Resampling Methods for Protein Structure Prediction

Benjamin Norman Blum

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-184
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-184.html

December 22, 2008

Resampling Methods for Protein Structure Prediction

by

Benjamin Norman Blum

B.S. (Stanford University) 2003

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Michael I. Jordan, Chair
Professor Sandrine Dudoit
Professor Yun S. Song

Fall 2008

The dissertation of Benjamin Norman Blum is approved:

Chair

Date

Date

Date

University of California, Berkeley

Fall 2008

ABSTRACT

Resampling Methods for Protein Structure Prediction

by

Benjamin Norman Blum

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Michael I. Jordan, Chair

Ab initio protein structure prediction entails predicting the three-dimensional conformation of a protein from its amino acid sequence without the use of an experimentally determined template structure. In this thesis, I present a new approach to ab initio protein structure prediction that divides the search problem into two parts: sampling in a space of discrete-valued structural features, and continuous search over conformations while constraining the desired features. Both parts are carried out using Rosetta, a leading structure prediction algorithm. Rosetta is a Monte Carlo energy minimization method requiring many random restarts to find structures near the correct, or *native* structure. Our methods, which we call *resampling* methods, make use of an initial round of Rosetta-generated local minima to learn properties of the energy landscape that guide a subsequent "resampling" round of Rosetta search toward better predictions. One of the main innovations of this thesis is to attempt to deduce from the initial set of Rosetta models not the entire native conformation but rather a few specific *features* of the native conformation. Features include backbone torsion angles, per-residue secondary structure, exposure of residues to solvent, and a three-tiered hierarchy of beta pairing features. For each feature there is one "native" value: the one found in the native structure. Native feature values are generally enriched in structures with low energy, as the native structure of a protein is significantly lower in energy than non-native structures and the energy of a protein is to some extent the sum of spatially local contributions. We have developed two methods for feature-space

resampling based on this observation. The first method employs feature selection methods to identify structural feature values that give rise to low energy, which are then enriched in the resampling round. The second, more sophisticated method updates the sampling distribution for all features at once, not just a selected few, by predicting the likelihood that each feature value is native. Our results indicate that both methods, especially the second one, yield structure predictions significantly better than those produced by Rosetta alone.

DEDICATION

To the memory of David Foster Wallace.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Michael Jordan, for the most fulfilling experience I've ever had or could ever hope to have of computer science research. I will forever be in his debt for his support and sage advice as I confronted the big questions about my career, and for the astonishing faith he demonstrated in sending an untested graduate student like myself to forge a relationship with a lab in a field entirely new to him. I would also like to thank David Baker, the guiding force behind that lab, for graciously hosting me for a summer in Seattle that somehow grew into two long, wonderful years. His enthusiasm was a constant spur to my own excitement about the work. Our day-to-day exchange of ideas proved the most exciting and productive intellectual relationship I've yet experienced in the scientific realm.

I would also like to thank the members of my dissertation committee, Sandrine Dudoit and Yun Song, for their insights and assistance, and all my friends and collaborators from the Baker lab and from Michael Jordan's group at Berkeley, which include Rhiju Das, David Kim, Phil Bradley, Bin Qian, James Thompson, Will Sheffield, Percy Liang, Guillaume Obozinski, Ben Taskar, and many, many others.

I would like to thank the National Science Foundation for their generous financial support throughout my graduate career.

Finally, I would like to thank my family—Al, Jude, Dan, Beth, and Leah—for their love and support, and all my friends in Seattle, San Francisco, Berkeley, New York, and elsewhere for not being too weirded out when they found out I was a secret scientist.

Contents

Chapter 1

Introduction

1.1 Proteins

Proteins are biological macromolecules that perform essential functions in all living organisms. They are composed of amino acid residues joined together by peptide bonds into long polypeptide chains. There are twenty naturally occurring varieties of amino acid that appear in proteins, each defined by a chemically unique *side chain*. Their precise sequence in a protein is encoded by the sequence of DNA base pairs in that protein's gene. This amino acid sequence is known as a protein's *primary structure*.

Proteins also have structure at other levels of resolution. Contiguous regions of the amino acid sequence form two main varieties of *secondary structure*, characterized by regular hydrogen bond patterns: *alpha helices* and *beta strands*. Multiple beta strands (possibly distant in sequence) bind together to form *beta pleated sheets*. Beta strands bind together in two orientations: anti-parallel and parallel. Occasionally one or more residues in one strand do not form hydrogen bonds to any residues on the opposite strand; such occurrences are known as *beta bulges*.

At the global level, the *tertiary structure* of a protein—its three-dimensional conformation—is formed by packing secondary structure elements together into one or more globular *domains*. During the folding process, the protein searches through its degrees of freedom for lower energy states. Each residue in an amino acid sequence has two primary degrees of freedom: rotation around the C_α–N bond, referred to as the phi torsion an-

gle, and rotation around the C_α–C bond, referred to as the psi torsion angle. The primary driving force behind the folding process is hydophobic burial—it is energetically favorable for polar side chains to be exposed to solvent and hydrophobic side chains to be buried in the protein's hydrophobic core. The protein backbone is itself highly polar, but within secondary structure elements all hydrogen bond donors and acceptors on the backbone are satisfied, so helices and sheets can pass through the core of the protein without incurring an energetic penalty.

Some proteins are composed of multiple polypeptide chains; the arrangement of these chains with respect to one another comprises the protein's *quaternary structure*.

For the reader interested in a very thorough and accessible introduction to protein structure and function, [Branden and Tooze, 1999] is an excellent reference.

