LOCAL MOTION AND LOCAL ACCURACY

IN PROTEIN BACKBONE

by

Ian W. Davis

Department of Biochemistry
Duke University

Date: _____
Approved:

_____
David C. Richardson, Supervisor

_____
Jane S. Richardson, Supervisor

_____

_____

_____

_____

Dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy in
the Department of Biochemistry in
the Graduate School of Duke University

2006

ABSTRACT

LOCAL MOTION AND LOCAL ACCURACY

IN PROTEIN BACKBONE

by

Ian W. Davis

Department of Biochemistry
Duke University

Date: _____
Approved:

_____
David C. Richardson, Supervisor

_____
Jane S. Richardson, Supervisor

_____

_____

_____

_____

An abstract of a dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy in
the Department of Biochemistry
in the Graduate School of Duke University

2006

Abstract

Proteins are chemically simple molecules, being unbranched polymers of uncomplicated organic compounds. Nonetheless, they fold up into a dazzling variety of complex and beautiful configurations with a dizzying array of structural, regulatory, and catalytic functions. Despite great progress, we still have very limited ability to predict the folded conformation of an amino acid sequence, and limited understanding of its dynamics and motions. Thus, this work presents a quartet of interrelated studies that address some aspects of the detailed local conformations and motions of protein backbone.

First, I used a density-dependent smoothing algorithm and a high-quality, *B*-filtered data set to construct highly accurate conformational distributions for protein backbone (Ramachandran plots) and sidechains (rotamers). These distributions are the most accurate and restrictive produced to date, with improved discrimination between rare-but-real conformations and artifactual ones.

Second, I analyzed hundreds of alternate conformations in atomic resolution crystal structures, and discovered that dramatic conformational change in a

protein sidechain is often coupled to a subtle but very common mode of conformational change in its backbone -- the backrub motion. Examination of other biophysical data further supports the ubiquity of this motion.

Third, I applied a model of backrub motion to protein design calculations. Although experimental characterization of the designs showed them to be unstable and/or inactive, the computational results proved to be very sensitive to changes in the backbone.

Finally, I describe how MolProbity uses my conformational distributions together with all-atom contacts and other tools to validate protein structures, and how those quality metrics can be combined visually or analytically to provide "multi-criterion" validation summaries.

For Katy

Table of Contents

List of tables

List of figures

List of symbols and abbreviations

3-D	Three dimensional
Å	Ångstroms (1Å = 10^{-10} meter)
API	Application programming interface
B	Crystallographic temperature factor
CD	Circular dichroism
CS	Computer science
DEE	Dead end elimination (an algorithm)
GBP	Glucose binding protein
GMEC	Global minimum energy conformation
GUI	Graphical user interface
H-bond	Hydrogen bond
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
k	The Boltzmann constant

KiNG	Kinemage, Next Generation (a computer program)
LB	Luria broth
mmCIF	Macromolecular Crystallographic Information File
NMR	Nuclear magnetic resonance
PDB	Protein Data Bank
PHP	PHP Hypertext Preprocessor (a programming language)
pre-Pro	Any amino acid immediately preceding a proline
R	The gas constant
RNA	Ribonucleic acid
T	Temperature
T_m	Melting temperature (e.g., of a protein)
SECSG	SouthEast Collaboratory for Structural Genomics
ϕ	A protein backbone torsion angle, C_{i-1}-N-Cα-C
ψ	A protein backbone torsion angle, N-Cα-C-N_{i+1}
τ	A protein backbone bond angle, N-Cα-C
$\chi 1$ - $\chi 4$	Protein sidechain torsion angles

Acknowledgements

I am deeply indebted to many people, both personally and scientifically, for their help and support over the past five years and beyond. I cannot thank them enough, and cannot mention them all, but I will try my best. To anyone I left out, your contributions may have done undocumented, but they did not go unappreciated!

My education has been generously supported by several institutions, and I would not have been able to do this work otherwise. I thank Vanderbilt University, Duke University, and the Howard Hughes Medical Institute for the scholarships they have given me, and I thank the National Institutes of Health for supporting the research program of the Richardson laboratory.

I have been very fortunate to speak and collaborate with brilliant researchers from around the world. I thank Wolfram Tempel and B.C. Wang for SECSG collaboration on using MOLPROBITY and the BACKRUB tool in structure improvement, Gerard Kleywegt for providing F_o-F_c as well as $2F_o$-F_c maps on the Electron Density Server site, and all the crystallographers who deposited their high-resolution structures and data. Scott Schmidler provided

invaluable assistance on the statistical analysis for the MolProbity score, and Rachael Brady was a font of visualization knowledge. I also thank Jack Snoeyink and his students for collaboration on inverse kinematics of protein backbone, and Phil Bourne and his group for letting us suggest PDB features and for featuring KiNG there.

I have had an excellent committee of professors offer me guidance and advice as I progressed through this Ph.D. program, and am grateful to all of them: Dan Gewirth, Homme Hellinga, Jan Hermans, Terry Oas, and Johannes Rudolph. I am especially grateful to Professor Hellinga for allowing me to do the protein design work in his laboratory.

My fellow Duke graduate students have made my time here a pleasure, and I would not have made it without their humor, friendship, and support. Several have also been collaborators: Jim Qiu and Mary Dwyer each spent months helping me through the wet chemistry of protein design; Loren Looger performed the initial BACKRUB design computations; and Brian Coggins has been a sounding board for many programming and visualization ideas.

Of particular note are the other graduate students in the Richardson lab, with whom I have been fortunate to work closely. Laura Murray and I entered the lab together, Jeremy Block and I have hashed through an endless array of research ideas, Vincent Chen has pushed and inspired me to write better and more innovative software, and Gary Kapral has tolerated my suggestions, questions, and quirks with good humor. I will miss all of them! Former students Mike Word and Simon Lovell also have my gratitude, for although I did not see them much in person, much of my work built upon theirs.

Other members of the Richardson group were also a joy and helped me get a real education. Bryan Arendall is one of the most thoughtful and curious people I know (yes Bryan, I mean both words both ways), and I admire his unflagging spirit. Michael Prisant has been generous with his time and computing knowledge on many occasions. And Lizbeth Videau has kept us all sane, focused, and our priorities in order, always with a joyful attitude, sometimes at risk to her personal sanity.

Dave and Jane Richardson are the best mad scientists I know, and the best mentors I could have asked for. They gave me both freedom and direction, and set an example and a standard that I will aspire to for the rest of my

career. They gave to me with the utmost generosity, and I stand deeply in their debt.

Of course, I am also tremendously indebted to my parents; they made it possible for me to come down this road, without pushing me down it, and they have supported me and believed in me the whole way. My father got me interested in both computers and the natural world at a young age, and my mother made me love learning and reading by her example. I hope this thesis will be a source of pride for them, as it is for me.

Finally, last but certainly not least, I thank my loving wife, Katy. Without her, I could not have done this, and it would not have been worth doing. With her, I know we can tackle any challenge and any adventure. I have been richly blessed, and I am grateful.

Chapter 1: Introduction

The theory and practice of x-ray crystallography were painstakingly built up over the first half of the 20th century, culminating in the late 1950's with the first structures of proteins -- of myoglobin by John Kendrew and of hemoglobin by Max Perutz, both at Cambridge. Even when only a handful of atomic models were available, investigators were studying the trends, patterns, and rules of structure that all proteins seemed to share (Ramachandran, Ramakrishnan et al. 1963; Ramakrishnan and Ramachandran 1965; Ramachandran and Sasisekharan 1968). Today, more than 37,000 macromolecular structures are known, with the coordinates centrally archived and available to the world (Berman, Westbrook et al. 2000). With this resource to draw on, new rules for protein structure are being discovered all the time, and yet we are unable to devise predictive models for many of their most basic features. Specifically, the "protein folding problem" is to accurately predict the 3-D configuration of a protein, given its amino acid sequence; the inverse problem (protein design) is to predict a sequence that specifically and stably adopts a specified conformation (and perhaps function). Although

progress has been made, both problems are far from being solved for the general case.

Proteins, of course, are the workhorses of the cell, performing the majority of catalytic, regulatory, and structural roles in modern organisms. As such, there is tremendous intellectual value in being able to examine them structurally, as miniature machines, to better understand their modes of action and interaction. More immediately, however, there is tremendous value to medicine in seeing proteins that are involved with specific diseases; many therapeutically important drugs specifically target a critical protein in some disease-relevant pathway. Thus, protein structures give us our first opportunity to create medicines in a fully rational way. Most major pharmaceutical companies now have significant structural biology programs, and structure-based drug discovery (SBDD) is seen as the way of the future.

Proteins, however deceptively simple they may be chemically, have layers of emergent properties and display enormous complexity. Thus, most real SBDD programs take a hybrid approach, with structural studies complimenting screening approaches (Jorgensen 2004). (In part, this is because we don't understand *drugs* well enough to accurately predict *their* properties either, such as absorption, distribution, metabolism, excretion, and

toxicity). The complexity of proteins, however, poses a serious barrier to solving the folding and design problems. For example, the Levinthal paradox says that exhaustively exploring all the possible conformations of even a small protein should take longer than the age of the Universe (Levinthal 1969). To access even a miniscule part of this conformational space, most investigators complement the judgment of human experts by relying heavily on computational algorithms.

One of the greatest sources of complexity in protein structure, and one of the major stumbling blocks for computation, is protein backbone. Backbone forms the major set of long-range, high-strength (i.e., covalently bonded) constraints on structure. As such, the conformation of backbone at one point may have long reaching effects on parts of the protein that are distant in both sequence and space. This stands in contrast to protein sidechains; although they often pack together in large, interdependent networks, the conformation of one is not *necessarily* linked to that of all the others. In fact, in the absence of backbone changes, sidechain interactions are approximately pairwise decomposable, which (with the appropriate algorithms) greatly simplifies the combinatorics of their conformations (Looger and Hellinga 2001). In general, however, moving any part of the backbone moves a large portion of the

remaining structure, such that many other atomic interactions must be reconsidered.

As a result, most computations around protein structure assume a single, fixed backbone conformation, usually based on an experimentally determined structure. Attempts to do otherwise in protein design have succeeded mostly for special cases only, as documented in Chapter 4. In attempts to predict new structures from their sequences based on our current database of sequence-structure pairs (homology modeling), the weakest results occur in loops, where the backbone conformation changes most quickly and radically (Martin, MacArthur et al. 1997; Venclovas, Zemla et al. 1997).

The work described here aims to improve our understanding of protein backbone, thereby enabling more accurate and efficient prediction of its behavior, with the long-term goal of finding generally applicable solutions to the protein folding and design problems. The core of this work is focused on the atomic-resolution three-dimensional structure of protein backbone at the scale of a few amino acid residues. From this core, the work also touches on a variety of other aspects of macromolecular structures, and on computational methods for exploiting them.

Specifically, I begin with a study of the allowed conformations for single residues in the tradition of Ramachandran (Ramachandran, Ramakrishnan et al. 1963), employing carefully quality-filtered data and sophisticated analysis to yield the most accurate and restrictive such description to date. I also address conceptually similar conformational distributions for protein sidechain and RNA backbone, both of which cluster into "rotamers". Next, I describe a survey of the highest resolution structures in the public database, which reveals a subtle but common and important mode of local change in backbone conformation (the "backrub" motion); additional evidence for this pattern is also documented from other sources of experimental data. I then relate the results of applying a model of backrub motion to the protein design problem, and describe the impact on both computational and *in vitro* biochemical characterizations. Finally, I return to the rotamer and Ramachandran distributions in the context of validating experimentally determined protein structures, and explore the impact of considering multiple criteria simultaneously when evaluating a structure. Additional information on the software developed for all of these studies is presented as an appendix.

Taken together, these studies show that careful scrutiny of the local conformations of protein backbone reveals many stringent constraints and at

least one common pattern of change, and suggest that both can be profitably leveraged by a variety of computational methods that tackle difficult problems rooted in protein structure. The aggregation of many such contributions by the wider community will hopefully lead to solutions to such problems, and in the long run, the resulting technologies promise us unprecedented control over the health of our own bodies and minds, and a deeper, more detailed, and more mechanistic understanding of the biological world.

Chapter 2: Torsion angle distributions

The three-dimensional structures of proteins and other macromolecules are almost entirely specified in the rotational degrees of freedom around covalent bonds, which are called torsion or dihedral angles. Ideal bond angle and length values are known from highly accurate small-molecule structures (Engh and Huber 1991; Engh and Huber 2002), and although bond angle variation can sometimes have significant consequences for accurate modeling (see Chapter 3 and (Bruccoleri and Karplus 1985)), it is nonetheless quite small by comparison.

The torsion angles of proteins and RNA have non-trivial distributions, such that some conformations are observed frequently and others, not at all. Furthermore, the angles generally have strong interdependence within the same residue and/or between neighbors, so one must study multi-dimensional distributions rather than studying each angle in isolation. For proteins, pairs of consecutive backbone dihedrals are linked, and all the dihedrals in a particular sidechain are linked. Sidechain and mainchain are mostly independent of each other, with certain exceptions. For RNA, the seven

dihedrals that connect neighboring bases are linked in a complex 7-D distribution (Murray, Arendall et al. 2003; Murray, Richardson et al. 2005).

The chemical forces that create torsion angle preferences in macromolecules are often sufficiently subtle that they are not captured by molecular mechanics forcefields; such calculations do a poor job of recapitulating the empirical distributions. Recent work has shown, however, that quantum mechanical simulations can produce much better agreement with the experimental data (Butterfoss and Hermans 2003; Hu, Elstner et al. 2003).

The work presented here is based on empirically determined protein structures rather than on theory. However, producing an accurate and reliable description of those torsion angle distributions requires considerable care, both in the selection and the processing of the data (detailed below). I first extracted the necessary data for creating the various distributions from the Richardson laboratory's Top500 database. Building on earlier work, I tested various criteria for quality filtering effectiveness, and adjusted the cutoff thresholds for an optimal balance of database size versus accuracy. I also developed a novel, two-pass, histogram-like method we call *density-dependent smoothing* for dealing with the large dynamic range found especially in the Ramachandran plots. Finally, I constructed software for using

these distributions as structure validation criteria in MOLPROBITY, which is treated in Appendix A.

Accurate limits on the conformations accessible to single residues provide a crucial reality check in thinking about conformational heterogeneity at larger scales, both for validating the conformational details of experimental structures and for constraining the algorithms during modeling or computational design exercises. Thus, this work forms the foundation for much of what follows in later chapters.

Database of structures

For all the torsion angle distributions described in this chapter, the data came from a carefully selected set of empirically determined protein structures. Selection occurred at two levels: the structures themselves were carefully screened, and then only high-accuracy regions of those structures were considered.

Structures for this study were taken from three sources: our previous database of 240 structures at 1.7 Å resolution or better (Lovell, Word et al. 2000), structures at 1.8 Å resolution or better from the 30% homology cutoff PDB Select list (Hobohm and Sander 1994), and new structures of 1.5 Å

resolution or better released from February to May, 2000. For closely related pairs of structures, the three lists were reconciled by using the one with the best combination of clashscore (number of van der Waals overlaps ≥ 0.4 Å per 1000 atoms (Word, Lovell et al. 1999a)) and resolution. If multiple identical chains were present, the first chain was chosen, unless the file header indicated that another was better ordered. Files with multiple, non-homologous chains were split only if each formed a separate compact unit. No NMR structures were used, because it is difficult to assess their accuracy in the first place and because very few such structures attain an accuracy comparable to the other structures in this database.

The list was further filtered by several additional criteria. Specifically, structures were rejected (1) if they had a clashscore ≥ 22 for those atoms with crystallographic B<40, (2) if they had a large number of distorted main chain bond angles (threshold defined as ≥10 main chain bond angles per 1000 atoms being ≥5 standard deviations from standard (Engh and Huber 1991) geometry), (3) if they had unusual amino acids with main chain substitutions (e.g., 1MRO, 1RTU), or (4) if they were subjected to free-atom refinement (e.g., 1NXB); each of these circumstances was rare. Wildtype structures were preferred to mutant if they were otherwise equivalent. We additionally

checked for large numbers of *B*-factors ≤1, which is an indication of the use of U^2 rather than *B* (which can be corrected by a constant term of $8\pi^2$), or of unrefined *B*-factors (which would be rejected); for this data set, however, none were found.

We named the resulting database the Top500; it contains 148 files from our previous database, 329 from the PDB Select list and 23 more recently solved files, giving a total of 500 structures and 109,799 residues (available at http://kinemage.biochem.duke.edu). All relevant data items are stored in and queried from MySQL database tables at the molecule level (e.g., PDB code, resolution) or the residue level (angles, *B*-factors, secondary structure). Dihedral angles were calculated with DANG (Word 2000) and secondary structure assignments made using the DSSP algorithm as modified in PROCHECK (Laskowski, Macarthur et al. 1993). Two- and three-dimensional visualizations of protein structures or of data plots were done interactively with kinemage graphics.

Various additional filters were applied at the residue level depending on whether the data were to be used for mainchain (Ramachandran, ψ-ϕ) or sidechain (rotamer) distributions. For the mainchain distributions, 1276 residues were eliminated because they are at chain termini and thus do not

have both ϕ and ψ defined. Residues were also eliminated if they had any mainchain atom with a crystallographic *B*-factor ≥ 30. A non-normalized cutoff on *B*-factor was used, for reasons previously discussed (Lovell, Word et al. 2000). Thus, 97,368 total residues appear in the final ϕ,ψ data set. For the sidechain distributions, methods are described in Lovell et al. (Lovell, Word et al. 2000) and included use of a *B*-factor cutoff of 40 for sidechain atoms. This is a compromise between accuracy and data set size (still a problem for individual amino acids), since it has been shown that the width of the χ angle distribution does not reach a minimum until $B \leq 20$ (Butterfoss, Richardson et al. 2005).

Ramachandran plots

The Ramachandran diagram (Ramachandran, Ramakrishnan et al. 1963), which plots ϕ vs. ψ backbone torsion angles for each residue in a protein, has been with us nearly as long as macromolecular crystal structures. (ϕ is the angle around the N–Cα bond, defined as C_{i-1}–N_i–$C\alpha_i$–C_i; ψ is the angle around the C–Cα bond, N_i–$C\alpha_i$–C_i–N_{i+1}.) This same coordinate system is used to show either empirical scatter plots of the conformations observed in the database of known 3D structures, or else contours of calculated energies or steric criteria as a function of ϕ and ψ for a dipeptide (as in Figure 1). Especially in recent

years, ϕ,ψ plots for individual proteins have also become central for structure validation, because ϕ,ψ values are not optimized in the refinement process and therefore provide a sensitive indicator of local problem areas (Morris, MacArthur et al. 1992). To the approximation of ideal covalent geometry and *trans* peptides, the ϕ,ψ plot encapsulates all information about backbone conformation in a remarkably concise and intuitive form; it has therefore been a key abstraction central to our growing understanding of protein structure, energetics, and folding.

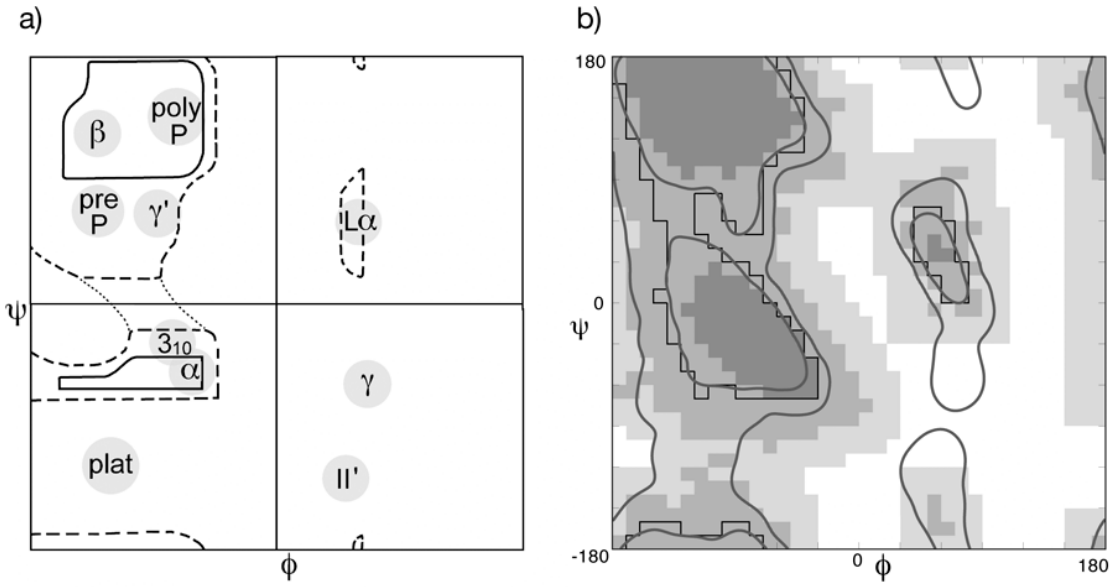


Figure 1: Classic ϕ,ψ treatments and definitions.

(a) Boundaries defined by hard sphere atomic overlaps, from Ramachandran and Sasisekharan (Ramachandran and Sasisekharan 1968); dashed lines enclose regions allowed with slightly smaller radii; dotted lines enclose regions allowed with small opening of the τ bond angle. Labels show approximate location of regions discussed by name here. (b) ϕ,ψ regions used for validation of experimental protein structures: areas shaded in dark, medium, and light gray are the "core", "allowed", and "generously allowed" regions, respectively, from ProCheck (Morris, MacArthur et al. 1992; Laskowski, Macarthur et al. 1993), whereas the stepped black outline encloses the single "strictly allowed" region from Kleywegt and Jones (1996). For comparison, the smoothe contours presented in this work are also shown.

Our knowledge of the overall empirical ϕ,ψ distribution has gradually improved in accuracy, both because a much larger number of structures are available and to an even greater extent because many of the recent structures are at extremely high resolution. The approximate location and shape of the most favorable, low-energy regions (consisting of $\alpha+3_{10}$, β+polyPro, and Lα, as labeled on Figure 1a, plus some areas bridging α and β) became clear very early (Ramakrishnan and Ramachandran 1965; Mandel, Mandel et al. 1977); their limits are produced by collision among mainchain and Cβ atoms and are, therefore, largely sidechain-independent except for Gly and Pro. Those particularly favorable regions have now converged to agreement in all treatments, except for a slight remaining tendency of the boundaries to shrink inward as accuracy continues increasing (Walther and Cohen 1999).

These ϕ,ψ criteria have become accepted as a central aspect of protein structure validation. They are routinely applied during deposition of structures to the Protein Data Bank (Berman, Westbrook et al. 2000) and have been made conveniently available on web sites such as the Biotech Validation Suite (http://biotech.embl-ebi.ac.uk:8400). Figure 1b shows the ϕ,ψ regions scored as "strictly allowed" for the Kleywegt and Jones single definition and the three "core", "allowed", and "generously allowed" regions for PROCHECK

(Morris, MacArthur et al. 1992; Laskowski, Macarthur et al. 1993). Inclusion of low-resolution, and especially of high-*B*, data in PROCHECK gave noise throughout the plot, producing unrealistic outlines for the two outer regions. In reaction, Kleywegt and Jones chose not to try distinguishing possible from forbidden conformations outside their 98% contour. However, several analyses (Herzberg and Moult 1991; Gunasekaran, Ramakrishnan et al. 1996; Pal and Chakrabarti 2002) have specifically studied a number of individual low-*B*, high resolution residues with ϕ,ψ in the outer regions (especially in the γ-turn, II' turn, and below-α plateau regions labeled in Fig 1a). They concluded the evidence is strong that such conformations are genuine, and also found that many of them occur at active or functional sites.

The γ-turn conformation (near ϕ,ψ = 75°, -60°, with CO_{i-1} to NH_{i+1} H-bond) was first described by Huggins in 1943 (Huggins 1943), although he envisioned a repeating series of residues with these torsion angles, producing a shallow helix with two residues per turn. Such a series has not been observed, although the mirror-image γ' conformation (near -75°, 60°) can sometimes repeat as a highly pleated β strand; the γ' region forms a small extension below the β region (Figure 1a). Némethy and Printz (Némethy and Printz 1972) suggested, on the basis of modeling studies, that the γ-turn

conformation may exist as a 3-residue chain reversal, and it was first described in a protein by Matthews (Matthews 1972) in thermolysin. Rose (Rose, Gierasch et al. 1985) and Milner-White (Milner-White 1990) have published reviews of the occurrence of γ and γ' turns.

The γ-turn and γ'-turn are also known as C_7^{ax} and C_7^{eq}, respectively, because the H-bond completes a 7-membered ring and the β-carbon is either axial or equatorial to the ring. Energy calculations for a dipeptide *in vacuo* (Cheam and Krimm 1990; Head-Gordon, Head-Gordon et al. 1991; Gould and Kollman 1992; Schäfer, Newton et al. 1993) find these two conformations as the overwhelming global energy minima in Ramachandran space, because the lack of water drives formation of the backbone H-bond. For proteins in aqueous solution, the γ' conformation is favorable but not optimal, and the γ-turn is accessible but rare (due to a minor steric overlap of Cβ and the carbonyl O).

In type II' tight turns, the second of four residues adopts ϕ,ψ angles near 50°, -125°. Gly is the most common residue in this position, although other residues are observed (Sibanda and Thornton 1985). As also seen in the γ-turn conformation, there is some steric overlap between the Cβ and carbonyl

oxygen. The serine of the catalytic triad of all α/β hydrolases and lipases has this conformation (Uppenberg, Hansen et al. 1994).

The analysis and proper treatment of these "outlier" conformations is still controversial, because it is very difficult to distinguish rare but genuine features of the molecules from errors in the models. Strained conformations should be expected, although the expectation is that they should be observed only rarely. Genuinely strained conformations are often useful indicators of biological significance, and certainly no attempt should be made to "fix" them. However, conformational outliers are to be treated with suspicion, since they may merely reflect deficiencies of the structure determination. This vital distinction between strain and error can, we propose, be approached statistically by behavior as a function of data quality and in individual cases both by the combination of atomic clashes and geometrical distortion and also by determining whether or not a more "normal" conformation could explain the experimental data equally well.

The pattern of relative frequencies within the ϕ,ψ distribution varies significantly among the 18 non-Gly, non-Pro amino acids (Richardson and Richardson 1989; Karplus 1996; Chakrabarti and Pal 2001), but the outlines of those regions are all nearly the same. In contrast, glycine and proline each

have very different ϕ,ψ distribution outlines than the other amino acids, with conformational constraints either significantly less (Gly, with no Cβ) or significantly more (Pro, with its pyrrolidine ring). However, their outliers have seldom been explicitly flagged and are not included in current overall-summary validation measures (Laskowski, Macarthur et al. 1993; Hooft, Vriend et al. 1996), since so much less data is available than for the general case. This is particularly unfortunate for glycine, because its lack of an observable Cβ atom makes its experimental ϕ and ψ values especially error-prone (Richardson 1981). Pre-Pro residues (those that precede prolines) also have a very distinctive ϕ,ψ distribution (Karplus 1996), and we treat pre-Pro as a separate fourth case here.

The current study revisits the Ramachandran plot for Gly, Pro, pre-Pro, the general case of the 18 other amino acids, and the residues in non-repetitive structure, using both new data and new techniques to resolve the above problems. From a database of 500 structures selected by resolution, homology, and other criteria of quality, the residues with high crystallographic *B*-factors for the backbone are omitted. Hydrogens are added and optimized (Word, Lovell et al. 1999b), and all-atom contact analysis (Word, Lovell et al. 1999a) is used to judge the reliability both of overall structures and of local regions. The

validity of sparsely populated regions is determined by analyzing their occurrence frequency vs resolution or *B*-factor and by examining a sample of cases for degree of bond-angle distortion, ambiguity of the electron density, and the presence of compensating interactions. The result is a set of remarkably well-defined empirical ϕ,ψ distributions which cleanly distinguish the truly disallowed regions from the disfavored but allowed regions.

Analysis by density-dependent smoothing

An accurate ϕ,ψ distribution is difficult to produce in practice because the data contain regions of both very low and very high density. For instance, there is a sparsely but evenly populated "shoal" of points below α helix; these points want large histogram bins or strong smoothing in order to make an even, continuous distribution. On the other hand, the α helix peak itself is more populated than other conformations by orders of magnitude, but conformations just to the right (greater ϕ values) are absolutely forbidden. Thus, α helix wants small histogram bins or very little smoothing so as to avoid bleeding over into forbidden regions. The trade-offs of different degrees of smoothing are illustrated in Figure 2.

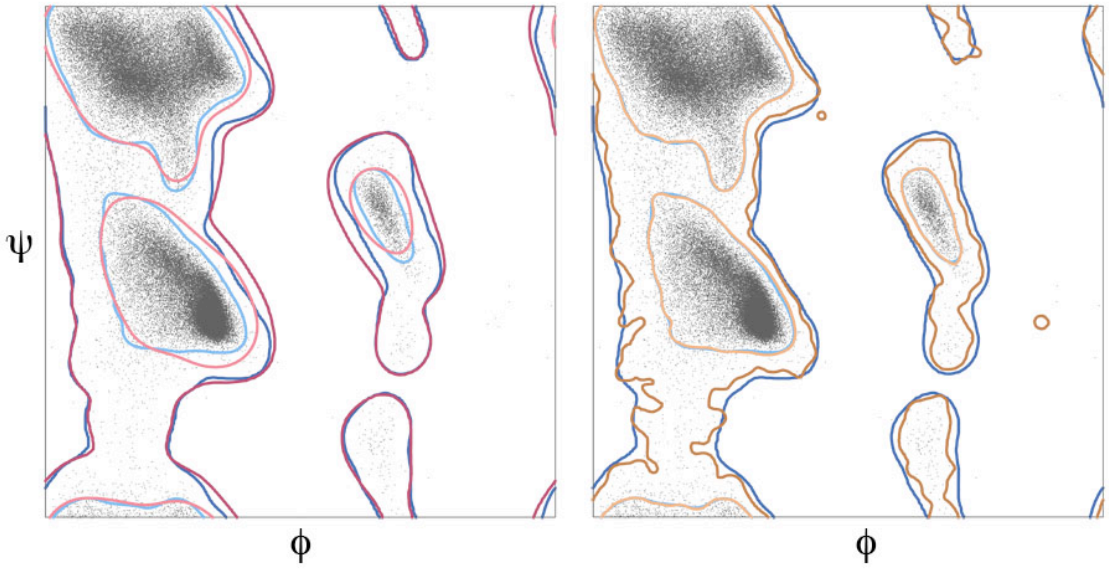


Figure 2: Comparison of density-dependent smoothing to one-pass kernel density estimation on general-case Ramachandran plots.

(left) Kernel density estimation (KDE, red contours) with α_0=28° matches the results of density-dependent smoothing (blue contours) in low population regions, but is too far outside the data edges in high density areas. (right) KDE with α_0=12° (orange contours) matches the results of density-dependent smoothing in high density areas, but performs erratically in sparsely populated zones.

To overcome these difficulties and produce accurate yet smoothly contoured boundaries for the ϕ,ψ distributions, I devised a density-dependent smoothing function. This function treats areas of sparse data as continuous regions of low, relatively constant density, while preserving the sharp transitions in regions where high density falls off rapidly. The smoothed distributions are used both for contouring the database distributions and (on the MOLPROBITY web site) for determining the ϕ,ψ quality values for each residue of a user-submitted structure.

The smoothed, normalized density of points, ρ, is expressed in general as a sum of the contributions of all N data points:

$$\rho(\phi,\psi) = \sum_{i=1}^{N} \sigma_i(\phi,\psi)$$

Each contribution σ_i is computed as a normalized cosine mask that depends on both ϕ and ψ, with a specified radius, α_i, according to:

$$\sigma_i(\phi,\psi) = \frac{\pi}{\alpha_i^2(\pi^2-4)}\{\cos(\frac{\pi x_i}{\alpha_i})+1\} \quad , x_i < \alpha_i$$
$$\sigma_i(\phi,\psi) = 0 \quad , otherwise$$

where

$$x_i = \sqrt{(\phi-\phi_i)^2 + (\psi-\psi_i)^2}$$

ϕ_i and ψ_i are the values for the *i*-th data point, and $\frac{\pi}{\alpha_i^2(\pi^2-4)}$ is the coefficient to normalize the mask volume to 1.0. Note the distinction between the many σ's, each of which simply spreads out the density of one data point, and the single ρ, which is the overall density function describing the distribution of populated regions on the Ramachandran plot.

The final density function ρ_2 is calculated in two iterations. In the first iteration, a density ρ_1 is calculated using the above equations, with identical $\alpha_i = \alpha_0$ for all *i*. The second iteration uses the same equations, but now the radius of each cosine mask varies according to:

$$\alpha_i = k\rho_1(\phi_i,\psi_i)^{-\lambda/n}$$

Since the objective is to smooth between the data points, we calculate the mask width from the function $\rho_1(\phi_i,\psi_i)^{-1/n}$, which approximates the average distance between points in *n* dimensions, modified by a linear factor *k* and an exponential factor λ. For this work, the number of dimensions in the data set, *n*, was always 2, and λ was fixed at 0.5, which produced contours that stayed suitably close to the steep transitions. The value used for *k* depended slightly on the data density in the sparse regions (see below). As implemented, ρ(ϕ,ψ) is approximated by summing all the σ_i on an evenly-spaced (2 degree) grid of

sample points to give $r(j,k)$. Any desired density $\rho(\phi,\psi)$ is found by linear interpolation from the four nearest $r(j,k)$. The end result is a good approximation of $\rho(\phi,\psi)$.

Contour levels are specified here as percentages. For example, a contour at 85% means that the data have been contoured at the largest density value not greater than $\rho_2(\phi_i,\psi_i)$ for 85% of data points i; that is, the contour encloses 85% of the data points and excludes 15%. Density values for contouring are determined after sorting all data points i by the value of $\rho_2(\phi_i,\psi_i)$; for example, the 85% value is the value of ρ_2 for the .85N-th point in the sorted list. Contours are calculated with the KIN2DCONT program (Word 2000).

For smoothing the general distribution, we used $\alpha_0 = 10°$ and $k = 13$. The final density (ρ_2) was then contoured at 99.95% (allowed) and 98% (favored) levels. The other distributions contained only 5-10% as much data, and so required a larger value of k for optimal smoothing. For the glycine, proline, and pre-proline distributions, we used $\alpha_0 = 10°$ and $k = 16$. The final density (ρ_2) was then contoured at 99.8% (allowed) and 98% (favored) levels. For the non-secondary-structure distribution in Figure 9, we again used $\alpha_0 = 10°$ and $k = 13$; ρ_2 was contoured at 99.9% (allowed) and 95% (favored) in order to

match the general-case contours closely, given the much smaller percentage of data in the helical region.

Ninety-eight-percent contours outlining the "favored" regions were defined for all four of the inclusive cases shown in Figure 3, allowing a summary statistic to be calculated for all residues. For the general-residue case, the 99.95% contour dividing the "allowed" from "outlier" regions is well behaved, but a 99.8% level had to be used for the single-residue distributions to avoid artifacts from small numbers. Residues in an individual structure are first assigned to Gly, Pro, pre-Pro, or general case and are then evaluated as favored, allowed, or outlier by comparing the interpolated density value for their ϕ,ψ to the relevant contour values. This procedure makes the use of smooth boundaries no harder than assignment by rectangular bins.

A precedent: adaptive kernel density estimation

Prior to developing the density-dependent smoothing algorithm, we searched in vain for a standard statistical method with the desired properties. Despite consulting with several statisticians, we were unable to locate any such method, although after completing this work we discovered the problem has been researched for at least two decades. Our method is in fact a kind of kernel density estimation (KDE) (Silverman 1986), which usually takes the

form of our first pass through the data, with a constant mask width (bandwidth) of α_0. One popular adaptive approach was invented by Abramson and implemented in STATA by Van Kerm (Abramson 1982; Van Kerm 2003); it is amazingly similar to our approach, using a per-point bandwidth inversely proportional to the square root of an initial estimate of the local density. (We used the 4th root instead, albeit for 2-D data instead of 1-D; our algorithm gives a square root for 1-D data as-is, or with $\lambda = 1$ would give a square root in the 2-D case).

Implementation of density-dependent smoothing

Our density-dependent smoothing algorithm has been implemented in the freely-available Java package SILK. The command-line interface to SILK supports numerous options for describing the input data space and choosing the type of smoothing (histogram, standard KDE, density-dependent KDE), the kernel type (cosine or gausian), smoothing parameters (α_0, k, λ), post-processing steps (e.g. convert probabilities to energies), and output format. As described above, SILK approximates the true $\rho(\phi,\psi)$ by evaluating it on a regular grid, the spacing of which can also be set. The full source code along with Makefiles for the analyses described in this chapter are available in the digital appendix.

ϕ,ψ for the general case

As explained above, we have developed a database of nearly 100,000 residues from 500 structures at ≤1.8 Å resolution, in order to determine which regions of Ramachandran space are populated for the best data: at very high resolution and low crystallographic *B*-factors. Plotting the ϕ and ψ of these data points for the general case of non-Gly, non-Pro, non-prePro (Figure 3a), we find the usual primary peaks in α, β and Lα conformations which have been known since Ramachandran (Ramachandran, Ramakrishnan et al. 1963). However, there are also significant, reproducible observations elsewhere on the plot which not only persist but even increase slightly in percentage as *B*-factors decrease, all the way down to *B*<10. The pattern does not change across the low-*B* data, but adding residues with backbone *B*≥30 leads to a dramatic increase in the amount of scatter, sparsely populating the entire ϕ,ψ plot including areas near ϕ = 0° which force large steric overlaps of mainchain atoms and are clearly not physically possible. The most important difference from earlier empirical studies of ϕ,ψ space, then, is the greatly improved signal-to-noise ratio (e.g., compare with Figure 5a of Morris *et al.* (Morris, MacArthur et al. 1992)). That improved signal allows a clear distinction

between truly disallowed "outlier" conformations and those that are rare but allowed.

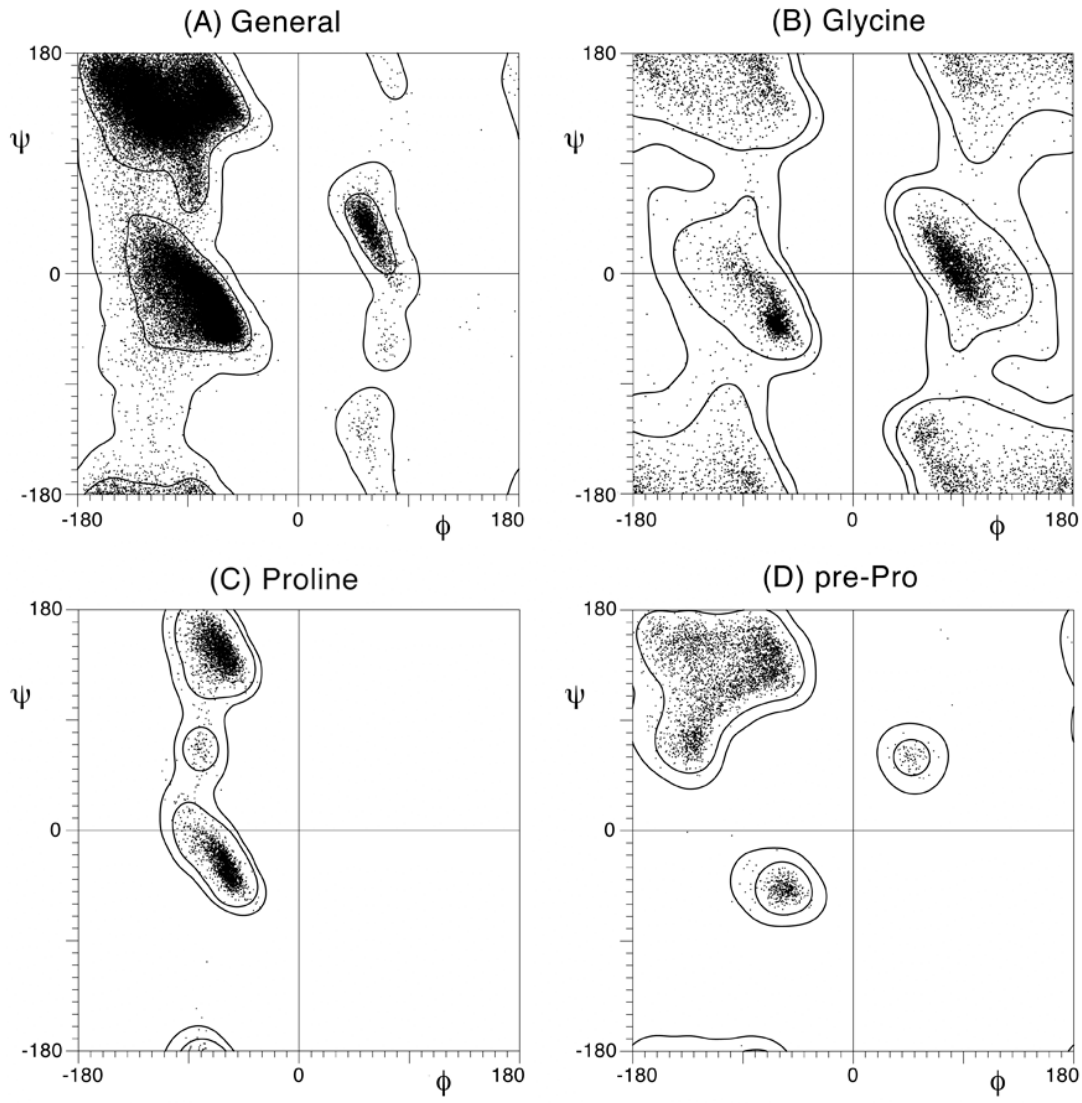


Figure 3: ϕ,ψ angle distributions for four cases.

Data includes 97,368 residues with backbone *B*-factor < 30 from the 500-structure high-resolution database, along with validation contours for favored and allowed regions. (a) The general case of 81,234 non-Gly, non-Pro, non-prePro residues. (b) The 7705 Gly residues, shown with twofold symmetrized contours. (c) The 4415 Pro residues with contours. (d) The 4014 pre-Pro residues (excluding those that are Gly or Pro) with contours.

Previous definitions of ϕ,ψ regions (Figure 1b) were done by counting data points within 10° angle bins. Although a bit coarse-grained, that system works quite well at high data density such as the PROCHECK "core" or the Kleywegt and Jones "strictly allowed" regions. However, at lower data density that method is unduly sensitive to statistical fluctuations, giving jagged edges that would presumably alter with the use of more or different data. In order to produce more robust and even edges, we choose to smooth and then contour the distribution. A second problem is produced by the contrast between very shallow and very steep edges, especially the extreme example of the diagonal edge to the right of a-helix, which goes from zero density to the global maximum in just over 20°. Data binning, uniform smoothing, or inclusion of low-accuracy data all tend to make this edge appear to spread outward into truly forbidden areas with large steric clashes. Therefore, we have used density-dependent smoothing (see above) to help the contours suitably hug the sharp edges while still smoothing the sparse, gradual edges.

The final choice is the level at which contours are drawn, defined by the percentage of the high-quality data they enclose. Our "favored" region includes 98% of the data and agrees almost exactly with the "strictly allowed"

region of Kleywegt & Jones (1996) as reproduced in Figure 1b, except for the absence of the left "leg" extending down from the β region; the correspondence would be complete if we had included pre-Pro residues in the general-case distribution. However, we agree with the PROCHECK authors (Morris, MacArthur et al. 1992; Laskowski, Macarthur et al. 1993) that in addition it is important to define an outer region that encompasses nearly all high-quality data, and we also believe that aim can finally now be successfully achieved for the general case; therefore, we have defined an "allowed" region that includes 99.95% of the data. The two contours enclosing favored and allowed regions are shown in Figure 3a, along with the 81,234 data points of the general-case distribution.

The favored region comprises 17% of the area of the plot; both favored and allowed regions together cover only 41.5%. However, our allowed region is significantly different in shape from either the "allowed" or "generously allowed" regions of Morris et al (1992). Our "outlier" region is both much larger and contains proportionately less data than their "outside" region. We agree that the plateau region below α should be considered allowed. We differ, however, on the forbidden nature of conformations near $\phi = 0°$ and on the acceptability of the sinuous, sparsely populated stripe of Ramachandran

space with positive ϕ, as seen in Figure 3a, which contains the controversial γ-turn and II'-turn conformations, as well as the favorable Lα region. Those conformations, although rare, do genuinely occur and are even enriched at active or binding sites. Most crystallographers have encountered at least one such example and have agonized more than necessary over its supposedly forbidden nature. These new, high-quality data show that although those conformations need some justification either by compensating favorable interactions or by functional need, they are nevertheless clearly allowed where such justification exists.

Although the conformations of the allowed but disfavored regions persist at low-*B* and high resolution, an even more compelling piece of evidence is the correlation with data quality: conformations that become more common as the data improve are to be believed, while those that become less common as data improve are to be treated with skepticism. Figure 4 plots the percent of data points (normalized by region area) versus *B*-factor for the γ, II', plateau, and outlier regions. As can be seen through the well-behaved range below *B*=30, there is a negative correlation with *B* for those conformations identified as genuine, while the outlier frequency is essentially zero; above *B*=30 and especially above *B*=36, the noise of random error spreads data points

everywhere. Even the present data, however, are not adequate for settling all possible cases: there are a few data points above Lα on the ϕ,ψ plot that we suspect are genuine, but they cannot at present be included within any well-behaved contour.

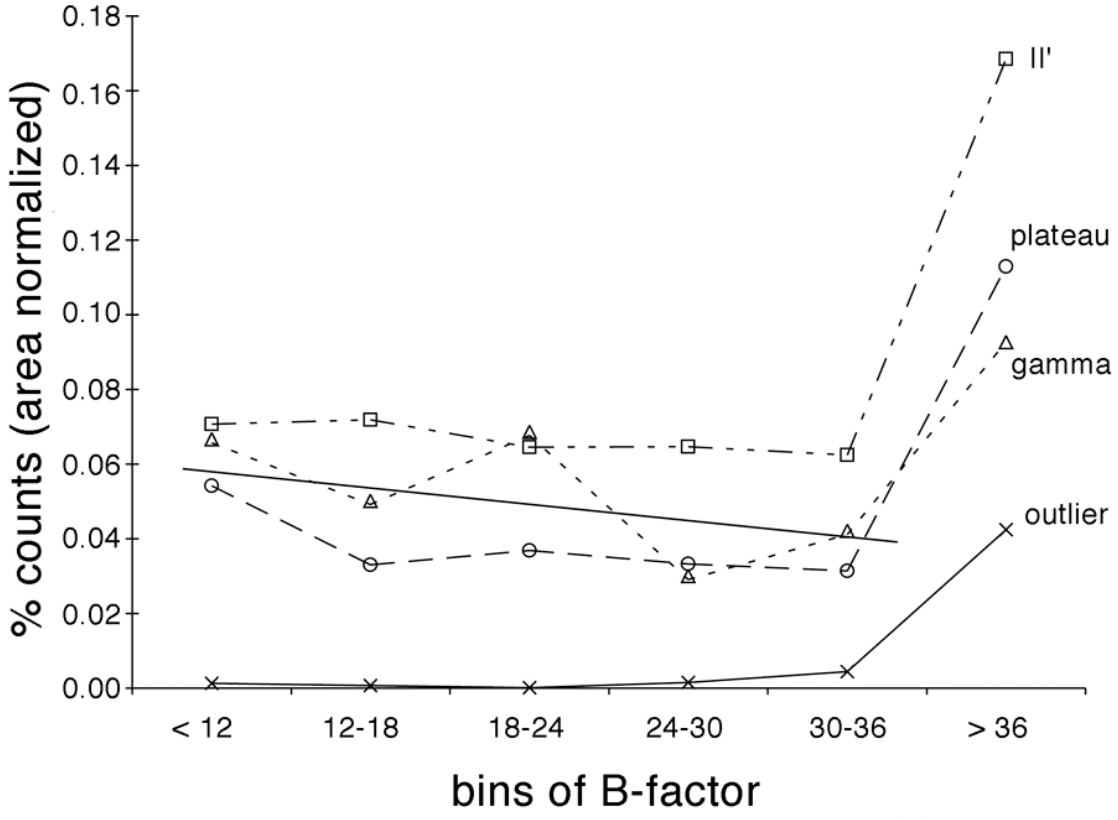


Figure 4: Occurrence of ϕ,ψ values in certain regions (for the general case residues), as a function of *B*-factor.

The lowest line shows outlier points (×) outside the allowed region, which occur at near-zero frequency up to *B*=30 but are quite prevalent at high *B*. In contrast, the solid straight line shows the inverse slope fit to the allowed but disfavored γ-turn (△), II′ (□), and plateau (○) regions, until they increase as well from noise in the highest range of *B*-factors.

Gly and Pro ϕ,ψ

Glycine and proline are significantly different from the other amino acids in their backbone stereochemistry. The lack of Cβ for Gly allows it to adopt a larger number of combinations of ϕ and ψ without steric clash, as compared to other amino acids. Conversely, for Pro the covalent bonding of the sidechain Cδ to the backbone nitrogen severely restricts the rotation about ϕ, allowing effectively a single value. This leads to the allowed and disallowed regions of the ϕ,ψ plot being of very different size and shape from those of the other amino acids: glycine ϕ,ψ is substantially less restricted than other amino acids, and proline substantially more so. Therefore, we calculate and evaluate Gly and Pro separately, as shown in Figures 3b and c.

The empirical distribution of ϕ,ψ for Gly is approximately twofold-symmetric around the central 0°,0° point, because the lack of a Cβ produces a mirror symmetry in the steric constraints for Gly. However, the data obey that symmetry only inexactly: the righthanded α-helix produces a small, intense peak around -60°,-40°, but the Lα to $L3_{10}$ region is even better populated presumably because it is a useful conformation accessible without any strain only for Gly. The steric constraints defining the outer limits of accessible ϕ,ψ regions should be symmetrical for Gly, and so we calculate the 98% and the

99.8% contours for Gly from a two-fold symmetrized version of the Gly ϕ,ψ data. The data points plotted in Figure 3b are unsymmetrized, to show that they fit well within the outlines defined by the symmetrized contours. Near ϕ = 180° there is not enough data to define the outer contour robustly, but toward ϕ = 0° the steric clashes are strong and the boundaries are clear. Therefore, although Gly conformation is more permissive than the general case, the outlier region covers 37% of the ϕ,ψ plot.

To the first approximation, Pro is very simple, because the ring closure restricts ϕ near -70°. However, as seen in Figure 3c, there is a rich detail in the distribution, with a ϕ width from about -50° to -100° and three distinct peaks in ψ. The two major peaks correspond to α and poly-proline II conformations, while the small central peak is in the γ' region. γ' is a slightly strained conformation stabilized by an NH_{i-1} to CO_{i+1} hydrogen bond. Figure 3c includes prolines preceded by either *trans* or *cis* peptide bonds; the *cis* examples have relatively more negative ϕ values and do not occur in the γ' peak, since they cannot form the γ' hydrogen bond.

Pre–Pro ϕ,ψ

As mentioned in the description of the general case, residues that precede proline are treated separately here because they have a distinctively different ϕ,ψ distribution, shown in Figure 3d. As originally pointed out by Karplus (Karplus 1996), pre-Pro residues preferentially populate a region near -130°, 80° (marked "pre-Pro" on Figure 1a) below the left side of the broad β region. The pre-Pro distribution in Figure 3d, indeed, shows more than twice as many data points in that region than the general distribution of Figure 3a, in spite of the fact that there are only 5% as many total pre-Pro residues; their preference for that region is thus 40-fold higher.

The only other regions populated by pre-Pro residues are β, poly-Pro, and two small, round areas at the very tip of α and of Lα. The constraints on pre-Pro conformation are produced by the Pro Cδ, which not only prevents H-bonding of the Pro NH but clashes with other backbone or Cβ atoms in many conformations. γ and γ' conformations are impossible for pre-Pro residues because the clash of CO(prePro-1) with Cδ(Pro) takes a large bite out of the pre-Pro ϕ,ψ distribution, producing the convex curve that forms the lower right edge of the β region for pre-Pro. As is often true, occurrence is enhanced just inside that sharp boundary, presumably because there is then a favorable

van der Waals contact for the same atom pair that clashes just outside the boundary.

Looking at the shape of the pre-Pro distribution in Figure 3d, it is hard to deny its significant difference from any of the patterns in Figure 3a-c; therefore, we treat pre-Pro ϕ,ψ separately as a fourth case for validation purposes. Examination of ϕ,ψ distributions for the other amino acids, either individually or in related groups, shows substantial differences in peak heights of various regions, especially for Lα; however, the contours enclosing favored and allowed regions do not differ enough to justify their separate treatment at the current database size.

Cross-peptide ψ,ϕ distributions

Although ϕ,ψ values are strongly linked within one residue, they have little influence on one another across the peptide bond (i.e., ψ of one residue and ϕ of the next). Substantial regions of the ψ,ϕ plot are nonetheless disallowed, especially the bands around $\phi = 0°$ and $\phi = 130°$. However, this information is largely redundant with that in the Ramachandran plot. It is thus not generally useful, but may have a special-purpose use in algorithms that build protein backbone one residue at a time (Hellinga 2005).

