'w');
fprintf (fid, 'REMARK Written by MATLAB on %s\n',datestr(now));
fprintf (fid, 'REMARK B factor column has been replaced by %s \n',field);
```

```
if findstr('dr',field)
    if findstr('ddr',field)                                          % determine which field and how many sets involved
        fieldsets = 'ddr_sets';numsets = 4;
    elseif findstr ('dr',field)
        fieldsets = 'dr_sets'; numsets = 2;
    end

    [setname protein synch res comp rfact lastr mos length refl atomnum rfr r]=textread('datasets.txt',...
        '%s  %s      %s    %f  %f   %f    %f    %f  %f     %d   %d      %f  %f','headerlines',1);

    for setnum = 1:numsets                                           % for each data set determine the corresponding
protein structure
        ind = find(strcmp(setname,str.(fieldsets)(setnum)));
        set(setnum) = setname(ind);                                  %   setname
        pro(setnum) = protein(ind);                                  %   protein name
        synchrotron(setnum) = synch(ind);                            %   synchrotron
        reflections(setnum) = refl(ind);                             %   # reflections
        resolution(setnum) = res(ind);                               %   resolution
        rfactor(setnum) = rfact(ind);                                %   r factor
        mosaicity(setnum) = mos(ind);                                %   mosaicity
        celllength(setnum) = length(ind);                            %   unit cell length
        rfree(setnum) = rfr(ind);                                    %   R free
        rother(setnum) = r(ind);                                     %   R
    end

                                                                     % Write out parameters: dataset, protein,
resolution ,Rfr/R
    fprintf (fid,'REMARK The data sets used for this calculation were: \n');
    fprintf (fid,'REMARK \t\tset    \tpro  \t\t\tres    Rfr\tR\n');
    for x = 1:numsets
        fprintf (fid,'REMARK \t\t%s \t%s   \t\t%4.2f    %3.1f / %3.1f\n',char(set(x)), char(pro(x)),resolution(x),
rfree(x), rother(x));
    end
end

for n = 1:str.totalatoms1                                            % write out ATOM lines: should execute even if
writing b factors
    switch char(str.chainid(n))
        case {'A','P'}
            if strcmp(str.ac(n),'A')|strcmp(str.ac(n),'B')       % if the line has an alternate conformation include
the ac
                fprintf (fid, 'ATOM  %5d  %-3s%s%3s %s%4d    %8.3f%8.3f%8.3f%6.2f%6.2f      %-4s\n', n,
char(str.atomid(n)), char(str.ac(n)), ...
                    char(str.res(n)), char(str.chainid(n)), str.resnum(n), str.x(n), str.y(n), str.z(n), str.occ(n),
str.(field)(n),char(str.segid(n)));
            else                        % if the line doesn't have an ac, then just print everything else.
                fprintf (fid, 'ATOM  %5d  %-3s %3s %s%4d    %8.3f%8.3f%8.3f%6.2f%6.2f      %-4s\n', n,
char(str.atomid(n)), ...
                    char(str.res(n)), char(str.chainid(n)), str.resnum(n), str.x(n), str.y(n), str.z(n), str.occ(n),
str.(field)(n),char(str.segid(n)));
            end
        case 'W'
            fprintf (fid, 'ATOM  %5d  %-3s %3s %s%4d    %8.3f%8.3f%8.3f%6.2f%6.2f      %-4s\n', n,
char(str.atomid(n)), ...
                char(str.res(n)), char(str.chainid(n)), str.resnum(n), str.x(n), str.y(n), str.z(n), str.occ(n),
str.bfactor(n),char(str.segid(n)));
    end
end
fprintf (fid, 'END');

status = fclose(fid);
```

```
function [poserr] = stroud(B,atoms,refl)
% This function calculates the positional error from the Stroud-Fauman
% formula (Protein Science, 1995, Vol 4, pp. 2392-2404), given the B
% factor, number of atoms, and the number of reflections.

a=atoms/refl;

p3 = 10*((epsilon(20,a)-epsilon(10,a))/(epsilon(30,a)-epsilon(20,a)));
p2 = (epsilon(20,a)-epsilon(10,a))/(exp(20/p3)-exp(10/p3));
p1 = epsilon(20,a)-p2*exp(20/p3);
poserr = p1 + p2*exp(B/p3);

function [epsi] = epsilon(B,a)
% values of k1-6 are those determined by Rama and I by fitting published
% curves with published formulas.
k = [-0.7238 -3.317e-5 3.6284 0.66709 0.0098103 9.9735];
slope = k(1) + k(2)*exp(B/k(3));
int = k(4) + k(5)*exp(B/k(6));
epsi = int + slope*exp(-2*a);
```

Vitae

Rohit Sharma was born on October 28, 1974 in Ludhiana, India. His parents, Ramesh and Savita Sharma, immigrated to the United States when he was two years old. His family which includes his sister, Neha, lived in Texas, Michigan, and Connecticut before moving to Fort Worth, Texas where he completed high school. Rohit attended the Massachusetts Institute of Technology where he attained a B.S. in Biology (1996). Afterwards, he gained admission in the M.D./Ph.D. program at the University of Texas Southwestern Medical Center in Dallas, Texas where he completed the work discussed in this dissertation. He and his wife, Charu Puri, have one daughter, Anisa Sharma.

Permanent Address: 2620 Country Creek Ln.
Fort Worth, TX 76123