The recent explosion in available genome data has brought with it an explosion in the number of known amino acid sequences of proteins. It has not, however, illuminated the precise *function* of these proteins. Secondary structure can be predicted with fairly high accuracy from sequence information, but it is the tertiary (and quaternary, if applicable) structure of a protein that most directly determines its biological function. Enzymes, for instance, which catalyze specific chemical reactions, depend on very precise catalytic geometry of an *active site* to bind to one or more *substrates*. Neither the location of this active site nor its geometry can be determined reliably without knowing the tertiary structure of the enzyme. Thus, in order to reap the full rewards from the new wealth of genome data, we must know the tertiary structure of the proteins that genes encode.

Unfortunately, the tertiary structure of a protein is quite challenging to determine. Experimental methods currently in use, including nuclear magnetic resonance spectroscopy [Wüthrich, 1990] and the higher resolution x-ray diffraction method [Kendrew *et al.*, 1958], are time- and resource-intensive. As a result, the number of known protein sequences now far outstrips the capacity of experimentalists to determine their structures. Fewer than 50,000 proteins have (at time of writing) had their structures experimentally determined [RCSB, 2008], out of a pool of about 1,000,000 known amino acid sequences [UniProt

Consortium, 2008]. In nature, the amino acid sequence of a protein uniquely (to a good approximation) determines the conformation it will fold into [Anfinsen, 1973]. If it were possible to predict this tertiary structure from primary structure using computational means, the impact on the current state of biological knowledge would be enormous. This is the protein structure prediction problem.

1.2 Protein structure prediction

Protein structure prediction has progressed mightily in the past thirty years. Although computational methods are not yet nearly as reliable as experimental methods, predicted structures are in some cases very close to the resolution of experimentally determined structures. Progress in the field has been particularly easy to measure since the establishment of the biannual meeting on Critical Assessment of techniques for protein Structure Prediction (CASP) [Moult *et al.*, 1995], a blind structure prediction benchmark in which essentially all leading researchers in the field participate. Every two years, a pool of proteins for which structures have been determined but not yet released are presented as challenges to the computational groups. Afterwards, the predictions are compared with the true structures and the various methods are assessed against one another.

Protein structure prediction is a wide and varied field, but historically algorithms have been subdivided into three primary categories: homology modeling, fold recognition, and ab initio modeling. In homology modeling, a target protein is modeled using a template protein with experimentally determined structure. The template, or "homolog," is identified by sequence similarity to the target. If no such sequence homolog exists, it may still be the case that the target protein adopts a fold similar to one in the database of solved structures; in this case, a suitable template might be identified using a fold recognition algorithm. These methods are also referred to as "threading" algorithms, since testing the match to the template typically involves threading the target sequence through the structure of the template and evaluating some simplified physical energy potential. The final category, ab initio modeling, refers to structure prediction in the absence of any structural template, and gen-

erally entails searching through conformation space for the global free energy minimum, as captured by some kind of energy function. This categorization of prediction methods was reflected in the categories of CASP competition up until CASP6; however, starting with CASP7 [Moult *et al.*, 2007], homology modeling and fold recognition have been joined into a single template-based modeling (also called comparative modeling) category, with the easiest targets (those having very high sequence similarity to their templates) placed in a "high-accuracy modeling" category. This reflects a shift in thinking within the field—homology modeling and fold recognition methods differ only in the distance of structural homologs that they can detect, and the primary distinction is between template-based modeling and ab initio modeling, with the former category accounting for about 85% of CASP7 targets [Moult *et al.*, 2007].

Applications of structure prediction are numerous, and depend on the accuracy of the prediction. At the atomic level of resolution—models within 1Å–1.5Å of the native conformation—the precise catalytic geometry of the active site of enzymes is in place, so catalytic mechanisms can be inferred. Protein-protein docking can be performed, and potential ligands can be screened automatically [Xu *et al.*, 1996]. This level of resolution is currently only reliably achievable by comparative modeling using close sequence homologs [Baker and Šali, 2001]. At the coarser resolutions currently attainable by ab initio methods, predictions can be used for molecular replacement in X-ray crystallography and hence to produce high-resolution structures [Qian *et al.*, 2007], or to identify likely active sites or functional relationships to proteins with similar structure.

1.2.1 Template-based modeling

Modeling based on templates predates the computational age; the first model derived from a template was built by hand [Browne *et al.*, 1969]. Comparative methods were among the first computational techniques for protein structure prediction [Blundell *et al.*, 1987]. Early influential methods include assembling large fragments of aligned structure from multiple templates [Levitt, 1992] and satisfying inter-residue distance constraints derived

from templates [Šali and Blundell, 1993].

Template-based modeling has four basic steps: identification of the templates, alignment of the target to the templates, building the model, and assessing the model [Martí-Renom *et al.*, 2000]. The historical distinction between homology modeling and fold recognition lies primarily in the manner in which the first two steps are carried out. In homology modeling, templates are found using simple sequence-sequence matching via BLAST or other methods [Altschul *et al.*, 1990; Jones *et al.*, 1992; Vingron and Waterman, 1994] or sequence-profile matching via PsiBLAST [Altschul *et al.*, 1997]. Many sophisticated methods exist for finding templates much more distant in sequence, including profile-profile matching [Godzik, 2003; Jaroszewski *et al.*, 2000a; 2000b], Hidden Markov Models [Karplus *et al.*, 1998], threading the target onto the proposed template structure [Jones, 1999; David *et al.*, 2000; Skolnick *et al.*, 2004; Zhou and Zhou, 2005], and "meta-server" predictions combining all of the above [Wallner *et al.*, 2003; Fischer, 2003; Ginalski *et al.*, 2003]. Both optimal alignment to the template and refinement of the model once it has been built from the template remain large unsolved problems. CASP7 was the first occasion in which the majority of submitted predictions for each target were better than the best experimental template with a perfect alignment but no refinement [Moult *et al.*, 2007]. Leading methods for refinement include minimizing an all-atom forcefield [Misura *et al.*, 2004] and assembling large fragments from multiple templates [Zhang, 2007]. Once the backbone of the model has been built, specific methods exist for modeling loops between secondary structure elements [Fiser *et al.*, 2000] and placing side chains [Bower *et al.*, 1997], although these steps are often embedded within the model-building and refinement steps in comparative modeling algorithms.