Figure 5 shows the general case ψ,ϕ plot for all residues except Gly and Pro; the plot does not change significantly when repetitive secondary structure is removed (data not shown). Because of their unique behavior in the Ramachandran plot, Gly and Pro were tested separately for any influence on the ψ,ϕ plot, in both pre- and post-peptide positions: X-Gly, X-Pro, Gly-X, and Pro-X. As expected, X-Gly relaxes the constraints on ϕ, allowing values around 130° (Figure 5, center). On the other hand, Gly-X matches the general distribution (except for Gly-Gly). Proline is fairly uninteresting: Pro-X matches the general distribution, and X-Pro closely resembles the Ramachandran plot for Pro. Subsets of X-Pro display some interesting trends: the ϕ_{i+1},ψ_i = -65°, -165° region of the X-Pro plot is exclusively Gly-Pro, which also seems to avoid the -65°, 90° region that other X-Pro pairs inhabit. Pro-Pro pairs gather around -65°, 150° (Figure 5, right). This information could potentially be useful in structure validation, but it applies to a relatively small number of peptides, thus limiting its usefulness.

We also tested pairs of aromatics (Phe, Tyr, Trp) and pairs of branched-β residues (Val, Thr, Ile) in the ψ,ϕ plot. Due to the small numbers of data points available, no significant differences from the general distribution could be observed.

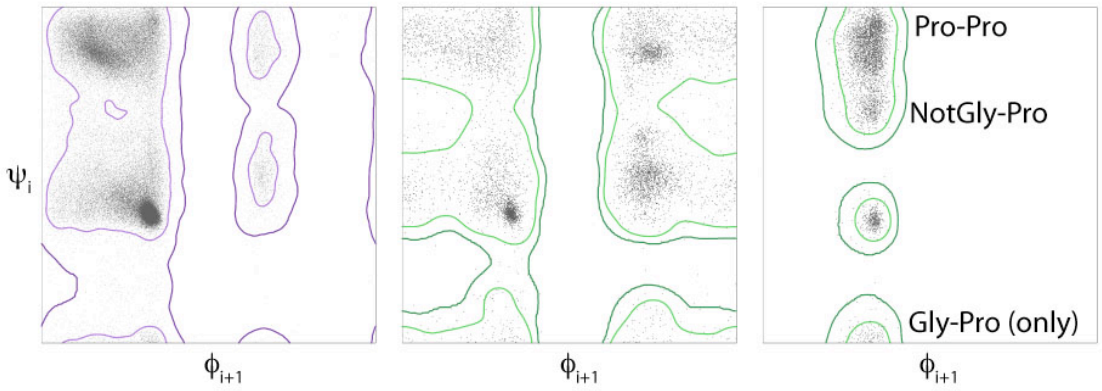


Figure 5: Cross-peptide plots for three cases.

(left) The general case ψ,ϕ (cross-peptide) plot for all non-Gly, non-Pro residues. (center) The ψ,ϕ plot for X-Gly residues. (right) The ψ,ϕ plot for X-Pro residues, with preferences for "X" marked.

Sidechain rotamers

The dihedral angles of protein sidechains (χ1 - χ4) also cluster very strongly when plotted in *N* dimensions, perhaps more so than the backbone dihedrals. The cluster shapes tend to be more regular than the regions of the Ramachandran plot (often approximately round), and many place most or all of the χ angles in staggered conformations (near -60°, 180°, or +60° for bonds between sp^3 hybridized atoms). Thus, the analysis of sidechain conformations is similar to that of backbone, but not identical.

Sidechain conformations were also studied by Ramachandran early in the history of structural biology (Ramachandran, Ramakrishnan et al. 1963), but the concept of rotamers was first introduced in 1987 by Ponder & Richards (Ponder and Richards 1987). Their library of rotamers contained a representative conformation from the center of each multidimensional cluster. Although their library had few artifacts, the relatively small database that was available limited the completeness -- most amino acids could only be analyzed in χ1 and χ2.

Many other groups have since developed their own rotamer libraries, each with its own strengths and shortcomings. Briefly, most of them contain rotamers with severe internal steric clashes, and/or were constructed with

unfiltered or otherwise inappropriate datasets. Some of the incorrect entries in these libraries are the result of systematic misinterpretations in the experimental data (e.g. the backwards leucine conformers *tt** and *mp**), and hence have polluted future structures in the database. For these reasons our laboratory developed the "penultimate" rotamer library (Lovell, Word et al. 2000) using high-quality filtered data. Lovell et al. discuss previous rotamer libraries in more detail and introduce the *m,p,t* naming system for χ angles, which will be adopted here.

The goal of the current work, however, is somewhat different in nature. Rather than cataloging and naming the local minima on the *n*-dimension energy surface of sidechain conformations (i.e., rotamers), we primarily seek to describe the distribution of well-validated, empirically observed conformations in this *n*-D space.

This distribution has several applications. First, like the Ramachandran plot, it can be used for structure validation: parts of the model that fall outside well-populated regions are usually indicative of fitting errors, but may indicate real but strained conformations that are stabilized by surrounding structure. Such strained conformations often occur at active or binding sites and may be important to the protein's function. Second, rotamer distributions

can be used for probability-weighted sampling of sidechain conformations in computational protein design, homology modeling, docking, etc. Third, the probability distributions could be converted to empirical energy functions for similar applications. In all these scenarios, it is important that the distribution be as accurate as possible. In particular, it is more important to exclude all invalid conformations than it is to include 100% of the valid ones.

Methods and parameters for rotamer analysis

Sidechain rotamers were also analyzed with the density-dependent smoothing algorithm developed for the Ramachandran plots. Although the rotamer populations show a smaller dynamic range, the density-dependent analysis does produce somewhat tighter contours. Parameters varied slightly depending on how many χ angles the sidechain has, as shown in Table 1. In particular, sidechains with more χ angles required a coarser sampling grid in order to fit within available computer memory.

χ angles	Amino acids	Grid (°)	α0 (°)	k (°)
1	Ser, Thr, Val, Cys, Pro[1]	1	10	14
2	Ile, Leu, His, Trp, Asn, Asp, Phe+Tyr[2]	5	10	14
3	Met, Gln, Glu, Pro[1], Lys[3], Arg[3]	8	10	14
4	Lys[3], Arg[3]	10	10	14

Table 1: Parameters used for calculating sidechain rotamer distributions

(1) The proline ring has only one degree of freedom and two rotamers (Cγ endo and Cγ exo), which can be distinguish based on χ1 alone. The 3-D plot confirms the expected two peaks.

(2) Phe and Tyr were the only sidechains with indistinguishable distributions. As such, their data were pooled to get more sample points and improved accuracy.

(3) Lys and Arg were plotted in 3-D for visualization purposes (one χ123 plot, and three χ234 plots with χ1 = m, p, or t) but the 4-D distribution is used for calculations.

The raw data for these distributions were taken from the Top500 database (the same one the Ramachandran analysis was done from), and restricted to sidechains with all *B*-factors less than 40. *B*-factors are generally somewhat higher for sidechains than mainchain, so the cutoff was relaxed somewhat (versus B≤30 for Ramachandran plots) to ensure that enough data points were available. Unlike the smaller database used in developing the Penultimate rotamer library (Lovell, Word et al. 2000), for this analysis sidechains were not filtered out based on all-atom steric clashes, as such filtering did not significantly alter the distributions.

No further filtering was needed for most amino acids, but the leucine "decoy" rotamers *tt** and *mp** required special treatment. These two conformations show up as substantial clusters in the 2-D χ angle distribution, but they are actually the result of systematically misfitting *tp* and *mt* conformations, respectively. The error results from ambiguous density at the end of a Leu sidechain, which may appear T-shaped. In that case, it is easy for the crystallographer to accidentally fit the end of the sidechain in the backwards orientation. Because of their prevalence in the database, these false peaks cannot feasibly be eliminated with stricter filters on resolution and/or *B*-factor. Because they are systematic rather than random errors, they cannot be

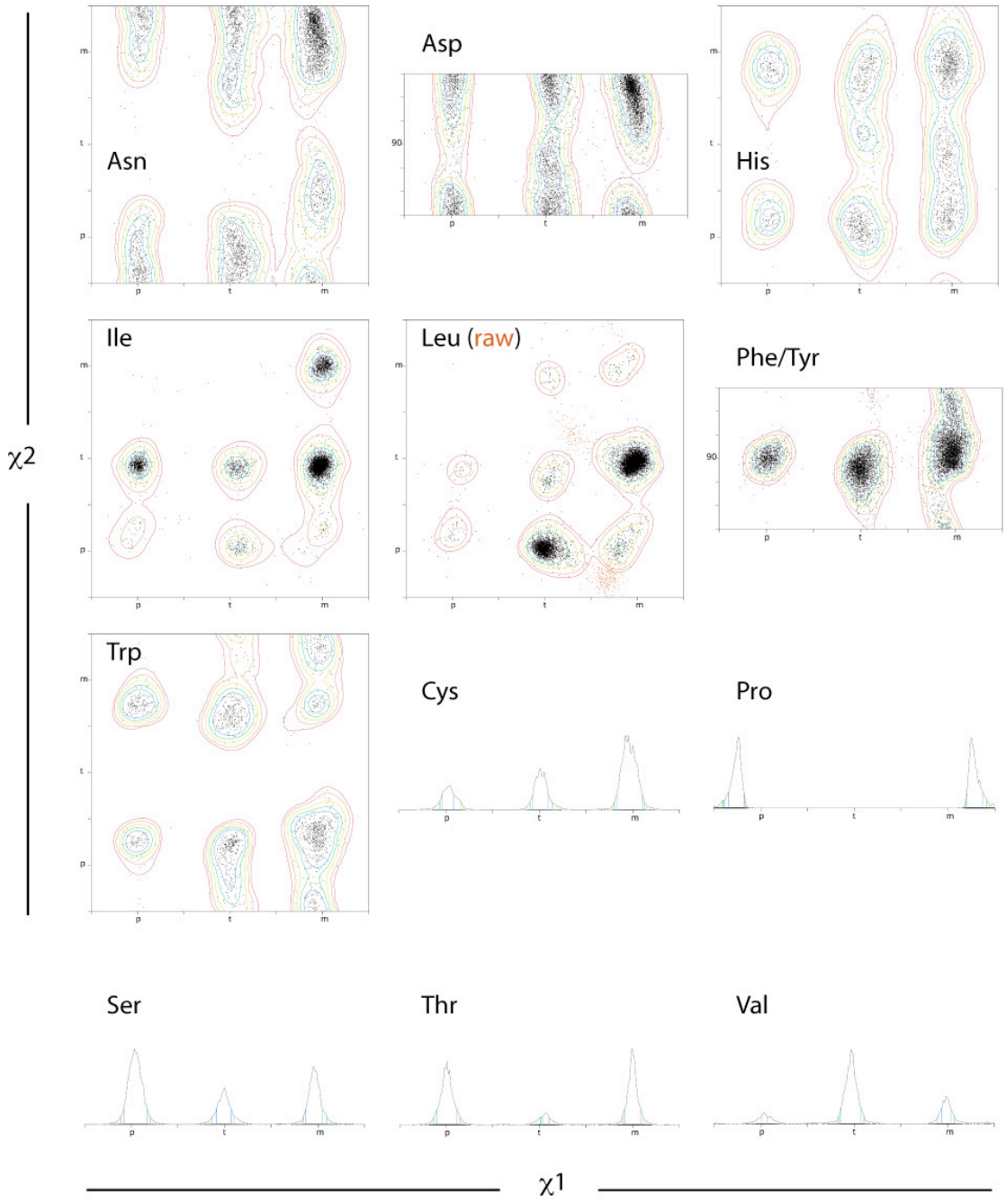
eliminated statistically. As a result, we manually removed from our data set any conformation within a 34°-radius circle of χ1,χ2 = 214.3°,215.6° (*tt**) or within a 40°-radius circle of 253.0°,10.4° (*mp**). These ranges were chosen by inspection after reviewing many of the individual structures in those regions. It is possible that this additional filtering eliminated a few valid data points in addition to the systematic errors, but on the whole it leads to a much more realistic distribution.

Figure 6a and b shows the rotamer distributions for all amino acids, including the raw data for leucine overlaid with the filtered contours.

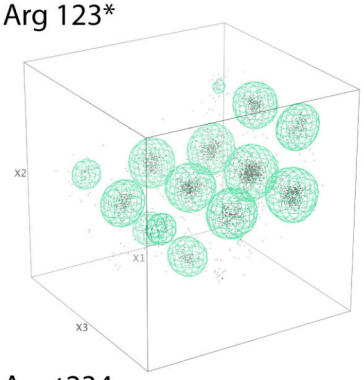
Figure 6: Rotamer distributions for all sidechains.

(a) All sidechains with one or two χ angles. Contour levels are 1 (red), 2 (orange), 5 (yellow), 10 (green), and 20 (blue) percent of data points excluded. Systematically misfit Leu conformations that were removed by manual filtering are shown as red points in that plot.

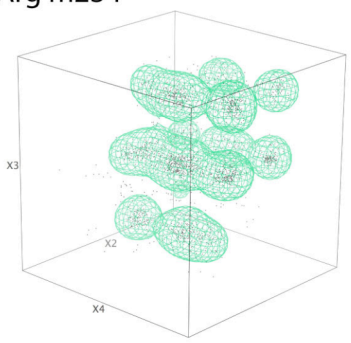
(b) All sidechains with three or four χ angles. Contour levels are 5 (brown) or 10 (green) percent of data points excluded.


Asn
Asp
His
Ile
Leu (raw)
Phe/Tyr
Trp
Cys
Pro
Ser
Thr
Val
χ2
χ1
m
t
p
90

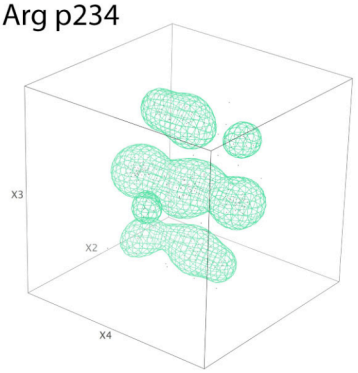
Arg 123*



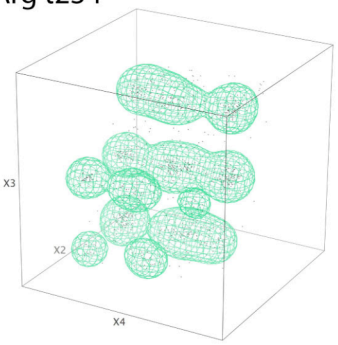
Arg m234



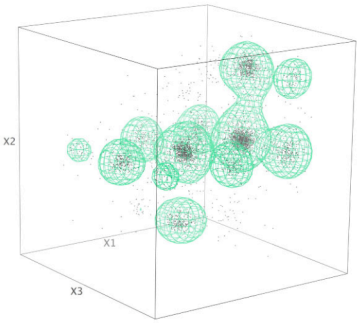
Arg p234



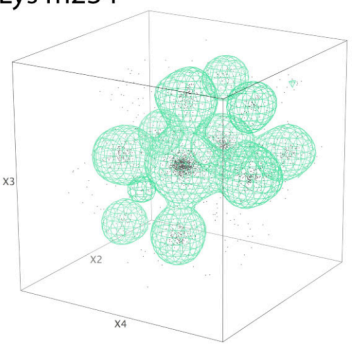
Arg t234



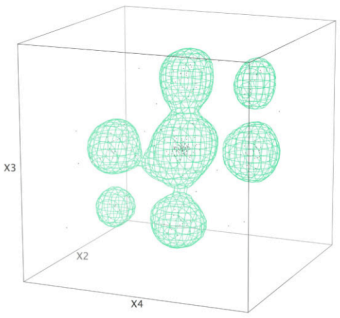
Lys 123*



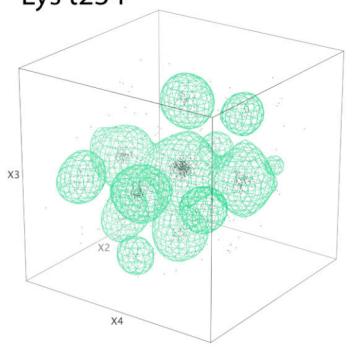
Lys m234



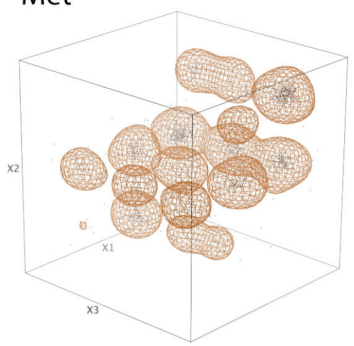
Lys p234



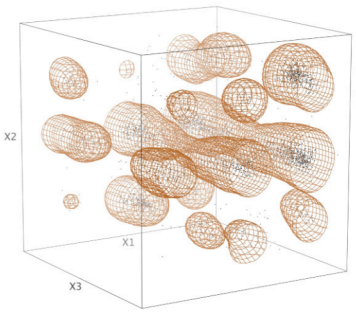
Lys t234



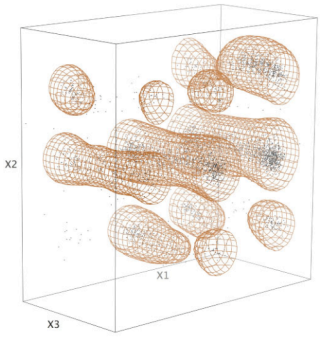
Met



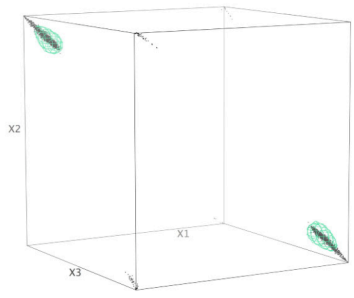
Gln



Glu



Pro 123



RNA backbone rotamers

The work of Laura Murray in the Richardson lab has shown that RNA backbone conformations cluster into distinct rotamers in much the same way that protein sidechains do (and that protein backbone does not) (Murray, Arendall et al. 2003; Murray, Richardson et al. 2005); these rotamers affect the overall structure of the molecule rather than just the local details. RNA backbone rotamers are also significantly more complicated than the other cases. For instance, only ~8600 residues of data are available, and mostly at much lower resolution than the ~100,000 protein residues used for the rotamer and Ramachandran studies. The data space is also much larger, with seven dimensions. These are the seven dihedral angles from one sugar to another, including both ring puckers. Laura found that this base-to-base unit (a "suite") showed much stronger clustering that the traditional phosphate-to-phosphate residue, and that most steric clashes occurred within a suite.

The tools for density-dependent smoothing were also useful for studying RNA rotamers, but we were limited to 2- or 3-variable projections of the full space. Using linked pairs of 3-D plots in KiNG and MAGE, Laura was eventually able to identify 42 backbone rotamers. Even calculating the 7-D distribution challenges our resources: a 10° granularity grid would require

$(360°/10°)^7 \times 4$ bytes = 313 GB of RAM! Recent extensions to SILK allow it to use a sparse grid, reducing memory requirements to manageable levels. However, it is still very difficult to determine whether the result is reasonable, since direct visualization is impossible.

Naming rotamers

Especially for RNA backbone, but also for protein sidechains, it is sometimes desirable to automatically assign a rotamer name or label to an empirically observed conformation (e.g. *mp, 3'emmtp3'*). For instance, it would allow robust comparison of protein structures to locate sidechains that were in different rotamers, more accurately than e.g. looking for χ angles that change by >40°. In the RNA case, it could greatly ease the task of identifying rotamers in the first case. Also, the RNA's 3-D fold could be converted to a 1-D string of rotamer names. This might open the way to using traditional string-based (i.e., sequence) bioinformatics algorithms on structural data. Additionally, sequence and structure codes could be combined for a richer description.

Such 1-D sequence-structure encodings have been proposed for RNA backbone (Hershkovitz, Tannenbaum et al. 2003; Murray and Richardson 2006), but the problem of automatically assigning rotamer names is still not

solved satisfactorily. Thus, the raw data for a bioinformatic analysis are not available. One simple approach would be to compute the Euclidean distance in angle space between a given conformation and each rotamer, assigning the conformation to the nearest rotamer. This may fail if (1) instances of a given rotamer are not distributed normally about a single point (e.g. Phe/Tyr, Figure 6), (2) rotamer clusters are normal but have different radii, or (3) changes in one angle are more important than changes in another (i.e. the space is not isotropic). In practice, all of these are likely to be problems for protein sidechains and RNA backbone, although (3) could probably be solved by weighting the different dimensions appropriately. Furthermore, many conformations will not correspond to any legal rotamer, and so maximum distance cutoff(s) must also be specified.

A more elaborate scheme involves hill climbing on the distribution: for any conformation that is a valid rotamer, follow the gradient of the distribution towards higher density until the peak is reached. All conformations that lead to this peak are then grouped as a single rotamer. However, this scheme is very sensitive to the choice of smoothing parameters. If the distribution is insufficiently smoothed, the algorithm will identify spurious "sub-rotamer" peaks; the same problem might also arise by bad luck from random noise in a

sparse distribution. If the distribution is over-smoothed, on the other hand, minor true peaks may become mere shoulders of their neighbors. For example, Asn rotamers *αm-80°* and *αm-20°* are in some danger of this, although they interact with surrounding structure differently (by H-bonding and van der Waals packing, respectively) (Lovell, Word et al. 1999). There may be cases where a minor conformation falls so close to a major one that although they are performing different structural roles, no choice of smoothing parameters can resolve them. Of course, in this case other methods are likely to miss the distinction also, and further segmentation of the data is called for (e.g. by protein secondary structure).

Preliminary support for hill climbing classification of rotamers has been added to the SILK software. This approach looks promising for protein sidechains, many of which have irregularly shaped clusters but reasonably large data sets. One downside is that the namable rotamers are defined by peaks in the data; while spurious peaks can be named the same as their parent, it is not possible to define new peaks by hand. As a result, the namable peaks may not match up exactly with our Penultimate Rotamer Library (Figure 7). In particular, the additional "sample point" rotamers

defined for Glu, Gln, Asp, Asn, Phe, and Tyr do not correspond to real peaks in the data distribution.

Because the two δ angles and the γ angle are cleanly bi- and trimodal respectively, and because many values are forbidden for ε, it may be possible to do smoothing and hill-climbing for RNA in only 3 or 4 dimensions. On the other hand, the RNA data are still so sparse that statistical techniques may be unable to reliable identify clusters at all -- many backbone rotamers have only a half-dozen known instances. Fortunately, these clusters appear to be generally ellipsoidal, and so an appropriately weighted Euclidean distance to predefined peaks may be the best method of classification for those cases.

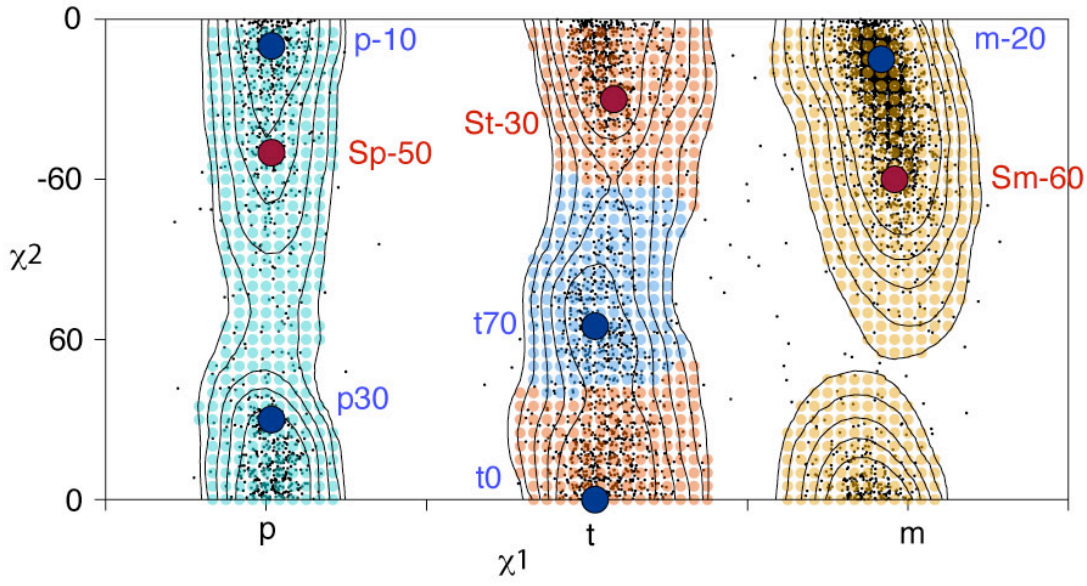


Figure 7: Asparagine rotamers.

Compare rotamers from the Penultimate library (balls and labels; blue are true rotamers and red are sample points) versus the rotamers defined by a hill climbing algorithm (pale "polka dot" shading). Notice how *p-10* and *p30* are not resolved into two separate rotamers by the hill climbing analysis, but *t70* and *t0* are.

Discussion

The studies of dihedral angles presented here are interesting in their own right for their insights into the energy patterns that make macromolecules fold, but one of the major practical applications of this data is structure validation. The process of validation serves several distinct purposes. The most traditional is for the benefit of journal and grant reviewers, lab directors, database entry, etc.: certifying whether a structure meets generally accepted current standards of good practice in the field. In macromolecular crystallography there is a reasonable consensus, for a given resolution range, about respectable values of residual and free R (Brunger 1992; Kleywegt and Jones 2002) (see also real-space fit of model to density at http://portray.smc.uu.se/eds), and for ideality of bond lengths, bond angles, and ϕ,ψ values (Laskowski, Macarthur et al. 1993; Kleywegt and Jones 1996). Such standards are extremely important, and we hope that the criteria developed here will become accepted by the community as additions to, or improvements on, current standards.

However, the present work is primarily addressed to strengthening two other important aspects of structure validation. The first is providing the end-users of 3D data with convenient but critical assessments of probable accuracy

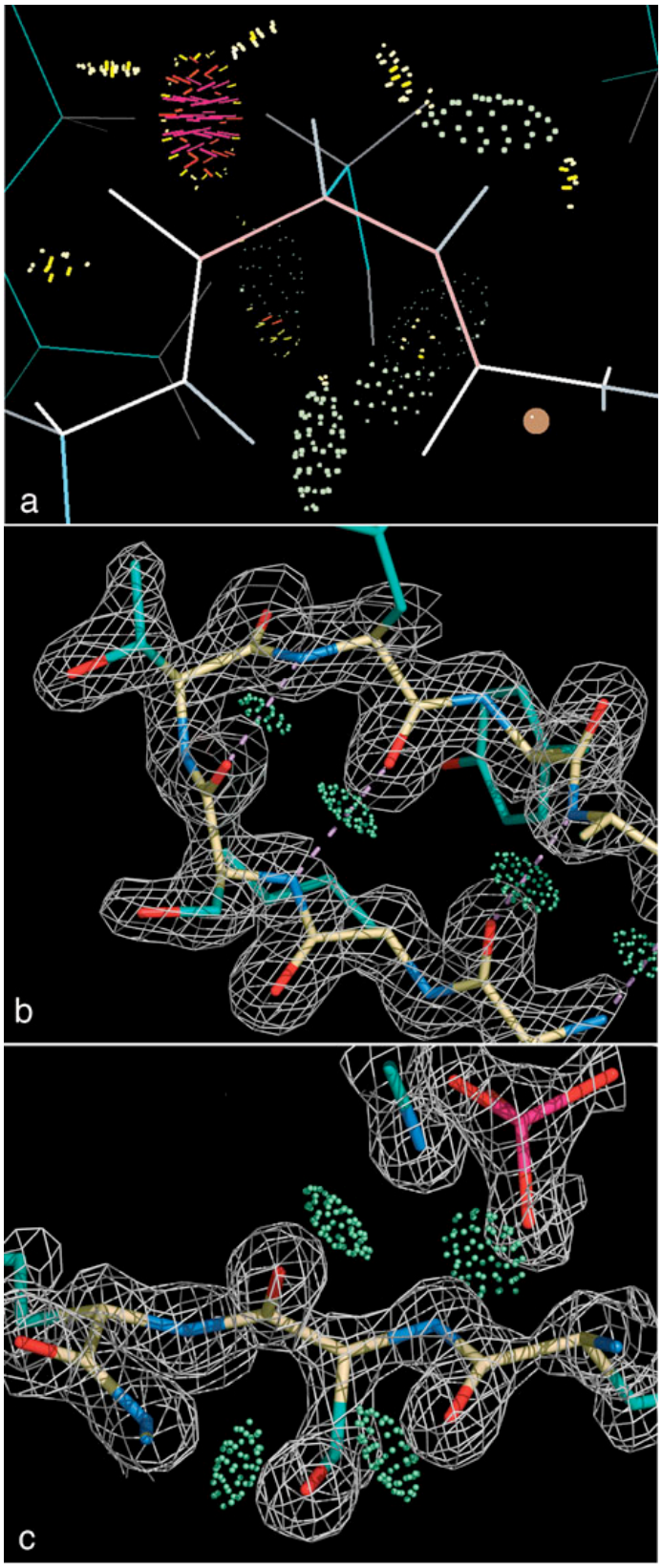
that apply to local regions as well as to overall structures. The second aspect is providing crystallographers themselves with a suite of tools for locating and fixing local problems during the process of fitting and refinement. The practical implementation of these goals is our MOLPROBITY web server, which is described in Appendix A. A major component of MOLPROBITY is all-atom contact analysis (Word, Lovell et al. 1999a; 1999b; Richardson and Richardson 2001), which has the advantage of using information (the hydrogen contacts) that is independent both of the usual refinement targets and of traditional validation criteria. That contact information, however, is most powerful if used in conjunction with suitable measures of geometric ideality, since choice of refinement strategy can to some extent trade off non-ideality between those two types of criteria. Therefore, there is a need for geometrical validation tools that are updated with large and quality-filtered datasets, that are tuned to complement all-atom contact analysis, and that are collectively optimized for sensitivity to backbone or sidechain conformations trapped in the wrong local minimum. The present analysis provides those geometric validation tools.

A residue with good fit to density, low *B*-factor, favored ϕ,ψ values, a rotameric sidechain, no atomic clashes, and ideal covalent geometry is almost

certain to be modeled correctly. Whenever several of those factors are far from optimal, however, an error should be suspected unless there are mitigating circumstances such as compensating favorable interactions, tight packing constraints, or functional requirements for a locally-strained conformation. One example where the combined validation criteria diagnose a clear error is the Ramachandran-outlier residue of Figure 8a, with ϕ,ψ values of +44°,-29° well inside the truly forbidden area near $\phi = 0°$, a bad all-atom clash, and two successive backbone bond angle opened up by more than 10°. It is an important argument that there is a normal, favorable conformation that could occupy nearly the same position in space and connect well with the continuing chain on either side. That also means this example is correctable, which is certainly not always the case but is made more likely by good rotamer libraries and multiple, independent validation criteria that are local and direction-specific.

Figure 8: Specific examples of unusual conformations either invalidated or validated by a combination of criteria.

(a) 2SIM Ser230 (Crennell, Garman et al. 1996) with ϕ,ψ +44°, -29°: low *B*-factors and some H-bonding, but a serious all-atom clash and two successive backbone bond angles (in pink) off by >10° invalidate this conformation. (b) 2TLX Thr26 (English, Done et al. 1999) γ-turn with ϕ,ψ +79°,-63°: small bond angle distortions, but good backbone H-bonds, no clashes, and clear electron density validate this conformation. (c) 1KA1 Ser264 (Patel, Martinez-Ripoll et al. 2002) outlier with ϕ,ψ +85°, +171°: modest bond angle distortion, but low *B*'s, good electron density, no clashes, and 4 good H-bonds validate this case.


a
b
c

These criteria are equally valuable for positively validating the correctness of well-placed residues with disfavored-but-allowed, or even outlier, conformations. Figure 8b shows the classic γ-turn residue from thermolysin, with ϕ,ψ of +79°,-63° classed as a serious outlier by PROCHECK, WHATCHECK, and Kleywegt & Jones (see Figure 1b). The new Ramachandran criteria class it not as an outlier but as allowed (although disfavored). It forms the γ-turn H-bond with no atomic clashes and only small bond angle distortions, at the end of a well H-bonded β-hairpin, and it has moderate *B*-factors (near 20) and clear, well-fit electron density. Figure 8c shows a serine with ϕ,ψ of +85°,+171°, which is a Ramachandran outlier even by the new criteria but is only just outside an allowed region. The other bond and torsion angles are reasonable, there are no bad clashes, the electron density is clear and well fit, the *B*-factors are low (about 7), and the residue makes two backbone and two sidechain H-bonds, one to help bind the adenosine-3'-5'-diphosphate product. In both of these examples the model is validated as clearly correct, because a single worrisome feature is outweighed by the total evidence of the other favorable indicators.

We are suggesting, on the basis of their reproducible but relatively rare occurrence patterns, that conformations outside the favored but inside the

allowed ϕ,ψ regions are modestly strained, with a significant but not huge energetic penalty relative to the favored α, β, and Lα conformations. Experimental evidence relevant to that claim can come from stability measurements of Ala vs Gly mutants for residues that start out in (and are likely to stay in) conformations favored for Gly but merely allowed for the general case. Several studies fulfill those conditions for the II'-turn and the below-α "plateau", and are discussed below. We have found no similar mutation studies for the γ-turn conformation, but in any case their interpretation would be less clear since the γ-turn region is not very well populated even for Gly.

Stites et al. (1994) mutated Gly79 of Staphylococcal nuclease (which has plateau ϕ,ψ of -102°,-145° in 1SNC) to Ala and found a decrease in stability of 1.3 kcal mol^{-1}. For residues with II' ϕ,ψ values that are also in position 2 of an H-bonded type II' turn, a Gly to Ala mutant of Gly68 in the V_L domain of antibody McPC603 (ϕ,ψ = 76°, -95° in 2IMM) was destabilized by 0.67 kcal mol^{-1} (Ohage, Graml et al. 1997), and an Ala mutant compared to a Gly mutant of Asn138 in Staphylococcal nuclease (ϕ,ψ = 41°,-108° in 1SNC) was destabilized by 1.2 kcal mol^{-1} (Stites, Meeker et al. 1994). The relative frequency of Gly is approximately 10-15 times higher than for Ala in these

regions, so that a simple pseudo-energy based on Boltzmann statistics ($E = RT \ln P$) would imply a difference of about 1.3-1.6 kcal mol^{-1}. This agreement within a factor of 2 seems very reasonable, given the theoretical uncertainty about the applicability of such a pseudo-energy and the experimental uncertainty about whether the energy difference might be lowered by a conformational change. [Note that an apparently anomalous mutation result, showing a 0.3 kcal mol^{-1} stabilization for Ala over Gly50 at ϕ,ψ 97°, -154° in 1SNC (Stites, Meeker et al. 1994), is not applicable to this issue: the loop containing Gly50 has very poor electron density and very high *B*-factors, so that Ala50 is likely to adopt an entirely different conformation.] Mutational results and empirical occurrence frequencies agree, therefore, that the II' and plateau conformations are allowed but disfavored for residues with β-carbons, by a significant but modest energy penalty on the order of 1-2 *kT*.

Comparison is also relevant with theoretical calculations for the energy or free energy of conformations accessible to the dipeptide, done with a water model or with intermediate values for the dielectric constant. Since the dipeptide cannot include longer-range interactions such as those involved in secondary structures, the most nearly appropriate comparison is with the empirical ϕ,ψ distribution found for non-repetitive protein structure, compiled

either just for Ala (without pre-Pro) or for the general case without Gly, Pro, or pre-Pro (shown in Figure 9). Aside from greatly lowering the enormous peak at α-helical ϕ,ψ, which is 10x the density of any other region, the non-repetitive distribution differs very little from the general-case distribution: favored and allowed contours for the non-repetitive (dark lines) and the general (thin lines) cases in Figure 9 coincide almost perfectly.

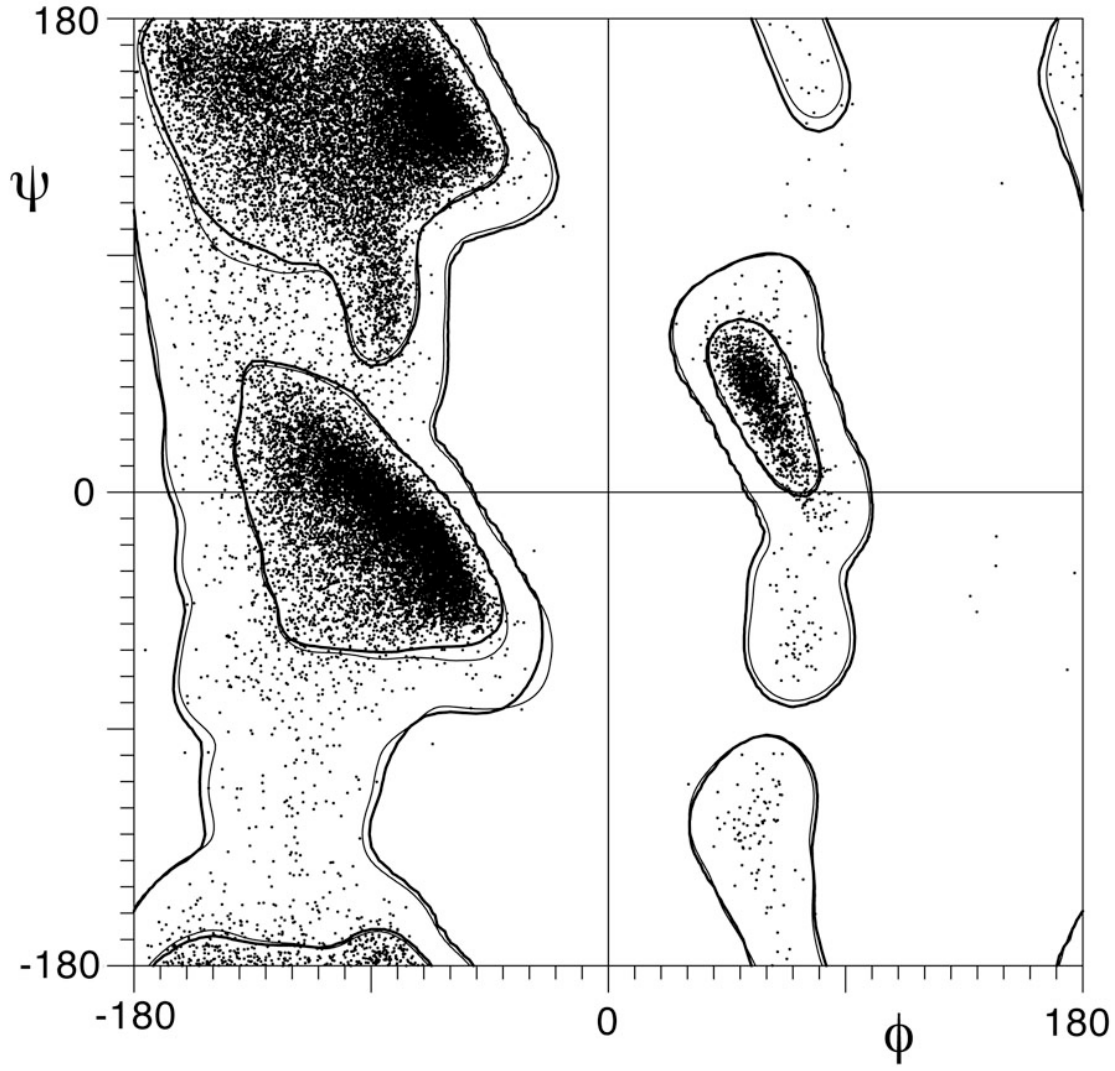


Figure 9: ϕ,ψ data points for general-case residues not in either helix or sheet secondary structure.

Shown for potential comparison with calculations of the local energetics of backbone conformation. Peak heights differ from Figure 3a, but the boundary contours (heavy lines) for non-repetitive residues are essentially indistinguishable from the general-case contours (thin lines).

The original Ramachandran calculations (Figure 1a) were purely steric, based on a hard-sphere model, with an outer contour from minor relaxation of bond angles. They show the three major areas clearly and form the basis for all later work. Our calculations of all-atom contact scores (Word, Lovell et al. 1999a) are also primarily steric, but have soft-sphere van der Waals repulsions and include an H-bonding term. All-atom contact scores as a function of ϕ,ψ calculated for Ala in ideal geometry are dominated by two deep elliptical troughs (negative scores are unfavorable) which cover the central region around $\phi = 0°$. These troughs involve truly dire atomic clashes of the O_{n-1} atom deserving the term "forbidden", while the unfavorable regions with positive ϕ are less extreme. In particular, there is a "shoal" that winds through $L\alpha$ and includes conformations such as γ-turn and II'-turn which are only slightly disfavored and which do occur in the empirical distribution.

Many energy calculations have been done for the Ala dipeptide in water; some of the excellent examples have used quantum mechanics (Peters and Peters 1981), a variety of molecular mechanics force fields (Roterman, Lambert et al. 1989), and estimates of free energy (D'Aquino, Gomez et al. 1996). All show the α, β, and $L\alpha$ regions and all show the "shoal" in positive ϕ, but the shapes and positions are not accurate (e.g., the strong diagonal shape of the α

and Lα regions is usually not evident). In particular, both steric and energetic calculations show as quite favorable a region left of α, near -150°,-60° on the ϕ,ψ plot, which is almost unpopulated in the empirical distributions. The two peptide NH groups are close in this conformation (see Figure 10), but the problem cannot just be electrostatic since those effects are included in all of the energy calculations. Our conjecture to explain this discrepancy between theory and observation is that the angle and crowding of the two NH groups near -150°,-60° permits H-bond donation to only a single acceptor, rather than the two H-bonds that are normally possible for two successive NHs .

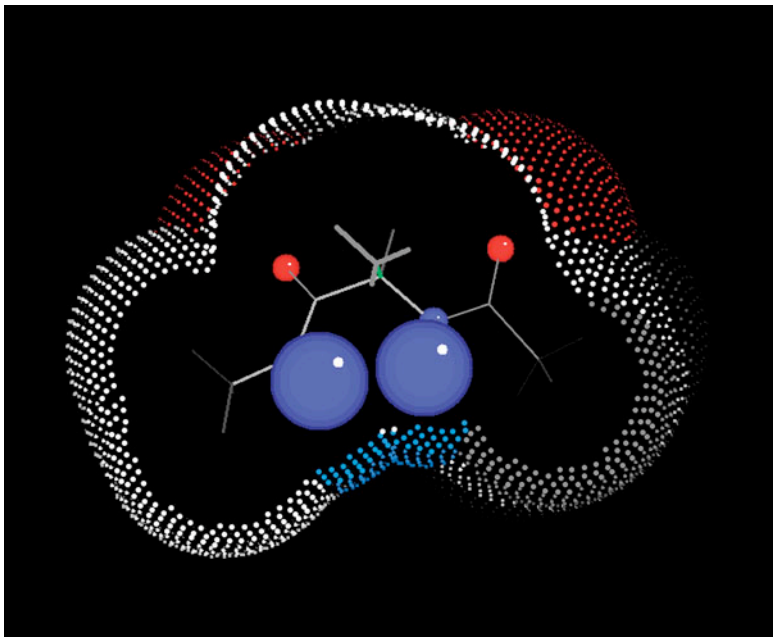


Figure 10: Stick figure and all-atom van der Waals surface for the conformation left of α near ϕ,ψ -150°, -60°.

The two peptide NH's (H balls in blue) are very close, their exposed surface (blue dots) is crowded by the surrounding Cα's and Cβ, and there is only room for one oxygen to approach as an H-bond partner.

Fortunately, there is now a new set of theoretical calculations, reported by Hu et al. (2003), which matches the empirical distribution (Figure 9) in very good detail. They performed dynamics simulations in which the Ala or Gly dipeptide portion was calculated quantum mechanically while the solvent and solvent-peptide energies were calculated with a molecular mechanics force field. Their theoretical distribution shows the forbidden troughs around $\phi = 0°$, the diagonal edges of α and Lα regions, and only a sparse population left of α (Figure 11). Even the Gly distribution matches the empirical data satisfactorily. Other recent work shows analogous improved matches for sidechain rotamers as well (Butterfoss and Hermans 2003; Butterfoss, Richardson et al. 2005). This achievement of closer agreement between theoretical and empirical dihedral distributions provides new hope that both approaches may now be converging toward correct treatments.

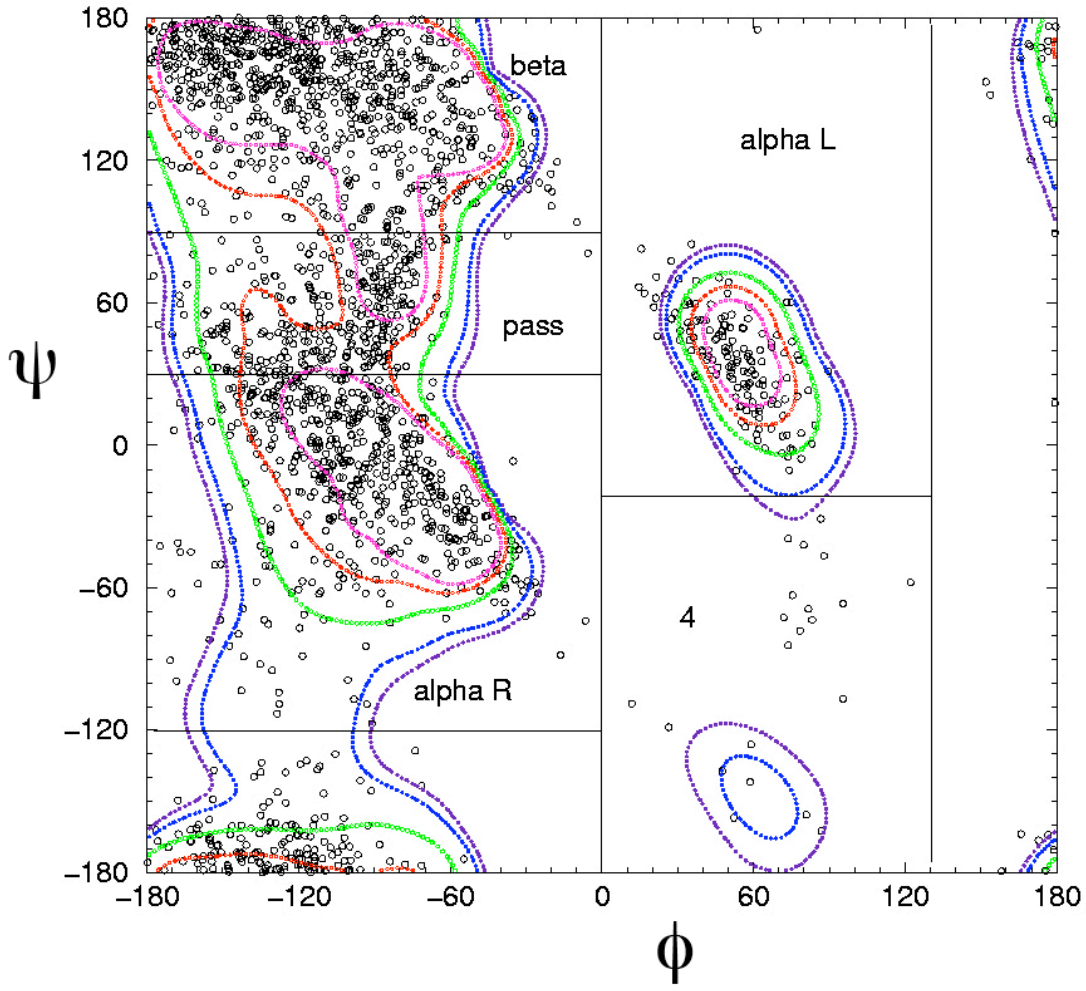


Figure 11: Comparison of theoretical and empirical distributions of backbone conformation.

Open circles show the sampled theoretical conformational distribution of Ace-Ala-Nme with QM/MM (SCCDFTB/amber). Successive contours enclose 99.8% (purple), 99.5%, 98%, 95%, and 90% (pink) of the empirical data points for alanine residues in non-repetitive secondary structure in the Top500 database. Figure from (Hu, Elstner et al. 2003).

Conclusion

One unifying theme of these proposed geometrical validation criteria is that bond angles and torsion angles are much more effective when analyzed in the appropriate local combinations than they are if treated individually. It has been evident from the first Ramachandran plot (Ramachandran, Ramakrishnan et al. 1963) onward to the current update that ϕ and ψ are not even approximately independent and must be analyzed together. The original insight of Ponder and Richards (Ponder and Richards 1987) in defining sidechain rotamers was that the sidechain angles are much more powerful if analyzed in combination rather than individually, and our recent rotamer analyses (Lovell, Word et al. 2000) confirm that principle even more strongly. RNA backbone provides a final example of this principle, where examining an angle in isolation may give a nearly flat distribution (e.g. ζ), but the same angle gives distinct clusters when analyzed with the other backbone dihedrals.

The geometrical structure-validation criteria described here are a revision and extrapolation of previous standards. The defined ϕ,ψ regions make a significant improvement by virtue of providing more stringent limits where that is needed but also by validating rare but quite acceptable conformations that most crystallographers have encountered at least once in an active site. It

is both appropriate, and perhaps even overdue, to use modern high-resolution, low-*B* data to define the standards which structures in general should aspire to approximate; these updated standards are especially compelling, because they have converged to agreement and remain constant from 1.8Å down to 0.5Å resolution and across all B-factor ranges <30. An acceptable protein structure at a given resolution can be defined by having suitably high overall percentages of favorable ϕ,ψ values and sidechain rotamers. However, the truly important claim was successfully demonstrated by work on a set of 29 structures from the Southeast Collaboratory for Structural Genomics (Arendall, Tempel et al. 2005): that by examining the outliers for each of those criteria in conjunction with all-atom contacts and electron density, and by correcting them when appropriate, essentially any protein structure can feasibly be rendered significantly more accurate than without these tools.

Chapter 3: The backrub motion

Background and introduction

Surprisingly, the frozen structures from ultra-high-resolution protein crystallography reveal a prevalent but subtle mode of local backbone motion — the "backrub" motion. This previously unrecognized, small-amplitude motion appears to be an influential type of local plasticity in protein backbone because it is coupled to much larger, two-state changes of sidechain conformation. The central sidechain changes accessible rotamers when concerted reorientation of two adjacent peptides swings it perpendicular to the chain direction, leaving the flanking structure undisturbed. In the alternate conformations of sub-1Å crystal structures, backrub motions account for 2/3 of significant Cβ shifts and 3% of the total residues in these proteins (126/3882), usually accompanied by two-state changes of sidechain rotamer and/or hydrogen bonding. Although this is an unconventional source of data on protein dynamics, high-resolution alternate conformations provide unmatched accuracy of atomic positions. Similar patterns in a variety of other structural data suggest that backrub motions are ubiquitous and play important roles in both protein dynamics and evolution.

A large body of experimental dynamics data, especially nuclear magnetic resonance (NMR) measurements (Wuthrich 1986; Cavanagh, Fairbrother et al. 1996), shows that a protein molecule in solution is quite mobile, at a range of size and time scales. A major driving force for residue-scale mobility is the constant bombardment by solvent and other molecules, felt especially by surface sidechains which dance between favorable conformations (rotamers) under that bombardment and transfer some of those forces to their local backbone. Indeed, sidechains are seen to be more highly mobile than backbone by NMR (Palmer 2004; Kay 2005), and surface sidechain mobility is also evident in crystallographic electron density maps even when it is not explicitly modeled. Analogous structural changes occur over the much longer evolutionary time scale, where the primary event is a sequence mutation (i.e., sidechain substitution) but the effects propagate to cause shifts in backbone conformation. The combination of exquisite packing (Word, Lovell et al. 1999a) and relaxed conformations (Lovell, Word et al. 2000; Lovell, Davis et al. 2003) in protein cores, along with the degeneracy of permissible sequences (Lim and Sauer 1989; Munson, O'Brien et al. 1994; Gassner, Baase et al. 1996), implies that the backbone must exhibit low-energy, localized modes of change

that co-adapt sidechain and backbone conformations to the new local structural requirements of sequence changes.

In either the dynamic or evolutionary case, however, even something as "simple" as backbone accommodation to local sequence or rotamer change is surprisingly difficult to model accurately. For large-scale backbone motions, various methods can produce approximate results close enough to be of practical utility: molecular dynamics (Lipari, Szabo et al. 1982; Karplus and McCammon 2002), elastic networks (Bahar, Erman et al. 1997), inverse kinematics (van den Bedem, Lotan et al. 2005), iterative simulation of fragment combinations (Rohl, Strauss et al. 2004), and systematic secondary-structure deformations (Qian, Ortiz et al. 2004). However, accurate prediction of local backbone changes has been elusive, which is a problem for several fields. Crystal structures of mutant proteins show conclusively that small backbone rearrangements are indeed very common (Baldwin, Hajiseyedjavadi et al. 1993; Matthews 1995), but energy calculations are still unable to predict those changes accurately. The detailed interpretation of backbone order parameters measured by NMR dynamics is hampered by the lack of reliable alternative models for local backbone motion. The notable successes of protein redesign (Desjarlais and Handel 1995; Dahiyat and Mayo 1997; Looger, Dwyer

et al. 2003) mostly depend on limitation to a completely rigid backbone scaffold taken from a known natural structure. Similarly, homology modeling (Tramontano and Morea 2003) generally works best if the core backbone of the template structure is left unchanged, although we know that the backbone will in fact accommodate somewhat (Mooers, Datta et al. 2003). Modeling of small local backbone motion is usually done either with molecular dynamics or with a set of pre-defined geometrical "moves" such as peptide flips or crankshaft ϕ/ψ motions (Fadel, Jin et al. 1995). Problems with these approaches include allowing unrealistic backbone motions (Hu, Elstner et al. 2003), holding fixed the sidechains and surrounding structure, and not explicitly considering correlated motions of adjacent peptides or coupling with changes of sidechain conformation.

Our attention was first drawn to backbone-sidechain coupling when manipulating brass "Kendrew" models, where a correlated twist of adjacent peptides is effective for swinging Cα-Cβ bonds perpendicular to a β strand. Similar shifts of Cβ directionality were invoked to account for altered near-neighbor packing of aromatic residues in the lab's protein design work (Richardson, Richardson et al. 1992). When the rebuilding of backward-fit sidechains in crystal structures required adjustment of the Cα-Cβ direction

(Richardson, Arendall et al. 2003), we investigated plausible backbone motions and prototype software tools to accomplish those changes (Methods; see also (Noonan, O'Brien et al. 2004)). These small backbone and Cβ shifts have been an important factor in the success of our structure-improvement methods (Arendall, Tempel et al. 2005). Until now, however, there was no direct evidence as to whether this motion actually occurs in the molecules themselves.

This chapter describes the small-scale, local "backrub" motion; shows that the backrub is the most common local backbone change seen in ultra-high-resolution protein structures; provides evidence for the same type of change over evolutionary time; describes the BACKRUB algorithm for modeling such shifts; and shows that this low-energy, small-amplitude concerted shift of the backbone atoms in two successive peptides is coupled to much larger-scale, two-state change of conformation for the central sidechain (Davis, Arendall et al. 2006). Backrub motions are also analyzed in the context of examples from point mutant structures, evolutionary homologs, non-crystallographic symmetry pairs, and NMR or molecular dynamics ensembles. The backrub constitutes one important example of a new paradigm of local backbone motion that is both demonstrably realistic and explicitly sidechain-coupled.

Methods

Survey of ultra-high-resolution crystal structures

The database consisted of all proteins with deposited structure factors at ≤ 0.9Å resolution available in the Protein Data Bank (Berman, Westbrook et al. 2000) as of May 2004, excluding duplicates at ≥ 50% identity and short peptides with unusual amino acids. The resulting 19 proteins, containing 3882 residues, are listed by resolution in Table 2 (see Results). All 19 structure determinations used synchrotron data collected at cryogenic temperatures and refined anisotropic B-factors. Phasing methods varied, but all except two were refined with ShelXL (Schneider and Sheldrick 2002).

Sidechain rotamers are defined and named as for the Penultimate rotamer library (Lovell, Word et al. 2000); "p", "t", and "m" in those names refer to χ angles near +60°, 180°, and –60° respectively. Structure analysis and creation of display files was done on-line in the MOLPROBITY web service at http://kinemage.biochem.duke.edu (Richardson, Arendall et al. 2003; Davis, Murray et al. 2004). If not already present, all hydrogen atoms were added and optimized with REDUCE (Word, Lovell et al. 1999b), but without flips of Asn/Gln/His orientation. All-atom contacts were calculated by PROBE (Word, Lovell et al. 1999a), giving scores and displays for H-bonds, van der Waals

contacts, and the rare instances of bad steric overlap in these high-accuracy structures. Kinemage 3D display files were made with the "multi-criterion" function in MOLPROBITY, which highlights alternate conformations, poor rotamers (Lovell, Word et al. 2000), Ramachandran outliers (Lovell, Davis et al. 2003), and serious all-atom clashes (overlaps ≥ 0.4Å), with user-controllable extra detail. The multi-criterion kinemages were viewed in the KiNG Java display program (Davis, Murray et al. 2004), along with $2F_o$-F_c and F_o-F_c maps obtained from the Electron Density Server at http://eds.bmc.uu.se (Kleywegt, Harris et al. 2004).

For each protein, Table 2 lists the number of residues for which an alternate conformation was defined for at least one atom (569/3882 residues total, or 15%). Alternate conformations involving concerted movement for the backbone of two or more neighboring residues were not analyzed in this study; omitting those 24 cases (93 residues) gives the 476 single-residue alternates. All single residues with alternate conformations defined for the Cβ atom (404 residues) are potential candidates for backbone movement, because a true shift of Cβ means the backbone must have moved, although small backbone changes are often modeled only as anisotropic B factors. After a preliminary survey, it was determined that a shift of at least 0.2Å in Cβ

position was necessary in order to draw unambiguous conclusions about the specific changes in backbone conformation. 200 single-residue alternates had Cβ shifts ≥ 0.2Å (see Table 2). 34 of them were omitted; half were not relevant because both sidechain conformations could actually be well fit from a single ideal Cβ position, implying no backbone motion (e.g. 1N9B Leu30, shown in the digital appendix; 1SSX Met213), while for the others electron density for the lower-occupancy conformation was visible for too few atoms or too poorly-shaped to allow reliable inferences about the geometry of backbone changes (e.g. 1GWE Val217, 1EJG Pro36, 1US0 Glu64 and Lys307). The 166 clearly interpretable single-residue examples with verified Cβ shifts ≥ 0.2Å were then classified as exhibiting either backrub movement (as described below) or some other type of movement.

Definition of backrub movement and BACKRUB modeling

A backrub motion shifts the position of the Cα-Cβ bond vector for residue *i* by a backbone change that is low energy (i.e., maintains essentially ideal bond lengths, bond angles, and peptide planarity) and is purely local (i.e., with essentially no motion of $C\alpha_{i\pm1}$ and none at all beyond $C\alpha_{i\pm2}$). The BACKRUB modeling algorithm, described below, closely satisfies these constraints using 3 rotations around Cα-Cα axes. Successful modeling of a putative backrub

motion using the BACKRUB software means that all clearly visible sidechain and backbone atom positions in both conformations can be well fit by two states of a single, closely-ideal model which differ only by adjustment of the 3 variable BACKRUB parameters, plus χ angles for sidechain *i*.

CHIROPRAXIS algorithm for intial exploration

A preliminary exploration of backrub motions was done by our CHIROPRAXIS algorithm, a brute-force sampling of ϕ,ψ,τ space that builds model halves separately inward from $C\alpha_{i-1}$ and $C\alpha_{i+1}$ to meet at a resulting $C\alpha_i$ within 0.02Å, disallowing Ramachandran outliers and limiting non-ideality of τ (Figure 12). This search produces a fan of permissible conformations that swing the central Cα–Cβ vector perpendicular to the chain direction, as anticipated, with essentially no Cβ motion in other directions. However, the ϕ,ψ plots of Figure 12 show that the relationships are non-linear and complex. The curve shapes differ for different starting conformations and for residue *i*-1 versus *i*+1, and they show much more spread with variation of $\tau_{i\pm1}$ in the β than in the α region. (Banding is an artifact of a 3° sampling interval in τ.) Interestingly, ϕ and ψ change very little for the central residue *i* whose sidechain is moving. These complex relationships prevented derivation of an analytical expression for this motion.

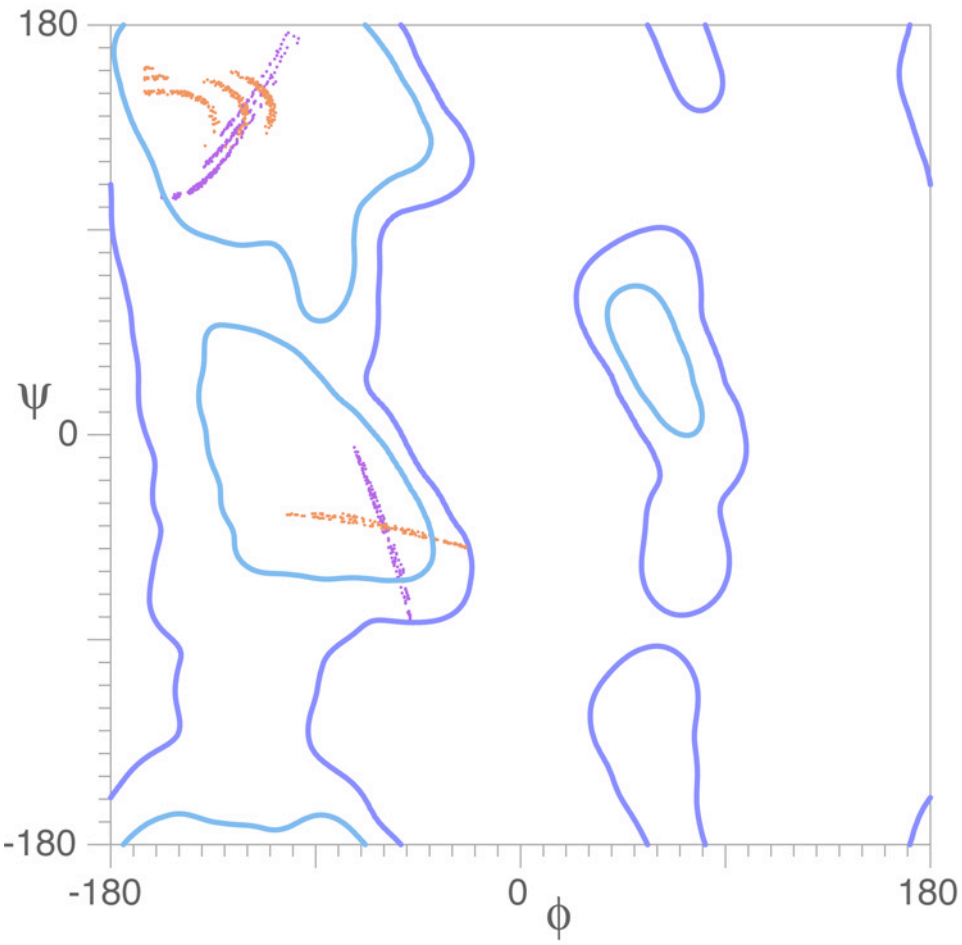


Figure 12: A brute-force exploration of the backbone conformations.

ϕ,ψ values for the *i*+1 (*orange*) and *i*-1 (*purple*) residues of conformations generated by a brute-force search of ϕ,ψ,τ space with invariant $C\alpha_{i-1}$ and $C\alpha_{i+1}$ positions; starting from either ideal α-helix (below center) or ideal β-sheet (top left). The ϕ,ψ angles are plotted within the contours of the updated Ramachandran plot from Lovell et al. (Lovell, Davis et al. 2003); parallel streaks of points result from coarse sampling of τ. All four dihedrals display complex, nonlinear relationships that are highly dependent on the starting conformation.

Expanding the search area to five residues (i.e., $C\alpha_{i-2}$ to $C\alpha_{i+2}$) allows the previously fixed $C\alpha_{i\pm1}$ to move a small distance from their starting points. This has little effect on the available range of motion for the center residue; it does, however, allow the center sidechain to tilt ~5° to either side (along the chain direction). This could become significant for longer sidechains, and may explain the "diagonal" motion of some non-backrub alternate conformations in the survey of high-resolution structures.

When CHIROPRAXIS creates conformations for the two model halves independently, up to ~10^6 conformations can be enumerated for each side (~10^{12} conformations total). Many of these can be automatically culled because they cannot possibly connect up with any conformation on the other side; even so, stochastic sampling was necessary to connect up the two halves in a reasonably efficient manner.

Given these constraints, we had to allocate sampling among the various bond angles and dihedrals. ϕ and ψ are the primary descriptors of backbone conformation and the only ones with a large, relatively free range of motion. However, we found that introducing slight variation in the other parameters (±1 standard deviation (Engh and Huber 1991)) led to a considerably wider range of conformations. Varying τ (N-Cα-C) produced the largest result.

Varying Cα-C-N and/or C-N-Cα angles produced less effect individually and almost no additional effect when combined with τ variation. On the other hand, varying ω (the peptide dihedral) increased the range of conformations almost as much as τ, but again it added little value when combined with τ variation. Thus, we chose to approximate all the small distortions of protein backbone with variation in a single bond angle, τ. Incidentally, the BACKRUB algorithm also introduces bond angle distortions into τ, and only into τ. This echoes an observation made 30 years before in considering the loop closure problem, where it was shown that introducing small amounts of variation in the bond angles led to more reproducible and realistic results (Bruccoleri and Karplus 1985).

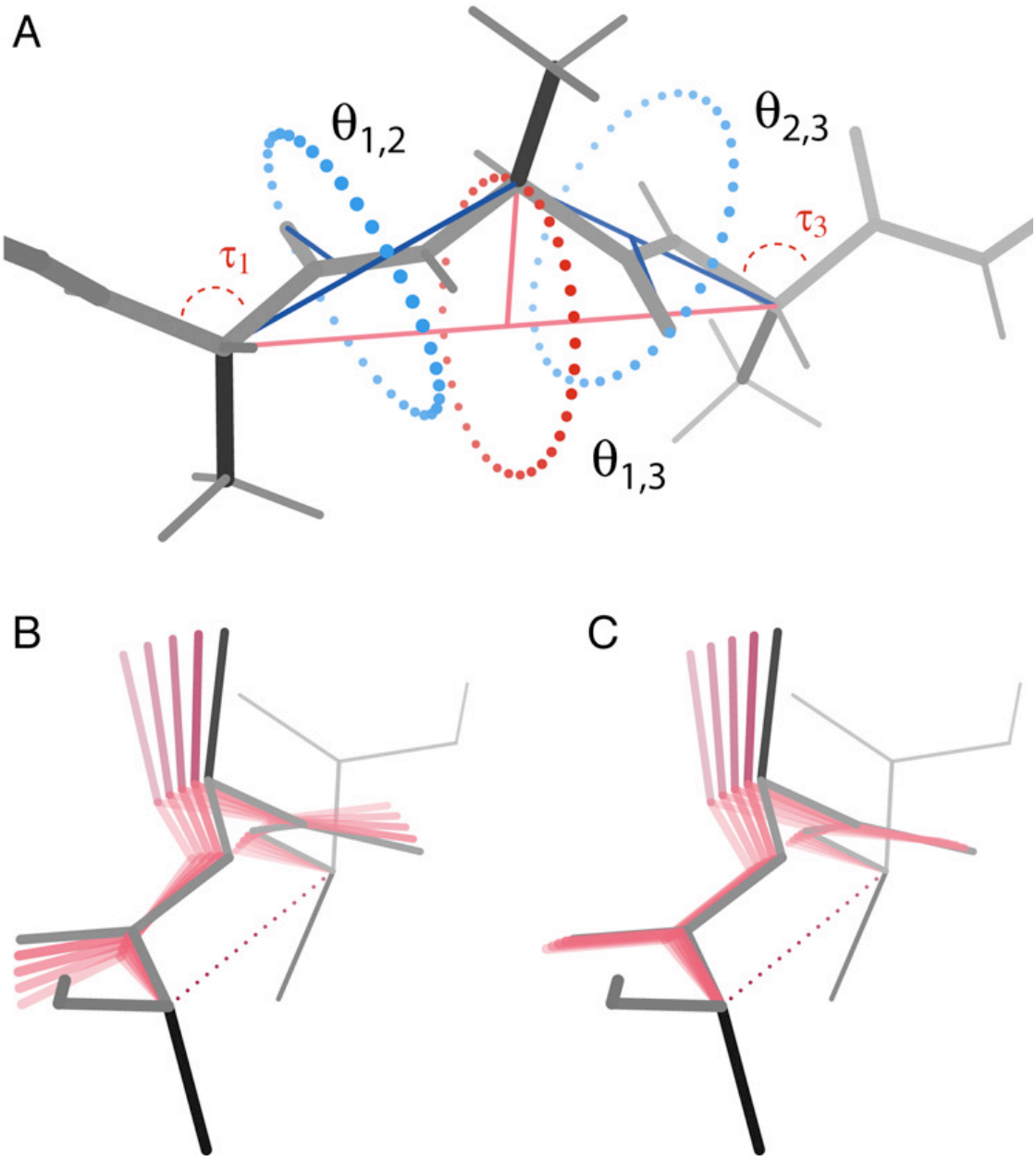
BACKRUB algorithm for software tools

The simplified BACKRUB algorithm (Figure 13) was developed for practical use, with only three independent variables. It produces an extremely close approximation of a backrub motion by user control of rigid-body rotations around Cα-Cα vectors: a two-peptide rotation and two single-peptide rotations. The BACKRUB algorithm was implemented in Java as a tool in KiNG (see Appendix A). Its use requires specifying the appropriate PDB-format coordinate file (with hydrogens) and activates a dialog box to control the

rotations, with informational displays that warn of Ramachandran outliers (Lovell, Davis et al. 2003) or τ angles > 1σ from ideal (Engh and Huber 1991). The BACKRUB tool can be active at the same time as the sidechain-rotamer and rotation tool in KiNG, which is similar to the sidechain tool described for MAGE (Word, Bateman et al. 2000; Richardson, Arendall et al. 2003). The KiNG software and related programs are available, free and open-source, from http://kinemage.biochem.duke.edu.

Figure 13: A schematic diagram of the BACKRUB motion.

(a) The primary rotation ($\theta_{1,3}$) moves the central residue and its adjacent peptides around the red axis ($C\alpha^{i-1}$ to $C\alpha^{i+1}$) as a rigid body, causing the central Cα to trace out the dotted circle. Secondary rotations ($\theta_{1,2}$ and $\theta_{2,3}$) move the individual peptides as rigid bodies around the blue Cα-Cα axes. A small amount of distortion is introduced into the τ angles (N-Cα-C), but they generally remain well within the range of values seen in typical crystal structures. (b) A series of backbones generated with BACKRUB by making 5° steps around the primary rotation axis (hydrogens not shown). (c) Another series of backbones generated with BACKRUB by making 5° steps around the primary rotation axis, while also rotating each peptide to roughly maintain the H-bonding position of the NH and CO groups.


A
θ1,2
θ2,3
τ3
τ1
θ1,3
B
C

The geometry used by the BACKRUB algorithm is diagrammed in Figure 13a. It acts on a central residue *i* and the two flanking peptides. For each of three different rotational components, a subgroup of atoms moves as a rigid body around some (virtual) axis. The primary component of motion rotates all atoms between the Cα's of residue *i*-1 and *i*+1 around an axis between $C\alpha_{i-1}$ and $C\alpha_{i+1}$. This produces a wide arcing motion of the sidechain roughly perpendicular to the overall local chain direction. The two secondary components rotate the four central atoms of a peptide group around an axis between the Cα's on either end, which helps alleviate τ-angle or H-bond strain introduced by the primary motion. All bond lengths and angles are invariant except for the three τ angles. In Figure 13b the substates differ only by the primary rotation, while in 13c the peptides have been rotated to help preserve the H-bonding CO and NH positions.

The BACKRUB algorithm closely approximates the common backrub mode of local backbone plasticity (see Discussion). However, it should be noted that there are many other changes in backbone conformation to which it is not applicable. Its area of effect is deliberately small, and so it is not suited for motions of large chain segments. Since it assumes fixed anchor points at either end, it cannot be used for domain hinges or immediately next to chain ends. A

generalization of BACKRUB to act between arbitrary Cα's was explored, but it introduces further complications while adding only a few additional successes at fitting changes in longer loops (see Discussion).

To aid in the assignment of backrub motion vs. other motion for each alternate conformation example, modeling of changes was tested using the BACKRUB tool in KiNG for more than half of the 166 cases, including all large or complex motions and multiple examples of each recognizable pattern of change (e.g., serines similar to Fig. 14b). If backbone atoms had been assigned alternate positions in the PDB file, then the conformation with more ideal peptide geometry (usually A) was taken as the starting point for modeling the second (B) alternate conformation, using only the three BACKRUB rotations and the sidechain χ dihedrals with idealized sidechain geometry, and emphasizing fit to the atoms most clearly observed in the electron density. In some cases, both original conformations had substantial but opposite distortions in covalent geometry; a more nearly ideal intermediate would make a better BACKRUB starting point, but we very seldom added such a step (e.g. Kin. 6 in the digital appendix). Criteria of good fit were the same as would be applied in crystallographic rebuilding at this resolution. If alternates had been assigned starting only at Cβ, then the common backbone conformation was

taken as the starting point, with an idealized Cβ (Lovell, Davis et al. 2003) (usually about halfway between the two assigned Cβ positions) and ideal-geometry sidechain (Engh and Huber 1991); then both A and B conformations were modeled using BACKRUB motions in opposite directions. These BACKRUB models are shown in orange in the figures, where their fit to deposited models and to electron density can be judged.

For correction of an experimental model misfit into the wrong local-minimum conformation during crystallographic refinement (e.g., Figure 18), first the new sidechain rotamer was chosen, next the backbone was shifted with the BACKRUB tool, and finally both backbone and sidechain rotations were adjusted to optimize all-atom contacts and electron density fit.

Methods for comparing independent structures

To look for local backbone motions that occur over evolutionary time, one must superimpose two or more independently determined structures. The choice of superposition method and parameters can have a significant effect on the perceived differences between structures. Furthermore, the median Cα travel for backrubs observed in the alternate conformation data was 0.2Å, equal to the best Cα RMSDs obtained when comparing evolutionarily related structures. Thus, backrub motions and any similar conformational shifts are

nearly lost in the noise of independent structure determinations (DePristo, de Bakker et al. 2003); alternate conformations have the advantage of totally avoiding this type of noise.

The best superposition for detecting local backbone shifts depends on the details of the structures to be compared. If the core structure of the protein is rigid and the shifts of interest occur within the core, then superimposing on all the core Cαs gives a more robust alignment. If, on the other hand, there are global "breathing" motions or the shift of interest occurs in a mobile (sub)domain, a local superposition on a subset of Cαs may be more appropriate. For alignment over the whole structure, either we used the LOCK2 web server (Shapiro and Brutlag 2004), or we used Lesk's sieve-fit method (Lesk 1991) to choose 75% of Cαs as the alignment core, followed by least-squares alignment in LSQMAN or ProFit. For local alignments, we used a least-squares superposition with an interactively selected set of 3-6 Cαs in the immediate vicinity of a putative backbone shift. Those details are described on a case-by-case basis below.

There do exist techniques for comparing structures without superimposing them, such as difference-distance plots (Richards and Kundrot 1988) and using internal coordinates (bond angles, dihedrals, etc). Indeed, distance-

difference plots are often useful for spotting domain-hinging motions that would complicate a superposition attempt. However, it is difficult to directly determine the nature of a local backbone shift directly from such plots. Also, for residues on the surface (where most backrubs occur) information is distributed anisotropically in space (i.e., all reference points are in the protein core and none are in the solvent). Internal coordinates seem appealing, but in practice there is too much random fluctuation in these parameters to make them useful. Furthermore, there is no direct correspondence between the magnitude of change in an angle and the magnitude of real-space motion; for instance, it is often difficult to distinguish significant structural changes from the noise when looking at $\Delta(\phi,\psi)$ plots (e.g. Figure 24).

Results

We first present a survey of backbone plasticity as revealed by the alternate conformations in a set of sub-1Å x-ray crystallographic structures totaling nearly 4000 residues. These data show that backrub motions are quite common, and that the BACKRUB algorithm is sufficient to create realistic models that closely match the experimental observations. We then illustrate the process of using BACKRUB for refitting during the structure determination process. Finally, we show evidence for backrub motions in point mutant

structures, evolutionary homologs, non-crystallographic symmetry pairs, and NMR and molecular dynamics ensembles. A table of examples, 3D kinemage graphics, selected coordinates, and a README file are available in the digital appendix.

Table 2: The set of protein structures surveyed for alternate backbone conformations.

Alt res: residues with one or more atoms in alternate conformations

Single alts: alternate conformations that encompass at most one residue plus its peptides

Cβ >0.2Å: single alternates with Cβ positions separated by at least 0.2Å in the deposited model

Classifiable alternates: alternates with verified Cβ >0.2Å and clear enough electron density to infer mode of backbone change reliably

Backrubs: classifiable alternates that displayed backrub motion instead of some other type of motion

* PDB file 1IX9 has no published journal reference; the depositors are B.F. Anderson, R.A. Edwards, M.M. Whittaker, E.N. Baker, and G.B. Jameson (2002).

PDB code	Description	Reference	Resol. (Å)	Total res.	Alt res.	Single alts	Cβ >0.2Å	Classifiable alternates	Backrubs
1EJG	Crambin (valence electron density)	(Jelsch, Teeter et al. 2000)	0.54	48	19	17	7	6	**5**
1UCS	Type III antifreeze protein RD1	(Ko, Robinson et al. 2003)	0.62	64	6	6	0	0	**0**
1US0	Human aldose reductase, with NADP+ and inhibitor	(Howard, Sanishvili et al. 2004)	0.66	314	79	45	18	16	**10**
1R6J	Syntenin PDZ2	(Kang, Devedjiev et al. 2004)	0.73	82	21	12	3	2	**2**
3AL1	Designed peptide α1, racemic P1-bar form	(Patterson, Anderson et al. 1999)	0.75	24	11	11	7	7	**6**
1IUA	*Thermochromatium tepidum* HiPIP	(Liu, Nogi et al. 2002)	0.80	83	6	6	2	2	**2**
1PQ7	Trypsin at pH 5 in borax	(Schmidt, Jelsch et al. 2003)	0.80	224	15	12	7	7	**5**
1NWZ	Photoactive yellow protein	(Getzoff, Gutwin et al. 2003)	0.82	125	25	12	8	7	**6**
1N55	E65Q mutant of *Leishmania* triosephosphate isomerase, with 2-phosphoglycolate	(Kursula and Wierenga 2003)	0.83	249	33	31	22	18	**9**
1SSX	α-lytic protease at pH 8	(Fuhrmann, Kelch et al. 2004)	0.83	198	23	23	17	13	**11**
1MC2	K49 phospholipase A2 homologue	(Liu, Huang et al. 2003)	0.85	122	10	10	4	3	**3**
1G6X	Bovine pancreatic trypsin inhibitor with mutated loop	(Addlagatta, Krzywda et al. 2001)	0.86	58	12	12	2	2	**2**
1MUW	Xylose isomerase	(Fenn, Ringe et al. 2004)	0.86	386	79	70	24	16	**12**
1DY5	Deamidated bovine pancreatic ribonuclease	(Esposito, Vitagliano et al. 2000)	0.87	246	39	39	23	22	**21**
1GWE	*Micrococcus lysodeikticus* catalase	(Murshudov, Grebenko et al. 2002)	0.88	498	47	40	9	8	**5**
1F9Y	*E. coli* HPPK with MgAMPCPP and 6-hydroxymethylpterin	(Blaszczyk, Li et al. 2003)	0.89	158	21	21	3	3	**2**
1IX9	*E. coli* Mn(III) superoxide dismutase mutant	*	0.90	410	41	34	18	15	**11**
1N9B	Extended-spectrum SHV-2 β-lactamase	(Nukaga, Mayama et al. 2003)	0.90	265	53	51	19	14	**11**
1OEW	Native endothiapepsin	(Erskine, Coates et al. 2003)	0.90	328	29	24	7	5	**3**
total				**3882**	**569**	**476**	**200**	**166**	**126**

The 19 structures surveyed are described in Table 2. Of alternate conformations that shift backbone for a dipeptide or less, 76% are backrub motions (126/166, listed in the digital appendix to this thesis). Globally, then, 3.2% (126/3882) of all the residues in these structures are well modeled as a backrub change in backbone conformation. This is a conservative estimate, omitting examples that can be modeled successfully without Cβ shift (e.g. 1N9B Leu30, Figure 15), but not considering the complementary cases best modeled as significant Cβ shifts but not deposited as such (e.g. 1PQ7 Cys57, Figure 16). Serine is far the most common amino acid seen, making up 25% of backrubs. Two factors could explain their predominance: Ser is the smallest rotatable sidechain, and it has many choices of donor or acceptor H-bond partners, some of which are local enough to constrain these slight backbone shifts. Most alternate conformations of any type occur at the protein surface, and polar sidechains predominate; Lys, Arg, and Glu contribute 8% of backrubs each. Of the hydrophobics, Val, Met, and Leu occur most often.

Other types of backbone motions observed include peptide flips (a 3-peptide change in which the central peptide rotates by 90-180°), usually in tight turns. They can be identified even within long stretches of concerted motion, but are still much rarer than backrubs, with only 4 cases in this dataset: 1GWE 105,

1IX9 135, 1MUW 174, 1US0 93. Intermediate rotations of single peptides are identifiable from large displacements of the carbonyl O density: many can be fit well with the BACKRUB tool (e.g. 1PQ7 Ser132, 1US0 Arg40) although the sidechain does not always move (e.g. 1N9B Ser41, which preserves two helix N-cap H-bonds through waters; coordinates in the digital appendix). The single-residue "other" cases include four large movements of chain-terminal residues, which are unconstrained on one side and thus have very different properties.

Figure 14a and b compare two different relationships seen for alternate-conformation serines, the most frequent backrub amino acid. These and other examples are available as animated, 3D kinemage graphics in the digital appendix. Figure 14a shows Ser34 in an αβ loop of the 1N9B TIM barrel, with a large Cβ shift of 1.02Å. Alternates had been defined for all atoms in residue 34 but not the rest of the two peptides. This represents the changes reasonably well but gives highly non-planar peptides for which there is no direct evidence; a pair of BACKRUB models fits the density equally well with nearly ideal geometry. Electron density for the carbonyl oxygen is highly anisotropic, indicating a single-peptide rotation that reinforces the primary backrub rotation rather than the usual compensation that preserves backbone H-

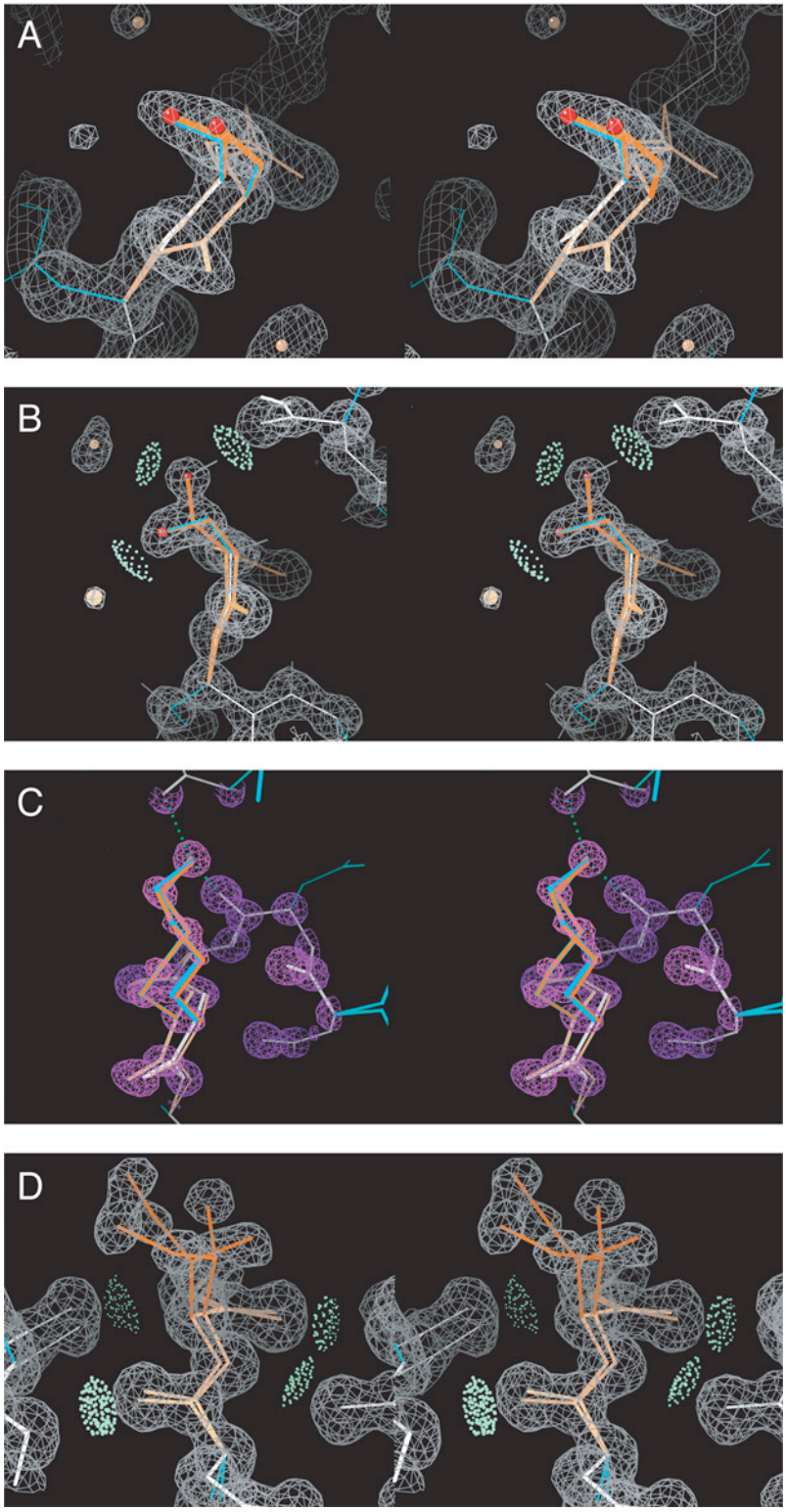
bonding (Fig. 13c). Ser34 alternate conformations are coupled to those of Arg250 (not shown), making or breaking an H-bond to the Ser Oγ. The Ser sidechain moves but does not change rotamer conformation; this is unusual, since 108 out of 126 backrub cases (86%) show distinct sidechain rotamers, as is also true for most single-residue "other" cases. 87 of the backrubs with distinct rotamers have χ1 in distinct local minima, as in Fig. 14b and d. Backbone alternates without a rotamer change usually show distinct H-bond states for either sidechain (Fig 14a) or backbone.

Figure 14b shows 1DY5 SerA15, a more typical serine backrub example with distinct rotamers and H-bonds from a clear 0.46Å Cβ shift, implying a backbone movement. Alternates were defined in the PDB file for Cβ but not for any backbone atoms (*white*), producing bond-angle distortions up to 9° around Cα and Cβ. Here also, a pair of BACKRUB models (*orange*) fits the electron density equally well with τ deviations under 1σ (3°) and completely ideal geometry otherwise. The asymmetrical T shape of the sidechain electron density for SerA15 is the usual pattern for serines with Cβ alternates (e.g. 1MUW Ser69, 1DY5 SerB50 in digital appendix). One conformation (here, with Oγ pointing left) has no strong positive or negative constraints, its rotamer is excellent, and its backbone conformation is presumably relaxed. The second

conformation makes a favorable but constrained H-bond (to a backbone CO) which is not accessible with a good rotamer and good geometry from the other Cβ position. Thus, the Cβ is pushed back significantly (causing a backrub shift) and the sidechain finds a compromise between H-bond strength and favorable χ1 angle. The Ser15 backbone CO stays in position and H-bonded. Ser15 in chain B shows the same pattern, which is usually but not always the case between subunit pairs with non-crystallographic symmetry.

Figure 14: Examples of backrub motions.

These examples were observed in the alternate conformations of atomic-resolution crystal structures for two serines (the most commonly occurring backrub residue), a lysine, and an isoleucine. Original models are in white and cyan; BACKRUB-fit models are in orange. $2F_O$-F_C maps contoured at 1.2σ are shown in gray; hydrogens are shown only in part b. See text for further details. (a) Ser 34 from 1N55. This residue moves in concert with Arg 250 to make/break an H-bond, but does not change rotamer. (b) Ser A15 from 1DY5, which changes both rotamer (χ1 *m* vs. *p*) and H-bonding state to nearby backbone and to waters. (c) Lys 100 from 1US0, with rotamers *mppt* and *mtmm* both ending at the same H-bonded Nζ position. Atoms Cβ, Cγ, and Cδ show clearly separated density peaks at 3σ (*purple contours*). (d) Ile 47 from 1N9B, in rotamers *tt* and *mm*. BACKRUB models were fit for both A and B alternate conformations (*peach/orange*). The deposited model (not shown here) was fit without backbone alternates but had a Cβ shift of 0.6Å and the same two sidechain conformations as shown.


A
B
C
D

Lys100 from 1US0 displays even clearer and more extensive alternate-conformation electron density (Fig. 14c), with separate 3σ peaks (*purple*) for three of the sidechain atoms in each conformation, including the Cβs 0.97Å apart. Strong anisotropy of backbone and Cε density indicates motion of those atoms also. Lys100 combines backbone and sidechain motion to leave Nζ in place (top right in Fig. 14c), maintaining two H-bonds to backbone carbonyls and a weaker interaction with Asn52 Oδ. The B alternate is a good rotamer (*mtmm*) (Lovell, Word et al. 2000) and the A alternate an acceptable one (*mppt*). Lys100 forms the C-cap of an α helix (Richardson and Richardson 1988), and its backbone H-bonds are apparently preserved by a concerted smaller backrub motion of Ser97. Backrub motions are less common and smaller within helices, where steric constraints from neighboring turns limit the magnitude of motion in the backrub direction. They are most common in β strands or extended loops. Another case of concerted backrub motion between two interacting residues is Tyr378 of 1GWE (see digital appendix), one conformer of which would intersect its 2-fold equivalent in another subunit if both were occupied simultaneously.

Fig. 14d shows Ile47 of 1N9B. Well-separated electron density peaks for all 6 Cγ and Cδ atoms in the two distinct sidechain rotamers unambiguously

mandate two Cβ positions and two backbone conformations, which are fit by the two BACKRUB models (*orange/peach*) with near-ideal geometry for both backbone and sidechain. This is a prototypical case in which the two peptides counter-rotate somewhat against the primary BACKRUB rotation and preserve all 4 β-sheet H-bonds (lenses of green contact dots, with both conformations overlaid in the figure). Another example of sidechain motion with backrub-maintained β sheet is Thr268 of 1R6J.

Figure 15, on the other hand, shows a case where alternate Cβs had been fit in error -- the backbone does not actually move. Leu30 in 1N9B had been modeled with a Cβ shift of 0.26Å diagonal to the local chain direction; it would not be well fit by a BACKRUB model. However, the density is fairly ambiguous, and a different and much more favorable pair of Leu rotamers fit the density better than the original and do not require any shift of the backbone or Cβ. Fitting errors like this were excluded from the tally of true backrub motions (Table 2).

In this analysis, several amino acid types are special cases. Glycine was ruled out by definition, since it has no Cβ to show a shift. Only one alanine is represented (1N9B Ala257), presumably because Ala sidechains have neither

rotamer nor H-bond states to drive local backbone shifts. Cystine alternates are frequently driven by breaking the disulfide; rotamers may or may not change (e.g. Figure 16, 1PQ7 Cys41; also 1N9B Cys123). Cys or Met alternates can often be analyzed at somewhat lower resolutions, if the heavy S atoms are clearly visible. Eight prolines have Cβ alternates fit >0.2Å apart, each with one alternate in each ring pucker (Cγ endo and Cγ exo). Half of them can be fit well by BACKRUB, including the skewed relationship of Cγ positions, but it is unclear whether that is the best description. Analysis of proline alternates is hampered by overlapping density for most atoms and by errors (Engh and Huber 2002) in geometry values currently used for proline (Engh and Huber 1991), which would cause some bias even at these high resolutions. Arginine displays especially varied and elegant patterns of alternate conformations because of guanidinium size and H-bonding (e.g. Figure 17, 1SSX Arg192; 1MUW Arg204).

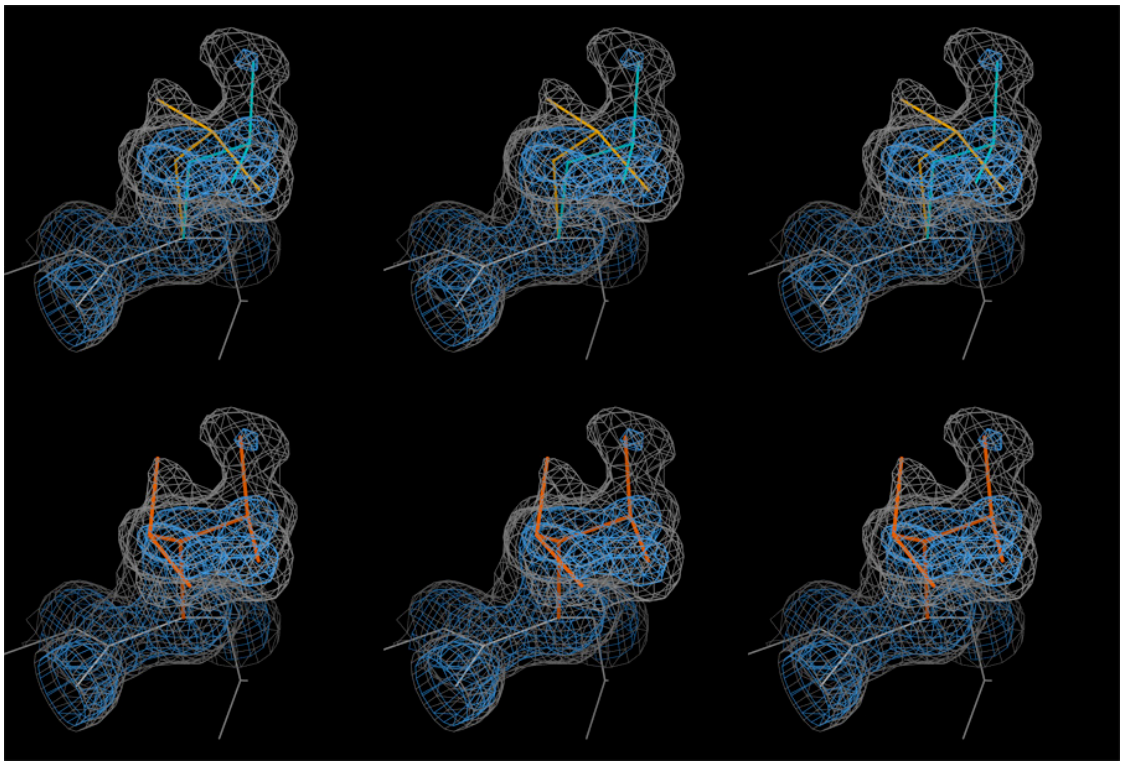


Figure 15: Example of a modeled alternate-conformation Cβ shift that does not actually require backbone motion.

Leu30 in 1N9B had been modeled with a Cβ shift of 0.26Å diagonal to the local chain direction (upper panel); it would not be well fit by a BACKRUB model. However, a different and much more favorable pair of Leu rotamers fit the density better and do not require any shift of the backbone/Cβ (lower panel). Left pair is cross-eye stereo, right pair is wall-eye stereo.

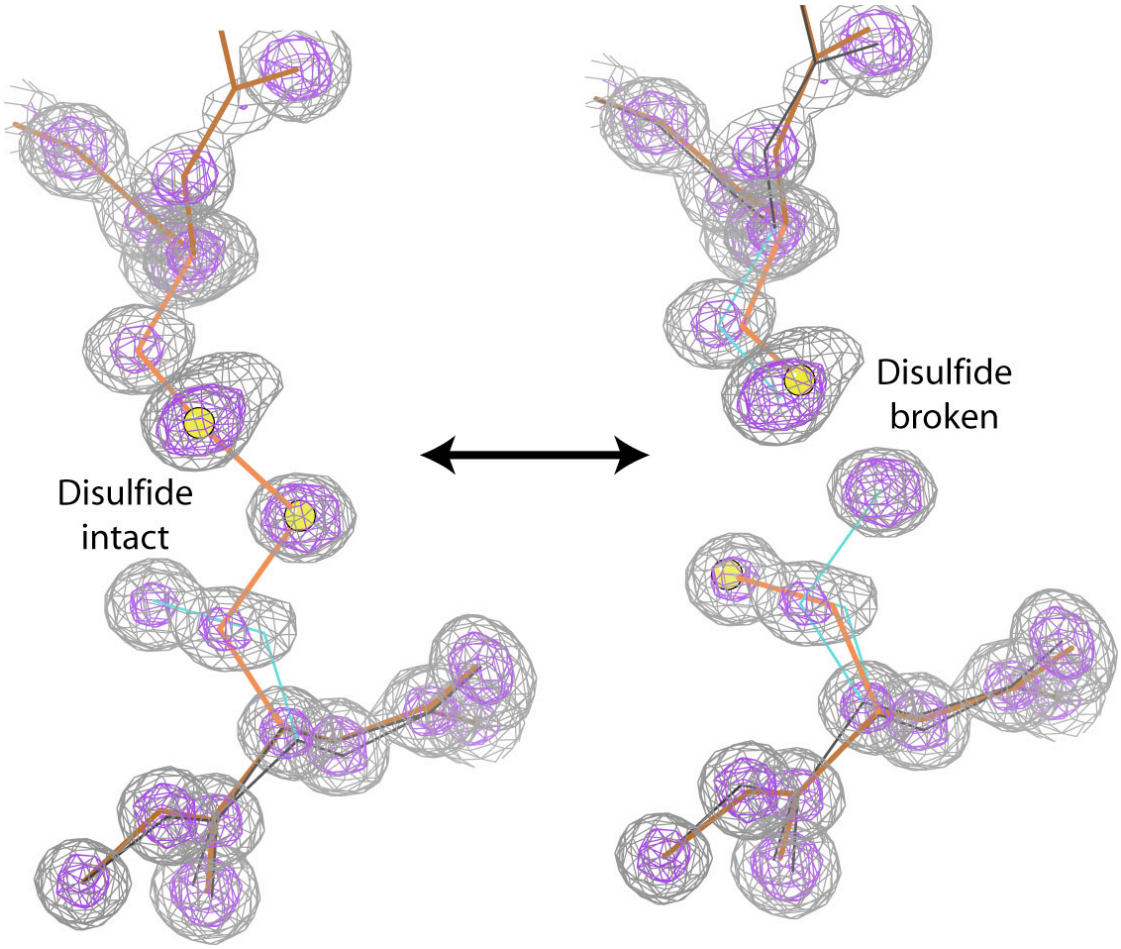


Figure 16: Backrub motion accompanies making / breaking of a disulfide bond.

Shown are Cys41 and Cys57 in 1PQ7. Note that 57 (top) was not originally modeled with two sidechain conformations, and so is not tallied as a backrub in Table 2.

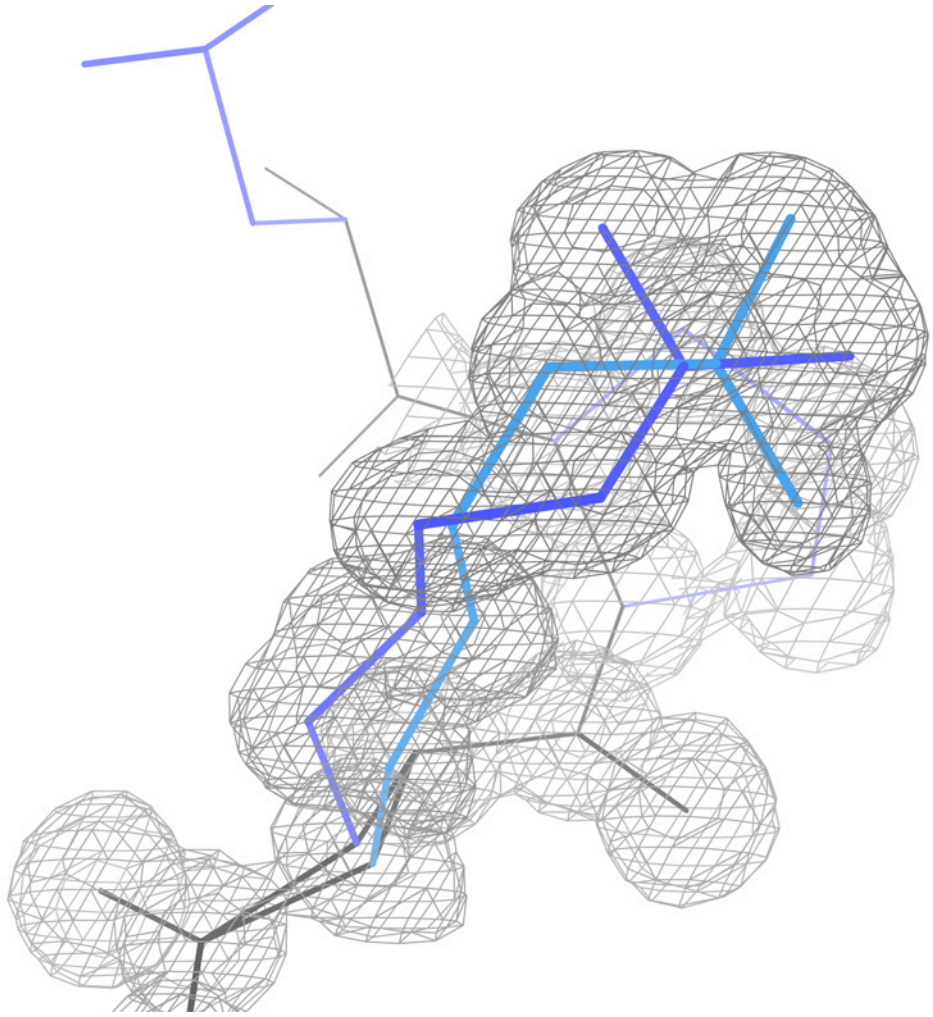


Figure 17: Elegant arginine alternates.

Shown is 1SSX Arg 120A, with separate electron density peaks for each of the six sidechain atoms. The guanidinium group occupies roughly the same space in either conformation, but coming from two distinct rotamers. This pair of alternates resembles the correction of an "arginine flip" misfitting sometimes found in lower-resolution x-ray structures. (Models shown are the deposited ones.) In this case, the alternates have different H-bonding patterns to nearby waters and a partial occupancy sulfate.

Figure 18 shows an example of using the BACKRUB software tool for model rebuilding during crystallographic refinement: IleA120 from 1MO0. In addition to representing real backbone plasticity as shown above, BACKRUB is also very effective for rebuilding because it moves one Cα and the neighboring peptides by a small amount, but dramatically alters the accessible rotamers by swinging the whole sidechain on a long lever arm. In the original conformation of Fig. 18a, all-atom contacts (Word, Lovell et al. 1999a; Word, Bateman et al. 2000) indicate severe steric clashes with surrounding atoms, and difference peaks in the F_o-F_c map suggest that χ1 is off by 120°. When the correct sidechain rotamer is fit on the original backbone (Fig. 18b), the serious steric clashes are all on one side, with space on the other side. A BACKRUB movement swings the sidechain to establish excellent packing interactions all around (Fig. 18c). Additional evidence strongly supports the correction: sidechain rotamericity improves, difference density is satisfied, and the new model matches Ile120 of chain B. Final confirmation comes from improved re-refinement, which has been done for many such cases (Arendall, Tempel et al. 2005).

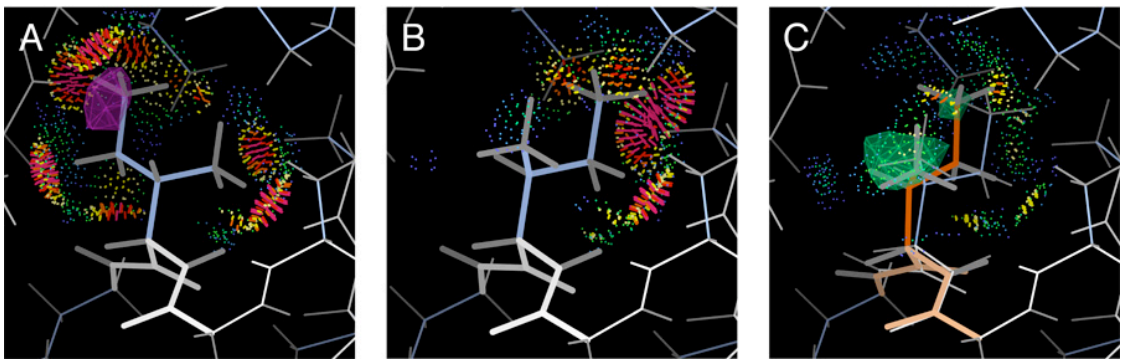


Figure 18: The BACKRUB tool in KiNG (Davis, Murray et al. 2004) was used to rebuild Ile A120 of 1MO0.

(a) The original conformation had serious steric clashes (*pink spikes*) with the surrounding residues and occupied a negative peak in the difference density (*magenta*). (b) The *pt* rotamer has clashes on one side and a small cavity on the other. (c) The BACKRUB model (*peach/orange*) shifts Ile A120 into that empty space and establishes good packing contacts (*green and blue dots*) with its neighbors, as well as satisfying positive peaks in the original difference density (*green*).

Backbone motions in other contexts

Although backrub motions are most clearly observed in the alternate conformations of high-resolution x-ray crystal structures, they also occur in other kinds of structural data. Since we expect backrubs to play a role in the evolutionary adaptation of structure to sequence, some point mutations should cause backrub motions when compared to the wildtype structure. Below, we show evidence for backrub motions in point mutant structures, evolutionary homologs, non-crystallographic symmetry pairs, and NMR and molecular dynamics ensembles.

In the PDB, there is a large group of point mutants of the *Staphylococcus aureus* nuclease, at resolutions from 1.6 - 1.9Å. These structures were analyzed at and around the mutation sites for evidence of local changes in backbone conformation. The most compelling example is 1EY0 (wildtype, 1.6Å) vs 1EY7 (S128A, 1.9Å). Despite their modest resolution compared to the structures analyzed for alternate conformations, these models superimpose on 101 Cαs (75%) to an RMSD of 0.08Å (presumably because 1EY7 was phased by molecular replacement from 1EY0 and had only local changes). As shown in Figure 19, Ser 128 falls in the middle of an α-helix, where it makes a sidechain hydrogen bond to Glu 101 in the neighboring helix. Mutating Ser 128 to Ala

seems to cause Glu 101 to move its H-bond one turn down the helix from the Ser Oγ to His 124, resulting in a backrub motion at both 124 and 128. The Cαs shift by 0.36 and 0.22Å, respectively; the Cβs shift by 0.46 and 0.31Å. These motions are small, but significant relative to the superposition RMSD of 0.08Å; the evidence is made more compelling by the change in H-bond partners, which is so often seen for backrubs in the alternate conformations.