1.2.2 Ab initio modeling

Ab initio modeling starts with the assumption that the native conformation is the global free energy minimum [Anfinsen *et al.*, 1961; Anfinsen, 1973], although there are in fact important exceptions to this rule [Baker and Agard, 1994]. In theory, then, the native

conformation can be found by energy minimization in conformation space without recourse to a structural template. The idea is appealing—an accurate ab initio structure prediction method would be wholly general and hence would make template-based modeling methods unnecessary. However, in practice the ab initio modeling problem is much harder than the comparative modeling problem and current methods do not approach the accuracy of template-based methods. Conformation space is very high-dimensional and the energy landscape is riddled with local minima.

Important research in this area concentrates both on improving the accuracy of energy functions and on simplifying the search space via discretization or reduced representations of structure. Energy potentials fall into two categories: physical terms, including electrostatic, solvation, and van der Waals interactions, and statistical terms derived from the set of experimentally determined protein structures [Sippl, 1995; Koppensteiner and Sippl, 1998]. Interactions between protein and solvent are typically captured using an implicit solvent model rather than with explicit solvent molecules. The most sophisticated energy functions now include a mix of statistical and physical potentials, and appear increasingly capable of discerning the native conformation from other conformations [Vorobjev *et al.*, 1998; Lazaridis and Karplus, 1999; Rapp and Friesner, 1999; Petrey and Honig, 2000; Lee *et al.*, 2001].

The choice of the conformation space in which to search is a crucial one. If it is too reduced, the native structure might not be contained within it, and the closest point to the native in the reduced space might not be discernible as near-native by the energy function. Search-space reduction is generally necessary, however, because the full conformation space is too large to search effectively. Backbone torsion angles can be limited to a discrete set of commonly observed values [Park and Levitt, 1995] or drawn from fragments of true protein structure from proteins in the database of experimentally determined structures [Sippl *et al.*, 1992; Bowie and Eisenberg, 1994; Jones, 1997; Simons *et al.*, 1997]. Side chains in nature typically assume conformations from a discrete pool of rotamers, so search over side chain conformations can be discretized as well [R. L. Dunbrack and

Karplus, 1994]. An even greater simplification can be achieved by abstracting side chains as super-atoms located at the centroids of the side chains or at the beta carbon, with statistical interaction potentials between side chains that average out internal degrees of freedom [Simons *et al.*, 1997].

Search itself is usually carried out using some kind of Monte Carlo procedure, including simulated annealing [Simons *et al.*, 1997] and genetic algorithms [Pedersen and Moult, 1995]. Many current methods produce, rather than a single local minimum of the energy function, a pool of candidates resulting from numerous search trajectories. A single prediction is then chosen from this pool by one of a variety of methods [Park and Levitt, 1996; Huang *et al.*, 1996; Samudrala and Moult, 1998].

Our work is built upon Rosetta [Simons *et al.*, 1997], a particularly successful ab initio modeling algorithm which generated a great deal of excitement for the promise of ab initio methods when it significantly advanced the field in CASP4 [Bonneau *et al.*, 2001]. Since then, progress in Rosetta has been only incremental, although no other methods have convincingly overtaken it [Moult *et al.*, 2007]. We discuss Rosetta in some detail in the next section.

1.3 Rosetta

Rosetta is one of the leading methods for ab initio protein structure prediction today. Rosetta uses a Monte Carlo search procedure to minimize an energy function that is sufficiently accurate that the conformation found in nature (the "native" conformation) is often the conformation with lowest energy.

Each Rosetta search trajectory proceeds through two stages: an initial low-resolution search stage in which side chains are represented as centroids without internal degrees of freedom, followed by a high-resolution refinement stage in which all atoms are placed and the energy function is closer to the true physical energy. Although the low-resolution energy function is not physically realistic and cannot generally distinguish the native conformation from Rosetta local minima, the global conformation largely comes together in

the low-resolution stage. The high-resolution stage alters the global conformation in minor ways, largely to accommodate the placement of side chains. The output of the low-resolution stage can be regarded as a proposed backbone on which to place side chains; the refinement stage evaluates the proposal, subjecting it to minor modification along the way.

One might suggest using the all-atom energy function throughout search. The low-resolution model cannot, however, be generated using the all-atom energy function, for two main reasons: first, there are too many degrees of freedom when all atoms are included, so search in this space is very slow; second, the high-resolution energy function is much rougher and prone to energetic traps. In order to allow the folding protein to arrive at the final folded state, some degree of flailing is required, and the all-atom energy function, with its strict adherence to physical laws, is not sympathetic to flailing. The native conformation is generally lower in all-atom energy than Rosetta local minima that have gone through the high-resolution refinement stage.

In the low-resolution stage, the primary search move is a *fragment replacement* move, in which a sequence of contiguous residues—either three or nine, although in principle any size would work—have their backbone torsion angles replaced with angles drawn from a fragment of protein structure in the PDB (the database of experimentally determined protein structures). This is the key innovation that enables Rosetta's success. Rather than search over individual torsion angles, the conformation can jump between locally viable structures. For a new target on which Rosetta is to be run, a *fragment pool* is generated ahead of time. This pool contains, for every *frame* of three or nine residues in the protein, a set of 200 fragments drawn from the PDB. In a protein of length n residues, there are $n - k + 1$ frames of length k, for a total of $(n - k + 1) * 200$ fragments of length k in the pool. The fragments are chosen by sequence similarity to the target protein's sequence within the frame, and by matching the predicted secondary structure of the target to the actual secondary structure of the fragment in the protein from which it derives. The same fragment pool is used for every search trajectory, with move proposals drawn out of it at random. After a fragment replacement, local minimization is performed and then the move

is accepted or rejected based on a Metropolis-style energy criterion.