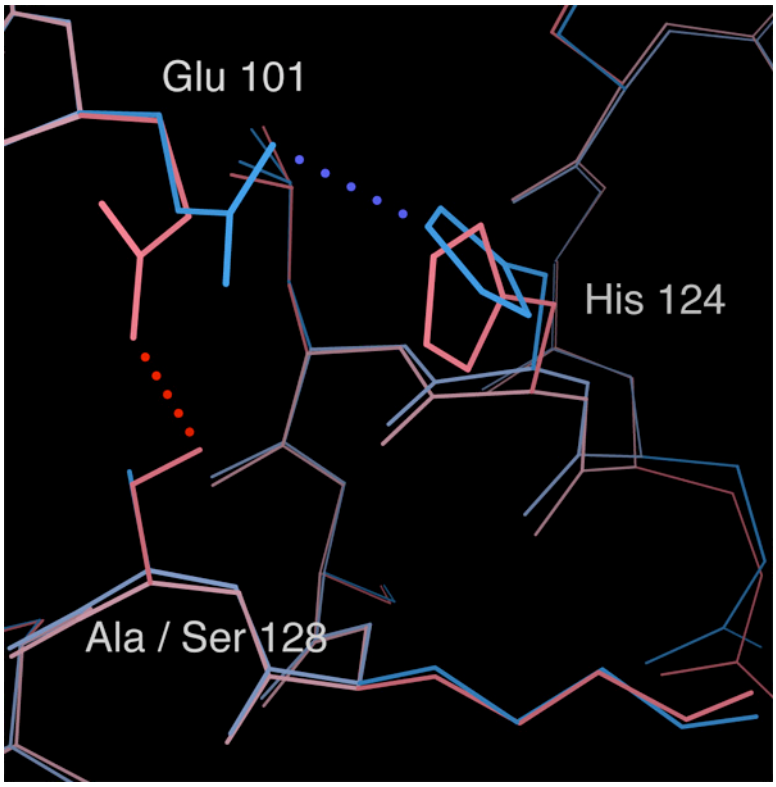


Figure 19: A comparison of Staph. nuclease structures reveals backrub motions caused by the S128A point mutation.

Ala128 packs closer into the helix since the H-bond to Glu101 is broken, while His124 shifts in order to make a new H-bond to Glu101. The motion at both positions 124 and 128 is well modeled by the BACKRUB algorithm.

Another trigger for backrub motions involves the packing of Phe and Tyr residues across the strands of a β-sheet, which depends on the sequence of the neighboring strand (Richardson, Richardson et al. 1992). If the Phe/Tyr sidechain rests on top of a glycine, it packs down closer to the sheet than if positioned over a bulky residue (Richardson, Richardson et al. 1992); Ala gives an intermediate result because its Cβ has only H projecting off it. To achieve these different packing positions, the aromatic sidechain needs to rotate slightly around the β strand, which is what happens with a backrub motion -- thus prompting this further study.

The streptavidin family provides an excellent example of this effect. Consider two mutants of wild-type streptavidin, 1SWU (Y43F, 1.1Å, *apo* form) and 1NBX (Y43A, 1.7Å, *apo* form). In the Ala 43 variant, Phe 29 from the neighboring β-strand packs its ring down on top of the Ala Cβ (Figure 20a). In the Phe 43 variant, Phe 29 has to rock back and also change rotamer in order to accommodate this much bulkier neighbor. The backrub motion for Phe 29 is obvious with a global superposition (0.43Å Cα RMSD), but it becomes even clearer with a 6-Cα local superposition.

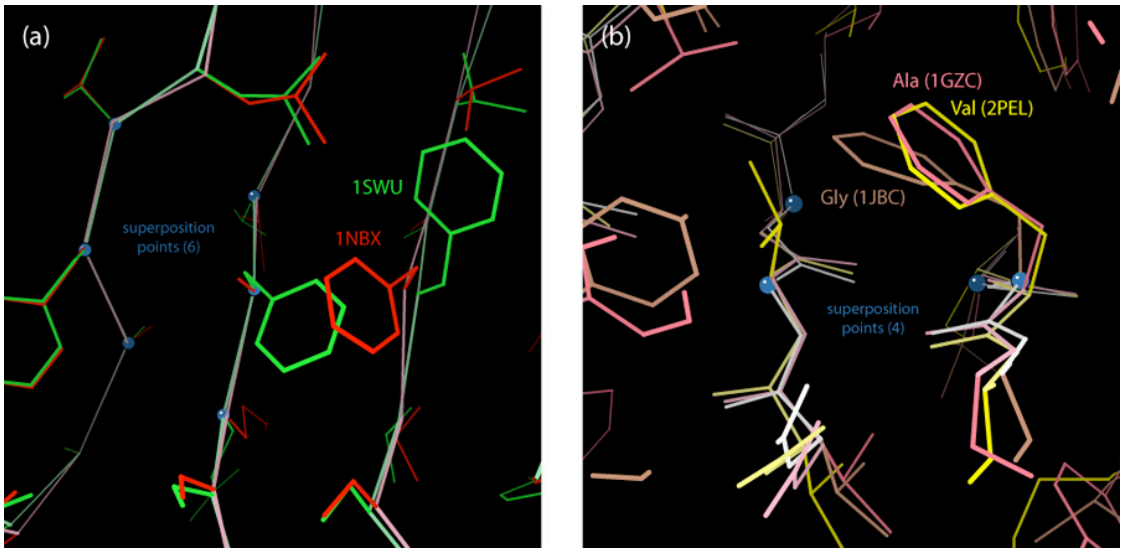


Figure 20: Mutations across from aromatic residues in β-sheets can cause backrub motions.

Cαs used for the local superpositions are marked with blue spheres. (a) In streptavidin, replacing an Ala (1NBX, red) with a bulkier Phe (1SWU, green) causes the neighboring Phe to change rotamer and undergo a backrub motion. (b) In concanavalin A, a conserved Phe can pack down more tightly on a Gly (1JBC, brown) than it can on the bulkier Ala (1GZC, pink) and Val (2PEL, yellow). The Phe undergoes a backrub motion in order to change its packing position.

A still more complete example of this packing phenomenon is present in three legume lectin / concanavalin A structures: 1JBC from jack bean at 1.15Å has Phe 197 packing on Gly 48; 1GZC from cockspur coral tree at 1.6Å has Phe 76 packing on Ala 166; and 2PEL from peanut at 2.25Å has Phe 71 packing on Val 161. (Although the sequence numbering varies, the regions are homologous and structurally equivalent.) Despite relatively low sequence identity, the structures superimpose well with LOCK2: 1JBC vs 1GZC gives 43% identity and 1.52Å RMSD; 1JBC vs 2PEL gives 44% identity and 1.55Å RMSD; and 1GZC vs 2PEL gives 38% identity and 1.55Å RMSD. As seen in Figure 20b, the structures were superimposed on 4 Cα in the core of the β-sheet to provide the best comparison of local structure. The difference between the Phe packing on Ala vs Val is minimal, but both show a marked contrast to the Phe packing on a Gly. The shift of the Phe is clearly a backrub-like motion, although it cannot be well modeled by the BACKRUB algorithm due to other conformational changes at that end of the sheet (see figure). Nonetheless, this example provides important evidence that backrub motions are an important component in the structural evolution of natural proteins.

Recently, Lindorff-Larsen and coworkers have refined an ensemble of ubiquitin structures using molecular dynamics (MD) to generate realistic

variation and NMR data to constrain the ensembles as a whole (Lindorff-Larsen, Best et al. 2005); they call the technique Dynamic Ensemble Refinement (DER). This approach should marry the best features of each field, and so it was here that we looked for further evidence of backrub motions.

Analysis of such ensembles presents special difficulties. First, there are likely to be additional, non-backrub motions that overlap with regions of interest, as for the concanavalin case above. Furthermore, a backrub motion is really a relationship between two conformational states. There is no simple, reliable, internal or external frame of reference that allows one to calculate a "backrub angle", which could then be compared across the ensemble. Instead, we pursued two complementary approaches. Both used a span of five continuous Cαs as a local reference frame for the comparison, numbered 1-5 with 3 being the backrub position. The first approach was an all-pairs approach: For a particular position along the chain, all possible pairs of models were superimposed on Cαs 1, 2, 4, and 5. The angle of backrub rotation was calculated as the dihedral 3-2-4-3', where 2 and 4 were averaged from the two models; obviously, this number is meaningful only if the RMSD of superposition is small. Empirically, an RMSD < 0.05Å and an angle > 20° indicated interesting pairs. A list of such pairs appears in Table 3.

Furthermore, we had access to seven distinct ensembles calculated with different parameters -- with or without S^2 order parameters, with or without J-couplings, and enforcing restraints on each model or on the ensemble as a whole. This allowed us to cross-validate the significance of several backrub motions we observed. For instance, a Gln 62 backrub occurred multiple times in 6 of the 7 ensembles, including the final published data, showing up to 30° rotation of the Cα-Cβ vector. Several other sites frequently displayed backrub motions, as listed in Table 4; those sites are shown mapped onto the ubiquitin structure in Figure 21.

Residue	Model 1	Model 2	Cα rmsd (Å, 1245)	backrub angle (°)
A 4 PHE	49	118	0.037	21.003
A 6 LYS	23	67	0.048	-23.049
A 6 LYS	115	116	0.047	-30.182
A 9 THR	64	125	0.032	-22.954
A 9 THR	80	81	0.046	-22.194
A 10 GLY	24	111	0.045	36.037
A 10 GLY	41	50	0.048	-36.58
A 14 THR	24	93	0.039	-24.952
A 14 THR	59	93	0.049	-31.702
A 17 VAL	14	126	0.048	34.098
A 17 VAL	29	44	0.021	-25.666
A 33 LYS	104	114	0.046	25.056
A 36 ILE	49	87	0.044	25.543
A 42 ARG	86	119	0.04	-20.543
A 44 ILE	73	123	0.048	21.742
A 47 GLY	21	55	0.048	-22.574
A 48 LYS	59	62	0.046	27.315
A 49 GLN	57	82	0.044	22.164
A 61 ILE	5	8	0.045	20.946
A 61 ILE	28	38	0.032	-23.092
A 62 GLN	10	59	0.039	21.395
A 62 GLN	14	86	0.043	22.646
A 62 GLN	25	30	0.045	21.362
A 62 GLN	62	64	0.038	30.797
A 63 LYS	9	65	0.041	-26.344
A 64 GLU	86	105	0.05	-20.929
A 66 THR	12	90	0.043	-22.063
A 66 THR	22	113	0.031	20.531
A 66 THR	28	61	0.035	-30.467
A 66 THR	35	86	0.048	-28.692
A 66 THR	55	116	0.048	31.503
A 66 THR	81	109	0.041	-34.475
A 66 THR	84	112	0.046	-45.271

Table 3: Pairs of models displaying backrub motions (1XQQ)

Residue	Total # of backrubs	# of ensembles (out of 7)
A 62 GLN	16	6
A 17 VAL	16	5
A 45 PHE	13	6
A 14 THR	13	3
A 66 THR	10	3
A 42 ARG	8	5
A 15 LEU	8	4

Table 4: Most frequently observed backrub motions (all ensembles)

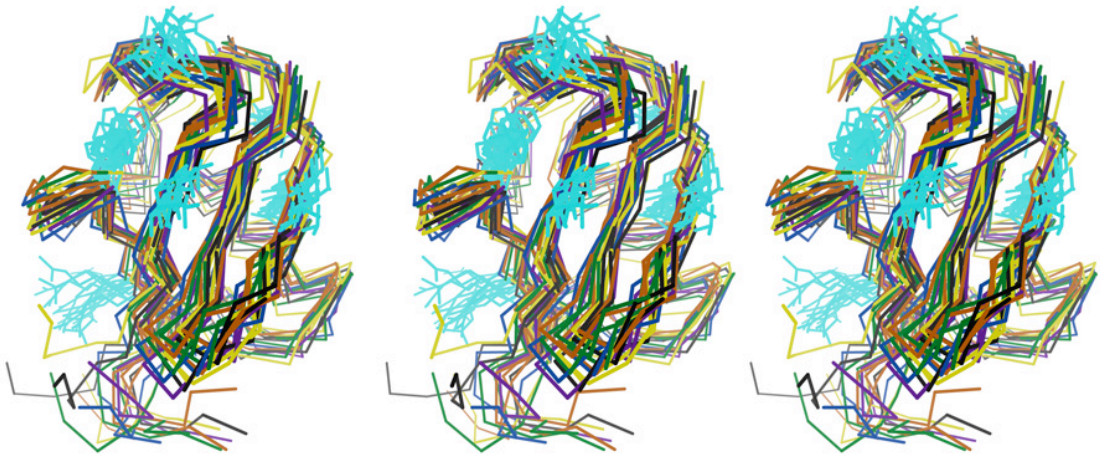


Figure 21: Most common sites of backrub motion in ubiquitin DER structures (1XQQ, first 16 models).

See Table 4 for complete list. Left pair is cross-eye stereo, right pair is wall-eye stereo.

Although the pairwise search does show that some structures within the ensemble are locally related by a backrub motion, it does not say anything directly about how prevalent backrub motion is at a particular site over the ensemble as a whole. To answer that question, we developed a simple statistical analysis. For each 5 Cα window along the sequence, we computed the standard deviation among models in Cα-Cα distances, angles, and dihedrals. Windows with low variation in the 2-4 distance, 1-2-4 and 2-4-5 angles, and 1-2-4-5 dihedral (relative to their peers) formed a good environment to observe backrub motion at the Cα3 position: They established a relatively static local environment that could be meaningfully compared across models. Of those windows, ones with large variation in the 1-2-4-3 and 3-2-4-5 dihedrals (again, relative to their peers) were flagged as probable sites of backrub motion; variation in those dihedrals is only meaningful if the surrounding structure is relatively static. Confirmation of the motion was performed visually, with a local multiple alignment in PROFIT on Cαs 1, 2, 4, and 5 (example in Figure 22). This ensemble for Gln62 shows a marked rotamer preference associated with the motion: *mt-30* at one end of the movement and *pt20* at the other. This is very similar to the rotamer-coupled backrubs seen in alternate conformations. Qualitatively, the statistical analysis

agreed with the pairwise one, finding Gln 62, Val 17, and Phe 45 all near the top of the list of probable backrubs.

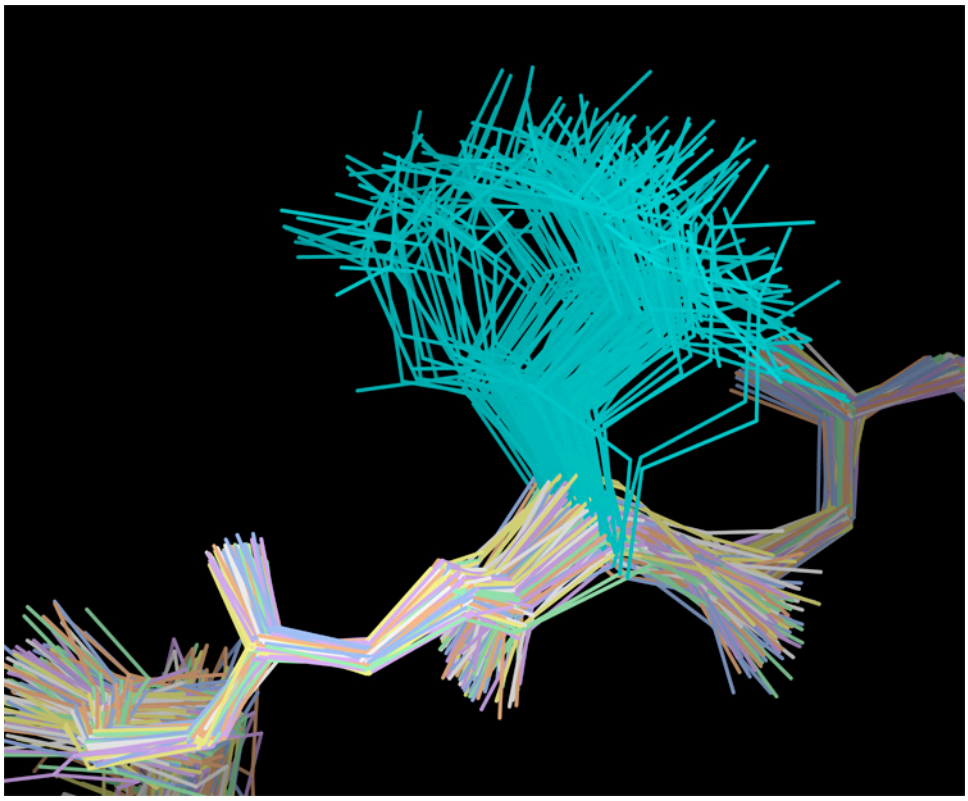


Figure 22: A likely backrub in an NMR structure of ubiquitin.

Multiple alignment on Cαs 60, 61, 63, and 64 suggests a backrub motion at position Gln62 in the DER ubiquitin ensemble (PDB code 1XQQ): most models toward the front of the backrub range show an *mt-30* sidechain rotamer, while most at the back end are *pt20*. For clarity, other sidechains are not shown.

Finally, we turned our attention to larger scale motions. Studying crystallographic alternate conformations at high resolution revealed that x-ray data were a rich and largely untapped source of dynamics information for identifying backbone motion. By analogy, we hypothesized that similar motions might be observed between identical chains within the same asymmetric unit, related by non-crystallographic symmetry (NCS). Because NCS pairs may experience distinct packing environments, such motions might be larger and more dramatic than backrub motions.

The data set for this study was the 66 PDB entries (as of October 2005) at or below 1.2Å resolution, with two or more identical chains. These structures faced the same superposition issues as for the evolutionarily related examples above. Superposition RMSDs were fairly low, typically in the 0.25 - 0.75Å range.

There were often significant differences between NCS pairs, especially in exposed regions; however, there were no consistent, striking patterns. The closest thing to a pattern was that 9 peptide flips were observed for the 34 structures actually examined. A peptide flip involves reorientation of a carbonyl oxygen by >90°, which involves shifting at least 3 peptides worth of backbone (e.g. Figure 23). The cases were 1C9O Glu 36, 1CZP Cys 46, 1IX9

Gly 135, 1MN8 Gly 72, 1NH0 Met 36 and Ile 50, 1NKI Pro 53, 1ODV Gly 115, and 1OH0 Gly 64. All appear to be real flips: they are supported by the electron density and are free of Ramachandran outliers and serious steric clashes. Several flips were at glycine residues, and several occurred in turns connecting β strands (the classic locus), but neither was true for a majority of cases. Exposed regions were also prone to shifting by 1-3Å relative to the rest of the structure without any peptide flips, and some peptides wagged up and down without flipping or appeared to participate in backrub motions. However, there was generally sufficient structural noise that it was unclear whether these observed structural differences were meaningful.

Other interesting motions emerged from the NCS comparisons, but they were unique to a single structure. An intriguing example is the α-helix from 41 to 49 in 1O7J (Figure 24). All four chains are identical in sequence, but A and D share one conformation, while B and C share another that is displaced by 0.7Å. This is not surprising until one discovers that A and C actually have the same arrangement of components packed around them in the crystal, simply offset by a few Ångstroms; the same is true for B and D. Thus, it appears there are two favorable helix conformations separated by some sort of

barrier, probably linked to rotamer changes for Val 43 and Val 46. In that way, this unique motion resembles the ubiquitous backrub motion.

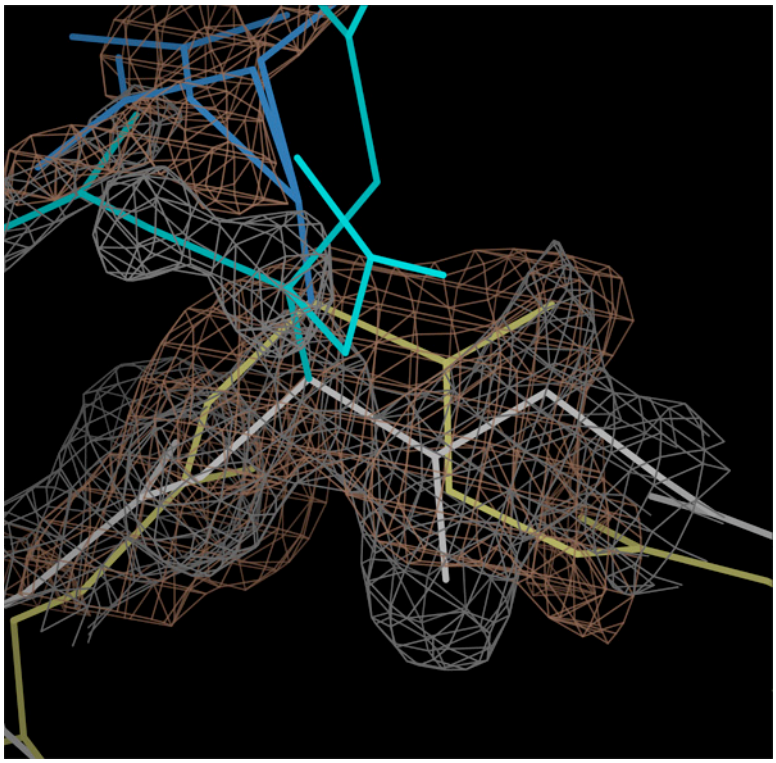


Figure 23: Example of a peptide flip between NCS-related identical subunits.

The flip occurs in 1C9O between Glu36 and Gly37. The B chain and its $2F_o$-F_c map (yellow/brown) have been superimposed on the A chain (white/gray). Note the carbonyl O densities nearly 180° apart.

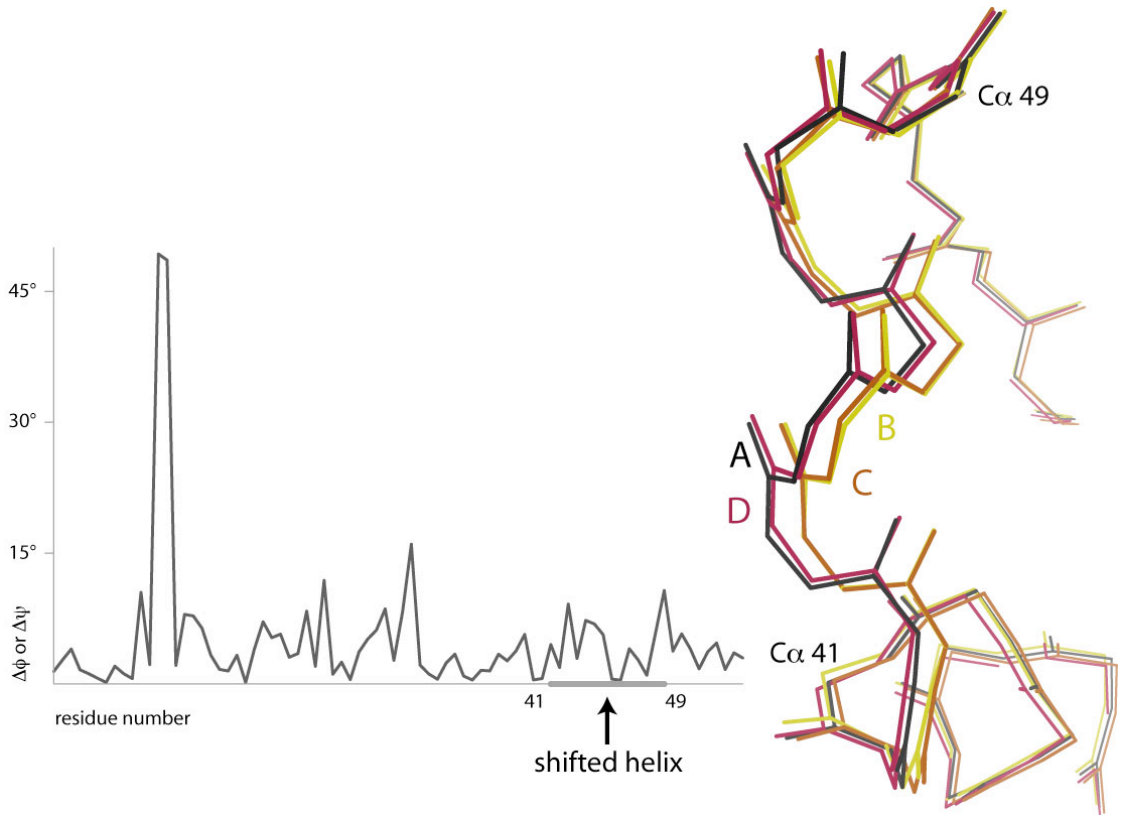


Figure 24: Shift of an α helix between NCS-related subunits in 1O7J.

(left) Although the helix shift is large relative to superposition error in Cartesian space, it does not stand out above the noise in a Δϕ,ψ plot. (right) Chains A and D take a different conformation than B and C, even though A and C have similar crystal packing environments to one another (as do B and D).

Discussion

The alternate conformations seen in sub-1Å-resolution crystal structures show unambiguously that protein backbone often exhibits highly localized, small-amplitude plasticity that is tightly coupled to larger, 2-state conformational change of the sidechain. By far the commonest case is a "backrub" motion, in which one residue and its adjacent peptides twist slightly around the backbone; this is usually driven by a change in sidechain rotamer and/or hydrogen bonding partners, leading to significant sidechain motion perpendicular to the chain direction. Over the 19 proteins and 3882 residues studied here, 1 residue in 30 clearly shows a backrub motion (126 examples), and backrub motions are surely even more prevalent under physiological or solution conditions than in frozen crystals.

The BACKRUB algorithm described here produces geometrically and sterically reasonable models that fit the electron density extremely well. Its utility in crystallographic rebuilding is demonstrated in Figure 18 (see above) and in Arendall et al. (2005). In practice, BACKRUB is most useful either for defining alternate conformations at very high resolution or for correctional rebuilding of backbone in the 2 to 3Å resolution range. At resolutions <2Å, backbone atom positions are strongly constrained, so a sidechain misfit into

the wrong rotamer produces distorted bond angles instead (Lovell, Davis et al. 2003). At resolutions >3Å, one cannot address such fine detail, due to surrounding inaccuracies.

Backrub motion is large for the central sidechain, moderate for its backbone, and decreases rapidly on either side. To accomplish a "true" low-energy backrub movement with essentially pure ϕ,ψ variables, as proteins presumably do, extremely small changes would propagate past the $i\pm1$ Cα atoms. This involves at least five pairs of ϕ,ψ angle changes and an intractable level of complexity. As the figures show, however, BACKRUB-generated models are remarkably good at fitting the electron density of alternate conformations with only three variables besides sidechain rotamer. Sidechain rotamers (Lovell, Word et al. 2000) are essentially always favorable, and all bond lengths, bond angles, and peptide planarity can be kept ideal except for the τ bond angles at Cα i-1, i, and i+1. Those τ distortions seldom exceed one standard deviation (Engh and Huber 1991), and sometimes ideality is improved. Of course, any BACKRUB refitting during solution of a crystal structure would be submitted to further refinement afterward. A small percentage of cases appear to require non-ideal rotamers or geometry (especially buried Met or disulfides) and may actually have both

conformations strained by the tight surroundings rather than both favorable. Different approximate fitting procedures that are commonly used for crystallography include allowing a sidechain to shift independently of the backbone or allowing one residue to shift independently of its neighbors. Those techniques are not significantly simpler than the BACKRUB, but they produce very large distortions of bond angles or peptide planarity not supported by the data.

The BACKRUB algorithm is the natural result of simple protein geometry when considering the protein backbone as a Cα trace; the region of interest is comprised of one central Cα with two other Cαs as neighboring anchors. This is a powerful representation because peptide units are effectively of constant length: 3.807 ± 0.026Å for *trans* peptides in the Top500 database (Lovell, Davis et al. 2003). If we assume that the neighboring Cα positions are absolutely fixed, then the central Cα must lie on the surface of a sphere 3.8Å from the left neighbor, and likewise for the right neighbor. Thus, the only allowable positions for the center Cα are at the intersection of those spheres, which is a ring centered at the midpoint of the line between the neighbor Cαs and perpendicular to it. Assuming the structure starts from a valid conformation, the center Cα is already positioned on that ring; it can be moved to any other

allowable position by rotating it around the axis between its neighbors. If all the atoms between those neighbors are rotated as a rigid group (instead of just the Cα), then one gets the primary rotation from the BACKRUB algorithm.

However, a Cα trace cannot tell the full story of backbone motions, because it necessarily ignores the atoms between the Cαs. In fact, a Cα trace does not contain enough information to accurately and reliably reconstruct an all-atom backbone model: although several groups have developed algorithms, it is not uncommon that they place one or more peptides backwards (off by >90°) (Iwata, Kasuya et al. 2002; Kazmierkiewicz, Liwo et al. 2002). This led David Richardson to propose that an "azimuthal" Cα trace, with four parameters per residue, would contain enough information to trivially and accurately reconstruct an all-atom backbone (unpublished). In addition to *x,y,z* coordinates for the Cα, such a trace would include an "azimuthal" dihedral angle that defined the peptide orientation relative to the rest of the backbone (e.g. $C\alpha_0$-$C\alpha_1$-$C\alpha_2$-O_1). Thinking of the azimuthal angle as a variable results in rotating the peptide group around the axis between its flanking Cαs; this is exactly the secondary motion in the BACKRUB algorithm.

Taken together, these insights from Cα traces and the azimuthal angle are sufficient to construct the basic BACKRUB algorithm. However, both the primary and secondary BACKRUB motions consist of defining an axis between two Cαs, and rotating the intervening atoms around that axis as a rigid body. Thus, one can easily extend this algorithm to larger segments of protein backbone by defining axes between all pairs of Cαs within some region; unfortunately, that extension is of little practical usefulness. First, the number of such axes grows quickly, as the square of the number of Cαs involved. Since one needs only a limited number of coordinates per residue (that is, per Cα) to define the backbone conformation, it is clear that in some sense these Cα axes are redundant with one another for large stretches of backbone. (On the other hand, it offers a possible qualitative insight into why no larger-scale patterns of backbone movement were discovered during this work -- there are simply too many possibilities for strong patterns to emerge.) Second, in the generalized case bond angle distortions become a bigger problem. In the BACKRUB case, the axes of rotation are roughly aligned with rotatable bonds, which minimizes distortion of the τ (N-Cα-C) angles; in the general case, the axis may be perpendicular to the local bonds, which results in much larger distortions. Third, rotations about the axes are not independent of each other,

because one rotation may move the fixed endpoint for another. Unlike the BACKRUB case, then, conformational changes cannot be described by simply specifying a value for each rotation, because the order of rotations matters in the generalized case. Despite these theoretical flaws, the generalized algorithm might have had some practical usefulness -- for example, in crystallographic rebuilding. However, in practice it did not prove useful, and so the generalization of BACKRUB was not pursued further.

It seems likely that other modes of local backbone plasticity remain to be discovered. This is particularly true of α-helices, where backrub motions are less common than in extended structure. Some examples of helix motions combine winding and unwinding to shift a local region sideways without disrupting the overall helix (e.g. 1EJG 6-9), but no other local helix modes were common enough for classification.

On the evolutionary time scale, local backrub shifts could be an important component of protein robustness to point mutations, accommodating sidechains of different sizes and shapes without radically altering the backbone scaffold. However, it is difficult to observe backrub motions directly by comparing structures of point mutants because the coordinate error between two "identical" but independently refined structures is a few tenths

of an Ångstrom, comparable to the size of the backrub conformational changes (Kleywegt 1999; DePristo, de Bakker et al. 2003). Thus, we turned instead to the more accurate and quite numerous examples of backrub motion found within single crystal structures. The magnitude of such backbone movement is small: 90% of the examples shift Cβ < 0.8Å, and 50% by < 0.4Å, while Cα and other backbone atoms move half that much or less. However, essentially all cases leverage sidechain atom shifts of 1 to 8Å (2.8Å on average), quite like the change necessitated by a sequence difference. Nearly all local backbone motions in alternate conformations are coupled to sidechain switches between rotamers (86%), which have steric and electrostatic consequences on par with the effects of a point mutation. Thus, we believe alternate conformations provide a good model of how a backrub motion could be involved in preserving a protein's structure as its sequence evolves. That belief is supported by the shifts observed in homologous and point mutant structures. Of even more immediate practical relevance, the backrub motion should provide a conservative backbone "move", of well documented occurrence in accurate experimental structures, to help protein design and homology modeling calculations provide the local backbone adjustments required for successful accommodation of sequence changes.

These observations are also directly relevant to protein dynamics, in spite of their origin in data from highly ordered crystals at cryogenic temperatures. Individual crystal structures are not usually thought of as dynamic, both because crystallization selects only a subset of the conformations populated in solution, and also because at most accessible resolutions alternate conformations are manifested only by lowered electron density and are thus seldom modeled in the coordinates. At very high resolution, however, multiple conformations become directly visible (2 or occasionally 3 copies, down to perhaps 10% occupancy in the best cases). At the cryogenic temperatures typical of modern data collection (near 100 K) presumably no large dynamic fluctuations occur in individual molecules, so alternate conformations represent static disorder between molecules, a sample of the conformations present in the room-temperature crystal. In the other direction, however, it is quite certain that the conformations seen in these structures are also present in solution—thus they show the geometry of a valid subset of protein motions.

Crystallographic alternate conformations imply that the states must have comparable energies, since their observable fractional occurrences lie between 1:1 and at most 10:1. (an energy difference of kT yields a 3:1 ratio). For these

Cβ-shift cases that imply backbone motion, the mobile sidechain atoms nearly always show clearly separated peaks, implying an energy barrier between the sidechain states. The backbone atoms, in contrast, nearly always show continuous density between positions close in space, implying that the backbone stays within a single local energy well. As shown above, backrub motions also have the important ability to preserve NH and CO orientations that control backbone H-bonds, thus preserving secondary structure despite substantial sidechain movement. This set of properties is here only shown to hold for the low-energy subset of motions manifested in high-resolution crystal structures, but it matches well with the properties seen by NMR order parameters and other dynamical measurements: relatively less motion of the highly H-bonded peptide NH's than for the carbonyl–Cα bonds (Wang, Cai et al. 2003), and greater and more complex motion for the sidechains (Palmer 2004; Kay 2005), most of which visit multiple separate rotamer states (Chou, Case et al. 2003).

Other recent NMR studies also agree closely with our findings. Blackledge and coworkers used extensive residual dipolar coupling (RDC) and hydrogen bond scalar coupling (HBC) data in a 3-D Gaussian axial fluctuation model to determine backbone motions for protein G on the sub-microsecond to

millisecond timescale (Bouvignies, Bernado et al. 2005). They find that peptides primarily rotate around the Cα-Cα axis, and the largest motions occur in β-sheet regions, both of which are also true for the backrub motions we observe. Although their model does not account for possible motion of the Cαs themselves, they do find that backbone motions are correlated across β-sheets through hydrogen bonds; we have predicted that backrub motions play a role in maintaining secondary structure H-bond networks as individual residues move about. As a detailed description of a common, experimentally verified local movement, backrub fluctuations provide a new model -- in addition to out-of-plane amide vibrations (Palmo, Mannfors et al. 2003) or crankshaft peptide motions (Fadel, Jin et al. 1995) -- for the analysis of NH order parameters and backbone RDCs, with the valuable feature of built-in coupling to larger sidechain motions.

Spectroscopic methods dominate the experimental study of protein dynamics because they can measure time scales, relative magnitudes, and even energetics of motions. However, any inference about the pattern of movement in space is highly indirect. In contrast, high-resolution crystal structures can give no insight into time scales, but they show a direct image of what atoms are moving where, and they constitute a valuable and largely

untapped source of dynamic information. Here we have used crystallographic alternate conformations to demonstrate a form of local backbone plasticity that appears to dominate small-scale accommodation to sidechain rotamer fluctuations and perhaps also to single-site mutations. These results also strongly imply that backbone and sidechain dynamics should not be analyzed in isolation, since for at least one common mode the two are tightly coupled.

Chapter 4: Applying BACKRUB to computational design

> *The revolution in molecular biology, along with other technical advances, has led to a far more active and meddlesome approach to both the study and the exploitation of proteins.*
>
> — Jane Richardson

A complete understanding of protein folding is one of the outstanding challenges for modern biology, and attempting to design new proteins is a stringent test of our progress. Additionally, there is a great practical payoff in protein design: since protein enzymes catalyze nearly all of the chemistry that makes life go, industry is justifiably enthralled with the promise of designable cheap, efficient, specific, and environmentally responsible catalysts.

The field of protein engineering or design is now several decades old. From the beginning there has been a division between studies of structure and studies of binding and catalysis. On the binding/catalysis side, Gutte early on produced a miniature ribonuclease (Gutte 1975; Gutte 1977) and proteins that bound polynucleotides (Gutte, Daumigen et al. 1979) and DDT (Moser, Thomas et al. 1983). There has also been much work on tuning the activity,

specificity, and stability of natural enzymes, particularly proteases like subtilisin used in detergents (Bryan 2000). On the structural side, many other researchers focused on α-helical structure, producing single helices (Kaiser and Kezdy 1983), coiled coils (Talbot and Hodges 1981), and multi-helix bundles (DeGrado, Wasserman et al. 1989); helical proteins continue to be popular targets for design. Some of the first attempts to design proteins with native-like sequences came from our laboratory, including a four-helix bundle and an antiparallel β barrel (Richardson and Richardson 1987; Richardson and Richardson 1989; Hecht, Richardson et al. 1990). While they often adopted the right global fold, these early designs were often less stable or soluble, and nearly always less cooperatively-unfolding than their natural counterparts, problems which continue to this day.

In general, it seems that designing truly native-like proteins with a single, well-ordered equilibrium structure is extremely challenging. However, extensive work by Hecht and colleagues has shown that simple hydrophilic / hydrophobic patterning of secondary structure elements plus specific linkers is sufficient to create α-helical bundles, such that randomly generated sequences of sufficient length with the appropriate hydrophilic / hydrophobic helical pattern form well-ordered, native-like structures more often than not

(Hecht, Das et al. 2004). On the other hand, β proteins are more difficult: they tend to form amyloid-type fibrils and monolayers unless specific strategies are used to prevent edge-strand aggregation (Wang and Hecht 2002). However, all of these semi-random proteins are low-resolution designs that rely on combinatorial sampling for their success: their atomic structures were not anticipated ahead of time, and although some of them exhibit some ligand binding or catalytic activity, these are fairly generic and were unintentional.

Most other protein design work has instead turned to redesigning existing proteins, either to change their structures or their activities. For instance, extensive studies have been done on repacking the hydrophobic core of T4 lysozyme, complete with crystal structures of many of the designs (Baldwin, Hajiseyedjavadi et al. 1993). Handel and coworkers did core-repacking designs for ubiquitin (Lazar, Desjarlais et al. 1997), while Mayo and colleagues have redesigned zinc finger motifs (Dahiyat and Mayo 1997). The Hellinga group has reengineered the binding pockets of several bacterial periplasmic binding proteins (PBPs) to bind small molecules other than their native ligands with high affinity and selectivity (Looger, Dwyer et al. 2003). They have also used PBPs as scaffolds for designing specific and quite active enzymes (Dwyer, Looger et al. 2004).

To date, however, most of the really successful protein (re)designs have focused exclusively on sidechains, assuming exactly the same backbone structure as the crystal structure of the wild type scaffold. This is due to both the computational cost and the difficulties in modeling backbone. Computationally, accounting for changes in backbone conformation is difficult because it interferes with the pairwise-additive property of the forcefields, which design programs exploit to efficiently decompose the enormous computational complexity of the design problem. Most simple kinds of backbone motion have global consequences for the protein structure; therefore backbone motion at site 3 likely changes the energy of interaction between sidechains 1 and 2. Such multi-body interactions reduce the effectiveness of algorithms like Dead End Elimination (DEE), which DEZYMER uses to find the global minimum energy design in a reasonable amount of time.

Beyond computational cost, however, modeling backbone realistically is difficult. Desjarlais and Handel tried to use small stochastic steps in ϕ,ψ space while repacking the core of T4 lysozyme, but found that energies from the Amber/OPLS forcefield had no correlation with the experimental stabilities of their designs (Desjarlais and Handel 1999); a good correlation is obtained

when backbone motion is omitted. This is consistent with other work that finds none of the molecular mechanics-style forcefields reproduce the conformational preferences of protein backbone (Hu, Elstner et al. 2003). Even with a more conservative model for backbone motion, Desjarlais and Handel found that it lead to less stable designs overall.

The general consensus in the field is that backbone flexibility degrades the quality of computational designs, but there are a few counter-examples. A family of α-helical bundle proteins was designed on an ensemble of backbone structures generated from a simple algebraic parameterization (Harbury, Plecs et al. 1998). Since there were only a few degrees of freedom, that family of backbones could be explored relatively completely within the limits of the parameterization. However, since it exploits the special symmetry of these structures, this technique has yet to be generalized. Kuhlman and coworkers have achieved the most ambitious backbone design to date, creating *de novo* a mixed α/β protein with a topology unknown in nature (Kuhlman, Dantas et al. 2003); the protein is found to be quite stable and its structure has been verified by crystallography. Their algorithm alternates between designing mainchain conformation and sidechain conformation and sequence, but the

exact reason for their remarkable success is somewhat mysterious: it could be a superior force field, a superior search strategy, or simply good luck.

Where there are experimental structures of designed proteins, it is often found that significant shifts have occurred in the backbone, even though the calculations were performed using the wild type structure. In repacking the core of T4 lysozyme, Matthews and co-workers found backbone RMSDs of 0.2 - 0.6Å in the designed proteins versus wild type (Baldwin, Hajiseyedjavadi et al. 1993). They also note the backbone shifts lead to larger shifts of 1 - 2Å in sidechain positions; although this resembles our backrub findings (Chapter 3), the structures are not sufficiently accurate for us to reliably classify the motion. Kortemme and Baker note that small backbone shifts in a designed protein-protein interface (PDB code 2ERH) led to incorrect prediction of sidechain conformations in the interface and an artificially high predicted energy (Kortemme and Baker 2004; Joachimiak, Kortemme et al. 2006).

The present work aims to extend the highly successful DEZYMER system for receptor design by incorporating small local backbone perturbations based on the BACKRUB model. If it had been fully successful, it would be (as far as we know) the first example of simultaneously redesigning backbone structure and protein function. It is clear that incorporating backbone flexibility can

greatly extend the power of computational design when done well; it is also clear that backrub motions are very common in real proteins and have a significant impact on sidechain position with a minimal disruption of the backbone. This work shows there is promise in combining Backrub and Dezymer, and will hopefully lay the groundwork for future exploration.

Scaffold construction

To test the impact of small shifts in backbone conformation on the design process, eleven nearly identical scaffold structures were submitted to the standard DEZYMER ReceptorDesign protocol. Those scaffolds consisted of the wildtype *Salmonella typhimurium* glucose/galactose binding protein (GBP; PDB code 1GCA) plus ten variants of that structure. In those variants, the backbone of one or more residues had been manually shifted using the Backrub tool in KiNG. Only evolving zone residues (i.e., those in the binding pocket) were candidates for shifting. Based on steric and geometric validation criteria, I modeled one or more additional plausible conformations for each residue's backbone. A few of the evolving zone residues were sufficiently constrained in the original structure that no alternative conformations could be safely proposed (residues 14, 236, and 256). All of the alternates used are listed in Table 5 and shown graphically in Figure 25.

Label	Res #	Backrub rotation (°)	N peptide rotation (°)	C peptide rotation (°)	Cβ travel (Å)
10.2	10	+10	0	-14	0.73
16.2	16	+10	0	+7	0.55
91-92.2	92	+10	-6	0	0.60
91-92.4	91	-8	+3	-14	0.28
152.1	152	-5	-8	+10	0.29
154.2	154	+10	+7	0	0.64
158.2	158	+5	0	0	0.34
183.1	183	-5	0	+10	0.44
211.3	211	+15	0	-20	0.56

Table 5: Alternative conformations modeled in 1GCA using Backrub

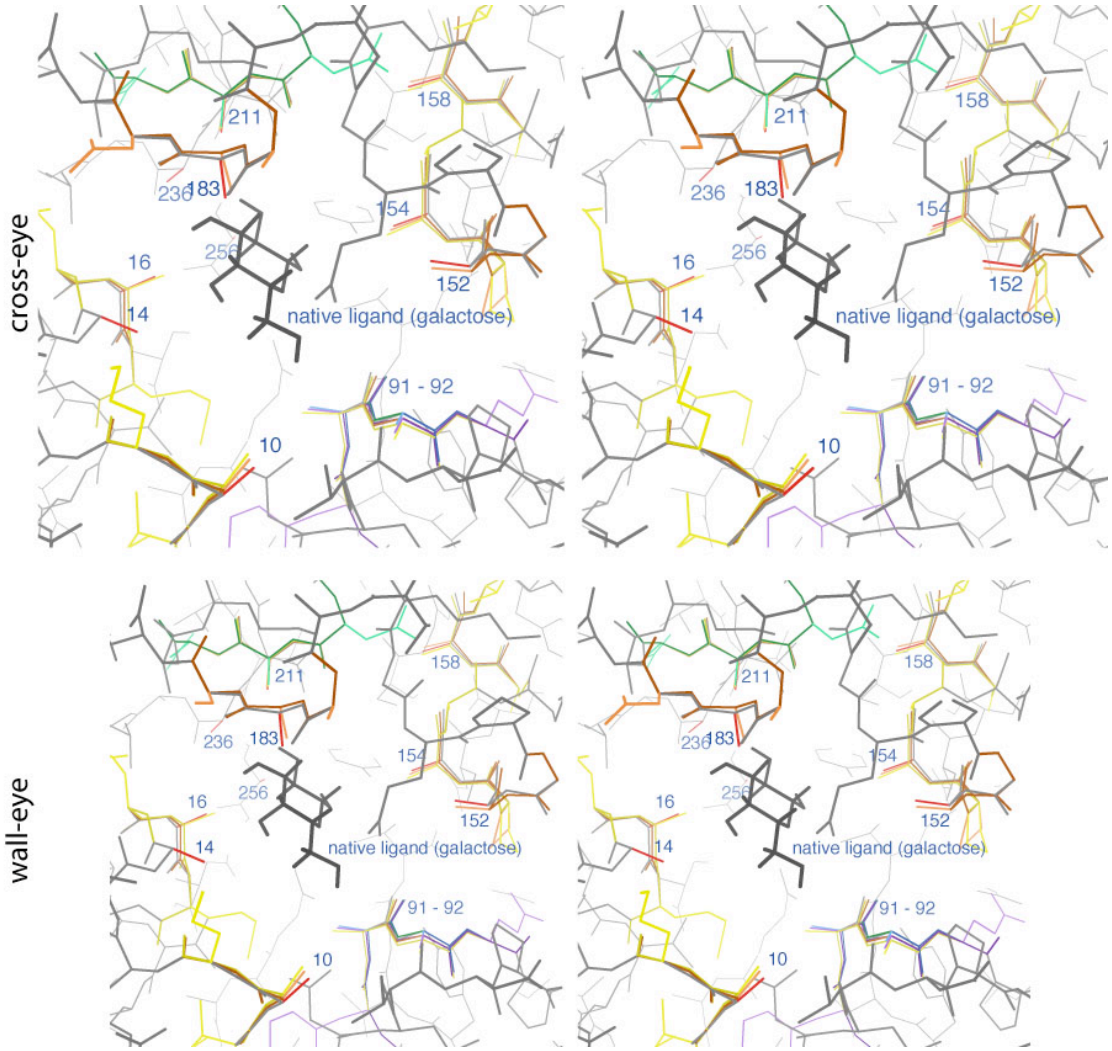


Figure 25: Alternative conformations modeled in 1GCA using Backrub.

Colors are used to distinguish multiple conformations at a single site: first is red, second is orange, etc. (The full two peptides affected by BACKRUB are colored.) Conformations in the same color at different sites are not related, except in the case of immediately adjacent residues (91 & 92). Note that the positions of residues 14, 236, and 256 are marked for reference, although they were not moved.

S. typhimurium GBP (stGBP) was used for the computational design, although *E. coli* GBP (ecGBP) was used for expression. stGBP is 94% identical to ecGBP and has no differences near the binding site residues. Furthermore, the stGBP structure is higher resolution (1GCA, 1.7Å) than the ecGBP structure (2GBP, 1.9Å) and scores better on the MOLPROBITY validation criteria. However, the primary reason for this choice was to facilitate comparison with the results of earlier experiments, which used the same approach to build a lactate binding GBP using only experimentally determined coordinates from 1GCA for the backbone scaffold.

Using the alternative backbone conformations from Table 5, ten variant scaffolds were constructed (GW1 – GW10). Some had just one residue shifted, while others had two or three residues shifted. The groups of two or three were chosen by inspection to be residues that were likely to interact with one another in the final design. One variant (GW1) had most evolving zone positions shifted, in an attempt to ascertain the maximum effect of backbone perturbation on the design process. The composition of all eleven scaffolds is listed in Table 6.

Name	10	14	16	91-92	152	154	158	183	211	236	256
GBP											
GW1	10.2		16.2	91-92.4		154.2	158.2	183.1	211.3		
GW2			16.2								
GW3				91-92.4							
GW4							158.2				
GW5								183.1	211.3		
GW6			16.2						211.3		
GW7			16.2					183.1			
GW8				91-92.2	152.1						
GW9			16.2					183.1	211.3		
GW10						154.2	158.2		211.3		

Table 6: Composition of the scaffolds

Dotted entries refer to backrub-modified conformations listed in Table 5; blank entries indicate the deposited conformation was used.

Design execution and evaluation

The computational protocol used to produce designs for each scaffold was identical. This protocol was heavily based on the standard ReceptorDesign scripts and parameters; however, the scoring procedure was modified and a few parameters were modified. Some changes to the parameters were suggested by Dezymer's author as better defaults: NonpolarExposedPenalty was set to 300 instead of 52 cal/Å^2; PleSize and FleSize were increased from 5,000 to 20,000 for improved sampling of ligand poses; and ExclusionRadius was reduced from 3.5Å to 2.4Å to allow lactate access to more of the binding pocket.

The effects of other possible modifications were also explored, but on the whole they all made the designs worse. Reducing VdwStretch from 1.2 to 1.0 produced somewhat fewer bad clashes but many more internal cavities. Overweighting the electrostatics terms (10x) increased the number of bad clashes, made many hydrogen bonds too short, and eliminated most of the sequence diversity in the results. On the other hand, underweighting the electrostatics terms (0.5x) increased the number of cavities, made many hydrogen bonds too long, and again reduced the sequence diversity in the

results. At this point, further efforts to optimize the parameters were abandoned and the actual design calculations were begun.

Because so many designs were produced -- 20,000 GMECs and 20,000 wells for each of 11 scaffolds -- considerable effort was required to evaluate the results. Looking at the wildtype scaffold results, I judged the wells to be significantly better than the GMECs, so I focused my efforts there. To further simplify the evaluation process, however, I also developed new tools in three areas: ranking the designs produced for any one scaffold; comparing the sequence spaces explored by different scaffolds; and evaluating the packing within the binding pocket.

The generally accepted goal in ranking designs is to favor those that score well by many different criteria: GMEC, receptor, and ligand energy; number of unsatisfied H-bonds; size of cavities and solvent-exposed surface; etc. Dezymer usually does this by computing the statistical rank for each criterion, adding up each design's rankings, and sorting on that sum. My alternative method is to compute a Z-score for each criterion (i.e. number of standard deviations from the mean), add those up, and sort on that sum. The difference between them is that the plain rankings are only sensitive to order, while the Z-scores are also sensitive to magnitude of difference between designs. They

did a roughly equal job of selecting designs that appeared good upon inspection, and they were well correlated with each other ($R^2 \sim 0.9$). Therefore, I choose to use both, selecting for further review the 100 designs in the top N^{th} percentile by both scores.

To illuminate the sequence relationships among these top 1100 designs (100 designs * 11 scaffolds), two kinemage visualizations were developed. The first scatters sequences in 3-space, puts harmonic restraints between them based on the number of mutations between them, and then minimizes the "energy". After several trials, the graph with the lowest final energy is retained (Figure 28). (This is a simple but common technique for graph layout that is used in other fields, such as for clustering social-science relationship networks.) The second visualization shows the 2D tree produced by a hierarchical clustering of sequences, where distance between sequences is calculated from BLOSUM62 scores (Henikoff and Henikoff 1992) and the criterion for merging clusters is the mean distance between their members (Figure 27). Both visualizations can be run on results from just one scaffold or from more than one, in which case color and translucency help distinguish scaffolds from one another.

Once a manageable number of high-scoring sequences was selected, the designs to be actually built and characterized were selected by inspection. Although all-atom contacts show steric clashes and hydrogen bonds effectively, a different visualization is needed to reveal cavities and packing defects. Extensive cavities could compromise both the stability and the specificity of the designs, but existing cavity visualizations make it difficult to judge the shape and position of the cavity relative to the ligand. My alternative rendering is a pair of nested dot surfaces, one for the ligand, and the other for the pocket (Figure 29). Now one can accurately judge whether the ligand is firmly packed or free to wobble, how particular cavities might impact specificity, and what mutations might help fill them in. The ligand surface is generated by Probe, but the pocket surface is generated placing dots on DCR's "foo" (unpublished). The foo is like an expanding foam of spheres that starts from the ligand position and grows until it hits other atoms or the protein surface. Thus, it also indirectly shows solvent-accessible surface for the ligand by showing channels from the pocket to the surface. One limit to this approach is that it does not reveal cavities that are not directly connected to the binding pocket.

The utility of these new tools has not been proven, because the real test of each is whether the resulting designs are actually superior when characterized in the lab. Nonetheless, I feel that each one helped me to understand the design results better.

From all the design results, a total of 10 different sequences were selected for biochemical characterization. Attention was given to representing a variety of scaffolds, a variety of ligand poses, and a wide range of sequence space. The selected designs are shown in Table 7.

Name	Sequence	Scaffold(s)
Wild type *E. coli* GBP	Y D F N K H D R W N D N	n/a
Looger et al. G1	**K K** F **K L M H K K** N **A D**	n/a
Looger et al. G2	**K M K K L K K M K** N **A S**	n/a
wA	**R** D F **s** K **D K K** W **A W D**	GW4
wB	**K** D **S v E A H** R W **K A K**	GBP
wC	**K** D **G s E K K** R **E S W W**	GW2, GW6
wD	**R H** F **S A K K K** W **E A E**	GW8
wE	**K** D F **s S K K** R **K K A D**	GW5
wF	**E** D **G S S R M K K A W W**	GW1
wG	**R E N** N K **A Q K** W **A W K**	GW3
wH	**R** D **S G** K **A F K F A W K**	GW10
wI	**K E G v E E F** R **Q A W W**	GW7, GW9
wJ	**M N T A S L M** R **E Q A K**	GW1

Table 7: Designs selected for biochemical characterization.

For comparison, the wildtype sequence and two previous GBP-lactate designs are also shown (Looger, Dwyer et al. 2003). "Sequence" is the sequence at the evolving zone positions only, in order (10, 14, 16, 91, 92, 152, 154, 158, 183, 211, 236, 256). Bold letters indicate mutations from wild type. Lowercase letters were altered by inspection after the design calculation was complete. Underlined positions indicate where the backbone was shifted from the wild type. If multiple scaffolds are listed, the first one is primary, and the others returned similar sequences that influenced the choice of design and/or mutations by inspection.

Experimental methods

Molecular biology

The mutations from the ten selected designs of Table 7 (wA - wJ) were inserted into the sequence of wild type *E. coli* glucose binding protein, and the resulting sequence was synthesized from oligonucleotides by the Hellinga lab gene fabrication facility. Results were fused to a chlorampenicol resistence gene, inserted into a pCR-Blunt II - TOPO plasmid (Invitrogen), and verified by sequencing.

I amplified the genes from their plasmids by PCR, at the same time introducing a XbaI restriction site at the 5′ end and a short linker, 6x His tag, two terminator codons, and an EcoRI site at the 3′ end. PCR product was purified by electrophoresis on a 1.5% agarose gel. Bands were cut out and gel extracted (Qiagen QIAquick), then double digested overnight with XbaI and EcoRI at 37° in SureCut Buffer H (Roche), giving a final yield of 150-250 ng/μL. pET-21a plasmid (Novagen) was prepared in the same way and was not treated with alkaline phosphatase, yielding 50 ng/μL. 1 μL each of digested gene insert and vector were ligated in 20 μL buffer for 20 minutes at room temperature followed by 20 minutes at 65°. This mixture was incubated with 100 μL chemically competent XL1-Blue cells (Stratagene) on ice for 1 hour,

then heat shocked for 45 sec. at 42°, recovered on ice for 5 min., plated on 2xYT + 100 μg/mL ampicillin, and allowed to grow overnight at 37°. Cells were picked from these plates and grown in 3 mL of LB overnight at 37°, then miniprepped (Qiagen QIAprep) according to directions. The purified plasmid DNA was then sequenced to ensure integrity of the designed gene, and this stock was used to transform cells for protein expression.

Protein expression and purification

1 μL of purified plasmid DNA was mixed with 100 μL of chemically competent BL21(DE3) pLysS cells (Stratagene) on ice for 5 min, heat shocked for 45 sec. at 42°, recovered on ice for 5 min., plated on 2xYT + 100 μg/mL ampicillin + 34 μg/mL chloramphenicol, and allowed to grow overnight at 37°. Attempts to express in ZYM-5052 auto-inducing media gave poor yield (Studier 2005), but HyperBroth (AthenaES) gave good results, as follows. 10 mL of 2xYT + 100 μg/mL ampicillin in a 125 mL flask was inoculated from the plate and allowed to grow for 3-4 hours at 37° and 250 rpm. This starter culture was then added to 500 mL HyperBroth + 100 μg/mL ampicillin + 1mM $CaCl_2$ in a 2 L baffled flask, which had been prewarmed to 37°. (Wild type glucose binding protein has a calcium binding site, so Ca^{2+} is maintained in all buffers throughout purification.) That was allowed to grow at 37° and 250

rpm for 30-60 minutes, until reaching an OD_{600} of 0.4. The flasks were then chilled in an ice-water bath for 30 minutes before protein expression was induced with 1 mM IPTG. Flasks were returned to a 22° incubator at 250 rpm and expression was allowed to continue overnight.

In the morning, cells were pelleted at 5500 rpm for 30 minutes in a Beckman JLA 10.500 rotor at 4°. Cell pellets were resuspended in 25 mL of 10 mM imidazole, 500 mM NaCl, 2 mM $CaCl_2$, 20 mM MOPS at pH 7.8. Cells were ruptured by 2 minutes of sonication on ice and then mixed with 250 μL PEI (polyethylenimine) to help precipitate DNA. Lysed cells were then pelleted at 16,500 rpm for 30 minutes in a Beckman JA 17 rotor at 4°.

Affinity columns were prepared with 5 mL Chelating Sepharose FastFlow slurry (GE Healthcare), washed with one column volume (CV) of water to remove ethanol, charged with 10-15 mL 100 mM $NiSO_4$, and equilibrated with 1CV of resuspension buffer. Cell lysate supernatant was syringe filtered (0.44μ) onto the columns. Columns were washed with an additional 20 mL of resuspension buffer, 16 mL of the same buffer at 60 mM imidazole, and 12 mL of the same at 100 mM imidazole. Purified protein was eluted with 6 mL of the same buffer at 400 mM imidazole (i.e., 400 mM imidazole, 500 mM NaCl, 2mM $CaCl_2$, 20 mM MOPS at pH 7.8). Eluted protein was dialyzed at 4° in

10,000 MWCO SnakeSkin (Pierce) against 2-3 changes of 3 L of 100 mM NaCl, 2 mM $CaCl_2$, 20 mM MOPS at pH 7.0. Purity of the resulting protein sample was verified by MALDI mass spectroscopy and/or SDS PAGE electrophoresis.

Protein characterization by circular dichroism (CD)

Because MOPS absorbs strongly in the UV region of interest for CD, purified protein was exchanged into a phosphate buffer (10 mM phosphate, 150 mM NaCl, pH 7.0) using 10 DG buffer-exchange columns (Bio-Rad). No calcium could be included because calcium phosphate is insoluble, but past experiments have nonetheless found GBP designs to generally be stable in phosphate buffer over the time span needed for the CD experiments. Final protein concentration was generally 15-30 μM. This was diluted with phosphate buffer as needed to achieve a CD signal of -50 to -100 at 222 nm, typically 500 μL protein solution plus 2.5 mL phosphate buffer.

Wavelength scans from 200 to 270 nm were used to look for evidence of secondary structure and thus, folded protein (1 nm steps, 1 s averaging, 25°). Temperature melts were performed in the absence of ligand to look for cooperative unfolding and determine a melting temperature (1.5° steps from 20 to 100°, 8 s averaging). Melts were repeated in the presence of 1 mM L-

lactate to look for a change in melting temperature vs. the *apo* form. Melt data was converted to fraction unfolded (*f*) according to $f = (\theta - \theta_{min})/(\theta_{max} - \theta_{min})$.

Computational results

The most striking feature of adding BACKRUB shifts to the DEZYMER receptor design calculation is how those small backbone changes radically expanded the sequence space of designs. For instance, there is very little sequence duplication between scaffolds for the top 100 best scoring designs of each. Since Dezymer simultaneously optimizes sidechain identity, sidechain position, and ligand pose, an individual scaffold may have significant duplication of sequences within its top 100 designed structures. For instance, GW5 has very little duplication (96 sequences for 100 structures), but GW7 has a great deal (44 sequences for 100 structures); this indicates that similar ligand poses often resulted in identical sequence designs for GW7 but not for GW5. The number of unique sequences in the top 100 is shown in the first row and column of Table 8. The rest of the table shows how many of these sequences are duplicated between pairs of scaffolds. Two of the three significant sequence overlaps (GW2 vs GW6, GW7 vs GW9) are in scaffolds whose backbones differ only at position 211; the remaining overlap (GBP vs GW4) has a backbone difference only at position 158. Contrast these examples to the

case of GBP vs. GW2, which differ only at position 16 but share just two designed sequences in common. Note that even for scaffolds that overlap significantly in sequence space, there are more unique sequences than duplicated ones. Globally, there are 637 unique sequences from pooling the top 100 scoring designs from each of these 11 scaffolds, compared to the expected 735 if there were no duplications between scaffolds (but accounting for duplication with scaffolds; i.e. the sum of the first row/column of Table 8).

	unique	GBP	GW1	GW2	GW3	GW4	GW5	GW6	GW7	GW8	GW9	GW10
unique		77	66	57	83	48	96	76	44	61	49	78
GBP	77		0	2	3	**20**	4	2	0	4	0	0
GW1	66	0		0	0	0	0	0	0	0	0	1
GW2	57	2	0		0	2	0	**27**	0	0	0	0
GW3	83	3	0	0		3	4	0	0	6	0	0
GW4	48	**20**	0	2	3		5	3	0	3	0	0
GW5	96	4	0	0	4	5		0	2	0	2	0
GW6	76	2	0	**27**	0	3	0		0	0	0	1
GW7	44	0	0	0	0	0	2	0		0	**24**	0
GW8	61	4	0	0	6	3	0	0	0		0	0
GW9	49	0	0	0	0	0	2	0	**24**	0		0
GW10	78	0	1	0	0	0	0	1	0	0	0	

Table 8: Sequence overlap for top 100 designs for pairs of scaffolds.

The row/column labeled "unique" indicates the number of unique sequences in the top 100 designs for each scaffold; this is less than 100 due to some design results having slightly different predicted structures but the same sequence. Cells in the body of the table indicate how many of those unique sequences are shared between any pair of scaffolds.

The sequence composition of the top 100 designs is shown graphically in Figures 26 and 27. Figure 26 shows one "sequence logo" (Schneider and Stephens 1990; Crooks, Hon et al. 2004) per scaffold, where the height of each letter reflects the prevalence of that amino acid at the particular position. (Only the evolving zone residues are shown.) They are arranged to demonstrate the global impact of shifting the backbone at residue 16 (third from left): not only does that position switch from favoring Phe to favoring Gly, but many other positions also switch their preferences markedly. Comparing Table 7 to Figure 26, one sees that for each scaffold there is general agreement between the selected sequence(s) and the overall profile of the top 100 designs, but the two do not match exactly.

Figure 27 shows a hierarchical clustering of the 637 globally unique sequences from the full set of scaffolds; sequences found in the top 100 of a particular scaffold are highlighted in each panel. Although the clustering scheme is crude relative to those used in constructing phylogenetic trees, it is immediately obvious that the 11 scaffolds populate very different regions of sequence space: not only is there little overlap from one scaffold to the next, but results are more similar within a scaffold than between scaffolds.

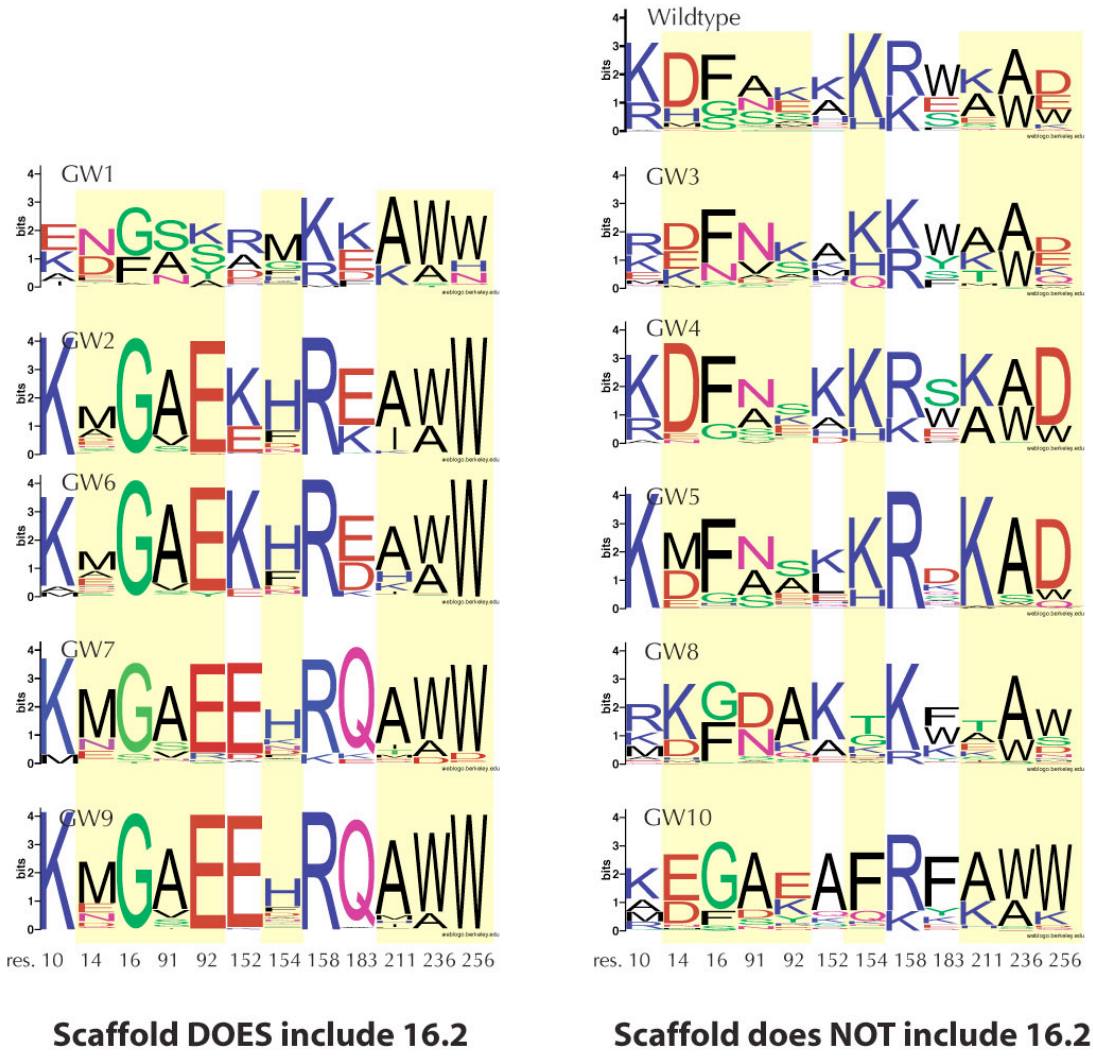
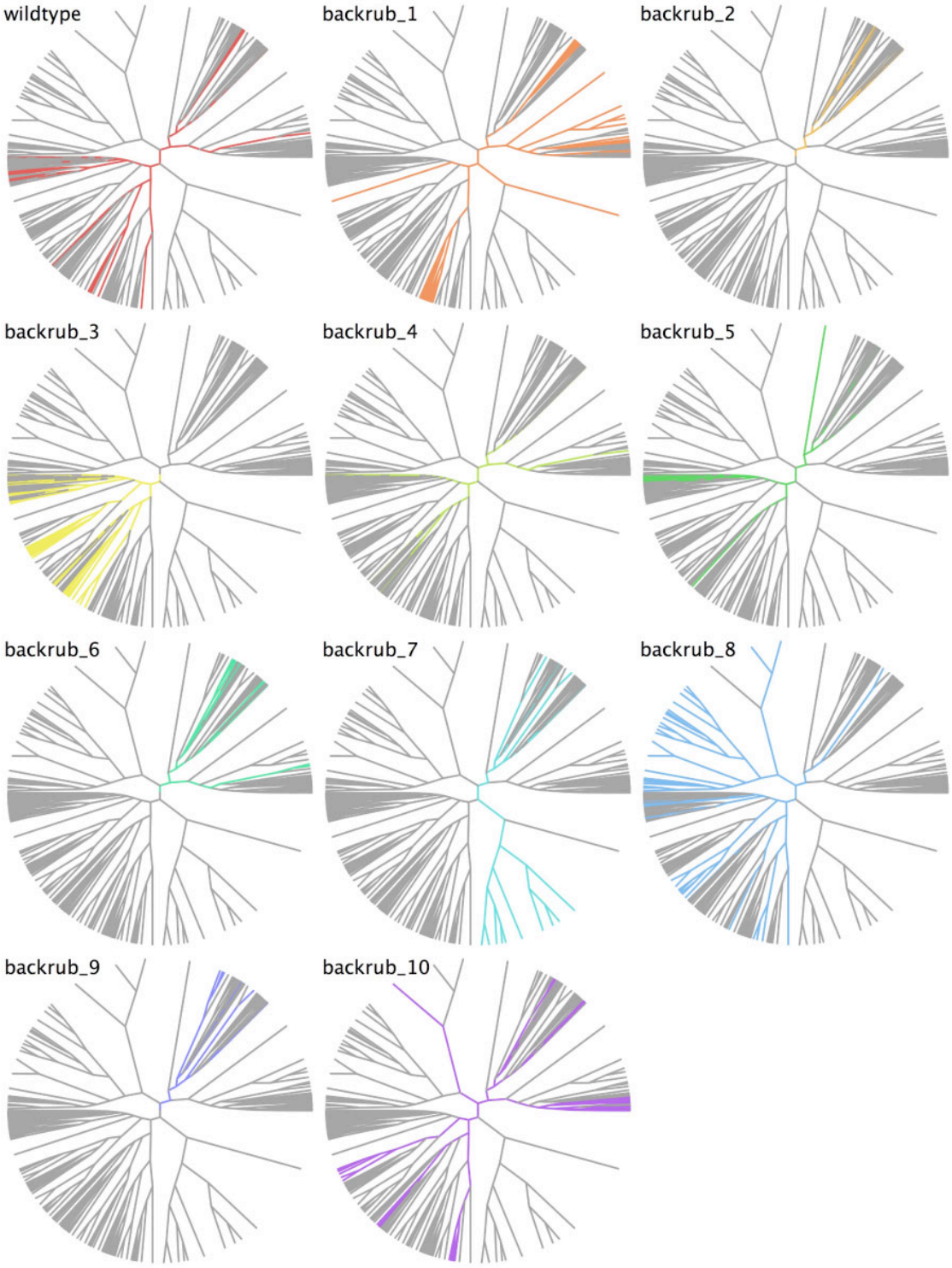


Figure 26: Sequence logos for the top 100 designs for each scaffold.

Only evolving zone residues are shown; at each site, the size of an amino acid code is proportional to the likelihood of it occurring there. Coloring is by residue properties. The left-hand column contains all scaffolds that moved the backbone of residue 16; the right-hand column contains all scaffolds that left the backbone of residue 16 in its experimentally observed position. The yellow bars highlight residue positions that were significantly impacted by the shift at 16.

Figure 27: Hierarchical clustering of the 637 unique sequences in the top 100 designs for all 11 scaffolds.

Points around the edge of the circle represent single sequences; moving toward the center of the circle, sequences and clusters of sequences are merged together. For each scaffold, the sequences and sequence clusters that occur in its top 100 designs are highlighted in color. Some scaffolds populate several of the major branches; others, just one or two. In all cases, they occupy quite different regions of the overall sequence space. The overlaps highlighted in Table 8 can be seen here graphically to some degree (compare GBP & GW4, GW2 & GW6, GW7 & GW9).


wildtype
backrub_1
backrub_2
backrub_3
backrub_4
backrub_5
backrub_6
backrub_7
backrub_8
backrub_9
backrub_10

In analyzing design results, it is also important to understand the relationships among the top sequences produced for a single scaffold. This challenge applies to all DEZYMER calculations, and is not unique to this particular project or approach. To examine the sequence space accessed by a single design calculation, we position the sequences in 3D such that the Euclidean distance between them roughly corresponds to the number of sequence differences between them (see Methods). Figure 28 illustrates the result of this process for the wildtype scaffold. For the top 100 designs in this set of scaffolds, we find that most produce 1-10 clusters of sequences; all members within a cluster are separated by 1-2 mutations, but clusters may be separated from each other by >6 mutations (out of 12 positions). Cluster size varies wildly, from single outlier sequences to clusters of 20 or more.

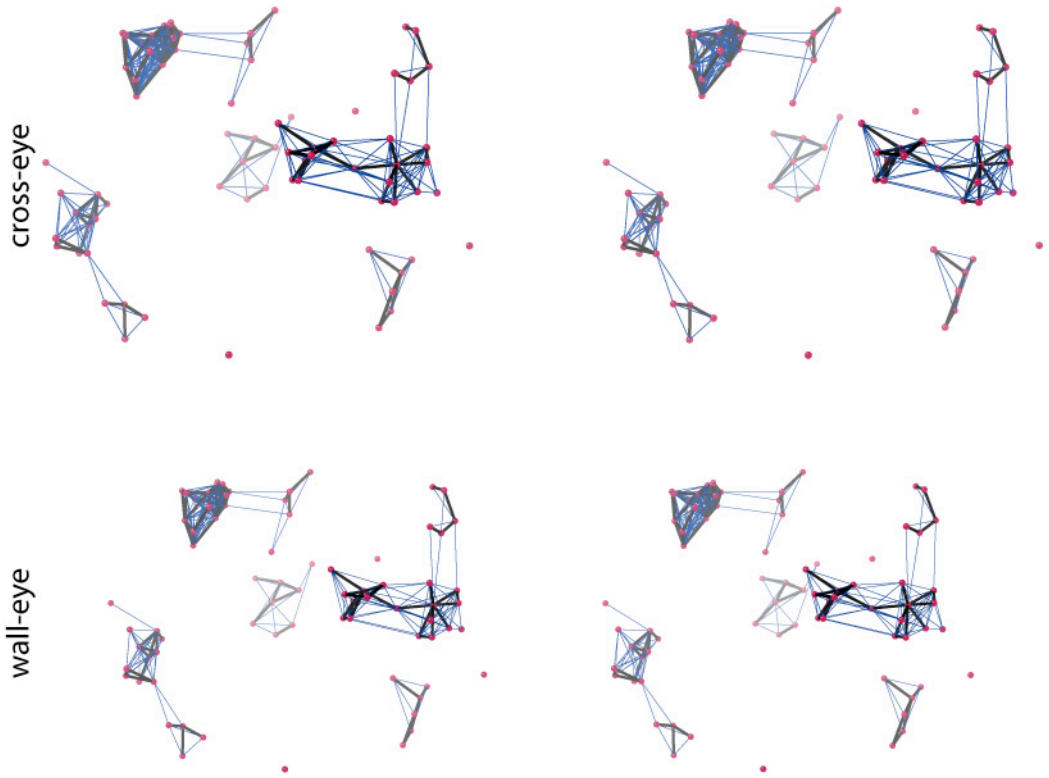


Figure 28: A spatial distribution of sequences for the top 100 designs on the wild type scaffold.

Each red ball represents a unique amino acid sequence; the distance between any pair is roughly proportional to how different the sequences are. Heavy black lines show single mutation steps, thin blue lines show double mutation steps. Note the wide variety of cluster sizes, from 17 members down to 1.

In addition to sequence analysis, the predicted structures of these designs must be scrutinized carefully; scoring the global energy narrows the field, but a single bad interaction in the binding site may wreck activity. Dezymer produces graphical representations of most terms from its scoring system, including electrostatics (particularly hydrogen bond donors/acceptors), van der Waals contacts and clashes, solvent-accessible surface for the ligand, and internal cavities. Although all of these are difficult to visualize effectively, cavities are particularly difficult, in part because they are not real objects, merely negative space between ligand and protein.

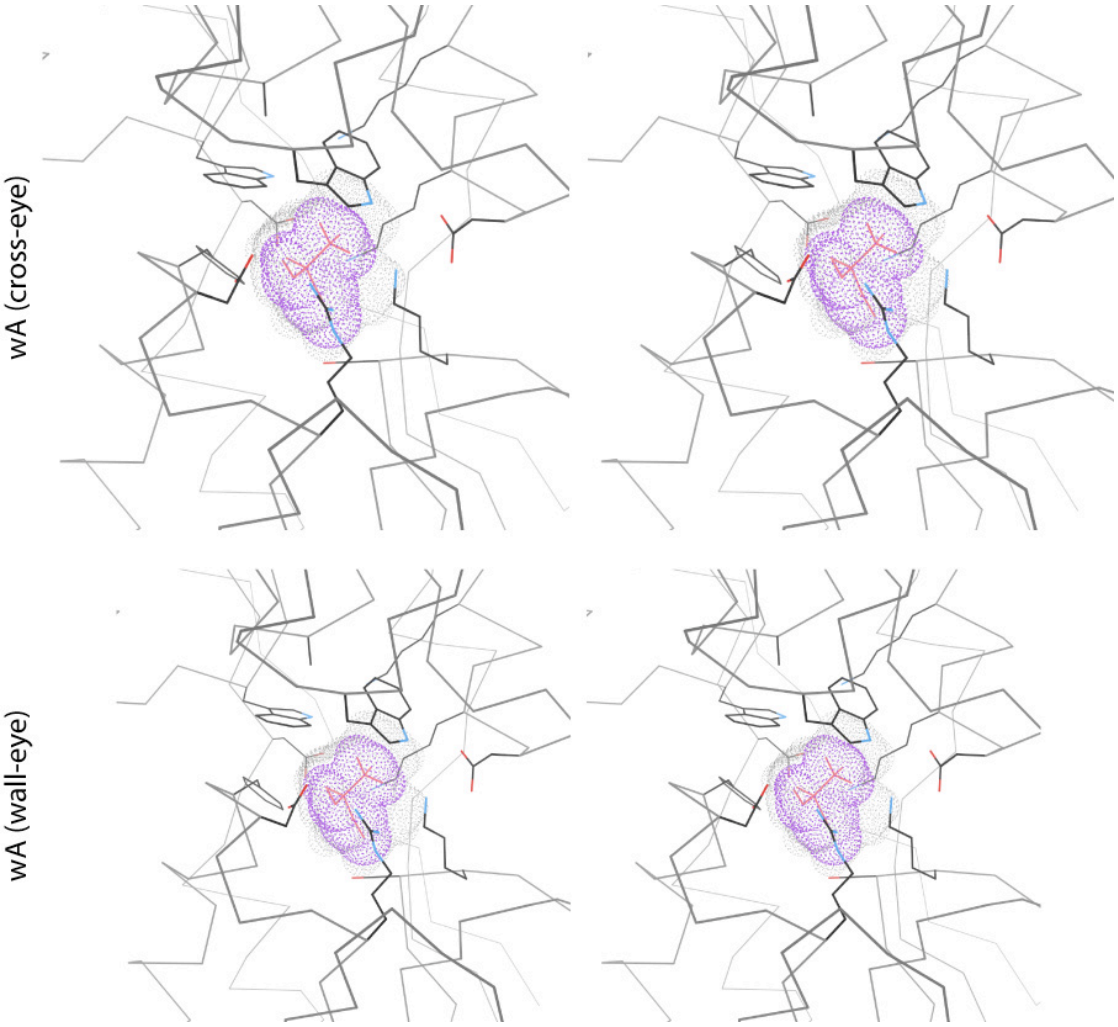
As part of this project, we developed a new method of visualizing cavities, using a pair of dot surfaces (Figure 29). The surface of the binding pocket is mapped by an expanding-foam "foo" (see Methods); the surface of the ligand is calculated by Probe (Word, Lovell et al. 1999a). By controlling dot size, color, and density one can see both surfaces simultaneously. This representation integrates well with all-atom contact dots, and the combination of the two clearly shows how well the ligand is braced in the binding pocket, what forces might hold it there, where there are empty spaces, how large those are, and how they relate to one another. This depiction also provides

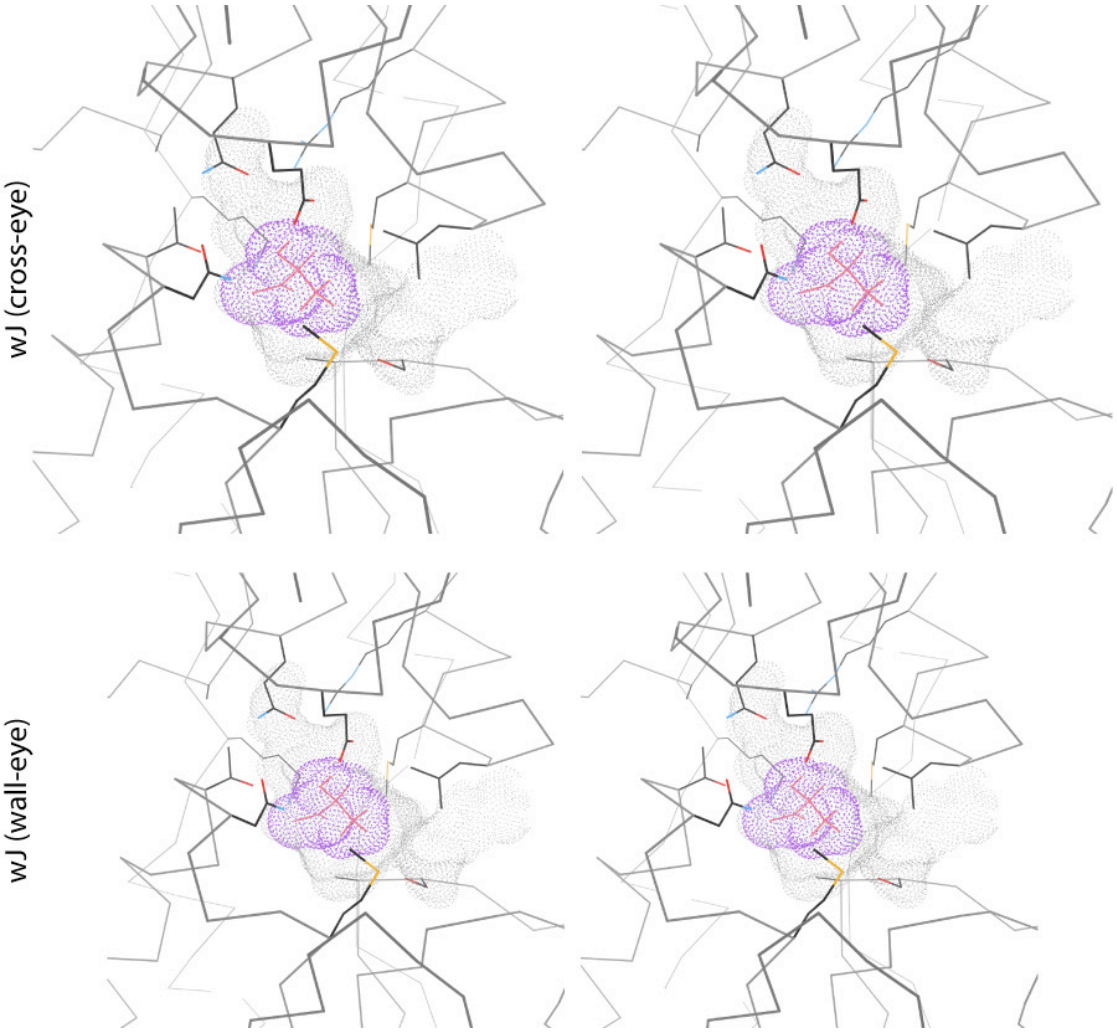
important context for the solvent-accessible surface calculation, and might be useful in predicting ordered waters in the binding pocket.

The selected designs (wA-wJ) show a variety of cavity sizes and distributions (Figure 29). In some the ligand is packed in tightly on all sides (e.g. wA), whereas others have more internal space and/or channels out to the solvent (e.g. wJ). Other designs had still larger cavities and channels, but these were disfavored during the selection process. It is believed that cavities negatively impact protein stability and ligand affinity, but the details are poorly known at this stage.

Figure 29: Comparison of binding pocket cavities in selected designs.

Gray dots show the surface of the binding pocket; purple dots show the van der Waals surface of the lactate. Design wA has very little predicted empty space and exhibits close packing on all side of the lactate. Design wJ is expected to have an internal cavity above the lactate and a channel out to the solvent as well.


wA (cross-eye)
wA (wall-eye)


wJ (cross-eye)
wJ (wall-eye)

Experimental results

Experimental characterization of the selected designs (Table 7) was inconclusive. Although none of the proteins appears to bind lactate with detectable affinity and several of them do not appear to unfold in a cooperative two-state manner (details below), the Hellinga lab routinely characterizes tens to hundreds of designs to find a "hit" for a given target ligand. Given the way backrub motions in the scaffold dramatically broaden the accessible sequence space, it would be premature to dismiss it just because this small set of designs did not show strong binding.

Our primary tool for characterizing the designed proteins was far-UV circular dichroism (CD) spectroscopy. This technique reports on chiral molecules and environments by monitoring relative differences in absorption of left vs. right circularly polarized light; in the range of peptide bond absorbance (240-180 nm), this gives information about protein secondary structure (Kelly and Price 2000). In particular, alpha helices produce a characteristic double minimum at 208 and 222 nm, and GBP is mostly helical. Figure 30 compares the CD spectrum of wild type ecGBP with the designed proteins. While many designs (wC, wD, wE, wF, wJ) reproduced the general shape of the double minimum shown by wild type (heavy black line), they did

so at reduced molar ellipticity, suggesting they have reduced helical content (i.e., are not fully folded). Designs wB, wH, and wI all seem to be missing the second α-helix peak at 208 nm, while wA comes quite close to reproducing the native spectrum in terms of both shape and intensity.

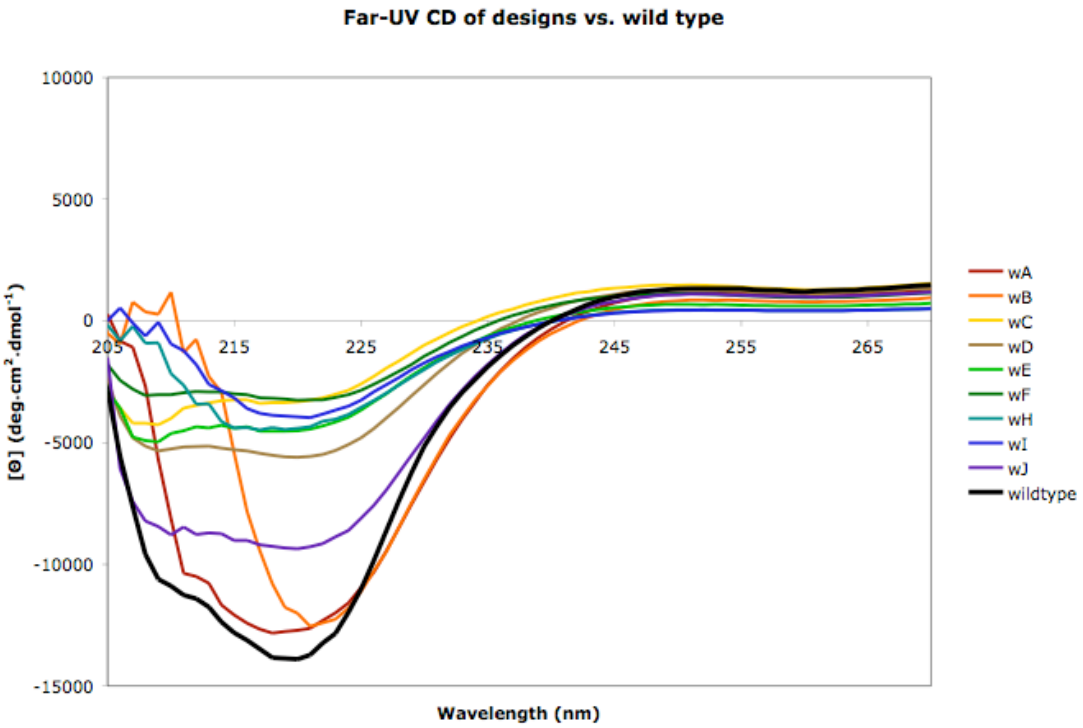


Figure 30: Far-UV circular dichroism wavelength scans for designed proteins (color) versus wild type (heavy black line).

Signal has been normalized by protein concentration. Designed proteins show less evidence of secondary structure than the wild type, to varying degrees.

To assess ligand binding in the designed proteins, we performed temperature melts while monitoring the CD signal at 222 nm. A melting temperature (T_m) can be established for proteins that show a cooperative unfolding transition, and an increase of T_m upon addition of ligand indicates that the protein binds that ligand (and is stabilized by the interaction). Figure 31 displays representative temperature melts for the designed proteins. Wild type ecGBP unfolds cooperatively and shows a ~15° shift in T_m upon addition of 1 mM D-glucose. wA, wB, wD, wF, and wJ also unfold cooperatively, but do not show any change upon addition of 1 mM L-lactate. wE may unfold cooperatively, but its T_m is so low that no initial baseline could be established. wH and wI display erratic, non-cooperative unfolding curves, further reinforcing the idea that they do not adopt a native-like structure. wC is an interesting case. Initially, it gave a two-state cooperative unfolding curve with a significant shift in T_m upon addition of lactate (~5°); however, later experiments did not reproduce that result, instead giving a weakly cooperative three-state unfolding curve that showed no response to lactate. We believe the initial data are (unfortunately) erroneous and possibly due to a failed stir control in the CD instrument.

To support the CD results, we also attempted a few other methods of analysis. ThermoFluor (Pantoliano, Petrella et al. 2001) runs on wB, wC, and wD were noisy and difficult to interpret, but none showed any significant response to lactate. (ThermoFluor follows thermal denaturation of proteins by monitoring fluorescence of an environmentally sensitive dye, which partitions into the protein as it unfolds; the result is a T_m value, analogous to the CD experiements.) Standard fluorescence studies of wC using Trp as an intrinsic fluorophore and isothermal titration calorimetry with wC both showed no change upon addition of L-lactate. Thus, we must conclude that wC does not bind lactate, and the original data are in fact in error.

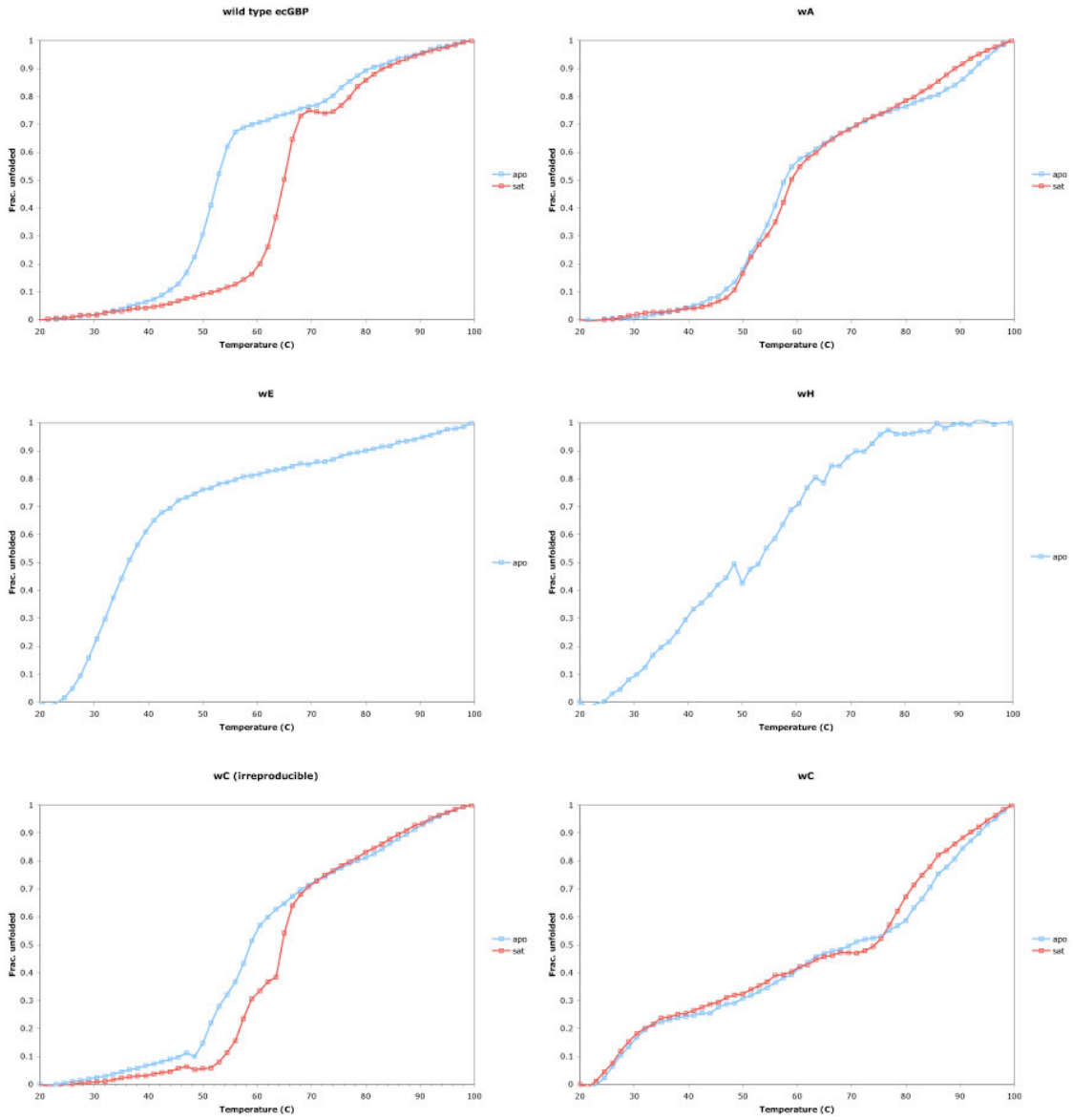


Figure 31: Temperature melts of wild type and designed proteins, monitored by CD at 222 nm.

GBP (top left) shows a significant increase in melting temperature upon addition of glucose, whereas designed proteins show no response to lactate; some designed proteins do not even unfold cooperatively. Initial data for design wC indicated stabilization by lactate (bottom left), but could never be reproduced.

Discussion

This test of using backrub motions in computational protein redesign was ultimately inconclusive: we found no protein that both was stably well folded and bound lactate with high affinity among the small set of designs we characterized experimentally. However, BACKRUB had a strong impact on the results of the design calculation itself. Other design projects in the Hellinga lab have needed to screen many tens of designs to find proteins that are stable, well folded, and have high affinity for their target ligand; only 9 were screened here. Additionally, the observations from x-ray crystallography strongly suggest that the backrub motion is ubiquitous and low energy, increasing the chances it could be successfully engineered into a site. Thus, we believe there is good reason for further study of this application.

Many of our results are supported by the findings of Desjarlais and Handel (Desjarlais and Handel 1999). Although allowing backbone flexibility somewhat degraded the experimental stability of their core repacking designs, the design calculation was able to access a wide range of sequences, many of them quite different from the results on a static backbone. They found that staying close to the wild type backbone structure was critical, and called for a better way of generating realistic backbone motion; we believe BACKRUB

answers that call and has many properties that make it a useful source of local backbone variation for design.

The most obvious impact of adding backbone flexibility to design seems to be the effect on accessible sequence space, and the sequence visualizations developed for this project reveal two interesting facts about the top 1100 designs (100 per scaffold) produced by DEZYMER: (1) Most scaffolds produce 1-10 sequence clusters (related by 1-2 mutations), which may be separated from each other by many mutations; and (2) Very few sequences appear in the top 100 designs for more than one scaffold. The fact that there are only a handful of really different, high-scoring designs for each scaffold makes comprehensive visual inspection much easier; bad clusters can be dismissed wholesale, while good ones suggest point mutations for a given starting point that may address specific flaws. On the other hand, the fact that each scaffold produces such different results suggests spending some combinatorial complexity on variations in scaffold structure may be quite fruitful.

In fact, these clusters of closely related sequences are very similar to the sequence families that emerge from studies of protein evolution using simple lattice models (Desjarlais and Handel 1999). For a single "fold", that work typically finds networks of sequences with equivalent fitness, related to each

other by single mutations. These so-called neutral networks have a central prototypical member with more immediate neighbors than any other; in these simplified models, the prototype is found to be the most stable, and sequences near the edges of the network are the least stable. The sequence clusters produced by DEZYMER usually also have a prototypical central member (see Figure 28). Additionally, evolutionary simulations show that the neutral networks associated with different structures are generally well separated in sequence space. Likewise, the well-separated sequence clusters produced by DEZYMER generally correspond to significantly different structures (i.e., ligand poses). Together, these facts suggest a purely statistical approach to selecting designs for characterization; it would be interesting to see how efficient this strategy is relative to subjective selection by a human expert.

Clearly these are observations about mathematical models rather than about Nature herself, but the approach taken by DEZYMER is typical of many other molecular modeling calculations with semi-empirical molecular mechanics-style force fields. Thus, our results may inform the general thinking about static scaffolds for molecular design and structure prediction. We have shown that even very small, very local motions of backbone (i.e. backrub motions) often move sidechains over relatively large distances. That in turn can easily

produce a large difference in the calculated energy of a structure (e.g. by introducing/relieving large steric clashes), so force field-based methods are likely to be quite sensitive to even small variation in backbone structure, a view shared by Saven and coworkers (Park, Yang et al. 2004).

And small variation in backbone structure is everywhere. It is clear that the atomic details of backbone structure diverge quickly as sequences evolve, even though the global fold is often retained far beyond the point where sequence homology can be detected (Chothia and Lesk 1986). Furthermore, the coordinate error in x-ray crystal structures is of the same order of magnitude as a backrub motion (DePristo, de Bakker et al. 2003), suggesting that for the moderate resolution scaffolds often used in design and modeling, the deposited coordinates may be at least as far from the true equilibrium structure as our BACKRUBbed scaffolds are from the standard GBP structure.

Given the impact of small backbone shifts and the imprecision of typical coordinate sets, it is amazing that computational redesign and homology modeling work as well as they do. Yet the consensus in both fields is that adding backbone flexibility to a calculation generally degrades the quality of the result (Venclovas, Zemla et al. 1997; Desjarlais and Handel 1999; Tramontano, Leplae et al. 2001). Thus we conclude that most of the common

approaches to modeling backbone flexibility simply are not realistic: molecular mechanics force fields do not reproduce the conformational preferences of a pair of peptides (Hu, Elstner et al. 2003), so they are unlikely to build reasonable models if given much latitude with a whole protein. The challenge is to find realistic models of backbone conformational change that support efficient computation, and to develop accurate ways of evaluating alternative backbone models. Although we have not been totally successful thus far, we believe these experiments suggest a promising direction for future work.

Chapter 5: Multi-criterion validation of protein structures

The discipline of structure validation has developed as a cross-check on the reliability of macromolecular structures, helping structural biologists during the process of structure determination and the entire community of users afterward. Programs such as PROCHECK (Laskowski, Macarthur et al. 1993), WHATIF (Vriend 1990), OOPS (Jones, Zou et al. 1991), and MOLPROBITY (Lovell, Davis et al. 2003; Davis, Murray et al. 2004) provide both overall statistical evaluations and flags of local problem areas, concentrating primarily on geometrical measures that can be analyzed from the model. Other validation utilities analyze aspects of the model-to-data agreement, such as SFCHECK (Vaguine, Richelle et al. 1999), real-space residuals (Kleywegt, Harris et al. 2004), water-peak analysis in DDQ (vandenAkker and Hol 1999), and the now almost universally utilized R_{free} value (Brunger 1992). Global validation measures serve the function of judging whether a structure meets accepted current practice, while local measures are especially important to users of structures, since no level of global quality can guarantee local accuracy in the region of your specific interest.

Structure validation is most useful if it is fully independent of the refinement target function, as is the case for R_{free} (if handled correctly), Ramachandran plots (Morris, MacArthur et al. 1992), and all-atom contacts (Word, Lovell et al. 1999a). Chapter 2 describes our work to develop dihedral angle criteria suitable for structure validation (Ramachandran and rotamer distributions) that also best complement all-atom contacts.

MOLPROBITY is a web server that brings together all the structure-validation tools from the Richardson lab (Lovell, Davis et al. 2003; Davis, Murray et al. 2004); it provides step-by-step guidance to make them easy to use -- and use correctly. MOLPROBITY also adds value by integrating information from disparate validation programs to create new tools, such as the multi-criterion displays and scores described here.

An important part of the MOLPROBITY philosophy is our focus on local accuracy. Many structural questions are answered by looking at just a few atoms or residues; thus, good overall accuracy cannot protect us from misinterpreting the structure if there is a local error in our region of interest. All of the fundamental validation tools in MOLPROBITY operate at the atom or residue scale: PROBE (Word, Lovell et al. 1999a) finds van der Waals contacts, H-bonds, and overlaps between pairs of atoms; RAMACHANDRAN and

ROTAMER report a per-residue percentile score for every protein backbone and sidechain (see Chapter 2); and PREKIN checks some critical aspects of the local geometry in proteins (Cβ deviation (Lovell, Davis et al. 2003)) and RNA (base-phosphate distance vs. sugar pucker (Murray, Richardson et al. 2006)). Outlier thresholds for these local measures have been tuned so as to jointly diagnose nearly all places where the structure is in the wrong local minimum conformation, but to give as few "false alarms" as possible.

However, MOLPROBITY supports a large and diverse community of users (see Results), and many of their applications benefit from global quality scores as well. Some validation metrics are not well suited to global scores because of their statistics. For instance, although single Cβ deviation outliers are diagnostic, the average is more indicative of refinement parameters than anything else. Also, most good structures have only 0 or 1 Ramachandran outliers, so percentage Ramachandran outliers behaves erratically as a global quality score, especially at short chain length. (Only about 1 in 2000 residues is expected to be a legitimate outlier.) On the other hand, we expect ~2% of residues to fall outside the favored Ramachandran region even in a perfectly correct structure, and we expect ~1% of sidechains to be non-rotameric. Likewise, clashscore (number of steric overlaps >0.4Å per 1000 atoms) is

useful as a global metric because almost all structures have some bad clashes; typical clashscores range from 5 to 30. Of course, many clashes can be eliminated by careful refitting and refinement. We know several crystallographers who watch clashscore as carefully as R_{free} during their refinements!

These global metrics -- clashscore, percent rotamer outliers, and percent Ramachandran favored/outlier -- appear in MOLPROBITY as a useful validation summary (Figure 32). There is, however, a problem of interpretation: what constitutes satisfactory scores given the experimental data that were available? What scores mark a structure as "good enough" to support homology modeling, molecular dynamics, or computational design? These are hard questions for us to answer, and even more so for MOLPROBITY novices. For crystal structures, we are able to provide for each global score a percentile ranking versus structures of comparable resolution; for instance, MOLPROBITY reports clashscore percentile as a function of crystallographic resolution. This has several problems, though. First, it endorses mediocrity: crystallographers may believe refinement is complete when a structure is in the 50th percentile for its resolution, but we have shown that with some careful attention it is possible to get almost any structure into the top 10% (or better) of scores for its

resolution (see Results and Arendall, Tempel et al. 2005). Second, such rankings can be done only for crystal structures; NMR and homology modeled structures must be ranked against the whole database because they have no practical equivalent to resolution. (NMR constraints per residue is somewhat like resolution, but the information is distributed very unevenly throughout the structure.)

Yet crystallographic resolution is the measure of accuracy that the structural biology community understands most intuitively. Thus, an alternative way of contextualizing the validation scores from MOLPROBITY: for each score, we can report what crystallographic resolution it is most typical of. We call this concept "effective resolution". It immediately solves the second difficulty, that of excluding NMR and homology models, and even allows comparison with x-ray models (albeit in a limited domain). We can also solve the first difficulty of percentile scores (mediocrity); instead of reporting the resolution where the score would be typical (50th percentile), we report the resolution where the score is excellent (top 10th percentile, say). Now having the effective resolution match the actual resolution requires that the crystallographer push the limits of accuracy possible with his/her data, a bit of harmless social engineering that will hopefully encourage more accurate structures.

The "effective resolution" concept works, of course, because each of the MOLPROBITY quality scores is correlated with crystallographic resolution (see Figure 35). Calculating the effective resolution is simply a matter of predicting resolution based on the value of a quality metric, such as clashscore. In this case, very reasonable fits can be had by a simple linear regression on the logarithm of clashscore, percent rotamer outliers, or percent Ramachandran not favored; we have eschewed more complicated fitting techniques in favor of this simple and comprehensible approach.

From this point, we are also very close to addressing an oft-requested feature: to provide a one-number global validation score that incorporates data from all the individual scores in MOLPROBITY. Part of the problem with this request is that it is unclear how to best combine the individual scores, and what weighting to give them. Our experience with effective resolution based on a single validation criterion immediately suggests that this unified score be an effective resolution calculated from multiple criteria. The most straightforward approach is then to do a multiple linear regression, which yields an easily understood functional form: a weighted sum of the individual scores, with the relative weights set to optimize the fit to resolution.

The challenge in structure validation is always to make sense of a tremendous amount of data without oversimplifying things, so here we present techniques for unifying and summarizing the outputs of MOLPROBITY visually, textually, and numerically. First, we allow users to quickly and visually locate alarmingly unusual regions in a particular structure, while preserving atomic detail about the nature of these probable problems, by constructing "multi-criterion" kinemages and charts. Second, we use all-atom steric contacts and multi-dimensional dihedral distributions to estimate the overall accuracy and reliability of a protein structure; the result is a unified, global validation metric presented as a single number on an intuitive scale -- the MOLPROBITY score, M_{ctm}.

Methods

Multi-criterion kinemages ("multi-kins") are assembled in multiple passes by combining output of PREKIN, PROBE, our rotamer and Ramachandran analyses, and PHP scripts within MOLPROBITY. To establish a frame of reference, there is an all-atom stick representation (including hydrogens) along with a Cα trace / pseudo-backbone. There is also a ribbon schematic, which may be colored N-to-C or by *B*-factor. Steric interactions and hydrogen bonding are shown with all-atom contact dots (Word, Lovell et al. 1999a),

although initially only bad clashes are visible. Ramachandran outliers have their backbone highlighted, and rotamer outliers have their sidechain highlighted. Large Cβ deviations (>0.2Å) are shown with balls centered on the ideal Cβ position. For nucleic acids, the base-phosphate distance is compared to the sugar pucker, and outliers are flagged. For crystal structures, all-atom stick models can be colored by *B*-factor and/or occupancy, and alternate conformations are highlighted. For NMR structures with experimental constraints available, NOE distance violations are shown. All these pieces are manipulated so that only a minimum depiction of the structure is initially presented to the user, with only the serious outliers flagged; additional details are turned on when desired. A typical example of a multi-criterion kinemage is shown in Figure 32.