Finding the global minimum of the energy function is very difficult because of the high dimensionality of the search space and the very large number of local minima. Rosetta employs a number of strategies to combat these issues, but the primary one is to perform a large number of random restarts. Thanks to a very large-scale distributed computing platform called Rosetta@home, composed of more than four hundred thousand volunteer computers around the world, up to several million local minima of the energy function (we will call them "models") can be computed for each target sequence. Computational costs for Rosetta are high. Each Rosetta model takes approximately fifteen minutes of CPU time to compute on a 1GHz CPU, and a typical data set for a single target consists of on the order of $100,000$ models.

1.4 Resampling

The fundamental insight behind *resampling* methods, which are the focus of the remainder of this thesis, is that a random-restart strategy throws away a great deal of information from previously computed local minima. In particular, previous samples from conformation space might suggest regions of uniformly lower energy; these are regions in which we might wish to concentrate further sampling. This intuition leads to a class of methods that we call *structure-based* resampling methods. We discuss past work in this area in Section 1.4.1, and illustrate some of the drawbacks of this approach with two of our own attempts at structure-based resampling methods. These efforts motivate the innovation that constitutes the main original contribution of this thesis: shifting search into a discrete-valued structural feature space, and identifying native-like *features* rather than attempting to identify native-like *structures*. In Section 1.4.2 we discuss the limited past work that has been performed in feature-based resampling—primarily genetic algorithms, which differ significantly from our methods—and introduce our own algorithms in this area.

1.4.1 Structure-based resampling

Structure-based resampling methods work by identifying either regions of conformation space or individual structures from the initial sampling round that show promise, and concentrating further search around them. The fundamental drawback of methods in this class is that they are limited to enrichment of regions of conformation space which have already been explored, whereas the native conformation will not generally lie within these regions.

In "conformation space annealing" [Lee *et al.*, 1997], a pool of random starting structures is gradually refined by local search, with low energy structures giving rise to children that eventually replace the higher energy starting structures. While the method does prove successful in some cases, it is limited to local exploration of areas of conformation space already sampled. New areas of conformation space are explored by the introduction of new random seed structures, but for larger proteins, the chance of a random structure being close enough to the native structure to give rise to near-native descendants may be vanishingly small. In [Brunette and Brock, 2005], a Rosetta-based resampling method is presented that operates by identifying "funnels" in conformation space and concentrating sampling on the low-energy funnels. Funnels are discovered by means of unconstrained conformational search, so this method too entails enrichment of regions already seen. On targets for which Rosetta produces occasional successes, their method significantly improves sampling of near-native conformations; however, it is not effective on proteins for which Rosetta produces no native-like structures.

Similar resampling strategies have been developed for general-purpose global optimization. These include fitting a smoothed *response surface* to the local minima already gathered [Box and Wilson, 1951] and using statistical methods to identify good starting points for optimization [Boyan and Moore, 2001]. Unfortunately, as we shall see in our own efforts in the next section, conformation space is very high-dimensional and very irregular, so response surfaces do not generalize well beyond the span of the points to which they are fitted. Generally, the correct (or "native") structure will not be in the span of the points seen so far—if it were, the first round of Rosetta sampling would already have been successful.

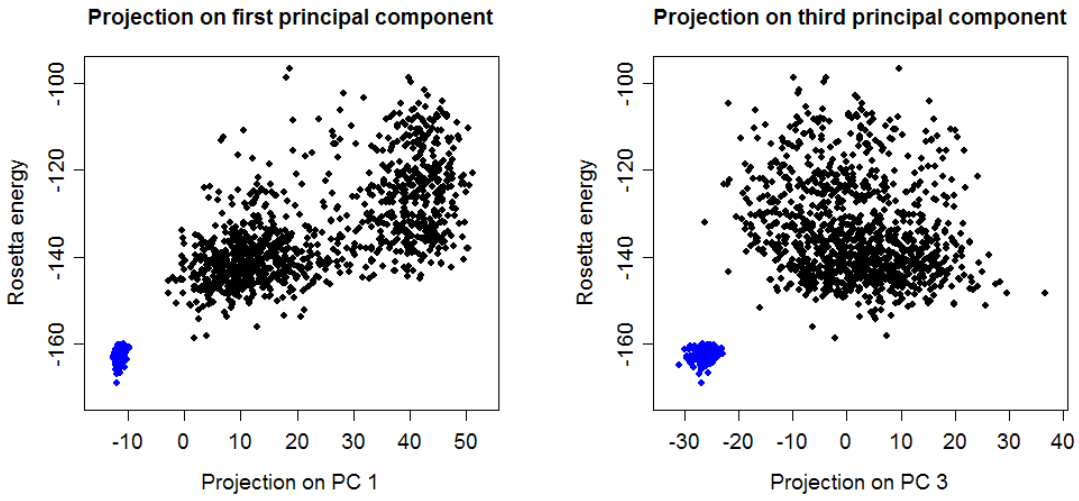
The method of [Boyan and Moore, 2001] is intriguing as a precursor to feature-based resampling, since it entails the careful design of features that identify good starting points for search. Search is divided into two stages: first, search in this feature space to identify a good starting point for further optimization, and, second, the optimization itself. However, the feature space response surface is fitted using points already seen, so, as in the case of response-surface fitting, does not necessarily generalize past the span of these points.