Multi-criterion charts ("multi-charts") contain the same basic information as multi-kins, but in a tabular, sortable text format: serious all-atom clashes, bad rotamers, Ramachandran outliers, large Cβ deviations, questionable sugar puckers, and high *B*-factors. The multi-chart can be printed out in any of its sorted states (default is by residue number), or it can be downloaded as an HTML file for viewing and re-sorting in a web browser. The outlier information can be presented within a crystallographic rebuilding program as

a "to do" list; that type of output is currently made available for COOT (Emsley and Cowtan 2004). A sample multi-chart is shown in Figure 32. Note that outlier entries are highlighted in hot pink, but that more detailed information is also reported. For each clashing residue, the overlap of its worst clash is given (that value is the sort index for the clash column), plus the identity of the two atoms producing the clash. The Ramachandran column reports each residue's status as favored, allowed, or outlier; its percentile score (the sort index); which of the four distributions it falls into (general case, Gly, Pro, or pre-Pro); and the actual ϕ and ψ values. RNA residues show *B*-factor, clashes, and pucker diagnosis; DNA, ligands, and waters show only *B* and clashes.

The multi-chart information is also summarized as per-criterion global scores, as shown in Figure 32. The summary scores are clashscore (all-atom and $B < 40$); clashscore percentile rank among structures of similar resolution; percent rotamer outliers; percent Ramachandran outliers and favored; number of sugar pucker outliers; and number of Cβ deviations. Each criterion is color-coded on a stoplight scale (red = bad, yellow = questionable, green = good) for rapid visual assessment.

Figure 32: Sample multi-chart and multi-kin for PDB file 2SIM.

(a) The validation summary lists global scores for all-atom clashes, rotamers, Ramachandran classes, and more. Scores are colored on a red-yellow-green "stoplight" scale. An early test of the MOLPROBITY score is labeled "MER" in this figure. (b) The multi-chart organizes validation criteria by columns, lists residues in sequence order by rows, and flags outliers in hot pink. The chart is sortable on any of the columns (e.g., to see all the worst rotamers). (c) The multi-kin shows all-atom contacts (hot pink spikes), rotamer outliers (gold), Ramachandran outliers (green), and Cβ deviations (magenta balls), all overlaid on the 3D structure.

a.

Summary statistics

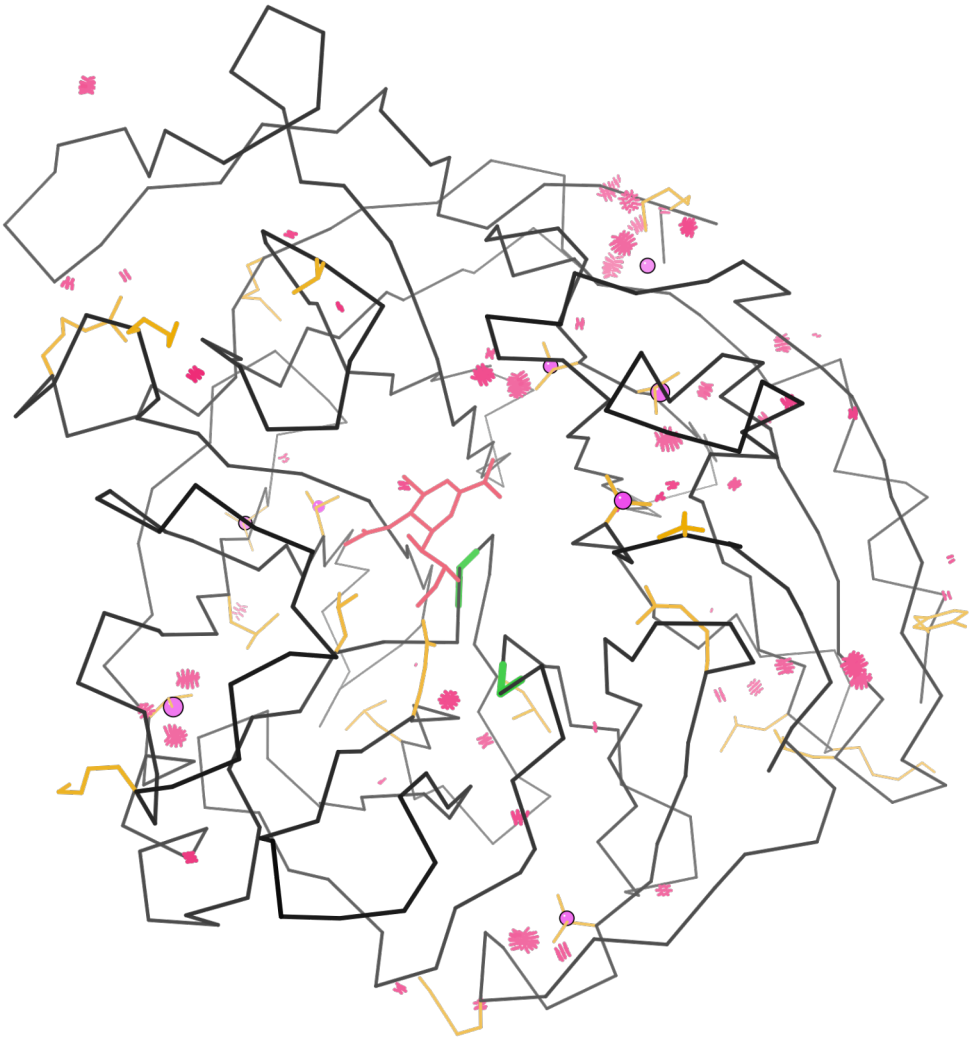
All-Atom Contacts	Clashscore, all atoms:	14.3	38th percentile* (N=718, 1.35Å - 1.85Å)
	Clashscore, B<40:	13.79	17th percentile* (N=718, 1.35Å - 1.85Å)
Protein Geometry	Rotamer outliers	8.56%	Goal: <1%
	Ramachandran outliers	0.53%	Goal: <0.2%
	Ramachandran favored	96.04%	Goal: >98%
	Cβ deviations >0.25Å	8	Goal: 0
	MER [ALPHA TEST - don't ask]	2.64	9th percentile* (N=7200, 1.35Å - 1.85Å)

* 100th percentile is the best among structures of comparable resolution; 0th percentile is the worst.

b.

Res	High B	Clash > 0.4Å	Ramachandran	Rotamer	Cβ deviation
	Avg: 18.74	Clashscore: 11.24	Outliers: 2 of 379	Outliers: 25 of 327	Outliers: 8 of 348
2 THR	35.69	0.916Å HB with 76 GLY 2HA	-	21.6% angles: -50.7	0.374A
3 VAL	35.31	0.666Å 1HG1 with 6 SER 1HB	Favored (21.75%) General case / -101.0,152.1	4.2% angles: 75.9	0.151Å
4 GLU	36.33	-	Favored (66.93%) General case / -72.6,-30.3	1.5% angles: -108.8,-158.4,52.6	0.08Å
5 LYS	33.47	0.922Å 1HZ with 367 GLU 2HB	Favored (8.43%) General case / -155.3,130.3	0% angles: -153.7,-87.3,87.1,51.9	0.121Å
6 SER	18.73	0.71Å N with 5 LYS 2HE	Favored (33.19%) General case / -150.2,160.8	57.3% angles: 72.5	0.046Å
7 VAL	14.37	-	Favored (44.03%) General case / -93.4,126.0	52.8% angles: -178	0.048Å
8 VAL	11.52	-	Favored (6.39%) General case / -85.7,-53.8	33.3% angles: -174.1	0.075Å
9 PHE	11.02	-	Favored (60.25%) General case / -128.1,131.8	64.4% angles: -67.1,-66	0.119Å
10 LYS	26.15	-	Favored (49.5%) General case / -104.3,118.8	23.7% angles: -176.2,70.7,-156.7,167	0.042Å
11 ALA	10.57	-	Favored (38.67%) General case / -55.4,133.9	-	0.1Å
12 GLU	17.1	-	Favored (5.14%) General case / 53.5,25.3	23.5% angles: -57.2,-162.3,34.4	0.044Å
13 GLY	15.99	-	Favored (4.47%) Glycine / -134.4,7.1	-	-
14 GLU	14.48	-	Favored (47.67%) General case / -117.4,142.7	71.9% angles: -60.4,-59.4,-17.4	0.096Å
15 HIS	20.13	-	Favored (63.97%) General case / -120.7,129.9	13.9% angles: -83.1,113.6	0.064Å
16 PHE	11.72	-	Favored (27.35%) General case / -116.2,157.1	89.7% angles: -64.2,-76.3	0.068Å
17 THR	15.31	-	Favored (27.63%) General case / -114.4,155.3	24.6% angles: 50.6	0.053Å
18 ASP	17.96	0.509Å 1HB with 596 HOH O	Favored (2.23%) General case / -86.1,-167.9	31% angles: 62.6,25.4	0.027Å
19 GLN	31.44	-	Favored (73.09%) General case / -62.8,-30.4	41.4% angles: -66.3,172.2,93.6	0.095Å
20 LYS	35.33	-	Favored (53.67%) General case / -87.5,-4.8	6.3% angles: -71.5,-61.1,-175.7,104.6	0.101Å

c.



To combine the various global scores from the multi-chart into a single number, we also developed the MolProbity score. The simple MOLPROBITY score M_{ctm} was determined by fitting a log-linear model of clashscore (c), rotamer outliers (t), and Ramachandran favored (m) to the crystallographic resolution, for all proteins between 0.75 and 3.50Å in the Protein Data Bank as of May 2006 (27,674 structures); the few structures available outside that resolution range are generally exceptional in one way or another, and we wished to avoid any bias from those unusual cases. Structures with fewer than 40 residues scored for rotamer or Ramachandran were excluded, as either (1) they were short peptides, (2) they were missing atoms (e.g. Cα only), or (3) there was a serious nomenclature problem. Minor nomenclature problems may have occasionally resulted in spurious clashes, but we expect this to be a relatively small effect. On the other hand, a large number of nominally low-resolution structures may have been phased by molecular replacement from higher resolution models, leading to systematically higher accuracy than would be expected for experimentally phased data. Because this is such common practice, however, we chose for our score to reflect the current reality of the database rather than use only experimentally phased structures. After all, many clients of MOLPROBITY will be validating structures solved by

molecular replacement. To be specific about the score components, *c* is the number of all-atom steric overlaps per 1000 atoms, regardless of *B*-factor (Word, Lovell et al. 1999a); *t* is the percentage of sidechain conformations classed as rotamer outliers out of the sidechains that can be evaluated; and *m* is the percentage of backbone conformations *not* classed as favored out of the residues that can be evaluated. Lower numbers mean better quality for all three score components.

Naïve fits exhibited a systematic bias in the residuals as a function of crystallographic resolution; that is, very high resolution structures typically scored worse than their actual resolution, and lower resolution structures were biased toward better scores. This bias is an artifact of the natural distribution of structure resolutions, and is exacerbated by the hard resolution cutoffs for this data set. The distribution is such that fitting a quality score as a function of resolution produces no bias, but the reverse does cause bias. (See schematic explanation in Figure 33). We were unwilling to accept a large bias of this type, because it seemed a minimum requirement for such a fit is that an average-quality structure within each resolution bin be assigned an effective resolution close to its actual resolution.

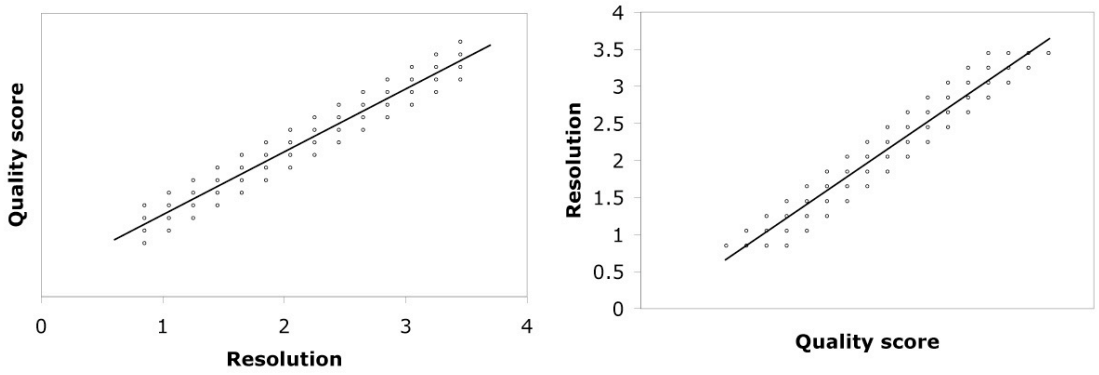


Figure 33: Schematic explanation of systematic bias in regression of resolution on quality.

(Left) Fitting quality as a function of resolution produces the expected straight line through the middle of the distribution. (Right) Fitting resolution as a function of quality (to determine an "effective resolution") leads to a skewed fit, because at the extremes of quality, data are "missing" for some resolution values. Put another way, standard regression fits try to minimize the squared residuals in *y*, at every *x*.

To avoid the bias, we instead performed a fit to points derived from the distribution. Structures were divided into 0.2 Å bins by resolution: [0.75 Å, 0.95Å), [0.95 Å, 1.15Å), …, [3.35 Å, 3.55Å). Initially, we tried one point per bin, where each score took on its median value within that bin. This technique eliminated the systematic bias by replacing a cloud of points with a line; however, the resulting fit was poorly correlated with the overall distribution, probably because too few points were available to get a robust fit. Instead, we created six points per bin, for each of which two scores took on their median value while the third took its first or third quartile value within the bin (see Table 9). The intuition was that this should "stabilize" the fit by sampling more than just the core of the distribution. This produced a fit that correlates well with the overall distribution but lacks the systematic bias as a function of crystallographic resolution (although the exact reason for this nice behavior is unclear). It should be noted that there is now a systematic bias to the residuals as a function of the fitted value itself (i.e., the correlation coefficient is slightly lower than for the naïve fit), but the shape of the distribution itself forces trading off one bias versus the other. Our modified fit follows the pattern that would be seen in the full distribution, rather than optimizing to fit the partial subset of the distribution that is actually observed.

Table 9: Derived data points used in fitting the MOLPROBITY score.

Within each resolution bin, the first, second, and third quartiles were calculated. (Second quartile is the median.) These were combined to generate 6 data points; for each point, exactly two scores take on their median value and the other takes on its first or third quartile value. The resolution value is the center of the resolution bin. Note that rotamer and Ramachandran scores were "clipped" as described in the text.

resol. (Å)	Num. data points	Clashscore c	Rotamers max(0,t-1)	Ramachan. max(0,m-2)
0.85	60	6.01	0.52	0.00
		6.01	0.52	0.64
		6.01	0.00	0.00
		6.01	1.42	0.00
		3.37	0.52	0.00
		10.85	0.52	0.00
1.05	346	5.77	1.16	0.00
		5.77	1.16	0.54
		5.77	0.23	0.00
		5.77	2.51	0.00
		3.83	1.16	0.00
		8.72	1.16	0.00
1.25	512	7.12	1.27	0.00
		7.12	1.27	0.53
		7.12	0.39	0.00
		7.12	2.66	0.00
		4.51	1.27	0.00
		10.68	1.27	0.00
1.45	1587	8.58	1.63	0.00
		8.58	1.63	0.92
		8.58	0.54	0.08
		8.58	3.21	0.08
		5.84	1.63	0.08
		12.31	1.63	0.08
1.65	2799	10.12	2.09	0.00
		10.12	2.09	1.09
		10.12	0.82	0.25
		10.12	4.26	0.25
		6.86	2.09	0.25
		14.17	2.09	0.25
1.85	5532	12.19	3.05	0.00
		12.19	3.05	1.43
		12.19	1.39	0.47

		12.19	5.60	0.47
		8.33	3.05	0.47
		16.93	3.05	0.47
2.05	4983	15.00	4.26	0.10
		15.00	4.26	2.17
		15.00	2.16	1.08
		15.00	7.54	1.08
		10.17	4.26	1.08
		20.89	4.26	1.08
2.25	3887	18.43	5.61	0.71
		18.43	5.61	3.25
		18.43	3.15	1.78
		18.43	9.31	1.78
		12.55	5.61	1.78
		25.75	5.61	1.78
2.45	3172	22.11	6.91	1.39
		22.11	6.91	4.98
		22.11	4.18	2.87
		22.11	11.49	2.87
		15.00	6.91	2.87
		30.77	6.91	2.87
2.65	1904	26.64	8.14	2.60
		26.64	8.14	7.38
		26.64	5.00	4.40
		26.64	12.69	4.40
		18.71	8.14	4.40
		37.54	8.14	4.40
2.85	1794	33.38	9.57	3.70
		33.38	9.57	10.31
		33.38	6.19	6.22
		33.38	15.07	6.22
		22.28	9.57	6.22
		46.31	9.57	6.22
3.05	1044	40.65	11.39	4.92
		40.65	11.39	14.03
		40.65	7.12	8.42

		40.65	18.66	8.42
		26.65	11.39	8.42
		56.65	11.39	8.42
3.25	441	49.12	12.54	6.99
		49.12	12.54	17.32
		49.12	8.18	12.29
		49.12	19.22	12.29
		31.72	12.54	12.29
		70.05	12.54	12.29
3.45	183	59.37	13.22	9.38
		59.37	13.22	23.33
		59.37	8.85	16.48
		59.37	20.24	16.48
		37.19	13.22	16.48
		82.69	13.22	16.48

Score components were transformed before fitting to improve the quality and relevance of the fit. All three scores individually underwent a log transform, as in each case that improved the linearity of the score-to-resolution relationship. Additionally, the rotamer and Ramachandran scores were "clipped" (at 1 and 2%, respectively) so as not to reward structures that were excessively ideal; due to the process by which rotamer and Ramachandran outliers are evaluated (Chapter 2), very good structures are still expected to have ~1% rotamer outliers and ~2% non-favored residues on the Ramachandran plot. Finally, after fitting the intercept term was set to 0.5 (instead of 0.088). This change makes M_{ctm} closer to the score of the best structures at a given resolution rather than the average structure at a given resolution, but because the fit coefficients are unchanged it does not alter the correlation with resolution. (By definition, those coefficients are selected to optimize said correlation.) It also means that "perfect" component scores lead to a MOLPROBITY score of 0.5, which is a physically attainable resolution and one at which we believe it should be feasible to attain perfect scores.

The final score function then takes the form

$$M_{ctm} = 0.42574 \cdot \ln(1+c) + 0.32996 \cdot \ln(1+\max(0,t-1)) + 0.24979 \cdot \ln(1+\max(0,m-2)) + 0.5$$

It has an R of 0.7 versus crystallographic resolution, which is higher than the R for any one of the scores individually (Table 10). It is slightly less than the R for the naïve fit, for the reasons described above. Plots of the residuals are shown in Figure 34. A pseudo-four-dimensional kinemage plot of resolution vs. *c*, *t*, and *m* is available in the digital appendix, with the fit "line" represented by stacked planes.

The methods used to test MOLPROBITY's ability to enhance the accuracy of x-ray crystal structures have been described in detail elsewhere (Arendall, Tempel et al. 2005), but they are summarized here. A working set of 29 structures from the Southeast Collaboratory for Structural Genomics (SECSG) was refined to completion using validation information from MOLPROBITY during every rebuilding cycle. A control set of 19 structures from the SECSG was refined to completion using similar protocols, but without MOLPROBITY; only PROCHECK (Laskowski, Macarthur et al. 1993) and standard crystallographic measures were used for validation. Additionally, a representative sample of experimentally-phased structures was taken from the

PDB for comparison. Characteristics of the three data sets are summarized in Table 11.

Score	Correlation (Pearson's *R*)
clashscore, *c*	0.537
log(1+*c*)	0.620
% rotamer outliers, *t*	0.508
log(1+*t*)	0.555
log(1+max(0,*t*-1))	0.555
% Ramachandran not favored, *m*	0.611
log(1+*m*)	0.659
log(1+max(0,*m*-2)	0.676
naïve fit to log(1+*c*), log(1+*t*), log(1+*m*)	0.712
final MolProbity score, M_{ctm}	0.702

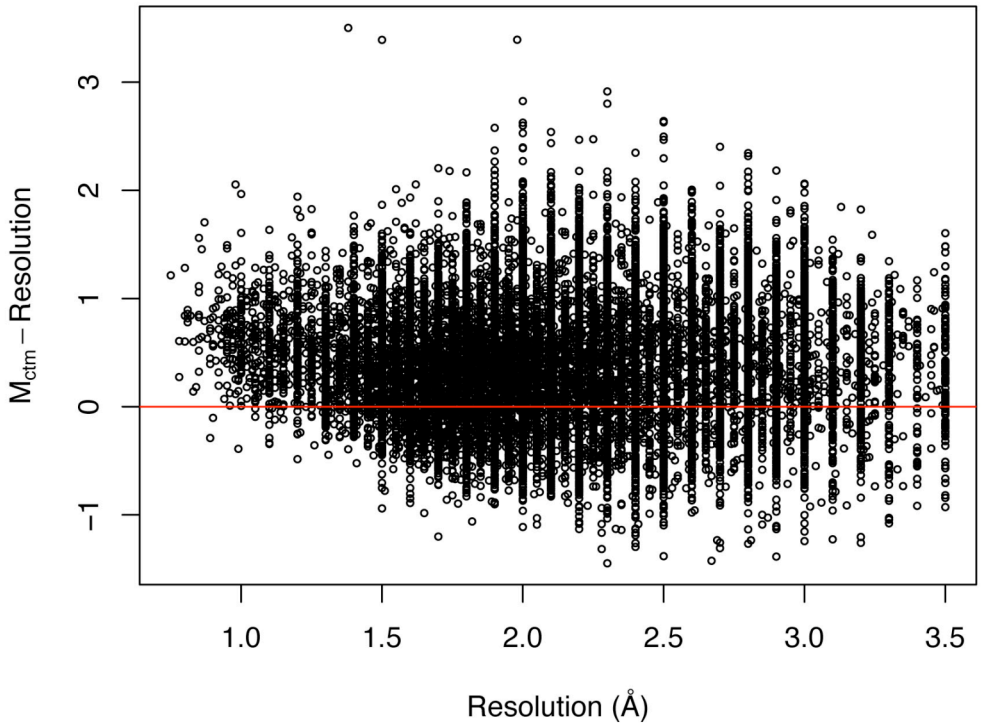
Table 10: Correlation of various quality scores with crystallographic resolution.

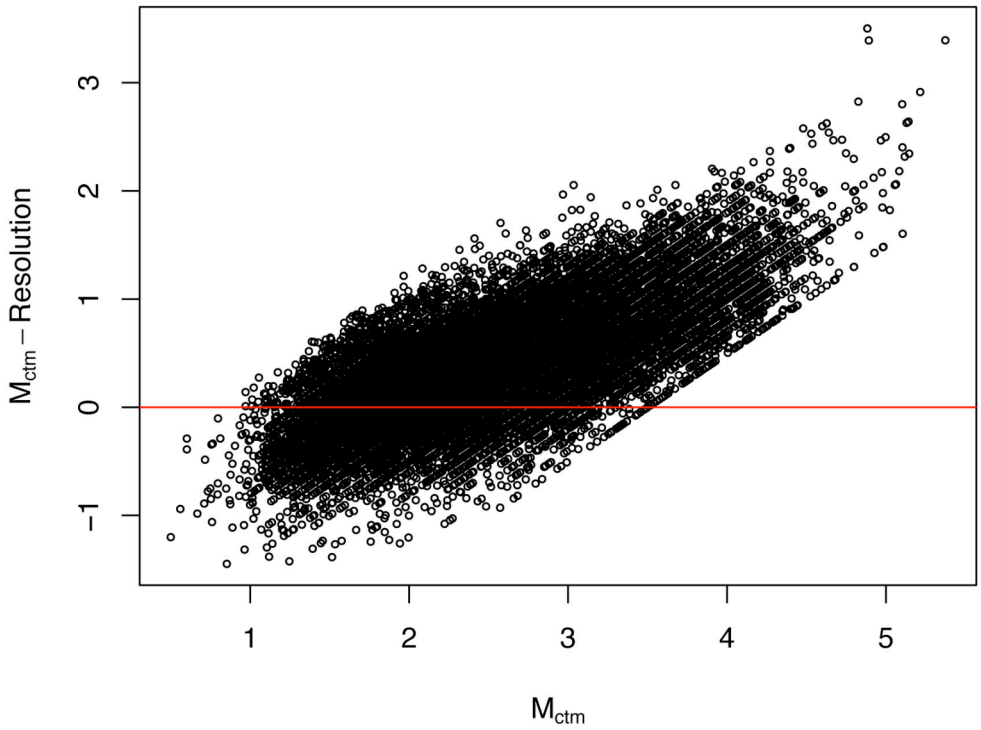
Figure 34: Residuals of the MOLPROBITY score compared to resolution.

(top) The difference between the MOLPROBITY score and the actual resolution is well distributed with regard to resolution, indicating a "fair" evaluation. The mean residual is greater than zero (~ 0.4) because the constant term of the fit equation was artificially raised to 0.5. (bottom) There is a trend to the residuals when plotted against MOLPROBITY score; such a trend must exist in this plot if the upper plot is to be flat, based on the characteristics of our data distribution (see Figure 33). This trend means that there is still room to make the fit follow resolution more closely, but doing so creates other, more troubling biases (see text).

Figure was created with the following commands in the statistics program R:

```
oldpar <- par(mar=c(5,6,4,2))

plot(pdb.clip12$resol, predict(pdb.clip12.lm,
pdb.clip12)+0.41245-pdb.clip12$resol,
ylab=expression(M[ctm] - Resolution),
xlab="Resolution (Å)", cex=0.5)

plot(predict(pdb.clip12.lm, pdb.clip12)+0.41245,
predict(pdb.clip12.lm, pdb.clip12)+0.41245-
pdb.clip12$resol, ylab=expression(M[ctm] -
Resolution), xlab=expression(M[ctm]), cex=0.5)

abline(h=0, col='red')

dev.print(device=pdf)
```


$M_{ctm} - Resolution$
3
2
1
0
−1
1.0
1.5
2.0
2.5
3.0
3.5
Resolution (Å)


$M_{ctm} - Resolution$
3
2
1
0
−1
1
2
3
4
5
M_{ctm}

Parameter	Working set	Control set	PDB sample
No. of structures	29	19	1784
Primary source	*P. furiosus*	*C. elegans*	
Resolution (min.)	1.2Å	1.47Å	0.54Å
Resolution (avg.)	1.91Å	1.97Å	1.86Å
Resolution (median)	1.90Å	1.80Å	1.80Å
Resolution (max.)	2.7Å	2.9Å	3.5Å
R_{work}	20.1%	20.%	
R_{free}	23.4%	24.4%	
RMS bond angle	1.26°	1.34°	
Real-space R	12.8%	13.4%	
Asn/Gln/His flips	0.0%	18.9%	19.3%
Non-rotamericity	1.2%	3.7%	6.04%
Rama. outliers	0.05%	0.54%	0.68%
Rama. favored	98.3%	96.9%	94.9%
Clashscore	3.16	17.6	18.5
MolProbity score, M_{ctm}	1.21Å	2.09Å	2.35
M_{ctm} - resolution	-0.70Å	+0.11Å	+0.48

Table 11: Summary of structures used to test MolProbity validation and structure improvement.

For the working-set structures, the validation and structure-improvement tools of the MOLPROBITY web service and related utilities (Lovell, Davis et al. 2003; Richardson, Arendall et al. 2003; Davis, Murray et al. 2004) were used throughout the refinement process, typically starting immediately after initial chain tracing. Subsequent model rebuilding at the SECSG focused on sections of the model where severe deficiencies were found in these tests: < 1% probability for side-chain rotamers, < 0.2% probability for ϕ,ψ values, or all-atom van der Waals overlaps > 0.5Å. The residue-by-residue real-space fit to the experimental data, as shown by SFCHECK (Vaguine, Richelle et al. 1999), also guided the refitting process. Cycles of validation, refitting, and refinement were repeated until further changes failed to improve the steric parameters or the fit to electron density.

After convergence, the coordinates and structure factors were sent to our group at Duke University. We examined the remaining problem areas on a multi-criterion kinemage in the KING (Davis, Murray et al. 2004) or MAGE (Richardson 2003) display programs, along with $2F_o$-F_c and F_o-F_c electron density, and did further rebuilding where feasible. Changes included correction of side-chains trapped in the wrong local minimum conformation (much less common toward the end of this study, when they were mostly

being corrected earlier), redefining solvent or ligands, and adding further parts of the structure that could be built reliably with the tools in KiNG and MAGE for small backbone movements and for interactive rotamer and contact evaluation while remodeling (Word, Bateman et al. 2000). If further changes were made, then one more brief round of refinement was done using the same options and parameters as before.

Results

MOLPROBITY's multi-criterion validation strategy has been adopted by the structural biology community with great enthusiasm; that is the best testament to its effectiveness. The public MOLPROBITY server hosts hundreds of different users every month and thousands of working sessions; several thousand different users have used it since its inception in 2001. In fact, more PDB files have been input to MOLPROBITY than are deposited in the PDB (42,300 vs. 37,300). About 75% of these were uploaded from the user's computer rather than pulled from the PDB, and most appear to be structures actively undergoing refinement. Additionally, several pharmaceutical companies (GlaxoSmithKline, Wyeth, Structural Genomix) and structural genomics centers (JCSG, NESG, CESG) operate their own MOLPROBITY servers for internal use.

Literature citations also reflect MOLPROBITY's popularity. Lovell et al. (2003) introduced MOLPROBITY (along with our new Ramachandran plot and the Cβ deviation), and according to the Web of Science was cited 14 times that year, 40 times in 2004, 87 times in 2005, and 42 times in 2006 (as of July 3rd). The paper by Davis et al. (2004) was dedicated solely to MOLPROBITY for RNA use, and was cited once in 2004, 11 times in 2005, and 7 times in 2006. Google Scholar finds 131 results for MOLPROBITY, 51 of them from 2005 and 22 from 2006. Finally, there are 69 structures in the Protein Data Bank that mention MOLPROBITY in their headers. An optional "REMARK 42" item has been tentatively defined for PDB file headers to report MOLPROBITY and perhaps other validation statistics; it is currently being evaluated by an international committee of the wwPDB. This further supports our belief that MOLPROBITY is being widely adopted.

As shown in Arendall et al. (2005), the new geometrical and all-atom evaluation tools in MOLPROBITY led to proposed changes in most of the working-set SECSG structures which not only improved the global quality scores (Table 11 and Figure 35) but which can be seen individually, once accomplished, to provide a clearly better local match to the electron density. Most of these refittings are major ones at the local scale, which add or delete

groups of atoms or change their positions by several Å, as in traditional model rebuilding. The difference is that there is now additional information both for locating the problem and for evaluating which potential solution is the best to try.

A variety of problems were discovered and fixed within the SECSG structures: sidechain amides or imidazoles flipped by 180° (Word, Lovell et al. 1999b); branched sidechains fit backwards into electron density, as in case studies of leucine (Lovell, Davis et al. 2003) and threonine (Richardson, Arendall et al. 2003); wrong backbone conformations, including sidechains fit into backbone density and vice versa; and mis-oriented or mis-identified ligands and solvent. All-atom contacts were useful for identifying a variety of problems; REDUCE uses them to automatically resolve the amide and imidazole flips. Cβ deviations and rotamer and Ramachandran outliers each found a smaller set of problems, but each identified some problems that were either not found by all-atom contacts, or were found but difficult to interpret.

Figure 35 shows the results of MolProbity on several global validation scores; in all cases, the control-set structures are about the same as the PDB average, while the working-set structures are significantly better, with scores along the bottom edge of the distribution. In the working set, there are at least

3 times fewer rotamer outliers, 10 times fewer Ramachandran outliers, and 5 times fewer serious clashes (see also Table 11). These structures also have 0% Asn/Gln/His flips (vs. ~20% for the controls), but as these can be fixed "for free" without damaging fit to data or geometry, this result is not surprising. On the other hand, it can be quite difficult to remove steric clashes, bad rotamers, and Ramachandran outliers while still maintaining or improving the covalent geometry and fit to the experimental data. When such a solution can be found, it is almost certain to represent the underlying molecular reality more accurately than the previous model. This assertion is supported by the last panel of Figure 35, which shows that the crystallographic cross-validation residual, R_{free}, was 1% lower on average for the working set than for the control set (23.4% vs. 24.4%). In other words, the models rebuilt using information from MOLPROBITY matched the experimental data better than their brethren that were rebuilt with conventional tools. Similar results were seen for 6 SECSG structures that were refined to completion with traditional methods and then subjected to MOLPROBITY analysis; their free-R values dropped by 0-4%, or about 1% on average.

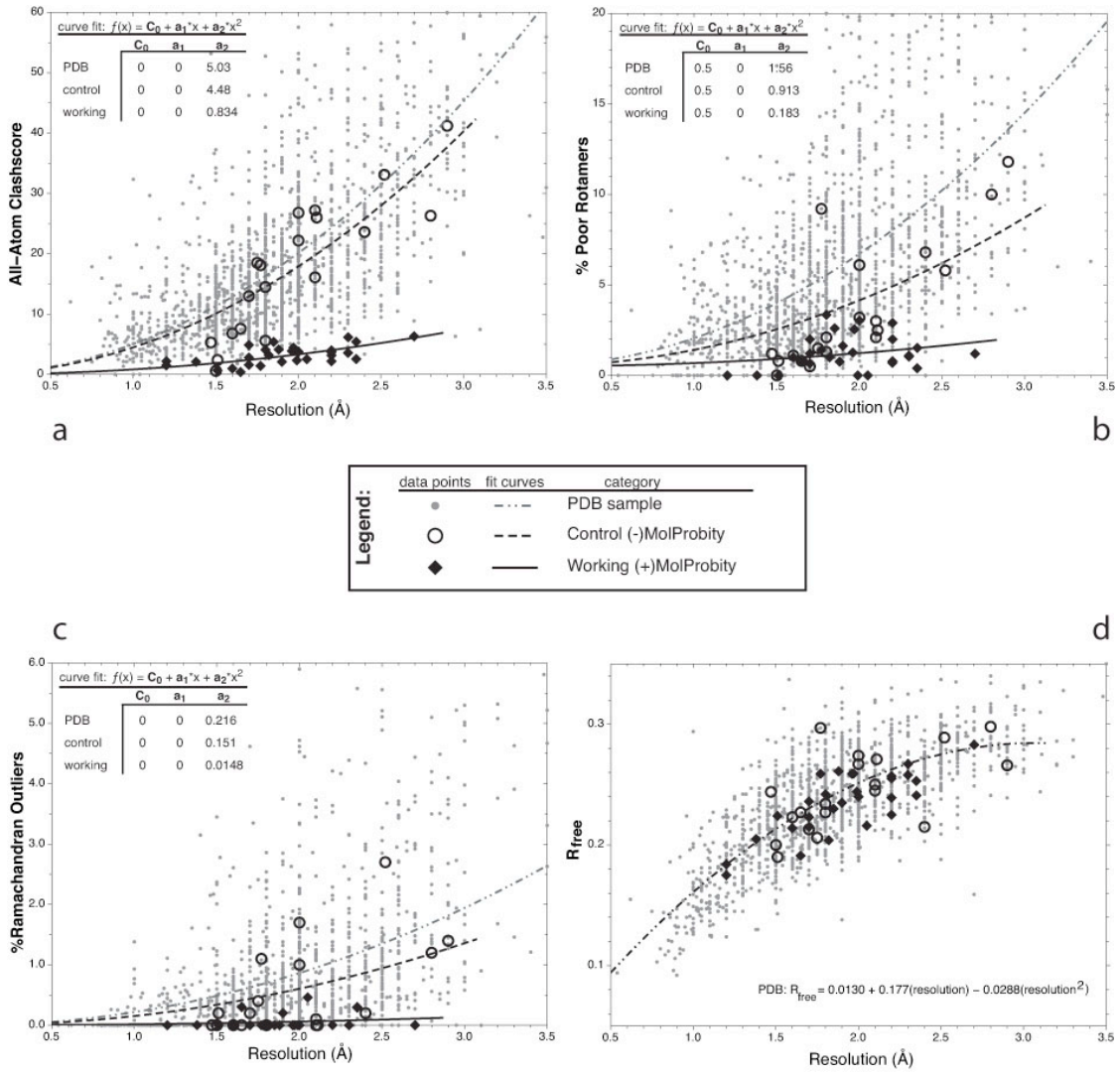
The structure with a 4% drop in R_{free} illustrates the usefulness of multi-criterion validation. Figure 36b shows a multi-kin of the N-terminal helix

along with difference density (F_o-F_c). The clustering of problems here -- clashes, rotamer outliers, difference peaks -- suggests a major misfitting that might be overlooked from the electron density alone (Figure 36a). In fact, the Asp 136 sidechain had been fit into density for its backbone and vice versa. Swapping the two and re-refining the structure revealed density for another turn of helix, resolved the various validation flags, and lowered R_{free} by [1%?] all by itself.

The MolProbity score (M_{ctm}) was developed after the SECSG experiments took place, so it was not used as a (pseudo-) target during the refinement and rebuilding. As expected, however, Table 11 and Figure 35 show that the MolProbity scores for the working set are much better than those for either the control set or the PDB sample, by 0.8Å and 1.2Å, respectively; again, the working set structures fall along the bottom edge of the distribution. One interpretation of this is that using MolProbity during rebuilding improved the effective resolution of structure determination by about an Ångstrom on average, though of course such a claim must still be regarded very skeptically at this point.

Figure 35: Plots of quality criteria vs resolution.

Shown with quadratic fit lines, for the three datasets of the PDB sample (gray dots), the SECSG control set (open circles), and the SECSG MolProbity working set (solid diamonds). a) All-atom clashscore (number of steric overlaps ≥ 0.4Å per 1000 atoms) (Word, Lovell et al. 1999a); b) percent poor sidechain rotamers (those with rotamer quality outside the 99th percentile contour defined by low *B*-factor, high-resolution data) (Lovell, Word et al. 2000); c) percent Ramachandran outliers (residues outside the 99.95% contour for general-case amino acids or the 99.8% contour for Gly, Pro, or pre-Pro) (Lovell, Davis et al. 2003); d) R_{free} value (Brunger 1992).


curve fit: f(x) = C0 + a1*x + a2*x2
C0 a1 a2
PDB 0 0 5.03
control 0 0 4.48
working 0 0 0.834
All-Atom Clashscore
Resolution (Å)
a
curve fit: f(x) = C0 + a1*x + a2*x2
C0 a1 a2
PDB 0.5 0 1.56
control 0.5 0 0.913
working 0.5 0 0.183
% Poor Rotamers
Resolution (Å)
b
Legend:
data points fit curves category
PDB sample
Control (-)MolProbity
Working (+)MolProbity
c
d
curve fit: f(x) = C0 + a1*x + a2*x2
C0 a1 a2
PDB 0 0 0.216
control 0 0 0.151
working 0 0 0.0148
%Ramachandran Outliers
Resolution (Å)
Rfree
PDB: Rfree = 0.0130 + 0.177(resolution) – 0.0288(resolution2)
Resolution (Å)

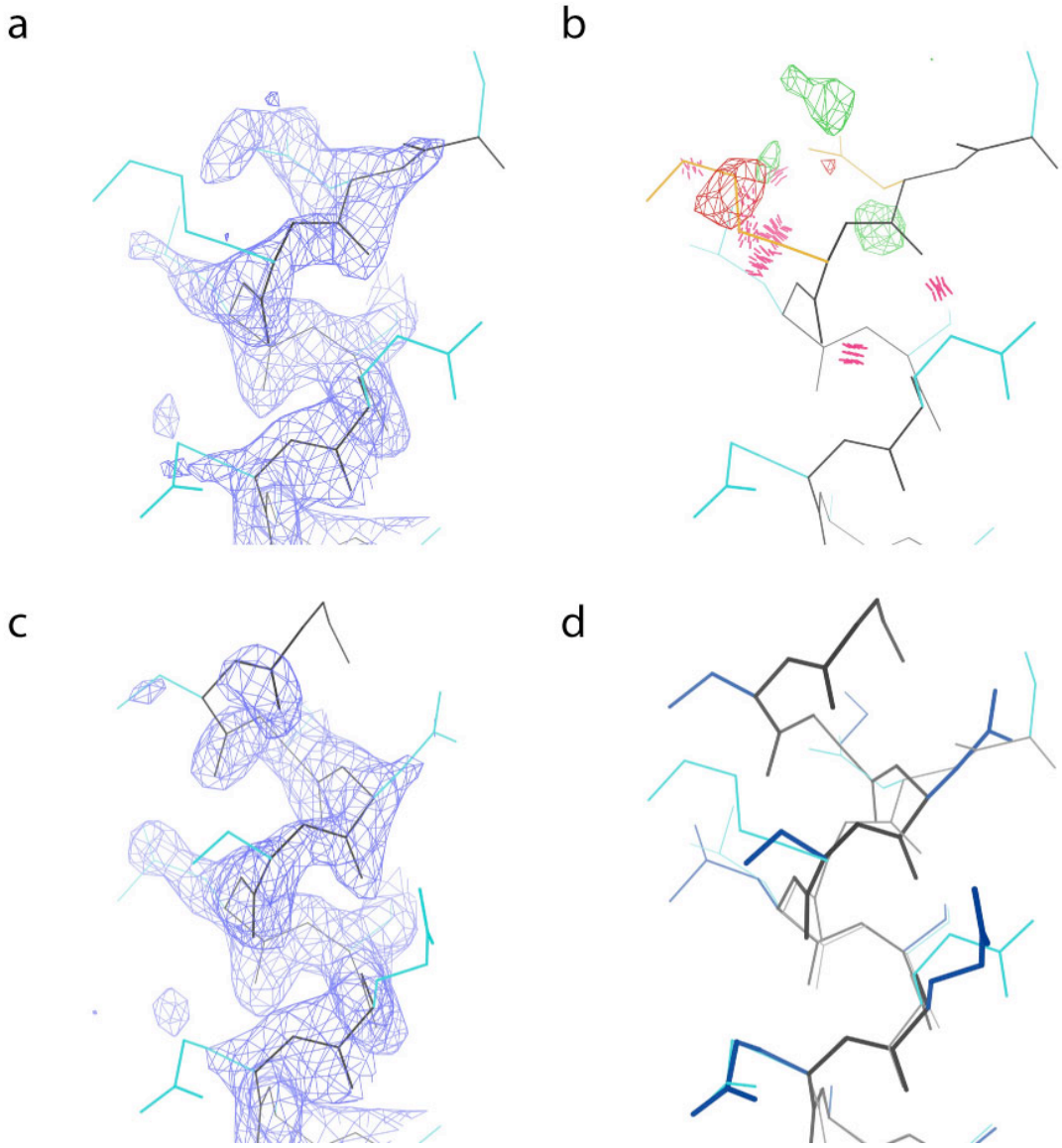


Figure 36: Rebuilding a structure using a MolProbity multi-kin.

This SECSG structure is deposited in the PDB as 1LPL before being repaired, and 1TOV afterwards. (a) The N-terminus of 1LPL shows reasonable fit to ambiguous density. (b) A multi-criterion kinemage from MolProbity highlights rotamer outliers (gold), steric clashes (pink), and difference density (red and green). (c) Swapping the sidechain and backbone of Asp 136 produces a better fit, and when re-refined, another turn of helix is revealed in the density (1TOV). (d) Overlay of 1TOV (heavy lines) and 1LPL (light lines).

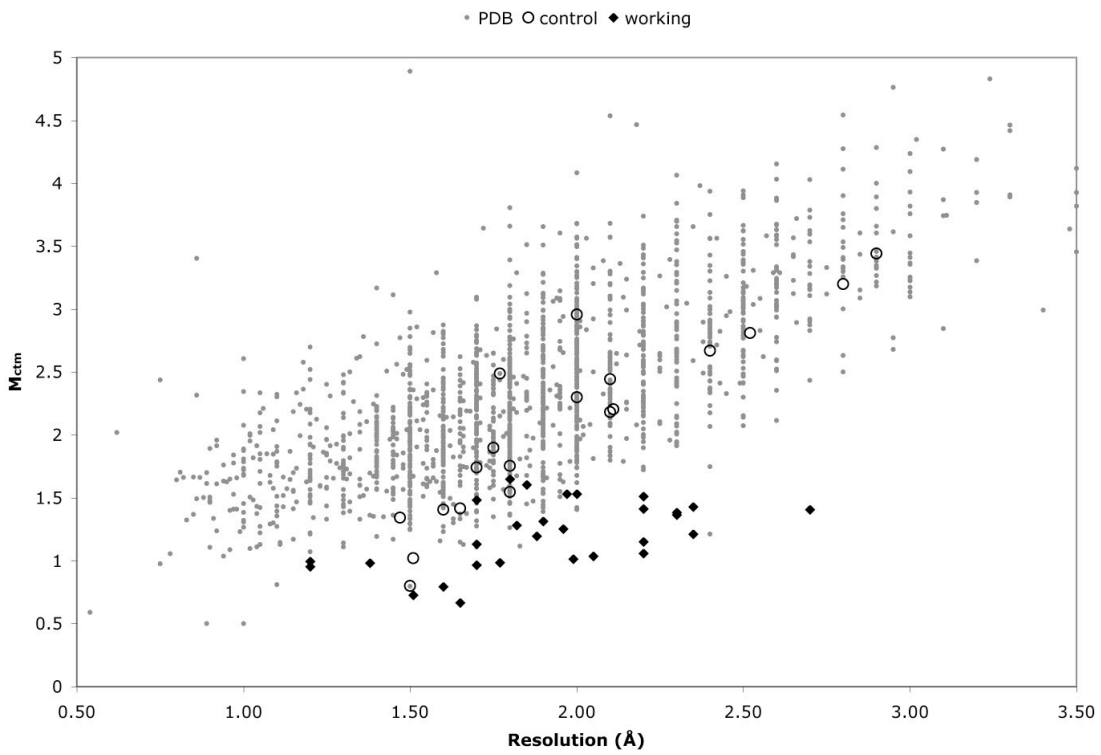


Figure 37: Plot of MolProbity score versus resolution.

Plotted are the working set (black diamonds, + MolProbity), control set (open circles, - MolProbity), and PDB sample set (gray dots).

Discussion

A major challenge in multi-criterion validation is one of data simplification with minimal loss of information: How does one present a report at once comprehensive and comprehensible? For multi-kins, we are highly dependent on color. Each type of outlier (clash, rotamer, Ramachandran, Cβ deviation, sugar pucker) is shown on a muted gray Cα trace of the protein in a bright, distinctive color (hot pink, gold, green, purple, and magenta, respectively). There is a balance to strike between making the colors different from one another and different from the colors used in rendering the structure itself (gray, white, cyan, etc). Currently, clashes, Cβ deviations, sugar pucker outliers, and non-water ligands are all similar colors; shape helps distinguish them somewhat, but there is still potential for confusion.

The kinemage color palette itself has undergone significant optimization, but in choosing color combinations and color scales for the multi-kin we have been guided by several useful resources from the visualization community (Tufte 1990; Tufte 1997; Ware 2000; Tufte 2001). In particular, ColorBrewer is useful for selecting *N* colors that are maximally distinct from one another, while also being color-blind friendly, reproducing well on a projector, etc. (Harrower and Brewer 2003). For encoding *B*-factors, we choose a black body

radiation scale, which avoids perceptual problems created by the ubiquitous "rainbow" scale (Bergman, Rogowitz et al. 1995; Ware 2000).

We would like to extend the multi-kins to show additional validation information, such as bond length and angle ideality, but this will make it even more difficult to distinguish the different kinds of information. One possibility would be to use a scheme like that for *B*-factors and occupancy, where a stick model of the structure itself is color coded. However, it is difficult to use those visualizations at the same time as the outliers, because they use some of the same colors; having to look at the analyses separately defeats the point of the multi-kin.

The multi-chart faces similar limitations as to how much information it can profitably display at once. Here, the major limitation is page width: each validation criterion gets its own column, but the goal is to visualize them all at once. Hence, the number of columns is limited by the page width. Font sizes can be reduced only so far while maintaining readability. One possibility is to change the table orientation, placing criteria in rows and residues in columns. Table cells are wider than they are high, so this allows for many more criteria, but the number of residues that can be seen at any one time is correspondingly greatly reduced. On the web, hypertext could be used to show a reduced

representation with details on demand, but (1) many crystallographers prefer to work from a paper copy of the chart, and (2) this again inhibits direct comparison of neighboring residues.

Anecdotally, we find most (though not all) crystallographers prefer the MOLPROBITY multi-chart to the multi-kin, using it as a checklist for their rebuilding efforts in O (Jones, Zou et al. 1991), XFIT (McRee 1999), COOT (Emsley and Cowtan 2004), etc. This may be because the kinemage graphics environments have limited rebuilding capabilities, or are simply less familiar than the standard tools. However, the graphical representations are often more information-rich than their textual counterparts (e.g., clashes have a physical location between two atoms, and other nearby atoms as context). We believe that an important future challenge will be to link the multi-chart to the graphical representation, so that a user can jump back and forth, getting the best of both worlds. It will be easiest to do this within our own software (i.e. with the multi-kin in KING or MAGE), but to serve our clients it would be very desirable to also integrate with crystallographic rebuilding programs. We already have a limited form of the multi-chart (just a list of outliers) integrated with COOT. Paul Emsley, the author of COOT, has also integrated all-atom contacts dots from Probe into COOT, so it may be possible to add graphical

markers for rotamer and Ramachandran outliers, etc. as well. We are also planning to tightly integrate MOLPROBITY with the PHENIX system for automated structure solution (Adams, Grosse-Kunstleve et al. 2002), which may require still other formats for the multi-criterion validation reports.

Another future challenge for multi-criterion validation will be NMR models. 10 or 20 models means 10 or 20 times as much information, plus NMR structures are prone to having large numbers of outliers. Jeremy Block in our lab is currently developing a 3D "multi-graph" that uses one dimension for model number and the other two for various multi-chart-like layouts. The multi-kin currently uses the same representation as for crystal structures, but with the added ability to animate through the models. As we learn how to best recognize, correct, and re-refine errors in NMR structures, we will hopefully discover how to best organize a multi-model multi-kin.

Turning now to global validation, the idea of expressing validation scores as "effective resolutions" will probably irritate a good number of crystallographers. (We chose the name "MolProbity score" instead for just this reason.) After all, almost any set of validation scores captures only a limited part of what constitutes structural accuracy, and any particular structure may have mitigating circumstances for its low score on a given test.

Despite these and other legitimate protests, we believe it to be an intuitive and useful way to summarize validation scores, and one that should be applicable to almost any structure-based validation scheme. If the functional forms are kept simple, as we have done here, then not only can effective resolution scores be compared to the actual crystallographic resolution, but scores can be compared to one another: a clashscore of 2Å with rotamers and Ramachandran at 1Å suggests rebuilding efforts should focus on steric contacts. If other validation programs also adopted an effective-resolution scale for their metrics, it would facilitate comparisons of methodology, such as MOLPROBITY's Ramachandran evaluation vs. PROCHECK's (Laskowski, Macarthur et al. 1993) or WHATIF's (Vriend 1990).

Structural quality and even structural accuracy are nebulous concepts, whose meanings likely depend on the methods and data used to determine the model as well as our intended use of it. For crystal structures, resolution is obviously a major determinant of quality, but is not synonymous with it. In fact, the relationship is probably not linear, to the extent that quality is quantifiable. Furthermore, the stated resolution is in part a decision made by the crystallographer; while there are rules of thumb, there is also some room for choice. Nonetheless, most experienced structural biologists have an

intuitive understanding of the *average* quality of a structure at a variety of resolutions; thus, effective resolution may be one of the most meaningful numbers we could use to express structural quality.

The validation criteria in MOLPROBITY capture one kind of structural quality, and comparing the combined MOLPROBITY score to resolution gives an R^2 of 0.5; this suggests that half the variation in MOLPROBITY score can be attributed to resolution. This leaves us with a typical nature / nurture result when considering x-ray data: about half of the final product seems to be due to the quality of the data collected, and half to the methods used and the care (or lack thereof!) exercised by the crystallographer.

As mentioned in the introduction, we semi-arbitrarily set the constant term in the MOLPROBITY score to 0.5. Because the coefficients of the other terms were not changed, this had no impact on the correlation with resolution, but it did have two benefits. First, it means the minimum possible score is a reasonable resolution value; the various attempted regression fits typically gave constants near zero or even negative, which were not meaningful to think of as crystallographic resolutions. Second, it means that most crystal structures will have a MOLPROBITY score somewhat larger than their resolution; this means we avoid endorsing as "good (enough)" structures

which are only average for their resolution. The definition of "most" varies a bit with resolution: 95% of high resolution structures in the database had a MOLPROBITY score higher than their resolution, but only 75% of the low resolution structures did. This is a consequence of the distribution of residuals (Figure 34), which is narrower and turns up slightly at the high-resolution end; this shows that the true relationship between resolution and the validation metrics is only approximately log-linear (although no other well known function gave a better fit).

An alternative to adjusting the constant term would be to fit to a high-quality subset of the data; for examples, all structures that have *c*, *t*, and *m* in the best quartile for their resolution bin. Since the average quality scores approach zero for high-quality, high-resolution structures, this might have naturally led to a reasonable constant term in the fit equation. However, this data filtering scheme greatly reduced the size of the working data set and gave an effective resolution function with poor correlation in the overall data set. Thus, we chose to fit the whole data set and subsequently adjust the constant term, as described above.