Response surface fitting

As an initial attempt at developing resampling methods for protein structure prediction, we investigated a response surface fitting approach. Our goal was to fit a smoothed energy surface to the Rosetta models seen so far and then to minimize this surface to find new starting points for local optimization of the Rosetta energy function.

The first task was to define the conformation space. The most natural space is defined in terms of the conformational degrees of freedom, the phi and psi angles. However, it is difficult to fit a response surface in the space of torsion angles because the energy function is highly irregular in this space; a slight change in a single torsion angle typically causes large global structural changes, which in turn cause large energy changes. Instead, we took the three-dimensional coordinates of the backbone atoms as our conformation space, with all models in the set aligned to a reference model. There are three backbone atoms per residue and three coordinates per backbone atom, so an n-residue protein is represented by a $9n$-dimensional vector. Even for small proteins of only around 70 residues this space is very high-dimensional, but we found that most of the structural variation in sets of Rosetta models was captured by the first 10 principal components. This step is related to the reduction of conformation space to principal components of structural variation by [Qian *et al.*, 2004]. Data were sufficient to fit a response surface in these 10 dimensions.

Along certain directions, energy gradients were detectable that pointed toward the native structure. One such direction was the first principal component for protein 1n0u (Figure 1.1.a; in this graph, the native structure is represented as an ensemble of Rosetta-



(a) (b)

Figure 1.1: (a) Rosetta models (black) and relaxed natives (blue) projected onto the first principal component. (b) Models and natives projected onto the third principal component.

minimized structures that started at the native conformation). However, in most directions the gradient did not point toward the natives (Figure 1.1.b). A response surface fitted to the Rosetta models shown in these graphs will therefore have high energy in the vicinity of the natives; and, in fact, minimization of the response surface did not result in near-native structures.

There are several lessons to be learned from this failure. First, these observations point toward a feature-based strategy: rather than fitting a response surface to all the dimensions jointly, one might more profitably identify a few dimensions that are associated with clear score gradients fit surfaces to these. Second, the highest-scoring models should be disregarded as uninformative; some steric clash or other avoidable structural flaw makes them inviable, and they should not be considered in the fit.

Neighbor score

These considerations were taken into account in our next attempt to fit a response surface. Rather than use Cartesian coordinates, we designed a structure representation to get directly at the main factor responsible for the energy gradient that spurs folding in nature: burial

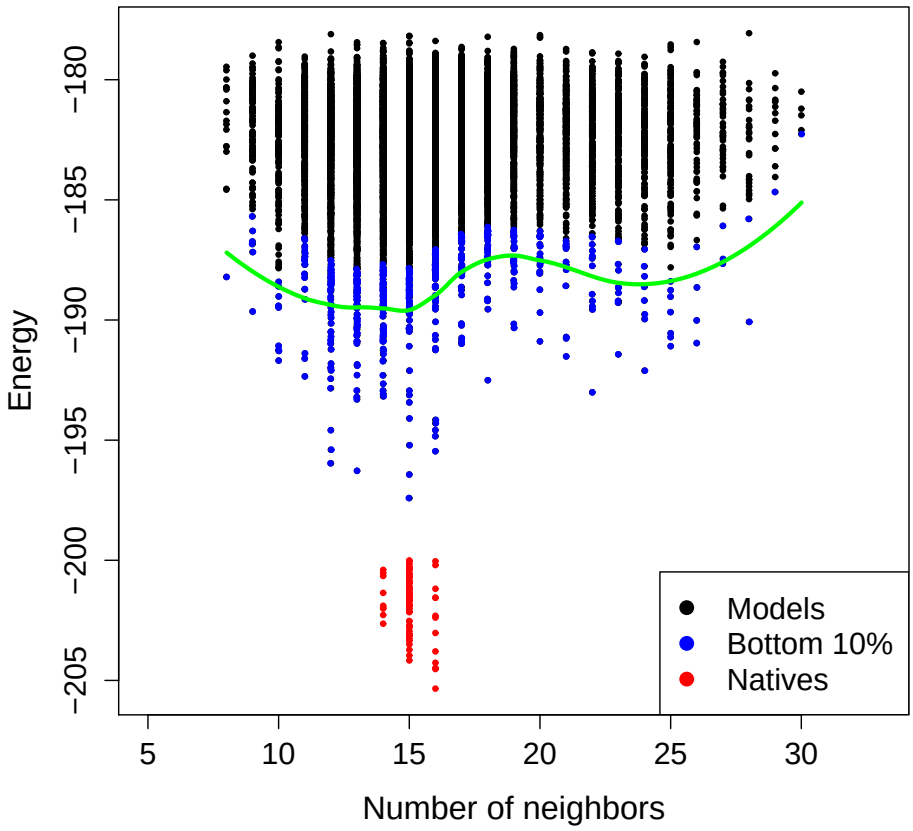
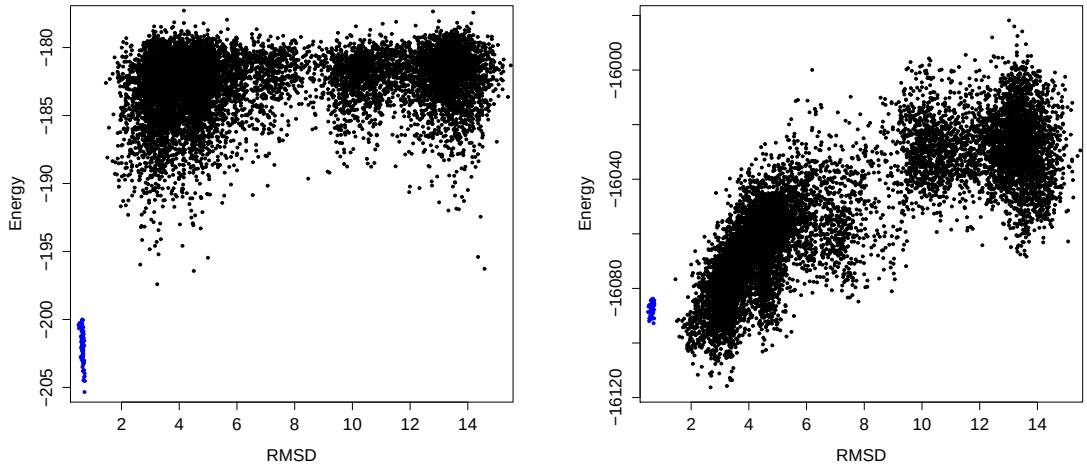