As we present it here, the simple MOLPROBITY score (M_{ctm}) is purely geometric and can be calculated for any model, regardless of source. This is a

strength in that it allows structure comparisons across disciplines, but a weakness in that it does not account for fit-to-data in experimental methods. Thus, in the future we may develop MOLPROBITY scores in the same vein that are specialized for x-ray structures, or NMR structures, etc. For x-ray, such a score might incorporate a term based on R_{work}, R_{free}, and/or the difference between them. For NMR, the score might consider number of restraints per residue, or what types of data were collected.

Finally, when is it reasonable to use global quality scores like M_{ctm}, and when should one use finer-grained evaluations? It depends on the type of information desired from a structure, the amount of structural expertise available, and the amount of time that can be spent evaluating a structure. For instance, a global score is valuable when large parts of the structure are important to answering the question, when many different properties may be examined, or when the question itself is loosely defined. If only a limited region (e.g. an active site) or narrow focus (e.g. hydrogen bonds) is relevant, more targeted evaluations are possible and are likely to give more reliable results. Additionally, global scores are useful to investigators with limited structural expertise who want to look at the "best" structure of X, especially if there are several structures of X at similar resolutions; more detailed

evaluations require considerable expertise to be used effectively. Finally, global scores are good for large-scale projects that look at hundreds or thousands of structures, where there is not time for a human expert to evaluate each one in depth. Thus, the first and third criteria (and unfortunately, sometimes the second) suggest that global scores, such as MOLPROBITY'S M_{ctm}, would often be useful in contexts such as bioinformatics studies; on the other hand, structural labs doing detailed work on an enzyme mechanism should primarily focus on detailed local evaluations.

Chapter 6: Conclusion and future directions

Taken together, these studies show that careful scrutiny of the local conformations of protein backbone reveals many stringent constraints and at least one common pattern of change, and suggest that both can be profitably leveraged by a variety of computational methods that tackle difficult problems rooted in protein structure. However, there are still many interesting questions opened up by this work that remain unanswered. In particular, there is an issue of techniques for studying backbone motion: other biophysical techniques could be brought to bear on backrub motions, while x-ray crystallographic data could be used to study other types of dynamics as well. Additionally, I have barely scratched the surface in applying the BACKRUB model to computational design, and have not even attempted applications to structure prediction, automated model (re)building, etc. BACKRUB's potential value in these cases is in being an accurate but greatly simplified parameterization of local backbone conformation. This raises the hope that future work may develop similar parameterizations for longer segments of backbone, especially for non-repetitive structure ("loops") joining well-anchored regions such as helices or strands. There are also remaining

challenges in model validation, such as integrating a large number of validation metrics and comparing multiple models sensibly. Finally, structure determination is becoming increasingly automated, and there is need for better automation both in validation itself and in (re)fitting informed by validation criteria. I will now consider some of these topics in more detail.

The value of crystallographic data for studying dynamics is often under-appreciated, but our study of backrub motion (Chapter 3) would have been impossible without it. There are many potential sources of data in crystallographic models for deducing the actual coordinate changes of motions. In order of increasing difficulty and increasing size-of-motion, they are crystallographic alternate conformations; non-crystallographic symmetry (NCS), TLS refinements, and multiple models; independent data collections; point mutations; and evolutionary homologs. Alternate conformations are best for studying small-scale motions, because there is no coordinate error from superimposing models. On the other hand, alternates are generally confined to one or a few residues, limiting them to highly local motions like backrubs. Unrestrained NCS models show variation at slightly larger scales, such as moving loops or breathing motions. Chapter 3 discussed a cursory survey of such motions, but I am not aware of a more detailed, systematic

study. The advantage of NCS over most pairs of independently determined structures is that all differences are known to arise from physical effects (crystal packing), rather than different data-processing or refinement protocols. There is a trend toward multiple models for analyzing motion in crystallography (DePristo, de Bakker et al. 2003; Terwilliger 2006); they could show both *N*-state and fluctuation-type motions, but there is not enough experimental data to trust the details of the modeled conformations, and the modeling methods are not yet mature (Terwilliger 2006). Such models represent both actual motion and positional uncertainty, as is also true for *B*-factors and NMR ensembles. Finally, point mutants and homologs are likely to show the largest conformational changes, but as the structures diverge it becomes increasingly difficult to deconvolute the many overlapping motions and extract any general principles.

Using traditional x-ray structures to study motion has advantages and disadvantages compared to other techniques. It is largely limited to two- or three-state motions. It cannot resolve the pathway(s) between states, as Laue diffraction may be able to. However, Laue experiments have proven much more difficult than standard crystallography, so there is not yet much data available. Also, their focus is change and motion, so the data have already

been fully exploited in that regard. Traditional x-ray structures can report accurately on motions only when end states are nearly iso-energetic; larger energy gaps quickly lead to more lopsided occupancies, making the minor state(s) unobservable. Finally, there is not information about rate of transition between states, as there is in NMR. Even if there were, rates in a frozen crystal are likely irrelevant to the cellular environment. On the other hand, crystallography has the critical advantage of locating atoms in space more precisely than any other method, which allows detailed characterization of the end states of the motion. This quality was essential for our backrub study.

As shown in Chapter 3, there is evidence for backrub motions from a variety of sources, including NMR. There are a number of intriguing NMR studies that suggest backrub-type motion of peptides in β sheets (Bouvignies, Bernado et al. 2005; Lakomek, Fares et al. 2005). These conclusions are generally based on various backbone order parameters, which (despite the name "model free") are usually interpreted based on a physically unrealistic model of diffusion within a cone. A more realistic model is provided by BACKRUB, and it would be interesting to interpret backbone order parameters based on this model. Furthermore, NMR might help determine how specific backrub motions are: that is, is the appearance of sizable backrub motions at a limited number of

positions an artifact of crystallography, or does it also hold in solution? Finally, NMR may be able to assign an approximate timescale to the backrub conformational transitions. We have assumed that it is comparable to the timescale of sidechain rotamer exchange, since that seems to be its driving force, but we have no direct evidence.

Another way to probe backrub motions is by attempting to design them, as was done in Chapter 4. There are three obvious extensions to this work. First, it might be necessary to integrate BACKRUB backbone motions into the combinatorial dead-end-elimination rotamer selection process, so that DEZYMER could access the full range of possibilities afforded by backbone motion, rather than a handful of hand-built scaffolds. Furthermore, such backbone moves could carry small penalties, to discourage random modification of the experimentally determined scaffold. Second, a static-backbone design trial should be carried out in parallel as a control. The control calculation for my work was several years old, so the major variable in results could have easily been the changes in DEZYMER that have occurred over that time. Finally, many more wet-lab trials are necessary to establish the statistical superiority / inferiority of design with BACKRUB. This may now be

possible, as the Hellinga laboratory develops the infrastructure for high-throughput design trials.

Alternately, one might try for a purely structural design, to deliberately induce (or remove) a backrub-type conformational change in a specific residue. The barrier to this approach is primarily the lack of an assay for backrub motion. We can reliably detect this motion only in the alternate conformations of very high resolution crystallographic structures. It might be possible to try such designs in some of the proteins surveyed in our study, but any mutations might either degrade the data quality to a poorer resolution or trigger a change in crystal form. Alternately, one might be able to obtain sufficiently accurate before-and-after sidechain residual dipolar couplings (RDCs) by NMR so as to detect the re-orientation of a single Cα--Cβ with respect to the protein frame, as well as detecting the coupled sidechain changes.

One obvious limitation of the backrub studies is that they only deal with small portions of the polypeptide chain; they are unable to describe movement of loops, hinges between domains, and the like. I am hopeful that future studies will develop simplified models in the spirit of our BACKRUB algorithm for longer stretches of backbone. As discussed in Chapter 3,

BACKRUB itself can be readily extended but is not useful beyond 3-4 residues. Thus, I believe we will need a conceptually different simplified parameterization for dealing with longer stretches of backbone such as loops.

MOLPROBITY checks a small (albeit important) subset of structural properties. It exists, of course, because other validation programs check different subsets, and so leave out some important criteria (as does MOLPROBITY). There will probably never be a Grand Unified Validation Program that diagnoses everything, and in any case, our ever-growing knowledge of What Proteins Do would quickly obsolete it. Even MOLPROBITY isn't a monolithic whole, but is instead a wrapper around many other programs that perform smaller parts of the overall function. Thus, there is a need within the structural biology community to create a meta-MOLPROBITY, a tool that incorporates the wisdom of many different validation packages, via aggregation rather than reimplementation. There are a great number of technical, social, and legal challenges to be overcome in doing so. Technically, it is unclear whether all the programs need to be installed together on one server, or whether a "web services" (e.g. SOAP) approach might allow them to operate semi-autonomously from different sites around the world. Personally, I believe the value of web services is unproven, and I suspect that all

programs will have to be local in order to achieve the necessary deep integration. The key to doing this successfully, after all, is really in distilling the results down to something understandable; merely concatenating the output of a dozen PROCHECKs on one web page makes only an incomprehensible mess. The work on multi-charts and multi-kins (Chapter 5) illustrates one style of condensing validation information, but to include significantly more information will require additional innovation.

Another future challenge for MOLPROBITY will be to better analyze groups of closely related models. Multiple-model PDB files are generally associated with NMR structures, but as discussed above they can also be generated from x-ray data. Just ten models means a multi-chart or multi-kin with ten times the validation data. Any validation display for multiple models must balance two competing user needs: to distinguish models (clashes, etc. from different models often overlap spatially in a multi-kin) and to compare models (sequential multi-charts make it hard to compare residue X in all models, and there's not enough room to list 20 models side by side).

Finally, I want to consider automation, in MOLPROBITY in particular and in structural biology in general. Much of the drive for automation has come from worldwide "structural genomics" initiatives that aim to solve large

numbers of structures in a high-throughput format. In addition to systems for expression, purification, and crystal/spectrum screening, the structural genomics centers (often collaborating with synchrotrons) have developed mostly automated systems for data collection and processing. Now we are involved in a concerted effort to develop a highly automated structure solution and refinement package, called PHENIX (Adams, Grosse-Kunstleve et al. 2002).

Our basic role in PHENIX will be to provide structure validation tools, which will be closely integrated with the refinement pipeline to maximize their effectiveness; the MOLPROBITY tools add new information not included in the target functions for crystallographic refinement. However, all decision steps in an automated system need comparable validation and (semi-)automated oversight. There is a stark contrast between biological systems and systems of our own design: we often marvel at the amount of energy and complexity that Nature expends on monitoring, repair, regulation, and redundancy; while our computer systems often have multiple points at which a single failure leads to catastrophe. The general problem is the subject of long term research in computer science, but in the meantime we can ensure that adequate validation is a part of all automated procedures, and that human

beings can quickly and effectively manually review automated processes and easily intervene where necessary.

Structural genomics centers are not the only clients for automated systems. As x-ray crystallography and NMR become more mature, focus is largely shifting from the techniques themselves to the biology they enable; as a result, many investigators with limited experience in structural biology are using them as yet another complementary technique. At the same time, the public database of structures is becoming an increasingly important computational resource for predicting unknown protein structures, unknown protein functions, and protein-protein and protein-drug interactions. As a consequence, the methods development community should be doubly motivated to distill the knowledge of expert human crystallographers and spectroscopists into expert systems that will both (1) maximize the knowledge obtained per experiment for inexperienced investigators and (2) promote the highest attainable accuracy for our shared public repositories of structures.

Appendix A: Design of MolProbity and KiNG

MolProbity

This document describes MolProbity 3.03 as of August 16, 2006. This is the version of MolProbity that appears in the digital appendix or very similar to it.

MolProbity began life as my rotation project in the Richardson lab, in the fall of 2001. At that time, it was little more than a file upload form and PHP scripts to call Reduce and Probe, producing the Richardson lab's all-atom contact analysis. KiNG had not been written yet, but small kinemages could be displayed online in JavaMage and large ones could be downloaded. Version 2 of MolProbity was created shortly thereafter as I learned more about how to architect PHP code: It powered a widely-used public server for several years, and incorporated the Ramachandran and rotamer analyses described in Chapter 2, as well as multi-criterion displays online in KiNG. The 2.x series suffered several shortcomings: the server was fragile and extremely difficult to transport or install elsewhere, only one PDB file could be worked on per session, sophisticated navigation and decision making was almost impossible, and code was not structured for reuse or to support multiple interfaces. The current version, the 3.x series, was begun in late 2003 to address these

concerns, and was the subject of a successful grant proposal to develop it further. This version totally replaced the 2.x server as the only publicly-available MolProbity sometime in 2006, and has performed very well thus far. I believe its architecture is now robust, extensible, and will support many years of future development.

PHP

MolProbity is implemented in PHP, a derivative of Perl that was developed specifically for constructing dynamic web page content on a server (as opposed to Javascript, which is often used for dynamic interaction once the web page is already in the client's browser). In retrospect, it was a good decision: PHP has continued to grow in popularity, to the point where it may now be the most used language for the web. With that comes relatively robust and well-supported code, lots of third-party libraries, and active ongoing development.

PHP is a nicely designed language in general, with a few notable warts. I'll discuss a few of each that are particularly relevant to working with the MolProbity codebase.

Like Perl, PHP is very strong on text wrangling; this is important for getting data in and out of the various command-line programs that MolProbity uses

(Reduce, Probe, etc.). It readily interconverts numbers and text strings, which is useful with said programs and also with web forms, where all data (including numeric data) is nominally transported as text. PHP's designers did the right thing by not overloading the + operator to mean string concatenation (as was done in Java, and worse, in Javascript). Thus, + always results in numeric addition, regardless of whether the quantities are numbers or strings. The idiom `$x = $s+0` is often used in MolProbity to convert a string (`$s`) to a number (`$x`). Unfortunately, string concatenation is instead accomplished with the dot operator, which has consequences for object-oriented programming (see below).

PHP has only one native data structure, the "array". However, it's much more than a C or Java array, and is so versatile I implemented something similar in Java (as driftwood.data.UberMap). PHP arrays can dynamically grow or shrink to any size. Like other arrays, they map unique indices (or keys) to potentially redundant values. Unlike most arrays, the indices can be numbers or strings (though not arbitrary objects), which allows them to serve most of the functions of hash tables. However, the indices/keys remain in the order they were inserted, so iteration order is predictable, unlike most hash tables. In Java terms, PHP arrays act as Maps. Their keys and values can be

separated and treated independently, so the value side acts like a Java List, and the keys side acts like a Java Set. This last point is somewhat non-obvious, but MolProbity frequently creates arrays where the keys are used to obtain a non-redundant set of strings, and the values are dummies.

```
$numbers = array(1, 2, 3, 1, 2, 4, 6, 3, 3, 2);
for($numbers as $n) $unique_num[$n] = "dummy";
$unique_num = array_keys($unique_num);
// now $unique_num = array(1, 2, 3, 4, 6);
```

Functions in PHP are not quite first-class citizens the way they are in e.g. Javascript, but they're much better off than in, say, Java. For instance, you can store the name of a function in a variable, and then use that variable to determine which function is called at runtime. This is used primarily in MolProbity's event-driven architecture for user interaction.

Object-oriented programming was tacked on to PHP as an afterthought, and it shows. Prior to PHP5 (which is still not widely adopted), class instances are really just fancy arrays that hold values and functions. In particular, serializing them to disk is dicey, so you're better off storing data you want to save in plain old arrays, which MolProbity does. Since the dot operator is used for string concatenation, one must use the awkward arrow operator (`->`) for member access. Scoping rules are also poor, so even within the class all

members must be accessed as **`$this->foo`**. I frequently cause bugs in MolProbity's display pages by defining functions and trying to call them without the "**`$this->`**" prefix.

On the other hand, classes are the only mechanism PHP has for defining namespaces. One must be careful when defining functions in included files, because it's an error to redefine any function. As a result, MolProbity's page display code is defined such that each page is a separate class. This avoids problems that otherwise arise when including two files that both define a **`display()`** function.

One last thing to be careful of is that arrays are always passed by value in PHP, and classes are passed by value in PHP4 but by reference in PHP5. This means that when an array is passed to a function, unless the programmer took special effort to ensure otherwise, any modifications will not affect the original. More surprising, the code below will not change the value in **`$a`** (as it would in, say, Java):

```
$a = array('foo' => 'bar');
$b = $a;
$b['foo'] = 'doh';
// now $a = array('foo' => 'bar')
// and $b = array('foo' => 'doh')
```

This is a big danger in MolProbity because most important user data is stored several layers deep in **`$_SESSION`**. It's important to read about and understand the reference operator (**`&`**) before doing much work in MolProbity.

Javascript

Despite its name, Javascript has very little to do with Java; most of its principles are quite different. Despite its reputation, Javascript is an interesting, powerful, and well-designed language, with one of the most flexible approaches to object-orientedness I've every encountered. However, web browser support for Javascript is inconsistent, incomplete, and plain ol' buggy. Futhermore, some people deliberately disable it, as it continues to be a security risk. For those reasons, it is important that MolProbity never *rely* on Javascript for its functionality. There are several places that I use Javascript to enhance the user experience, but I am committed to retaining full functionality even when it is disabled. For instance, Javascript auto-submits the model selector on the main page, suggests appropriate validation options for different structures (e.g. with and without H), and allows the multi-chart to be sorted offline.

If I decided to add more extensive Javascript to MolProbity, I would probably incorporate one or more of the major open-source libraries, such as Prototype / Scriptaculous or Dojo.

Debugging

Debugging web applications is generally more difficult than debugging desktop applications, and languages like PHP and Javascript are subject to certain types of bugs that don't appear in languages like C++ and Java (e.g. type mismatches, misspelled variable names). PHP error reporting can be tuned to different levels, which is sometimes helpful. PHP scripts that display directly to a web browser will generally write out any error messages that come up, but PHP scripts running as background jobs will often die silently, with nowhere for the errors to go. MolProbity's background job launcher does try hard to capture these messages in `system/errors` in the data directory for the current session -- this usually works, but some messages may slip through the cracks.

Because of MolProbity's unique event-driven navigation system (see below), it is possible to screw up a session in such a way that you have to kill the session and start over. For instance, this happens if you call **`pageGoto()`** with a destination that doesn't exist. You can also get in trouble if you fire off a

background job that dies instantly (e.g. the background PHP script has a syntax error) -- the job progress page will cycle forever, with no way to abort.

Debugging Javascript is even worse -- almost impossible. I'm impressed with the language but the tools to support it are horrible. Firefox has the most and best tools, including a full-fledged debugger ("Venkman") that can be installed as an extension. However, errors are still often silent, or the error message is uninformative and even misleading as to where the source of the problem is. There is no equivalent of stderr for unobtrusive debugging messages, so you're stuck with **`window.alert()`** or appending text to an HTML **`textarea`**.

Platforms

MolProbity is designed to run under Linux and Mac OS X, but not (for instance) Windows or Solaris. This is because MolProbity needs both a Unix-like environment, and a modern Sun Java virtual machine. Sun Java is not available on many minor platforms, although it is available for Windows. However, Windows doesn't have a natively Unix-like environment. It might be possible in the future to run MolProbity from within a Windows Linux "emulator" like CygWin, but it is unclear whether Java could also be run from within that environment.

Three tier architecture

A three-tier architecture for the web means that presentation, "business" logic, and data storage are well separated from each other. MolProbity approximately follows this model -- presentation is more cleanly separate than the other two, but each facet can still be considered independently.

MolProbity stores most of its data in regular files in the filesystem -- PDB files, kinemage files, etc. Some data is stored hierarchically in a single array (**`$_SESSION`**), which is also serialized to the disk. Because the entries in **`$_SESSION`** are defined ad hoc throughout the code, it's very important to document their structure in one place for later reference: that document is `doc/extending/variables.html`. Industry best practice dictates using a relational database for data storage, but this causes three problems for MolProbity. One, it would take a significant amount of work to design a database schema for the information stored in **`$_SESSION`**, and a significant amount of code to load and save it. Saving additional information becomes a lot more work, which is bad for a rapidly evolving project. This is partially offset by the mandatory documentation that such an approach creates. However, the second problem is that most of our data is in files, and the utility programs MolProbity relies on (Reduce, Probe, etc.) have to act on files. If

they were stored in a database, we would waste a lot of time pulling out the data, writing a temporary file, running a program on it, and slurping the results back into the database. Finally, it's no small effort to install, maintain, and secure a relational database like MySQL. We want it to be as easy as possible to distribute MolProbity and set it up on other machines, and requiring a relational database significantly ups the difficulty.

The business logic of MolProbity resides in a number of PHP files in `lib`. These functions run Richardson lab programs in standard ways, and parse their output to make it usable from PHP. They also perform various editing operations on PDB files and synthesize information from multiple sources. Wherever possible, these functions can be called without establishing a MolProbity session: they take an input file name and an output file name, and any other information they need to do their job. They do not refer to anything in **`$_SESSION`**, or expect any particular organization of data in the filesystem. This is really important for enabling multiple interfaces (e.g. web browser vs. command line), as discussed below.

MolProbity's presentation layer (i.e., the part that provides a web browser interface) is the most elaborate and complex of the three. There are two documents in `doc/extending` that specifically deal with it,

`UI_Framework.html` and `Adding_Pages.html`. They lay out in detail both the motivation behind the design and its technical details, both of which will be summarized briefly here. The system is designed to mimic the stateful, event-driven model of traditional GUI applications over the stateless HTTP protocol of the web. Because it is stateful, it must deliberately break the web browser's Back button, which assumes a stateless nature to the web: past actions have permanently altered the files stored by MolProbity, and so past displays are no longer accurate representations of the application state. Allowing the Back button would mean allowing arbitrary "undo"; this could possibly be permitted if MolProbity were implemented on top of a system like Subversion, but it would take a lot of work to make it robust.

Each web page the user interacts with is defined in a single file in `pages`. The file defines a uniquely named class, which has a **`display()`** method for rendering HTML and zero or more **`onXXX()`** functions, which respond to form submissions, clicks on hyperlinks, etc. Special functions allow one page to "call" another (analogous to popping up a modal dialog box) and return back again, as well as simply transfer control to a different page. The files in pages are accessed indirectly, via `public_html/index.php`, which coordinates the flow of control among them.

Figure 38 presents a visual description of the architecture.

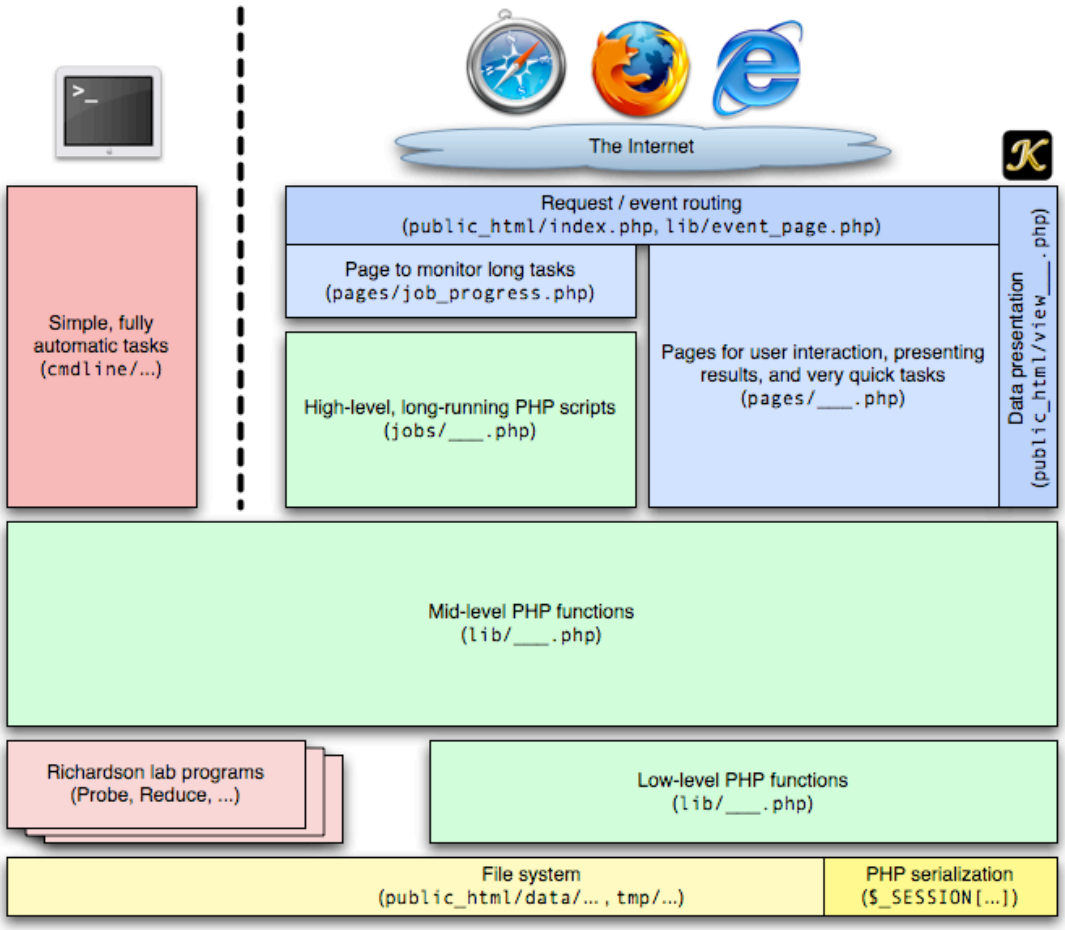


Figure 38: Overall architecture of MolProbity.

Upper layers are built on lower layers. Data storage is in yellow, business logic is in green, web UI is in blue, and command line UI is in red. The Richardson lab programs could also be considered part of the business logic, as indicated by their location.

Multiple interfaces

MolProbity has passed through three versions: a short-lived prototype constructed during my rotation in the Richardson lab (version 1.x), a long-lived public server focused around a main page (2.x), and the current public version described here (3.x). One of the major goals for the 2.x-to-3.x rewrite was to separate the functionality from the web interface, thereby allowing us to provide multiple interfaces. The interactive web / HTML interface is still the way that most people use MolProbity, but we now have a significant stable of scriptable, command-line tools as well. Because so much of the functionality is packaged into reusable, modular functions, the command line scripts are short and easy to write. Modularizing functionality requires a little extra effort during development, but maintaining that modularity is critical to the future development and maintainability of MolProbity.

I can imagine at least two additional interfaces for MolProbity. One is a web services interface, where we would basically expose MolProbity functions on the server in such a way that they can be called remotely, over the network. There are a variety of heavyweight approaches like SOAP that are built around XML, but I would instead favor a RESTful approach that basically uses HTTP GET and POST commands as function calls directly, using the

preexisting mechanisms for encoding data and passing variables. This is much lighter weight and more natural to implement in PHP, and the real-world experience of companies like Amazon and eBay has been that many more people use the REST interface than the SOAP one when both are made available. One important technical obstacle is that many MolProbity commands take a long time to complete, long enough that an HTTP request might time out while waiting for a response. Thus, we would have to implement a system analogous to the HTML job monitor, where clients are issued a "job ticket" that they can use to periodically check the status of a long-running job (adding hydrogens, making complex kinemages, etc.). I'm not convinced that there is (yet) enough demand for MolProbity web services to make this worthwhile, but it might be important in the long term.

More immediately, I would like to create a "MolProbity on a CD" package, which could be inserted on any Mac or Linux computer and launched in graphical mode (i.e., interaction via a web browser) without having to install or configure any software. Since CDs are read-only media, we would have to use some part of the host filesystem (perhaps `/tmp`) for storing PDB files, kinemages, etc. during the session. We would also have to provide a web

server and PHP, but there are lightweight web servers written in PHP, such as Nanoweb (http://nanoweb.si.kz/).

Directory navigation

One of the challenges with any web application is managing the hierarchy of directories and URLs where different components reside. For instance, publicly visible pages are in a separate directory from the libraries; for security purposes, a properly configured web server will allow outside access to that directory only, to protect most of the code from malicious attack. Likewise, libraries may be organized into several directories to give logical structure to the code, and all of the required external programs (Probe, Reduce, etc.) are stored in `bin/`, with one copy for Linux and one copy for Mac OS X. To let these various resources find each other, every PHP script that serves as an entry point (i.e., where execution could begin) is required to define the root of the MolProbity distribution tree, as **`MP_BASE_DIR`**.

User data -- PDB files, kinemages, etc. -- is stored in subdirectories of `data/`, which is in `public_html/` so it will be accessible to the world. Each user gets their own directory, which is named with a long string of random numbers and letters. Since there are so many possible unique IDs, it would be nearly impossible for an attacker to guess one. However, we must be careful that no

one can ever produce a listing `public_html/data/`, because that would reveal all the IDs and allow someone to browse the contents of all active sessions. For instance, Apache must be configured not to display a directory listing for `data/`. This scheme is still not invincible, because traffic to and from MolProbity is not encrypted. A determined attacker could probably sniff the session ID out of the network traffic. However, anyone with data that valuable shouldn't be using a public server anyway!

Security

Speaking of security holes, it has come to my attention while writing this document that all the benefits of writing a web application (cross-platform, no installation, always up-to-date, etc.) are offset by the additional security concerns they generate. The array of potential holes is endless, and the appropriate level of paranoia depends on the value of the data involved. (Which, in the case of MolProbity, shouldn't be that valuable.) However, one also should be concerned about attacks that could compromise the security of the server. For instance, for a long time MolProbity would allow upload of files with a `.php` extension. Since these were stored in web-accessible user data directories, subsequently requesting them would cause them to be executed on the server as the Apache user. With the help of a freely available

PHP script, this is equivalent to command-line access to the server machine, which could then be hijacked for sending spam, hosting malware or phishing sites, etc.

Monitoring and management

MolProbity 3.x has more tools for monitoring and management than its predecessors, but there's still room for improvement. The tools are available from `public_html/admin` (which should always be password protected), and they include tools to check PHP configuration, MolProbity configuration, currently active sessions, and some usage statistics. We've begun to supplement the built-in usage stats with data from Google Analytics as well, which covers a different set of information.

The major remaining need is for a tool to monitor resource usage, particularly processor usage and times when the system is overloaded. (There are already some built-in limits on disk usage that are managed per-session.) There is a crude system documented in `cmdline/uptime-plot.php`, but it has to be stopped and started by hand and provides fairly limited information; for instance, you can see processor usage, but not which jobs were causing unusually high usage.

Models and ensembles

Another major goal for the MolProbity 3.x rewrite was to robustly support multiple PDB files in one session. (In MolProbity 2.x, uploading a new PDB displaced the previous one; old results were still accessible but no further action could be taken on them.) To that end, MolProbity maintains a collection of "models", and the user can easily switch from working on one to another, and back again.

In addition, any material change to a PDB file results in creating a new model entry. (This is by convention only; programmers should take care to uphold it.) For example, using Reduce to add hydrogens or flip Asn/Gln/His residues results in a new PDB and a new MolProbity model. This way, a user could potentially run Reduce with and without the "-build" option, resulting in two different models which could then be compared.

A related problem is dealing with PDB files that contain multiple models. This is just rare enough and hard enough that most programs don't do it well or don't do it at all. After long consideration, we decided that to MolProbity "model" would mean one PDB file containing exactly one model, with no MODEL / ENDMDL cards. When multi-model files are input, MolProbity splits them into individual PDBs and creates one entry for each in

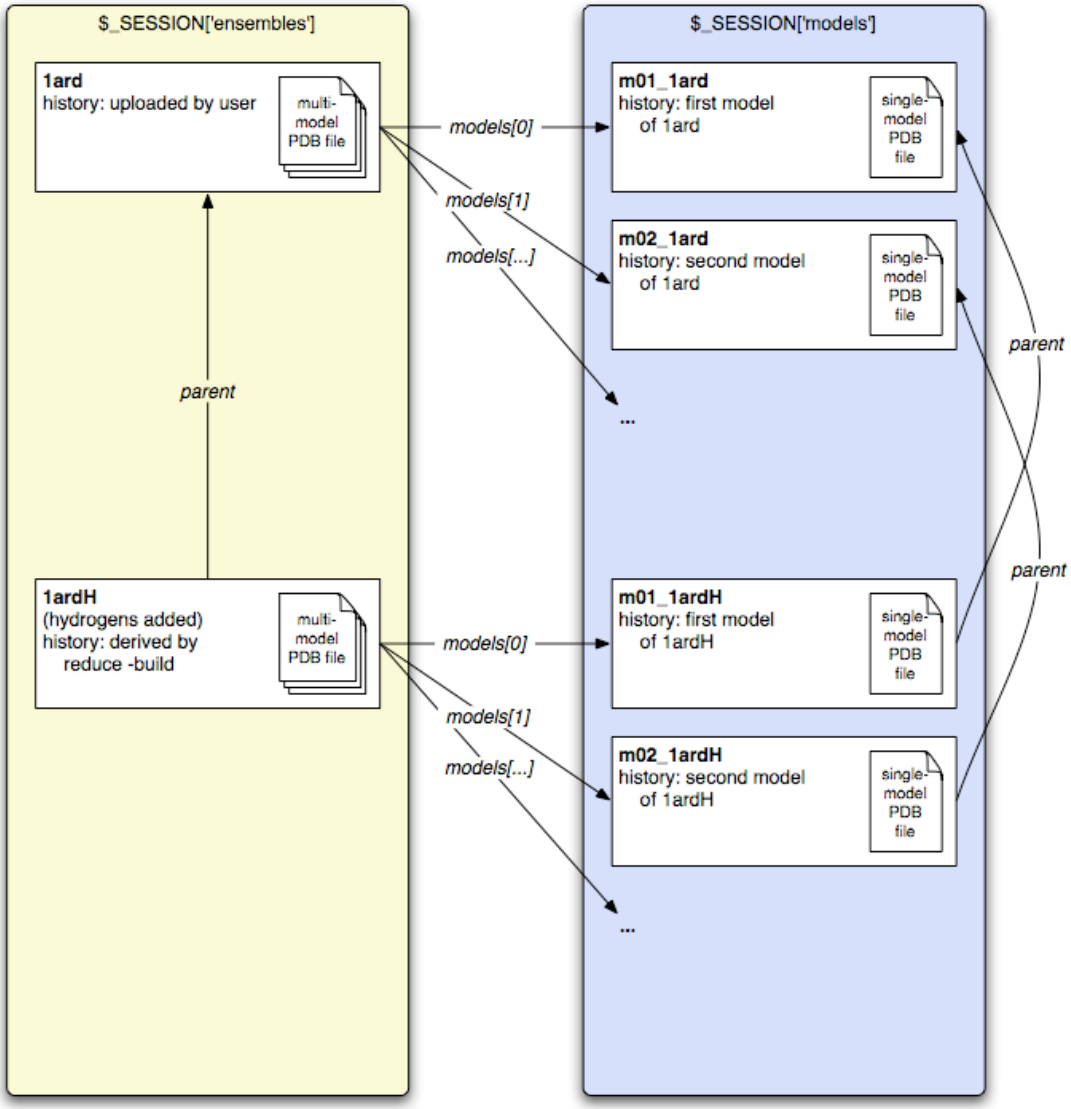
`$_SESSION['models']`. Each one can be analyzed separately and the results can be recombined later for presentation to the user.

We later realized that some programs needed a single PDB file with all the models in it. (For instance, the Java program that makes Ramachandran plot PDFs.) As a result, we introduced an "ensemble" data structure that parallels the "model" data structure. Both are associated with a single PDB file, but the ensemble is a multi-model PDB. The ensemble also has a list of its constituent models, and the two are kept in sync: any material change to a model results in a new ensemble, and any material change to an ensemble results in one or more new models. Functions are available for splitting and merging PDB files to facilitate this process. Figure 39 illustrates the relationship of MolProbity models and ensembles.

This is a powerful programming model, but it is complex and non-intuitive for users. MolProbity has room for improvement in how these concepts are represented in the user interface. My plan is that in the model / ensemble system will be the future mechanism for things like comparing versions of an x-ray structure at various points along its refinement trajectory: the user would upload the individual models, group them together as an ensemble, and then use ensemble-oriented tools for comparative analysis.

Figure 39: The relationship of models and ensembles in MolProbity.

This example depicts the arrangement of data for an NMR file (1ARD) which has had hydrogens added by Reduce. (MolProbity doesn't actually have the ability to add H to all models in an ensemble right now, but this is the way it will work in the future.) New models / ensembles are either uploaded by the user (or equivalently, pulled from an online database) or derived from some model / ensembles already in the system (the "parent") via adding hydrogens, refitting sidechains, etc. Given a new or modified ensemble or set of models, MolProbity can split or join PDB files to produce the matching counterpart(s).


$_SESSION['ensembles']
1ard
history: uploaded by user
multi-model PDB file
parent
1ardH
(hydrogens added)
history: derived by reduce -build
multi-model PDB file
models[0]
models[1]
models[...]
$_SESSION['models']
m01_1ard
history: first model of 1ard
single-model PDB file
m02_1ard
history: second model of 1ard
single-model PDB file
...
m01_1ardH
history: first model of 1ardH
single-model PDB file
m02_1ardH
history: second model of 1ardH
single-model PDB file
...
parent
parent

KiNG

This document describes KiNG 1.54 as of August 16, 2006. This is the version of KiNG that appears in the digital appendix or very similar to it. In addition to this document, anyone working on the code should consult `king/doc/hacking-king.pdf`; although it is somewhat out of date in places, it has much additional information not covered here.

I began writing KiNG in the spring of 2002 (?), initially as a way to teach myself the matrix math used for rotating three-dimensional objects. It was inspired and heavily influenced by the concepts in David Richardson's Mage program, but in many cases the code was developed without direct reference to the Mage C code. KiNG's architecture and role as a web-embedded applet were also influenced by Dave's JavaMage program. Version 1.00 was released to the world three years ago, August 15, 2003, although beta versions were in use well before that. Since then, a number of people have contributed to KiNG, mainly by authoring tools and plugins. Chief among them is Vincent Chen, who has also contributed other improvements throughout the code base.

KiNG is a fairly mature and stable software system, and has some reasonable features in place that allow it to grow and evolve in a modular

manner. However, there are a number of warts and design flaws remaining, many of them caused by my inexperience as a programmer at the time. Many of them are mentioned below, as appropriate. A complete or major rewrite would be desirable at some point in the future, but it will be a major effort.

Java

KiNG is implemented in Sun Microsystems' Java programming language. This has turned out to be a good decision for many reasons -- Java is fast enough to do complicated 3-D graphics, it was designed from the start to be cross-platform, there are many high-quality free libraries available for it, and the documentation is excellent. There are other languages that also match many or all of these criteria, such as C / C++. However, Java's ability to run real applications within a web browser ("applets") with very little additional effort or expertise has been extremely valuable: the KiNG applet is a central part of the MolProbity system. Most other languages are incapable of this or would require significant extra effort and expertise to develop plugins for the various browser/platform combinations, not to mention the fact that users would then have to specifically install those plugins. (90% of MolProbity visitors already have some version of Java enabled.)

Dependencies and modularity

Our goal is for KiNG to run on Sun Java 1.3 or later. Specifically, the core of KiNG should always be able to run as long as Java 1.3 is present. We make no effort to support alternative Java implementations (e.g. GNU Java), as they tend to be incomplete or incompatible especially in terms of graphics and GUI components. Some plugins (discussed below) may require a later version of Sun Java, such as 1.4 or 1.5; for instance, 1.4 introduced regular expression support and linked hash tables, both of which are very useful. Any piece of code that has requirements beyond Java 1.3, however, should be loaded using the reflection API (java.lang.reflect) in such a way that the program can function normally (albeit without the specific functionality). Although it takes more code to do this sort of dynamic loading in Java than it does in (for instance) PHP, KiNG uses this strategy for several bits of code that will not function on all machines (OpenGL rendering, Mac drag-'n'-drop, etc.).

In general, modularity and extensibility are important goals for KiNG's design. The core needs to be small because many users will download it as an applet over slow internet connections. Whenever possible, non-essential features are implemented as a tool or plugin (see below) that is loaded dynamically at run time through the reflection API. This gives users a chance

to mix and match functionality as they see fit, and it means individual components can often be updated without updating the entire application.

Modularity is enforced partly by package structure. At the base is the Driftwood libraries, which comprise several subpackages of commonly used utility functions (data structures, GUI helpers, string manipulation); some of the subpackages depend on one another, but none of them have external dependencies. Next up is the **`king.core`** package, which has all the data structures for parsing, storing, and displaying kinemages; it depends only on Driftwood. The **`king`** package contains all of the GUI scaffolding for a full-featured application: application and applet windows, drop-down menus, kinemage text editor, and various tools for manipulating the kinemage data. It depends on both **`king.core`** and Driftwood. Finally come the tools and plugins, which may be found in the **`king.tool.*`** family of subpackages, or in their own projects (e.g. Chiropraxis, ExtraTools, Molikin, etc.). These may have dependencies on any of the base KiNG components (and generally do), but no part of KiNG should depend on them. There is no way to enforce these dependency relationships; it is up to the KiNG core developers to maintain them.

Top level organization

I have yet to read any concrete advice on how to structure the internals of a desktop application. Given the amount of literature on three-tier web applications, I must assume desktop applications are far more complicated and variable as to their requirements. The design of KiNG is a response to the perceived shortcomings of its predecessor, JavaMage. From my point of view, one of the most irritating features of the JavaMage code was that it was very difficult for one part of the application to communicate with another part unless specific provisions were already in place. In response, KiNG is organized around a "hub" object (**`KingMain`**) that represents a single instance of KiNG. **`KingMain`** contains references to all the major subsystems, and (almost) every class in the **`king`** package and every tool or plugin hold a reference to **`KingMain`**.

Program execution begins in one of two places. For use as a desktop application, execution begins in **`KingMain.main()`**, which parses the command line arguments and then calls **`KingMain.Main()`** (note capitalization: **`main()`** vs. **`Main()`**). When running as an applet in a web browser, execution begins in the **`Kinglet`** class. It has several different ways it

can present itself, but they all end up creating a new **KingMain** object and calling its **Main()** method.

The **Main()** method loads configuration (**KingPrefs**) and then creates the GUI (**ContentPane**); the menus (**UIMenus**); a system for loading and saving kinemages (**KinfileIO**); a system for switching between various open kinemages (**KinStable**); the graphics area (**KinCanvas**); the kinemage text area (**UIText**); and a tree component for browsing kinemage hierarchies (**KinTree**). The various GUI components are then assembled into windows or applets panels as appropriate for the mode KiNG is running in. This code was recently modified so that external code can assemble the GUI components in a different layout; although ugly, this is the only mechanism currently available for embedding KiNG graphics into some other application.

The two most complex subsystems, file I/O and graphics, are described in detail later on.

KingMain is also home to an ugly, ad hoc, poorly designed system for sending messages throughout the application (the **notifyChange()** method). This is a critical service, because many different systems need to know when the kinemage is altered (for instance, to redraw the screen), and many systems

are capable to altering the kinemage. Calls to **`notifyChange()`** cascade down to the major subsystems, which are responsible for notifying their subsystems, and so on. This hard-coded hierarchy is inflexible, inefficient, and difficult to extend. I did develop a publish/subscribe messaging model (**`KinemageSignal`** / **`KinemageSignalSubscriber`**), but most of the code doesn't use it and the two systems are currently intertwined. Unfortunately, because the messaging system was so bad for so long, a lot of code just calls **`KinCanvas.repaint()`** when it has made "minor" changes. This results in the graphics being updated, but no other listeners are notified that the kinemage has changed. A complete rethink of the messaging system is an important goal for a major rewrite of KiNG. Ideally, any change to the kinemage would automatically result in a message to all interested parties, but the naïve implementation would create immense overhead whenever major edits were performed. (Incidentally, this is not the cause of intermittent failure to update of interactive Probe dots; that seems to be due to buffers for the standard input or output streams getting filled and deadlocking.)

Kinemage data structure

The core of KiNG is its classes for representing kinemage data (in **`king.core`**). They closely mimic the file format: a **`Kinemage`** contains

KGroups, which contain **KSubgroups**, which contain **KLists**, which contain **KPoints**. (The "K" prefix is to avoid collision with other Java libraries that use the common words Point, List, or Group.) Groups, subgroups, and lists are fairly simple containers that inherit most of their functionality from the **AGE** class (Abstract Grouping Element). **Kinemage** has more methods because it has to keep track of much file-level data, but it is otherwise quite similar. These classes form a bidirectional hierarchy: parents have lists of their children, and children know their parents. For reasons that are now obscure to me, building the hierarchy requires two separate method calls: one to add a child to a parent, one to set the parent for a child. I suspect it had to do with parsing the kinemage. Anyway, it's now firmly entrenched but a common source of errors.

All lists are **KLists**, but there are many implementations of **KPoint**: **DotPoint**, **VectorPoint**, **BallPoint**, etc. An alternative arrangement would be to have specialized lists and uniform points, but there are two reasons not to do it that way. First, Mage allows for mixed lists -- a ball in the middle of a list of vectors, and so on. Although KiNG doesn't currently support this, it could do so relatively easily under the current scheme. Second, generalized points would use too much memory. In a typical, well-designed kinemage,

there are many more points than there are lists. Each list can therefore afford to waste a few bytes of memory for fields it won't use (e.g. vector width for a list of balls). On the other hand, adding a field to every point in the kinemage just to support a rarely-used feature is fatal. Unlike C, Java does not support the **`union`** concept, so there is no way to make one field do double duty. Thus, each type of point is specialized, so that it allocates memory only for the properties it supports. To save even more memory, there is a mechanism that uses a very space-efficient hash table (**`tmGet()`** / **`tmPut()`**) to store rarely used attributes, like aspects and comments. Because it is somewhat slower than ordinary attributes, this is only a win if (1) relatively few points actually define the attribute (otherwise it will consume more memory than a dedicated field), and (2) the attribute doesn't usually need to be accessed during the painting cycle (which should be as fast as possible). For example, we can get away with using this system for aspects because most kinemages don't define aspects, and even if they do, the aspects don't need to be accessed in the painting cycle unless the user has chosen to view a particular aspect.

In the future, I would like to see the core of KiNG evolve into a self-contained library for 3D vector graphics that could easily be used by other programs; however, there are some design flaws to fix first. Two have

already been mentioned: two calls are required to establish a parent-child relationship, and there isn't a good messaging system for notifying listeners when the kinemage is modified. Additionally, all the descendents of **AGE** (groups, subgroups, and lists) are too tightly coupled to the KiNG application. First, they all contain a **JCheckBox** that records their on/off state, because (1) it would be difficult to coordinate the internal on/off state with the state showed by the GUI checkbox otherwise, and (2) if the GUI maintained a checkbox-to-**AGE** mapping there is an additional opportunity for memory leaks when the kinemage is discarded. Second, all the descendants of **AGE** implement **MutableTreeNode** so they can be used directly in the **JTree** component of **KinTree**. In some ways, this simplifies that part of the code, but it needlessly complicates the **AGE** class. It would be better to implement **MutableTreeNode** proxies or wrappers around the **AGE** objects, but then one has to work at keeping the two hierarchies synchronized when the user starts editing the kinemage.

File I/O and kinemage parsing

KiNG employs a hand-built custom parser and tokenizer for reading kinemage files. I am well aware that there are several state-of-the-art tools for creating Java parsers, such as JavaCC and ANTLR. However, the kinemage

language is not entirely regular, and these tools require deep knowledge of formal grammars and other CS voodoo, plus learning a whole other language. The added complexity and dependencies (what if those projects disappear?) still don't seem worthwhile.

After careful study, I did formulate relatively simple rules for correctly dividing up any kinemage into a series of "words" (tokens). Those rules are documented in the second half of `doc/format-kinemage.pdf`, and they are implemented in **`KinfileTokenizer`**. This at least is standard CS procedure, and allows the parser to consider sequences of symbols instead of just sequences of characters. The tokenizer is also used by **`MageHypertexter`** to aid in parsing Mage-style hyperlinks, which follow basically the same rules as the rest of kinemage format.

The actual parsing is done by **`KinfileParser`**, which uses the tokenizer to scan through a stream and constructs one or more new **`Kinemage`** objects. Every kinemage keyword starts with an "@" symbol; these divide the kinemage into blocks. Therefore the parser contains one function for dealing with each keyword it understands. If it encounters an unrecognized keyword, it skips ahead until it encounters another keyword token.

Like Mage, the KiNG parser tries to be tolerant and forgiving of mistakes. Since kinemage files are often created and edited by hand, it is important that the parser be as forgiving as possible. Unlike Mage in its default mode, however, KiNG prints error messages (with line numbers!) when it encounters syntax errors. The purpose is twofold: if the kinemage doesn't look as expected, the user / author has some idea where the problem could be; if the kinemage does look (approximately) as expected, the user / author is warned of a potentially subtle mistake. The trade-off is that inexperienced users may be inundated with error messages that they can't reasonably interpret and act on, but I still believe it is better for them to be aware that there may be a problem with the kinemage than for KiNG to fail silently.

In a perfect world, the KiNG parser would also be dynamically extensible via a plugin mechanism, so that plugins could support new file format features. This would be reasonably easy to do at the keyword granularity, because the parser is already designed on a one-function-per-keyword basis. It would be much harder to do at the attribute level, so that (for instance) a plugin could allow for some new property of groups or lists, but that might be desirable. There would have to be a corresponding mechanism to extend **`KinWriter`**, so that custom data could be saved as well.

Most code will interact with **KinfileIO** instead of **KinfileParser**, because the former can load kinemages in a background thread while displaying a progress dialog. Since it may take several minutes to load a large kinemage, it is important that the application not freeze up for that whole time. The choreography required to do this is more complicated than it should be, but most of the thread programming is isolated in the **KinfileLoader** class, while **KinfileIO** takes care of managing load / save dialogs and file names. On the other hand, the mechanism for dynamic Probe dots uses **KinfileParser** directly to load the output from Probe, because it is already running in a background thread and does not want to display a progress dialog.

Rendering

KinCanvas is mostly just a drawing surface, but it coordinates both the graphics rendering and the user input via mouse and keyboard. Input is described below, but graphics rendering is mostly handled by the **Engine** class. **Engine** handles all the calculations for rotating, translating, and scaling point coordinates. It also handles clipping, depth cueing, shortening lines that intersect balls, and various types of picking (pick at a point, pick within a ring, pick within a sphere, etc.).

Drawing uses a simple Painter's Algorithm that proceeds in two rounds. In the first round, coordinates are calculated for each point; every **`KPoint`** has fields to store its original (**`x0`**, **`y0`**, **`z0`**) and transformed (**`x`**, **`y`**, **`z`**) coordinates. For maximum efficiency, all the operations -- rotation, translation, scaling, and even perspective correction -- are collapsed into a 4-by-4 matrix (**`driftwood.r3.Transform`**), which is then applied once to each point. Points are sorted into bins based on their (transformed) Z coordinate, or discarded if they are outside the clipping planes. In the second round, the points are revisited in back-to-front order, and painted to the screen. For picking, the Z-sorted list is traversed front-to-back, so as to pick on the point closest to the front of the view.

It is well known that the painter's algorithm produces the wrong image for certain arrangements of overlapping objects, where no one is completely in front of the others. However, such arrangements generally do not occur in kinemages of macromolecular structure, so it is not usually a problem. A more elegant but computationally expensive answer is to record a depth value at each pixel ("z-buffering"). OpenGL does this with specialized hardware, allowing it to be blindingly fast. JMol developers claim to have done this completely in Java software, but it does seem to limit them to relatively simple

models on a relatively small canvas -- JMol at full screen size would be too slow. KiNG is capable of handling much larger, more complicated structures at larger screen sizes (especially with fast 2D primitives), at the occasional expense of rendering realism.

KiNG does currently feature OpenGL-accelerated graphics, but it does not use them to their full potential. For maximum speed, KiNG would use 3D OpenGL calls, so that transformation and depth-sensitive rendering were all done with specialized hardware. Instead, KiNG currently uses OpenGL as a library of 2D primitives that are faster than the equivalent Java2D methods; it still does transformations and depth sorting itself, in software. In fact, the work I did on this would have been totally unnecessary if Sun had just made their 2D graphics take advantage of modern graphics hardware in the first place. They're finally getting around to that, but it may still be years before it's available for production use. It would be nice for KiNG to take full advantage of the 3D OpenGL calls, but that would require creating a whole other **`Engine`** class for OpenGL drawing. (It's important that KiNG never rely exclusively on OpenGL or abandon the pure Java rendering pathway, so that it can continue to run seamlessly in a standard pure-Java browser environment.) Because KiNG would no longer do the transformations and

depth sorting, it would also require an alternative approach to picking, and additional logic would be required if the kinemage contained any translucent objects. There are established techniques for dealing with these problems, but they are not trivial.

In order to simplify the drawing code, I created the **`Painter`** interface to abstract away the differences between OpenGL 2D graphics and Java2D graphics. Painter has all the functions that kinemage points need to draw themselves. Behind the scenes, **`StandardPainter`** calls **`java.awt.Graphics`** methods while **`JoglPainter`** calls JOGL methods to achieve the same result. The alternative approach would be to put multiple drawing methods in each point: **`paintJava2D()`**, **`paintJOGL()`**, etc. This would be easy enough for **`DotPoints`**, which do very simple drawing, but it would cause massive code duplication for **`VectorPoints`**, which have to calculate line shortening and clipping before drawing to the screen. Additionally, with my arrangement it is easy to add other **`Painters`**, such as **`HighQualityPainter`** (which uses methods from **`java.awt.Graphics2D`** to make prettier-than-normal pure Java graphics). Since **`StandardPainter`** and **`HighQualityPainter`** are in turn built on the **`Graphics`** and **`Graphics2D`** interfaces, it is simple to swap in a

Graphics object that points to the printer, a JPEG image, or a PDF: this is how KiNG produces those kinds of output without much extra effort.