Figure 1.2: One component of the neighbor score, fitted to residue 20 of 1opd. The blue points indicate models taken into account in fitting the loess curve, shown in green. These points are the best 10% of models by energy within each bin on the horizontal axis. The red points indicate the relaxed native population.

of hydrophobic residues. The degree of burial of each residue can be approximated by the number of other residues whose centroids are within 10Å of the centroid of the given residue. The structure representation for each model is then a vector of length equal to the number of residues in the protein, with each entry being the neighbor count for the associated residue.

In keeping with the observations at the end of the previous section, we fit a separate response surface to each dimension of this space. One such response surface, for residue 20 of protein 1opd, is shown in Figure 1.2. Each black or blue point represents one of the

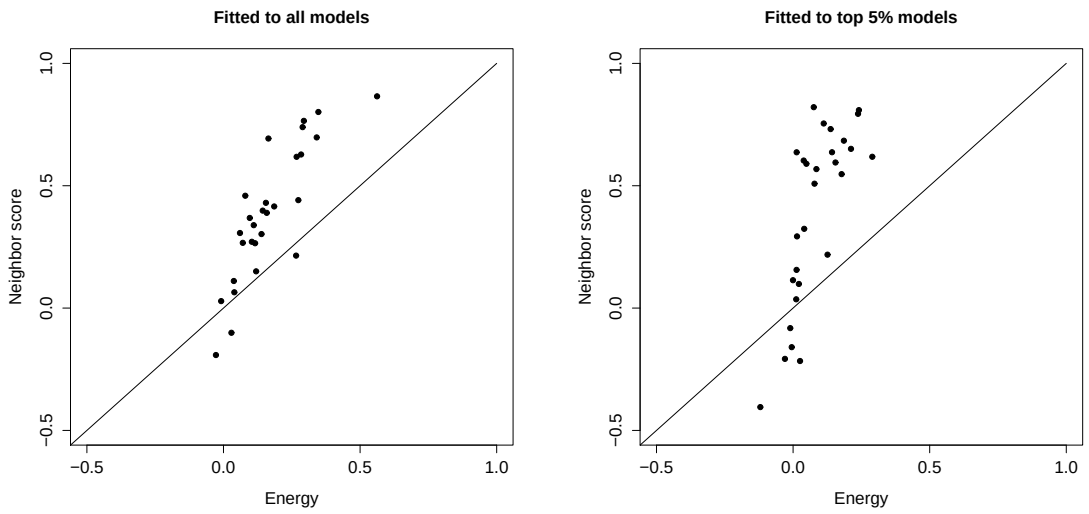


(a) (b)

Figure 1.3: (a) Energy versus RMSD for the top 5% of models for 1opd, out of a run of 200,000. (b) Neighbor score versus RMSD.

10,000 models generated in an initial sampling round. The blue points represent the best 10% of these models by energy in each of the possible integer bins along the horizontal axis. In keeping with the observation that only low-energy points are informative about the quality of regions of conformation space, these are the only points to which the curve is fitted. The curve itself is shown in green. Note that in this case the red points, the relaxed native population, are centered at the minimum of the curve. This holds generally true for nearly all residues. This is a remarkable general fact about Rosetta sampling—the consensus value for any individual feature is nearly always right, but there are few (if any) models that have all the consensus features.

The global neighbor score is derived by adding the per-residue neighbor scores for all residues. Since the natives are near the minima of most of these component scores, this in effect measures the number of residues in each model that are near the native value. The correlation of the resulting global neighbor score with RMSD to the native is, as shown in Figure 1.3.b, quite impressive; far superior to the correlation between energy and RMSD shown in Figure 1.3.a (note that these models are the lowest-energy 5% from a large sam-



(a) (b)

Figure 1.4: (a) Correlation of neighbor score with RMSD versus correlation of energy with RMSD for the 28 proteins in our benchmark, in runs of 20,000 models. (b) Same as (a) except limited to the top 5% of models from runs of 200,000.

pling round, which accounts for the flat energy cutoff at the top of the plot).

This holds generally true across our 28 protein benchmark set. In Figure 1.4 we show the correlation between the neighbor score and RMSD versus the correlation between energy and RMSD for all 28 proteins, in two different-sized sampling rounds. These results are somewhat remarkable—in extreme cases, the neighbor score has a correlation coefficient of around $.8$ while energy has a coefficient of 0.15.

Unfortunately, resampling with the neighbor score as an additional potential did not result in structures nearer to the native. The problem is two-fold: first, burial only occurs in the final stages of Rosetta search, so the new potential forces over-compression early on; second, and more importantly, the relaxed natives did not score as well as the lowest-scoring models for any of the 28 proteins in our benchmark. In most cases they scored *nearly* as well, but because they do not score lower, there is no score gradient that can be followed to find them. Thus, the neighbor score is useful only for enrichment of areas already seen—the classic pitfall of response surface methods.