User interaction

Every **KinCanvas** object has a **ToolBox** object associated with it, which is responsible for listening for mouse and keyboard events and dealing with them. **ToolBoxMW** is a dynamically loaded subclass that will enable mouse wheel support for Java 1.4 or later; Java 1.3 doesn't support the mouse wheel. **ToolBox** also defines the **overpaintCanvas()** method, which allows it to add extra information to the graphics once all 3D objects are rendered (e.g., the point ID in the lower left corner).

ToolBox doesn't actually handle mouse and keyboard events itself, because a typical kinemage viewer can be put into many "modes" -- a viewing mode, several drawing modes, a model refitting mode, etc. Each mode is represented by a "tool", a subclass of **king.BasicTool**. Only one tool can be active at any given time, and it receives all mouse and keyboard events at the expense of any other tools that may be loaded. Most of the common actions a tool would invoke (rotate, zoom, pick, etc.) are implemented in **king.ToolServices**, so that all tools can share them.

KiNG's tools are actually a superset of its plugins (i.e., **`king.BasicTool`** is a subclass of **`king.Plugin`**), so both can be discovered and loaded dynamically at run time. **`ToolBox`** contains all the machinery for discovering tools / plugins and loading them via the Reflection API. **`ToolBox`** follows the standard Java service provider (SPI) model for discovering new tools and plugins. Detailed documentation is provided, but in brief each JAR file with tools or plugins lists the fully-qualified names of those classes in a file named `META-INF/services/king.Plugin`. As long as that JAR is on the Java classpath when KiNG is launched, those tools and plugins will be loaded by KiNG.

The difference between tools and plugins is in user interaction. Tools receive exclusive rights to mouse and keyboard events when they are active. This approach was adopted to simplify matters, so that independently developed tools wouldn't simultaneously react to the same mouse event, with unanticipated consequences. On the other hand, plugins never get events from the mouse or keyboard, but they can add items to the Tools menu. They're often used for data importers or exporters, or they often provide a GUI for editing or manipulating the kinemage in some way that doesn't require point-and-click in the graphics.

Refitting and rebuilding tools

Strictly speaking, the refitting and rebuilding tools (for backrubs, rotamers, etc.) are not part of KiNG proper, but are tools and plugins that are defined in the Chiropraxis packages. However, because they are closely related to the scientific content of this thesis, they are described here. The process of user refitting with backrubs is described in Chapter 3 and in Davis et al. (2006), while its successful application in structural genomics use is reported in Arendall et al. (2005). They are also one of the most complex parts of KiNG as a whole, and need explanation if someone else wishes to continue their development in the future.

MolDB: a molecular "database"

All of the Java code I have written to deal with macromolecules uses data structures defined in **`driftwood.moldb2`**. Although the PDB format seems simple, there are a large number of complicated cases that the Richardson lab has observed and documented "in the wild". Designing a system that deals with even a majority of such cases requires a lot of effort, and hence future developers are strongly encouraged to extend this system rather than develop ad hoc PDB parsers.

In particular, the MolDB classes have evolved over several generations to address the thorniest problem for dealing with structures, namely, multiple conformations. Most structural biologists and structural biology algorithms default to thinking of molecules as objects in the classical sense, with a defined list of parts, each of which exists at exactly one well-defined location at any particular point in time. PDB (and mmCIF) files take this one step further by introducing MODEL / ENDMDL records, which allow multiple conformations to be defined. This isn't too bad, because each MODEL can be treated as a separate classical entity as described above. On the other hand, there is no guarantee that the contents of one MODEL are logically related to the contents of another -- they could be totally different proteins, as far as the file format is concerned. To complicate matters, PDB (and mmCIF) files allow for alternate conformations, which provide multiple locations for some atoms but not others. The ambiguities and potential for abuse are huge here. As a small sampling: Should an atom with alternate conformations define a default (" ") state? If two atoms both define an "A" and a "B" state, does atom 1 in state A imply that atom 2 must be in state A? If one atom defines states "A" and "B" and another defines "C" and "D", which state(s) of atom 2 are compatible with state A of atom 1? If the user is refitting a sidechain in alternate

conformation "A" and in the process moves some backbone atoms with no alternates defined, should that move affect only state A (i.e. define alternate conformation flags for the backbone atoms) or all states (leave backbone in state " ")?

MolDB defines a logical framework in which such questions can be addressed. The contents of a PDB or mmCIF file are regarded as a collection of **Model**s, analogous to a PDB MODEL record. Each **Model** contains one or more **Residue**s, which in turn contain one or more **Atom**s. **Residue**s may belong to a chain and/or a segment, but those units are not part of the hierarchy per se because they can be used in pathological ways (e.g. one segment (PROT) spanning multiple chains (A, B), one of which belongs to multiple segments (PROT, SOLV)). Thus a **Model** defines a "parts list" for a structure, but it does not define a specific conformation.

Conformations are defined by **ModelState**s. A **Model** may have multiple **ModelState**s, but each **ModelState** defines exactly one position for every **Atom** in the **Model**. For good reasons, it is possible to have a **ModelState** that leaves some **Atom** positions undefined, but these will generally lead to exceptions if used directly for calculations. **ModelState**s record **Atom** locations using **AtomState** objects. Accordingly, a **ModelState** has no more

than one **AtomState** for every **Atom** in a **Model**, but given that a **Model** may be associated with multiple **ModelState**s, an **Atom** may be associated with multiple **AtomState**s. In practice, an **AtomState** object is what a structurally-oriented programmer would think of as an atom, and corresponds one-to-one with an ATOM record from a PDB file. This arrangement is depicted in the top panel of Figure 40.

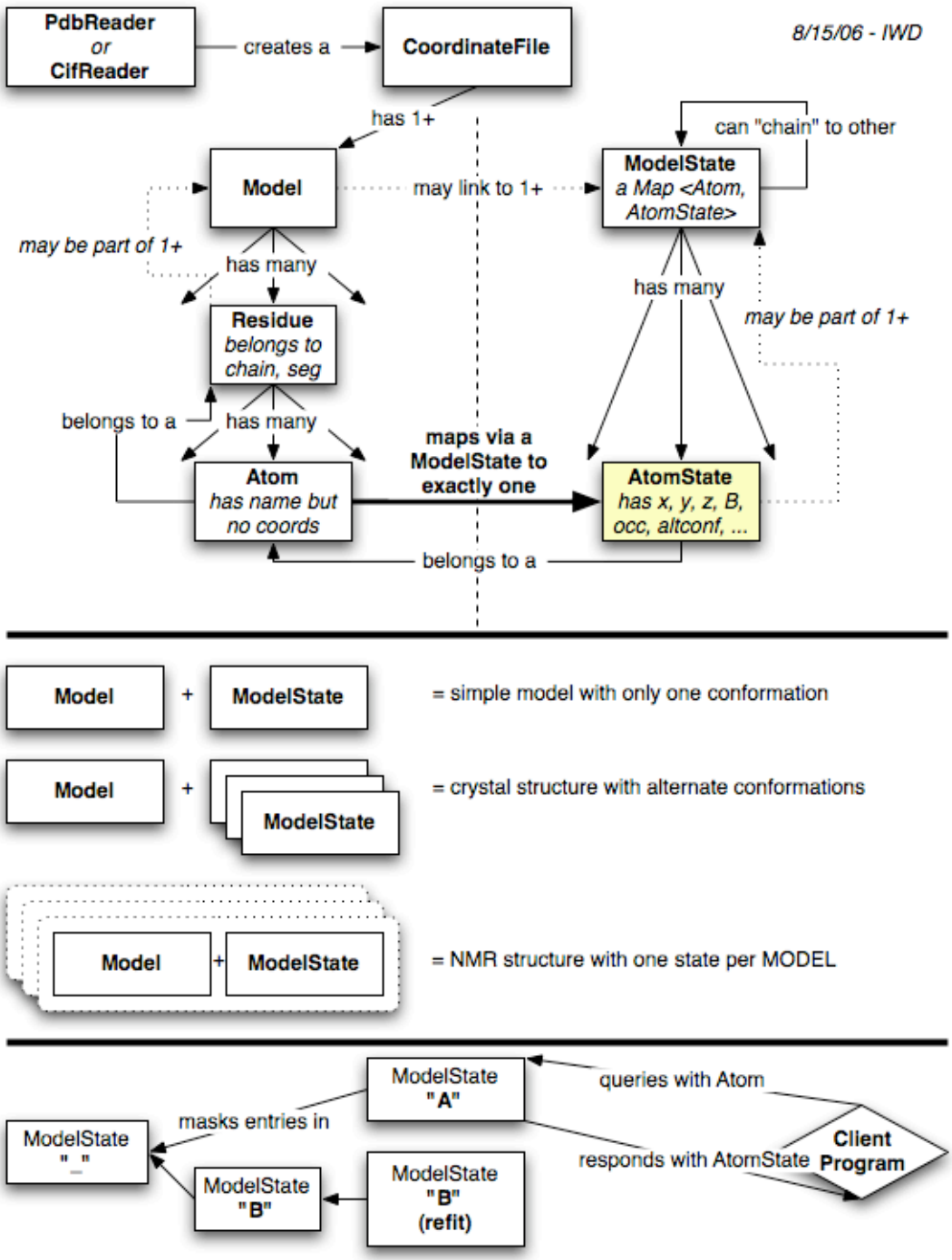
ModelStates may be linked together to form tree-like structures. If a **ModelState** is queried for the position of an **Atom**, it first looks to see if it has a matching **AtomState**. If not, it queries its parent for the position of that **Atom**, and so on. An exception is thrown only if none of the ancestor **ModelState**s contains an appropriate state for the **Atom**. This allows one to efficiently construct derivative conformations that differ only in a few details, without changing anything in the original **ModelState**. Unsurprisingly, this mechanism is used for refitting (e.g. the derivative state contains a new conformation for one sidechain) and for representing alternate conformations (the base state contains all " " atoms, and derivative states exist for the "A", "B", … atoms). This arrangement is depicted in the bottom panel of Figure 40.

Multi-MODEL PDB files such as NMR structures are represented differently: each MODEL entry generates a separate **Model** object. This is not

ideal, but I haven't devised a better answer yet because PDB MODELs can potentially contain totally different molecules rather than different conformations of the same molecule (e.g. the output of certain superposition programs). Since MODELs are not usually used together with alternate conformations (thank God!), each such **`Model`** generally has just one **`ModelState`**. The various arrangements created by common PDB files are depicted in the center panel of Figure 40.

Figure 40: The architecture of MolDB.

See text for details. Solid-line arrows represent actual object references held in the code. Dotted-line arrows represent logical relationships that are deliberately not captured directly in the code, to facilitate reuse of child objects in modified copies of their parents.


PdbReader
or
CifReader
creates a
CoordinateFile
8/15/06 - IWD
has 1+
Model
may link to 1+
ModelState
a Map <Atom, AtomState>
can "chain" to other
may be part of 1+
has many
Residue
belongs to chain, seg
has many
may be part of 1+
belongs to a
has many
Atom
has name but no coords
maps via a ModelState to exactly one
AtomState
has x, y, z, B, occ, altconf, ...
belongs to a
Model
+
ModelState
= simple model with only one conformation
Model
+
ModelState
= crystal structure with alternate conformations
Model
+
ModelState
= NMR structure with one state per MODEL
ModelState
"A"
queries with Atom
masks entries in
ModelState
" "
responds with AtomState
Client Program
ModelState
"B"
ModelState
"B"
(refit)

Despite all the work, MolDB still has significant shortcomings. I have not yet written any code that tries to work seriously with multiple MODELs, but I suspect that when I do the current arrangement will prove unsatisfactory. A possible solution would be to return a single **`Model`** with multiple **`ModelState`**s whenever the multiple **`Model`**s were identical, but there is some question as to how "identical" they would have to be for this to work -- completely, or just mostly? The current definitions of **`equals()`** for **`Model`**s, **`Residue`**s, and **`Atom`**s makes this more complicated (in general, two "identical" **`Atom`**s will not compare as equal), and those definitions interact strongly with the workings of **`ModelState`**, so changes must be made carefully. Those equality definitions also make it hard to reliably and unambiguously define parts of the model by name (e.g. "A GLU 13"). However, enabling name-based identity would require enforcing name-uniqueness restrictions, which further complicates the PDB and mmCIF parsers. It is not unheard of for a PDB file to redefine an atom with the same name (including the same alternate conformation flag, usually " ") but with different coordinates. Requiring name uniqueness leads to either data loss or arbitrarily creating alternate conformation assignments on the fly. On top of all this, even fully legal PDB files have a somewhat different data model than fully legal mmCIF files. For

instance, the mmCIF specification clearly allows residue "numbers" to be any arbitrary string. The MolDB classes have been reworked to provide for this in part, but it is still difficult to provide a "universal" data structure for disparate macromolecular structure formats. Once one considers all the technically illegal but widely used variations on those file formats, it becomes nearly impossible.

The Chiropraxis model manager

All of the Chiropraxis rebuilding tools share a common **`ModelManager`** that holds the current **`Model`**(s) and **`ModelState`**(s). Multiple simultaneous editing operations are generally allowed (e.g. a movable rotamer riding on a backrub motion), but locking mechanisms exist to avoid logical collisions (e.g. two overlapping backrub motions). Rebuilding tools change the current model by constructing a new, derivative **`ModelState`** object and populating it with new **`AtomState`**s for the atom(s) that have moved. This **`ModelState`** then becomes the current one in the **`ModelManager`**, and is available on a read-only basis to all the rebuilding tools. This is an important point: once a **`ModelState`** is created, it is never altered in the future. By walking back up the tree of **`ModelState`**s, the **`ModelManager`** can easily "peel off" layers of change to implement a robust Undo function. **`ModelManager`** actually maintains a stack

of **`Model`** / **`ModelState`** pairs, because a change will occasionally involve e.g. mutating a residue, which impacts both the **`Model`** and the **`ModelState`**. In these cases, the **`Model`** object is copied and then modified, so that the original remains unchanged.

The model manager is also responsible for managing shared, dynamically-updated quality indicators, such as all-atom contact dots from Probe or NOE violations from NOE-display. This is crying out for a plugin mechanism, but at the moment all the logic is hard-coded within the **`ModelManager`** class. This mechanism invokes the most complicated multi-threaded code anywhere in KiNG (**`BgKinRunner`**). This class starts a background thread which waits for update requests from the GUI thread, then launches a command line program, communicates with it via Unix standard input and output streams, and parses the kinemage output. It then creates a callback on the GUI thread that actually updates the kinemage and the display. It is capable of rudimentary command substitution, and can create PDB-format file fragments for mobile regions of the model on the fly. If updates are occurring rapidly, it can also abort a run part way through and immediately restart with the most current data. All this is made harder because the Java mechanisms for communicating with external processes over standard input / output seem buggy and prone to deadlock.

Things seem to have improved in the later 1.4 and 1.5 releases, but background update can still fail fairly routinely for inscrutable reasons. Also, what works on one platform (Linux) may not work on another (Mac, Windows).

Final notes

Here ends the documentation of the software systems I created for my Ph.D. research in the Richardson lab, but never fear: there is more documentation. For MolProbity, there is information on installing and maintaining the server in the MolProbity `doc` directory, and there is detailed information for developers in `doc/extending`. Most functions are documented individually, and the command line scripts in `cmdline` are generally documented in comments at the top of the file. (In the future they may have **`-help`** switches too.)

For KiNG and the other Java software, there are many Javadoc-formatted comments within the code, and per-project documentation in the per-project `doc` directories. Of particular note is the official kinemage format document, `format-kinemage.pdf`. It aims to cover all of the core kinemage concepts in sufficient detail to enable a semi-independent implementation, and I made a particular effort to document the tokenization rules for parsers. (This

document was authored and maintained with LyX, www.lyx.org). Also important is Hacking KiNG, which reviews all of the code in Driftwood, Chiropraxis, and KiNG on a package-by-package basis, describing the relationship of parts and the functionality available. This document also describes my general approach to organizing software projects as well as tools and procedures for editing source code, creating documentation, designing build systems, and packaging and deployment. Hacking KiNG is somewhat out-of-date but still accurate for much of the code base. Finally, I expect much of the future work in KiNG will be done as tools or plugins. Numerous examples exist, including the **`sample_plugins`** package designed specifically for illustrating the process. Prose documentation can be found in Hacking KiNG, as well as the comments for the **`ToolBox`**, **`Plugin`**, and **`BasicTool`** classes.

I sincerely hope this software is constructed well enough to stand the test of time, allowing other researchers and graduate students to build on the base it provides. There are some promising preliminary indications that this will happen. I will be happy to consult on further development projects to the extent allowed by my future schedule and employer; in the absence of a more current address, your best bet for reaching me is at `ian.w.davis@gmail.com`.

Appendix B: Digital resources

This thesis should be accompanied by an optical disc (CD or DVD) containing all the code and data described herein. If you didn't get one, you should feel cheated. Complain to David or Jane Richardson.

References

Abramson, I. S. (1982). "On Bandwidth Estimates in Kernel Estimates -- A Square Root Law." The Annals of Statistics **10**(4): 1217-1223.

Adams, P. D., R. W. Grosse-Kunstleve, L. W. Hung, T. R. Ioerger, A. J. McCoy, N. W. Moriarty, R. J. Read, J. C. Sacchettini, N. K. Sauter and T. C. Terwilliger (2002). "PHENIX: building new software for automated crystallographic structure determination." Acta Crystallogr D Biol Crystallogr **58**(Pt 11): 1948-54.

Addlagatta, A., S. Krzywda, H. Czapinska, J. Otlewski and M. Jaskolski (2001). "Ultrahigh-resolution structure of a BPTI mutant." Acta Crystallogr D Biol Crystallogr **57**(Pt 5): 649-63.

Arendall, W. B., W. Tempel, J. S. Richardson, W. Zhou, S. Wang, I. W. Davis, Z. J. Liu, J. P. Rose, M. Carson, M. Luo, D. C. Richardson and B. C. Wang (2005). "A Test Of Enhancing Model Accuracy In High-Throughput Crystallography." The Journal of Structural and Functional Genomics **6**: 1-11.

Bahar, I., B. Erman, T. Haliloglu and R. L. Jernigan (1997). "Efficient characterization of collective motions and interresidue correlations in proteins by low-resolution simulations." Biochemistry **36**(44): 13512-23.

Baldwin, E. P., O. Hajiseyedjavadi, W. A. Baase and B. W. Matthews (1993). "The Role of Backbone Flexibility in the Accommodation of Variants That Repack the Core of T4 Lysozyme." Science **262**: 1715-1718.

Bergman, L. D., B. E. Rogowitz and L. A. Treinish (1995). A Rule-based Tool for Assisting Colormap Selection. IEEE Conference on Visualization, Atlanta, GA.

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

Blaszczyk, J., Y. Li, G. Shi, H. Yan and X. Ji (2003). "Dynamic roles of arginine residues 82 and 92 of Escherichia coli 6-hydroxymethyl-7,8-dihydropterin pyrophosphokinase: crystallographic studies." Biochemistry **42**(6): 1573-80.

Bouvignies, G., P. Bernado, S. Meier, K. Cho, S. Grzesiek, R. Bruschweiler and M. Blackledge (2005). "Identification of slow correlated motions in proteins using residual dipolar and hydrogen-bond scalar couplings." Proc Natl Acad Sci U S A **102**(39): 13885-90.

Bruccoleri, R. E. and M. Karplus (1985). "Chain Closure with Bond Angle Variations." Macromolecules **18**(12): 2767-2773.

Brunger, A. T. (1992). "Free R-value - a novel statistical quantity for assessing the accuracy of crystal structures." Nature **355**(6359): 472-475.

Bryan, P. N. (2000). "Protein engineering of subtilisin." Biochimica et Biophysica Acta (BBA) - Protein Structure and Molecular Enzymology **1543**(2): 203-222.

Butterfoss, G. and J. Hermans (2003). "Packing imperfections in proteins probed by database statistics and side chain energetics." Protein Sci **12**: 2719-31.

Butterfoss, G. L., J. S. Richardson and J. Hermans (2005). "Protein imperfections: separating intrinsic from extrinsic variation of torsion angles." Acta Crystallogr D Biol Crystallogr **61**(Pt 1): 88-98.

Cavanagh, J., W. J. Fairbrother, I. Palmer, A.G. and N. J. Skelton (1996). Protein NMR Spectroscopy: Principles and Practice. San Diego, Academic Press.

Chakrabarti, P. and D. Pal (2001). "The interrelationships of side-chain and main-chain conformations in proteins." Progr. Biophys. & Mol. Biol. **76**: 1-102.

Cheam, T. C. and S. Krimm (1990). "Ab initio force fields of alanine dipeptide in four non-hydrogen bonded conformations." Journal of Molecular Structure **Theochem 206**: 173-203.

Chothia, C. and A. M. Lesk (1986). "The relation between the divergence of sequence and structure in proteins." Embo J **5**(4): 823-6.

Chou, J. J., D. A. Case and A. Bax (2003). "Insights into the mobility of methyl-bearing side chains in proteins from (3)J(CC) and (3)J(CN) couplings." J Am Chem Soc **125**(29): 8959-66.

Crennell, S. J., E. F. Garman, C. Philippon, A. Vasella, W. G. Laver, E. R. Vimr and G. L. Taylor (1996). "The structures of Salmonella typhimurium LT2 neuraminidase and its complexes with three inhibitors at high resolution." J Mol Biol **259**((2) Jun 7): 264-280.

Crooks, G. E., G. Hon, J. M. Chandonia and S. E. Brenner (2004). "WebLogo: a sequence logo generator." Genome Res **14**(6): 1188-90.

D'Aquino, J. A., J. Gomez, V. J. Hilser, K. H. Lee, L. M. Amzel and E. Freire (1996). "The Magnitude of the Backbone Conformational Entropy Change in Protein Folding." Proteins: Structure, Function, and Genetics **25**: 143-156.

Dahiyat, B. I. and S. L. Mayo (1997). "*De Novo* Protein Design: Fully Automated Sequence Selection." Science **278**: 82-87.

Davis, I. W., W. B. Arendall, 3rd, D. C. Richardson and J. S. Richardson (2006). "The backrub motion: how protein backbone shrugs when a sidechain dances." Structure **14**(2): 265-74.

Davis, I. W., L. W. Murray, J. S. Richardson and D. C. Richardson (2004). "MolProbity: structure validation and all-atom contact analysis for nucleic acids and their complexes." Nucleic Acids Research **32, Web Server Issue**: W615-W619.

Davis, I. W., L. W. Murray, J. S. Richardson and D. C. Richardson (2004). "MOLPROBITY: structure validation and all-atom contact analysis for nucleic acids and their complexes." Nucleic Acids Res **32**(Web Server issue): W615-9.

DeGrado, W. F., Z. R. Wasserman and J. D. Lear (1989). "Protein design, a minimalist approach." Science **243**(4891): 622-8.

DePristo, M. A., P. I. de Bakker, S. C. Lovell and T. L. Blundell (2003). "Ab initio construction of polypeptide fragments: efficient generation of accurate, representative ensembles." Proteins **51**(1): 41-55.

Desjarlais, J. R. and T. M. Handel (1995). "*De novo* design of the hydrophobic cores of proteins." Protein Science **4**: 2006-2018.

Desjarlais, J. R. and T. M. Handel (1999). "Side-chain and backbone flexibility in protein core design." J Mol Biol **290**(1): 305-18.

Dwyer, M. A., L. L. Looger and H. W. Hellinga (2004). "Computational design of a biologically active enzyme." Science **304**(5679): 1967-71.

Emsley, P. and K. Cowtan (2004). "Coot: model-building tools for molecular graphics." Acta Crystallogr D Biol Crystallogr **60**(Pt 12 Pt 1): 2126-32.

Engh, R. A. and R. Huber (1991). "Accurate Bond and Angle Parameters for X-ray Protein Structure Refinement." Acta Crystallographica, Section A **47**: 392-400.

Engh, R. A. and R. Huber (2002). Structure quality and target parameters. International Tables for Crystallography F. M. G. Rossman and E. Arnold.

English, A. C., S. H. Done, L. S. D. Caves, C. R. Groom and R. E. Hubbard (1999). "Locating Interaction Sites on Proteins: The Crystal Structure of Thermolysin Soaked in 2% to 100% Isopropanol." Proteins: Structure, Function, and Genetics **37**: 628-640.

Erskine, P. T., L. Coates, S. Mall, R. S. Gill, S. P. Wood, D. A. Myles and J. B. Cooper (2003). "Atomic resolution analysis of the catalytic site of an aspartic proteinase and an unexpected mode of binding by short peptides." Protein Science **12**(8): 1741-9.

Esposito, L., L. Vitagliano, F. Sica, G. Sorrentino, A. Zagari and L. Mazzarella (2000). "The ultrahigh resolution crystal structure of ribonuclease A

containing an isoaspartyl residue: hydration and sterochemical analysis." J Mol Biol **297**(3): 713-32.

Fadel, A. R., D. Q. Jin, G. T. Montelione and R. M. Levy (1995). "Crankshaft motions of the polypeptide backbone in molecular dynamics simulations of human type-alpha transforming growth factor." J Biomol NMR **6**(2): 221-6.

Fenn, T. D., D. Ringe and G. A. Petsko (2004). "Xylose isomerase in substrate and inhibitor michaelis states: atomic resolution studies of a metal-mediated hydride shift." Biochemistry **43**(21): 6464-74.

Fuhrmann, C. N., B. A. Kelch, N. Ota and D. A. Agard (2004). "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 **338**(5): 999-1013.

Gassner, N. C., W. A. Baase and B. W. Matthews (1996). "A test of the "jigsaw puzzle" model for protein folding by multiple methionine substitutions within the core of T4 lysozyme." Proceedings of the National Academy of Sciences of the United States of America **93**: 12155-12158.

Getzoff, E. D., K. N. Gutwin and U. K. Genick (2003). "Anticipatory active-site motions and chromophore distortion prime photoreceptor PYP for light activation." Nat Struct Biol **10**(8): 663-8.

Gould, I. R. and P. A. Kollman (1992). "Ab initio SCF and MP2 calculations on four low-energy conformers of *N*-Acetyl-*N*′-methylalaninamide." J. Phys. Chem. **96**: 9255-9258.

Gunasekaran, K., C. Ramakrishnan and P. Balaram (1996). "Disallowed Ramachandran conformations of amino acid residues in protein structures." Journal of Molecular Biology **264**: 191-198.

Gutte, B. (1975). "A synthetic 70-amino acid residue analog of ribonuclease S-protein with enzymic activity." J Biol Chem **250**(3): 889-904.

Gutte, B. (1977). "Study of RNase A mechanism and folding by means of synthetic 63-residue analogs." J Biol Chem **252**(2): 663-70.

Gutte, B., M. Daumigen and E. Wittschieber (1979). "Design, synthesis and characterisation of a 34-residue polypeptide that interacts with nucleic acids." Nature **281**(5733): 650-5.

Harbury, P. B., J. J. Plecs, B. Tidor, T. Alber and P. S. Kim (1998). "High-resolution protein design with backbone freedom." Science **282**: 1462-1467.

Harrower, M. A. and C. A. Brewer (2003). "ColorBrewer.org: An Online Tool for Selecting Color Schemes for Maps." The Cartographic Journal **40**(1): 27-37.

Head-Gordon, T., M. Head-Gordon, M. J. Frisch, I. Charles L. Brooks and J. A. Pople (1991). "Theoretical study of blocked glycine and alanine peptide analogues." J. Am. Chem. Soc. **113**: 5989-5997.

Hecht, M. H., A. Das, A. Go, L. H. Bradley and Y. Wei (2004). "De novo proteins from designed combinatorial libraries." Protein Sci **13**(7): 1711-23.

Hecht, M. H., J. S. Richardson, D. C. Richardson and R. C. Ogden (1990). "*De Novo* Design, Expression, and Characterization of Felix: A Four-Helix Bundle Protein of Native-Like Sequence." Science **249**: 884-891.

Hellinga, H. W. (2005). Personal communication.

Henikoff, S. and J. G. Henikoff (1992). "Amino acid substitution matrices from protein blocks." Proc Natl Acad Sci U S A **89**(22): 10915-9.

Hershkovitz, E., E. Tannenbaum, S. B. Howerton, A. Sheth, A. Tannenbaum and L. D. Williams (2003). "Automated identification of RNA conformational motifs: theory and application to the HM LSU 23S rRNA." Nucleic Acids Res **31**(21): 6249-57.

Herzberg, O. and J. Moult (1991). "Analysis of the steric strain in the polypeptide backbone of protein molecules." Proteins: Structure, Function, and Genetics **11**: 223-229.

Hobohm, U. and C. Sander (1994). "Enlarged representative set of protein structures." Protein Science **3**: 522-524.

Hooft, R. W. W., G. Vriend, C. Sander and E. E. Abola (1996). "Errors in Protein Structures." Nature **381**: 272.

Howard, E. I., R. Sanishvili, R. E. Cachau, A. Mitschler, B. Chevrier, P. Barth, V. Lamour, M. Van Zandt, E. Sibley, C. Bon, D. Moras, T. R. Schneider, A. Joachimiak and A. Podjarny (2004). "Ultrahigh resolution drug design I: details of interactions in human aldose reductase-inhibitor complex at 0.66 A." Proteins **55**(4): 792-804.

Hu, H., M. Elstner and J. Hermans (2003). "Comparison of a QM/MM force field and molecular mechanics force fields in simulations of alanine and glycine "dipeptides" (Ace-Ala-Nme and Ace-Gly-Nme) in water in relation to the problem of modeling the unfolded peptide backbone in solution." Proteins: Structure, Function and Genetics **50**: 451-463.

Huggins, M. L. (1943). "The structure of fibrous proteins." Chem. Rev. **32**: 195-218.

Iwata, Y., A. Kasuya and S. Miyamoto (2002). "An efficient method for reconstructing protein backbones from alpha-carbon coordinates." J Mol Graph Model **21**(2): 119-28.

Jelsch, C., M. M. Teeter, V. Lamzin, V. Pichon-Lesme, R. H. Blessing and C. Lecomte (2000). "Accurate Protein Crystallography at Ultra-High Resolution: Valence-Electron Distribution in Crambin." Proceedings of the National Academy of Sciences of the United States of America **97**: 3171-3176.

Joachimiak, L. A., T. Kortemme, B. L. Stoddard and D. Baker (2006). "Computational design of a new hydrogen bond network and at least a 300-fold specificity switch at a protein-protein interface." J Mol Biol **361**(1): 195-208.

Jones, T. A., J.-Y. Zou, S. W. Cowan and M. Kjeldgaard (1991). "Improved Methods for Building Protein Models in Electron Density Maps and the Location of Errors in these Models." Acta Crystallographica, Section A **47**: 110-119.

Jorgensen, W. L. (2004). "The many roles of computation in drug discovery." Science **303**(5665): 1813-8.

Kaiser, E. T. and F. J. Kezdy (1983). "Secondary structures of proteins and peptides in amphiphilic environments. (A review)." Proc Natl Acad Sci U S A **80**(4): 1137-43.

Kang, B. S., Y. Devedjiev, U. Derewenda and Z. S. Derewenda (2004). "The PDZ2 domain of syntenin at ultra-high resolution: bridging the gap between macromolecular and small molecule crystallography." J Mol Biol **338**(3): 483-93.

Karplus, M. and J. A. McCammon (2002). "Molecular dynamics simulations of biomolecules." Nat Struct Biol **9**(9): 646-52.

Karplus, P. A. (1996). "Experimentally observed conformation-dependent geometry and hidden strain in proteins." Protein Science **5**: 1406-1420.

Kay, L. E. (2005). "NMR studies of protein structure and dynamics." J Magn Reson **173**(2): 193-207.

Kazmierkiewicz, R., A. Liwo and H. A. Scheraga (2002). "Energy-based reconstruction of a protein backbone from its alpha-carbon trace by a Monte-Carlo method." J Comput Chem **23**(7): 715-23.

Kelly, S. M. and N. C. Price (2000). "The use of circular dichroism in the investigation of protein structure and function." Curr Protein Pept Sci **1**(4): 349-84.

Kleywegt, G. J. (1999). "Experimental assessment of differences between related protein crystal structures." Acta Crystallographica Section D **55**: 1878-1884.

Kleywegt, G. J., M. R. Harris, J. Y. Zou, T. C. Taylor, A. Wahlby and T. A. Jones (2004). "The Uppsala Electron-Density Server." Acta Cryst. D **60**: 2240-2249.

Kleywegt, G. J., M. R. Harris, J. Y. Zou, T. C. Taylor, A. Wahlby and T. A. Jones (2004). "The Uppsala Electron-Density Server." Acta Crystallogr D Biol Crystallogr **60**(Pt 12 Pt 1): 2240-9.

Kleywegt, G. J. and T. A. Jones (1996). "Phi/Psi-chology: Ramachandran revisited." Structure **4**: 1395-1400.

Kleywegt, G. J. and T. A. Jones (2002). "Homo Crystallographicus - Quo Vadis?" Structure **10**: 465-472.

Ko, T. P., H. Robinson, Y. G. Gao, C. H. Cheng, A. L. DeVries and A. H. Wang (2003). "The refined crystal structure of an eel pout type III antifreeze protein RD1 at 0.62-A resolution reveals structural microheterogeneity of protein and solvation." Biophys J **84**(2 Pt 1): 1228-37.

Kortemme, T. and D. Baker (2004). "Computational design of protein-protein interactions." Curr Opin Chem Biol **8**(1): 91-7.

Kuhlman, B., G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard and D. Baker (2003). "Design of a novel globular protein fold with atomic-level accuracy." Science **302**(5649): 1364-8.

Kursula, I. and R. K. Wierenga (2003). "Crystal structure of triosephosphate isomerase complexed with 2-phosphoglycolate at 0.83-A resolution." J Biol Chem **278**(11): 9544-51.

Lakomek, N. A., C. Fares, S. Becker, T. Carlomagno, J. Meiler and C. Griesinger (2005). "Side-chain orientation and hydrogen-bonding imprint supra-tauc motion on the protein backbone of ubiquitin." Angew Chem Int Ed Engl **44**(47): 7776-8.

Laskowski, R. A., M. W. Macarthur, D. S. Moss and J. M. Thornton (1993). "ProCheck - A program to check the stereochemical quality of protein structures." Journal of Applied Crystallography **26**: 283-291.

Lazar, G. A., J. R. Desjarlais and T. M. Handel (1997). "De novo design of the hydrophobic core of ubiquitin." Protein Sci **6**(6): 1167-78.

Lesk, A. M. (1991). Protein architecture: a practical approach. New York, IRL Press.

Levinthal, C. (1969). How to Fold Graciously. Mossbauer Spectroscopy in Biological Systems, Allerton House, Monticello, Illinois, University of Illinois Press.

Lim, W. A. and R. T. Sauer (1989). "Alternative packing arrangements in the hydrophobic core of λ repressor." Nature **339**: 31-36.

Lindorff-Larsen, K., R. B. Best, M. A. Depristo, C. M. Dobson and M. Vendruscolo (2005). "Simultaneous determination of protein structure and dynamics." Nature **433**(7022): 128-32.

Lipari, G., A. Szabo and R. M. Levy (1982). "Protein dynamics and NMR relaxation: comparison of simulations with experiment." Nature **300**(5888): 197-198.

Liu, L., T. Nogi, M. Kobayashi, T. Nozawa and K. Miki (2002). "Ultrahigh-resolution structure of high-potential iron-sulfur protein from Thermochromatium tepidum." Acta Crystallogr D Biol Crystallogr **58**(Pt 7): 1085-91.

Liu, Q., Q. Huang, M. Teng, C. M. Weeks, C. Jelsch, R. Zhang and L. Niu (2003). "The crystal structure of a novel, inactive, lysine 49 PLA2 from Agkistrodon acutus venom: an ultrahigh resolution, AB initio structure determination." J Biol Chem **278**(42): 41400-8.

Looger, L. L., M. A. Dwyer, J. J. Smith and H. W. Hellinga (2003). "Computational design of receptor and sensor proteins with novel functions." Nature **423**(6936): 185-90.

Looger, L. L. and H. W. Hellinga (2001). "Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics." J Mol Biol **307**(1): 429-45.

Lovell, S. C., I. W. Davis, W. B. Arendall, III, P. I. W. d. Bakker, J. M. Word, M. G. Prisant, J. S. Richardson and D. C. Richardson (2003). "Structure Validation by Cα Geometry: ϕ,ψ and Cβ Deviation." Proteins: Structure, Function and Genetics **50**: 437-450.

Lovell, S. C., J. M. Word, J. S. Richardson and D. C. Richardson (1999). "Asparagine and Glutamine Rotamers: *B*-Factor Cutoff and Correction of Amide Flips Yield Distinct Clustering." Proceedings of the National Academy of Sciences of the United States of America **96**: 400-405.

Lovell, S. C., J. M. Word, J. S. Richardson and D. C. Richardson (2000). "The penultimate rotamer library." Proteins: Structure, Function, and Genetics **40**: 389-408.

Mandel, N., G. Mandel, B. L. Trus, J. Rosenberg, G. Carlson and R. E. Dickerson (1977). "Tuna cytochrome *c* at 2.0 Å Resolution." Journal of Biological Chemistry **252**: 4619-4635.

Martin, A. C., M. W. MacArthur and J. M. Thornton (1997). "Assessment of comparative modeling in CASP2." Proteins **Suppl 1**: 14-28.

Matthews, B. W. (1972). "The γ Turn. Evidence for a new folded conformation in proteins." Macromolecules **5**: 818-819.

Matthews, B. W. (1995). "Studies on protein stability with T4 lysozyme." Advances in Protein Chemistry **46**: 249-278.

McRee, D. E. (1999). "XtalView/Xfit - A Versatile Program for Manipulating Atomic Coordinates and Electron Density." Journal of Structural Biology **125**(2-3): 156-165.

Milner-White, E. J. (1990). "Situations of Gamma-Turns in Proteins: their relation to alpha-helices, beta sheets and ligand binding sites." J. Mol. Biol. **216**: 385-397.

Mooers, B. H., D. Datta, W. A. Baase, E. S. Zollars, S. L. Mayo and B. W. Matthews (2003). "Repacking the Core of T4 lysozyme by automated design." Journal of Molecular Biology **332**(3): 741-56.

Morris, A. L., M. W. MacArthur, E. G. Hutchinson and J. M. Thornton (1992). "Stereochemical quality of protein structure coordinates." Proteins: Structure, Function, and Genetics **12**: 345-364.

Moser, R., R. M. Thomas and B. Gutte (1983). "An artificial crystalline DDT-binding polypeptide." FEBS Letters **157**(2): 247-251.

Munson, M., R. O'Brien, J. M. Sturtevant and L. Regan (1994). "Redesigning the hydrophobic core of a four-helix-bundle protein." Protein Science **3**: 2015-2022.

Murray, L. J., W. B. Arendall, III, D. C. Richardson and J. S. Richardson (2003). "RNA backbone is rotameric." Proc Natl Acad Sci U S A **100**(24): 13904-9.

Murray, L. J., D. C. Richardson and J. S. Richardson (2006). Unpublished work.

Murray, L. J. and J. S. Richardson (2006). Personal communication.

Murray, L. J., J. S. Richardson, W. B. Arendall and D. C. Richardson (2005). "RNA backbone rotamers--finding your way in seven dimensions." Biochem Soc Trans **33**(Pt 3): 485-7.

Murshudov, G. N., A. I. Grebenko, J. A. Brannigan, A. A. Antson, V. V. Barynin, G. G. Dodson, Z. Dauter, K. S. Wilson and W. R. Melik-Adamyan (2002). "The structures of Micrococcus lysodeikticus catalase, its ferryl intermediate (compound II) and NADPH complex." Acta Crystallogr D Biol Crystallogr **58**(Pt 12): 1972-82.

Némethy, G. and M. P. Printz (1972). "The γ Turn, a possible folded conformation of the polypeptide chain. Comparison with the β turn." Macromolecules **5**: 755-758.

Noonan, K., D. O'Brien and J. Snoeyink (2004). Probik: Protein Backbone Motion by Inverse Kinematics. Workshop on the Algorithmic Foudations of Robotics, Springer Verlag.

Nukaga, M., K. Mayama, A. M. Hujer, R. A. Bonomo and J. R. Knox (2003). "Ultrahigh resolution structure of a class A beta-lactamase: on the mechanism and specificity of the extended-spectrum SHV-2 enzyme." J Mol Biol **328**(1): 289-301.

Ohage, E. C., W. Graml, M. M. Walter, S. Steinbacher and B. Steipe (1997). "β-Turn propensities as paradigms for the analysis of structural motifs to engineer protein stability." Protein Science **6**: 233-241.

Pal, D. and P. Chakrabarti (2002). "On Residues in the Disallowed Region of the Ramachandran Map." Biopolymers **63**: 195-206.

Palmer, I., A.G. (2004). "NMR characterization of the dynamics of biomacromolecules." Chem. Rev. **104**(8): 3623-3640.

Palmo, K., B. Mannfors, N. G. Mirkin and S. Krimm (2003). "Potential energy functions: from consistent force fields to spectroscopically determined polarizable force fields." Biopolymers **68**(3): 383-94.

Pantoliano, M. W., E. C. Petrella, J. D. Kwasnoski, V. S. Lobanov, J. Myslik, E. Graf, T. Carver, E. Asel, B. A. Springer, P. Lane and F. R. Salemme (2001). "High-density miniaturized thermal shift assays as a general strategy for drug discovery." J Biomol Screen **6**(6): 429-40.

Park, S., X. Yang and J. G. Saven (2004). "Advances in computational protein design." Curr Opin Struct Biol **14**(4): 487-94.

Patel, S., M. Martinez-Ripoll, T. L. Blundell and A. Albert (2002). "Structural enzymology of Li(+)-sensitive/Mg(2+)-dependent phosphatases." J. Mol. Biol. **320**: 1087-1094.

Patterson, W. R., D. H. Anderson, W. F. DeGrado, D. Cascio and D. Eisenberg (1999). "Centrosymmetric bilayers in the 0.75 A resolution structure of a designed alpha-helical peptide, D,L-Alpha-1." Protein Science **8**(7): 1410-22.

Peters, D. and J. Peters (1981). "Quantum Theory of the Structure and Bonding in Proteins. Part 8. The alanine dipeptide." Journal of Molecular Structure **85 - Theochem**: 107-123.

Ponder, J. W. and F. M. Richards (1987). "Tertiary Templates for Proteins: Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes." Journal of Molecular Biology **193**: 775-791.

Qian, B., A. R. Ortiz and D. Baker (2004). "Improvement of comparative model accuracy by free-energy optimization along principal components of natural structural variation." Proc Natl Acad Sci U S A **101**(43): 15346-51.

Ramachandran, G. N., C. Ramakrishnan and V. Sasisekharan (1963). "Stereochemistry of Polypeptide Chain Configurations." Journal of Molecular Biology **7**: 95-99.

Ramachandran, G. N. and V. Sasisekharan (1968). "Conformation of polypeptides and proteins." Adv Protein Chem **23**: 284-438.

Ramakrishnan, C. and G. N. Ramachandran (1965). "Stereochemical criteria for polypeptide and protein chain conformations. II. Allowed conformations for a pair of peptide units." Biophysical Journal **5**: 909-933.

Richards, F. M. and C. E. Kundrot (1988). "Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure." Proteins **3**(2): 71-84.

Richardson, D. C. and J. S. Richardson (1989). Principles and Patterns of Protein Conformation. Prediction of Protein Structure and the Principles of Protein Conformation. G. D. Fasman. New York, Plenum Press**:** 1-98.

Richardson, J. S. (1981). The Anatomy and Taxonomy of Protein Structure. Advances in Protein Chemistry. C. B. Anfinsen, J. T. Edsall and F. M. Richards. New York, Academic Press. **34:** 167-339.

Richardson, J. S. (2003). All-atom contacts: A new approach to structure validation. Structural Bioinformatics. P. E. Bourne and H. Weissig. New York, John Wiley & Sons, Inc.**:** 305-320.

Richardson, J. S., W. B. Arendall, III and D. C. Richardson (2003). New Tools and Data for Improving Structures, Using All-atom Contacts. Methods in Enzymology, vol. 374. C. W. Carter, Jr. and R. M. Sweet. New York, Academic Press. **374:** 385-412.

Richardson, J. S. and D. C. Richardson (1987). Some Design Principles: Betabellin. Protein Engineering. D. L. Oxender and C. F. Fox. New York, Alan R. Liss, Inc.**:** 149-163-and 340-1.

Richardson, J. S. and D. C. Richardson (1988). "Amino Acid Preferences for Specific Locations at the Ends of α-Helices." Science **240**: 1648-1652.

Richardson, J. S. and D. C. Richardson (1989). "The de novo design of protein structures." Trends Biochem Sci **14**(7): 304-9.

Richardson, J. S. and D. C. Richardson (2001). "MAGE, PROBE, and Kinemages", chapter 25.2.8. International Tables for Crystallography. M. G. Rossmann and E. Arnold. Dordrecht, Kluwer Academic Publishers, The Netherlands. **Vol F: "Crystallography of Biological Macromolecules":** 727-730.

Richardson, J. S., D. C. Richardson, N. B. Tweedy, K. M. Gernert, T. P. Quinn, M. H. Hecht, B. W. Erickson, Y. Yan, R. D. McClain, M. E. Donlan and M. C. Surles (1992). "Looking at proteins: representations, folding, packing, and design." Biophysical Journal **63**: 1186-1209.

Rohl, C. A., C. E. Strauss, K. M. Misura and D. Baker (2004). "Protein structure prediction using Rosetta." Methods Enzymol **383**: 66-93.

Rose, G. D., L. M. Gierasch and J. A. Smith (1985). "Turns in peptides and proteins." Advances in Protein Chemistry **37**: 1-109.

Roterman, I. K., M. H. Lambert, K. D. Gibson and H. A. Scheraga (1989). "A Comparison of the CHARMM, AMBER and ECEPP Potentials for Peptides. II. ϕ - ψ Maps for N-Acetyl Alanine N'-Methyl Amide: Comparisons, Contrasts and Simple Experimental Tests." Journal of Biomolecular Structure & Dynamics **7**: 421-453.

Schäfer, L., S. Q. Newton, M. Cao, A. Peeters, C. V. Alsenoy, K. Wolinski and F. A. Momany (1993). "Evaluation of the dipeptide approximation in peptide modeling by ab initio geometry optimizations of oligopeptides." J. Am. Chem. Soc. **115**: 272-280.

Schmidt, A., C. Jelsch, P. Ostergaard, W. Rypniewski and V. S. Lamzin (2003). "Trypsin revisited: crystallography AT (SUB) atomic resolution and quantum chemistry revealing details of catalysis." J Biol Chem **278**(44): 43357-62.

Schneider, T. D. and R. M. Stephens (1990). "Sequence logos: a new way to display consensus sequences." Nucleic Acids Res **18**(20): 6097-100.

Schneider, T. R. and G. M. Sheldrick (2002). "Substructure solution with SHELXD." Acta Crystallogr D Biol Crystallogr **58**(Pt 10 Pt 2): 1772-9.

Shapiro, J. and D. Brutlag (2004). "FoldMiner and LOCK 2: protein structure comparison and motif discovery on the web." Nucleic Acids Res **32**(Web Server issue): W536-41.

Sibanda, B. L. and J. M. Thornton (1985). "Beta-hairpin families in globular proteins." Nature **316**: 170-174.

Silverman, B. W. (1986). Density Estimation for Statistics and Data Analysis. London, Chapman and Hall.

Stites, W. E., A. K. Meeker and D. Shortle (1994). "Evidence for strained interactions between side-chains and the polypeptide backbone." J. Mol. Biol. **235**: 27-32.

Studier, F. W. (2005). "Protein production by auto-induction in high density shaking cultures." Protein Expr Purif **41**(1): 207-34.

Talbot, J. A. and R. S. Hodges (1981). "Comparative studies on the inhibitory region of selected species of troponin-I. The use of synthetic peptide analogs to probe structure- function relationships." J. Biol. Chem. **256**(23): 12374-12378.

Terwilliger, T. (2006). Personal communication. Diffraction Methods Gordon Conference, Bates College, Maine.

Tramontano, A., R. Leplae and V. Morea (2001). "Analysis and assessment of comparative modeling predictions in CASP4." Proteins **Suppl 5**: 22-38.

Tramontano, A. and V. Morea (2003). "Assessment of homology-based predictions in CASP5." Proteins **53 Suppl 6**: 352-68.

Tufte, E. (1990). Envisioning Information. Cheshire, Connecticut, Graphics Press.

Tufte, E. (1997). Visual Explanations. Cheshire, Connecticut, Graphics Press.

Tufte, E. (2001). The Visual Display of Quanitative Information. Cheshire, Connecticut, Graphics Press.

Uppenberg, J., M. T. Hansen, S. Patkar and T. A. Jones (1994). "The sequence, crystal structure determination and refinement of two crystal forms of lipase B from *Candida antarctica*." Structure **2**: 293-308.

Vaguine, A. A., J. Richelle and S. J. Wodak (1999). "SFCHECK: a unified set of procedures for evaluating the quality of macromolecular structure-factor data and their agreement with the atomic model." Acta Cryst. D **55**: 191-205.

van den Bedem, H., I. Lotan, J. C. Latombe and A. M. Deacon (2005). "Real-space protein-model completion: an inverse-kinematics approach." Acta Crystallogr D Biol Crystallogr **61**(Pt 1): 2-13.

Van Kerm, P. (2003). Adaptive kernel density estimation. 9th UK Stat Users meeting, London, Royal Statistical Society.

vandenAkker, F. and W. G. J. Hol (1999). "Difference density quality (DDQ): a method to assess the global and local correctness of macromolecular crystal structures." Acta Crystallographica, Section D **55**: 206-218.

Venclovas, C., A. Zemla, K. Fidelis and J. Moult (1997). "Criteria for evaluating protein structures derived from comparative modeling." Proteins **Suppl 1**: 7-13.

Vriend, G. (1990). "WHAT IF: A molecular modeling and drug design program." Journal of Molecular Graphics **8**(1): 52-56.

Walther, D. and F. E. Cohen (1999). "Conformational attractors on the Ramachandran map." Acta Crystallographica Section D **D55**: 506-517.

Wang, T., S. Cai and E. R. Zuiderweg (2003). "Temperature dependence of anisotropic protein backbone dynamics." Journal of the American Chemical Society **125**(28): 8639-43.

Wang, W. and M. H. Hecht (2002). "Rationally designed mutations convert *de novo* amyloid-like fibrils into monomeric β-sheet proteins." Proceedings of the National Academy of Sciences USA **99**: 2760-2765.

Ware, C. (2000). Information visualization: design for perception. San Francisco, CA, Morgan Kaufmann Publishers.

Word, J. M. (2000). All-Atom Small-Probe Contact Surface Analysis: an information-rich description of molecular goodness-of-fit. Department of Biochemistry. Durham, NC, Duke University: 274.

Word, J. M., R. C. Bateman, Jr., B. K. Presley, S. C. Lovell and D. C. Richardson (2000). "Exploring Steric Constraints on Protein Mutations using Mage/Probe." Protein Science **9**: 2251-2259.

Word, J. M., S. C. Lovell, T. H. LaBean, H. C. Taylor, M. E. Zalis, B. K. Presley, J. S. Richardson and D. C. Richardson (1999a). "Visualizing and Quantifying Molecular Goodness-of-Fit: Small-Probe Contact Dots with Explicit Hydrogens." Journal of Molecular Biology **285**(4): 1711-1733.

Word, J. M., S. C. Lovell, J. S. Richardson and D. C. Richardson (1999b). "Asparagine and Glutamine: Using Hydrogen Atom Contacts in the Choice of Side-chain Amide Orientation." Journal of Molecular Biology **285**: 1735-1747.

Wuthrich, K. (1986). NMR of Proteins and Nucleic Acids. New York, Wiley.

Biography

Ian Wheeler Davis was born August 31, 1978 in Charlotte, North Carolina, to Jennifer Wheeler Davis and John Mahan Davis. He has one brother, Addison Reid Davis.

He made the decision to become a scientist at age 4 while attending Trinity Presbyterian Preschool; he wrote his first computer program in BASIC on a Commodore Vic 20 with 4 kilobytes of RAM at age 6 under the guidance of his father. He graduated as valedictorian from Sun Valley High School in Monroe, North Carolina in 1997. He attended Vanderbilt University in Nashville, Tennessee on a full tuition Harold Stirling Vanderbilt scholarship, where he researched the structure of TGF-β receptors by NMR with Dr. Andrzej Krezel. He graduated from Vanderbilt summa cum laude with a B.A. in Molecular Biology in 2001.

He married Katherine (Katy) Leigh Bitler on July 27, 2002 in Houston, Texas.

He was awarded a Howard Hughes Medical Institute Predoctoral Fellowship to support his graduate work, as well as a National Science Foundation Predoctoral Fellowship that he was unable to accept concurrently.

From Duke University, he received a University Scholars fellowship for interdisciplinary studies and a James B. Duke fellowship. He was also awarded scholarships to the Keystone Structure-Based Drug Design and Gordon Diffraction Methods conferences in 2006. He has published four papers in scholarly journals: "Structure Validation by Cα Geometry: ϕ,ψ and Cβ Deviation" (Proteins, 2003), "MolProbity: structure validation and all-atom contact analysis for nucleic acids and their complexes" (Nucleic Acids Research, 2004), "A test of enhancing model accuracy in high-throughput crystallography" (Journal of Structural & Functional Genomics, 2005), and "The Backrub Motion: How Protein Backbone Shrugs When a Sidechain Dances" (Structure, 2006).

Ian currently lives in Durham, North Carolina with his wife Katy, his dog Rip, and his guinea pig Beulah.