There are several lessons to be learned from this effort. First, restricting attention to

the lowest-energy models and treating features as independent were successful strategies. Second, some method is required for extending search into new areas of conformation space. Figure 1.2 contains a clue. The fitted curve is clearly bimodal; another minimum appears worth exploring. The same is true for the curves fitted to other residues. Perhaps the natives sit within a combination of minima that never appears within the initial sampling round. If there were a method to sample other combinations, we could explore structures with low neighbor scores in new regions of conformation space. Unfortunately, it is quite difficult to produce a structure with a specific string of neighbor counts. Other types of structural features, however, are easier to control.

1.4.2 Feature-based resampling

The intuition behind feature-based resampling methods is that even if no models from the initial sampling round are near the native conformation, they may contain native-like *features* which can be recombined to create new, more native-like structures. In the extreme case, a model for a two-domain protein may have one domain correct and the other incorrect; intermixing this model with another that has the other domain correct would result in a wholly correct prediction.

Feature-based resampling methods in the literature are largely restricted to genetic algorithms. After each round or "generation" of search, a new round of structures is created in which features of the best structures in the previous generation are recombined with one another. The feature recombination step in genetic algorithms is intrinsically random—no attempt is made to identify those features most responsible for the success of low-energy structures and to recombine these. Structure representations and feature types differ between methods. Methods exist for Cartesian space [Rabow and Scheraga, 1996], but nearly all methods in the literature represent proteins as strings of torsion angles [Dandekar and Argos, 1992; Judson *et al.*, 1993; Bowie and Eisenberg, 1994; Pedersen and Moult, 1995; Cui *et al.*, 1998], occasionally discretized on a lattice [Dandekar and Argos, 1992]. Per-residue torsion angle features are natural and easy to work with—we employ them too—but

they fail to capture larger-scale elements of protein structure. Although a string of torsion angles is a complete description of a protein's backbone conformation, the properties of protein structure that lead to strong variations in energy—for instance, hydrogen bonding between beta strands—depend on the precise spatial interrelationships between residues distant in sequence, and hence follow only indirectly from torsion angles. A more powerful feature-space representation would include these larger-scale features. This avenue has been explored in a limited way [Petersen and Taylor, 2003], but only in the search for the boundaries of secondary structure elements, not the way in which distant secondary structure elements interrelate. No matter what structure representation is used, genetic algorithms explore new regions of conformation space by feature recombination but do so in an undirected fashion.

Another approach to feature recombination is given by [Bradley and Baker, 2006], employing beta strand pairing features similar to (though cruder than) the ones we introduce in Chapter 3. An initial sampling round is used to generate a set of strand pairing features, defined as regions in the contact plot. In the resampling round, beta contacts are enforced via bridges in the *fold tree*, a constraint system we also employ. However, no systematic effort is made to identify beta contacts most likely to be present in the native structure; some common Rosetta sampling pathologies (for instance, a tendency to form beta hairpins) are avoided via the stochastic application of score penalties, but in large part resampling is spread evenly among a subset of pairings gleaned from low-energy structures in the first round and passing certain hand-crafted topology filters. For some targets, search in the resampling round only constrains a single beta contact, resulting in no feature recombination; for others, two contacts are constrained, allowing a limited level of recombination. Our methods aim to allow unlimited recombination of native features, to systematically constrain beta contacts likely to be native at a higher rate than beta contacts less likely to be native, and to resample other kinds of features (such as torsion features) within the same framework.

Our approach

We have developed an approach that avoids the limitations of structure-based resampling methods by recombining structural features to explore new regions of conformation space, and avoids the limitations of genetic algorithms and other feature-based resampling methods by predicting likely native features to recombine and the optimal rates at which to combine them. Our work rests on two assumptions. First, we assume that even if no single local minimum computed in the first round of search has *all* the native feature values, many or all features will assume their native values in at least *some* of the models. Second, we assume that energy contributions from features are partially independent, so that native feature values will be associated, on average, with lower-energy models.

In Chapter 2, we describe a resampling algorithm that employs feature selection methods, including both decision trees and Least Angle Regression (LARS) [Efron *et al.*, 2004], to identify structural features that best account for energy variation in the initial set of models. Certain of these feature values (those associated with low energy) are predicted to be present in the native conformation. Stochastically constrained Rosetta search is used to generate a set of models enriched for these key feature values. However, no attempt is made to quantify the probability of a feature value being native, and hence to discriminate between near-certain predictions and less likely ones. All predicted native feature values are enriched at the same rate.

In Chapter 3, we develop a statistical model for prediction of nativeness probabilities and a resampling technique that exploits the model to enrich native feature values for *all* features, not just a selected few. The model incorporates a variety of statistics gathered from an initial pool of generated models. In order to learn exactly how much weight to assign to energy differences, sampling rate differences, and other "meta-features," the model is trained on a pool of Rosetta structures for 28 alpha/beta proteins with known native conformations. The training process allows us to sidestep the vulnerability of structure-based resampling methods and our own previous work to pathologies in the energy function by learning exactly how much to trust energy as an indicator of nativeness. The feature distri-

bution in models generated by standard Rosetta can be regarded as Rosetta's initial beliefs about which feature values are native; the model yields an updated distribution that combines energy information with the other meta-features, yielding an improved assessment of nativeness.

By recombining features predicted likely to be native, our methods create models in the resampling round with novel combinations of native features. This is particularly apparent in the resampling of beta strand pairing features, in which native pairings never seen together in the control population are present in the resampled population. Our results show that this methodology leads to significantly improved structure predictions.

Chapter 2

Native feature selection

2.1 Overview

In this chapter we present a resampling algorithm in which feature selection methods are used to identify a few native feature values for enrichment in further search. The experiments presented in this chapter are on a smaller scale and the results less promising than those for the main work of this thesis, described in the next chapter. However, these results are an important precursor to our later work and are founded on many of the same ideas. The limitations of the algorithm described in this chapter are an important motivation for the methods presented in the next chapter.

The native feature selection method contrasts with structure-based resampling methods, which concentrate search around a few promising *structures* already seen, by concentrating search on promising *features*. For most targets, the first round of search will not generate any models with all the native features. However, many native feature values are present in at least some of the models. If these feature values can be identified and combined with each other, then sampling can be improved.

The algorithm has three steps, each mapping from one structural representation space to another (Figure 2.1). In the first step, described in Section 2.2, we project the initial set of Rosetta models from continuous conformation space into a discrete feature space. The structural features that we have designed characterize significant aspects of protein structure and are largely sufficient to determine a unique conformation. In the second step,

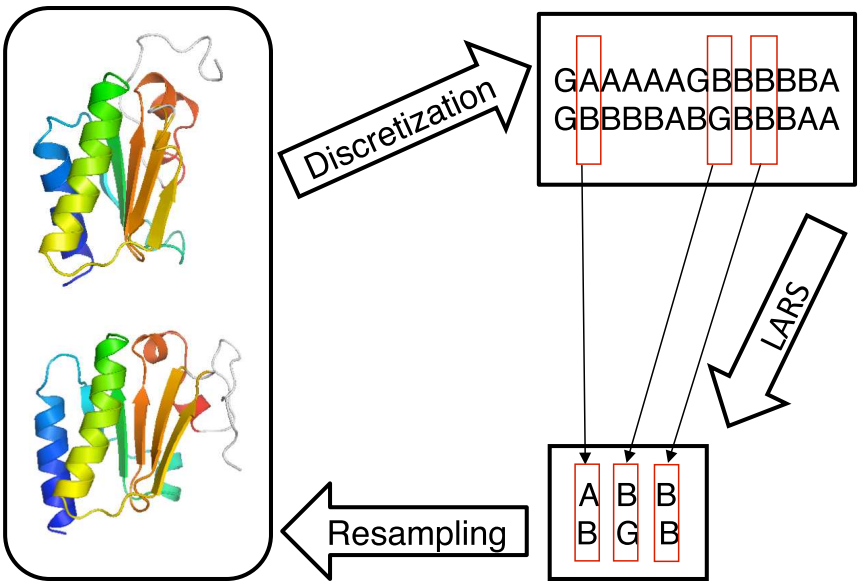


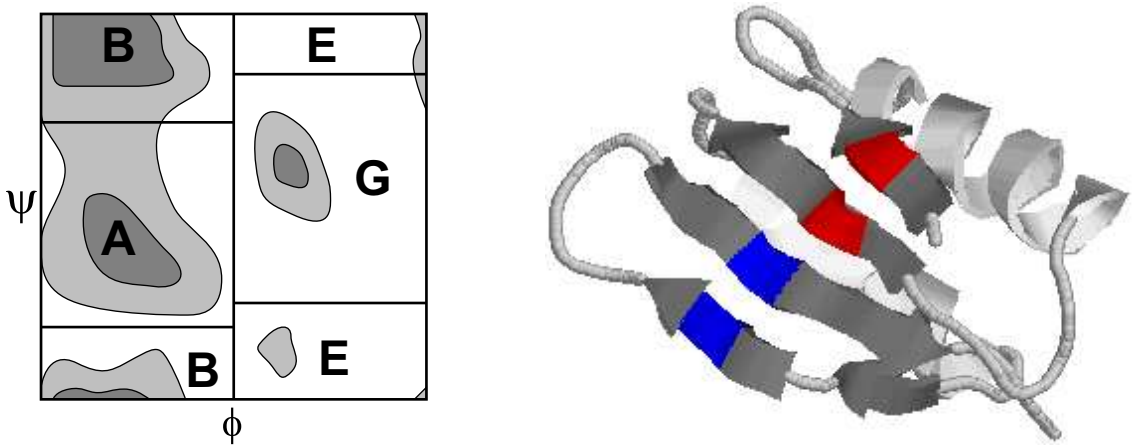
Figure 2.1: Flowchart of resampling method.

described in Section 2.3, we use feature selection methods including both decision trees and Least Angle Regression (LARS) [Efron *et al.*, 2004] to identify structural features that best account for energy variation in the initial set of models. We can then predict that certain of these feature values (generally, those associated with low energy) are present in the native conformation. In the third step, described in Section 2.5, we stochastically constrain these feature values in a new round of Rosetta search to generate a set of models enriched for these key feature values.

In Section 2.5, we show the results of Rosetta search biased towards selected feature values. In Section 2.6, we conclude with a discussion of the results achieved with this method, as well as some of the drawbacks that led to the development of the method described in the next chapter.

2.2 Discretization

The discretization step significantly reduces the search space while preserving essential structural information. For the purpose of the work described in this chapter, we make use of two types of structural features: torsion angle features and beta contact features.



(a) (b)

Figure 2.2: (a) Bins in Ramachandran plot. (b) Structure of 1dcj. Two helices are visible behind a beta pleated sheat consisting of four strands, the bottommost three paired in the anti-parallel orientation and the topmost two paired in the parallel orientation. In this "cartoon" representation of structure, individual atoms are not rendered.

2.2.1 Torsion angle features

Torsion features are residue-specific. The observed values of the ϕ and ψ angles for individual residues are strongly clustered in the database of solved protein structures (the *PDB*), as illustrated in the Ramachandran plot. In order to discretize the possible torsion angles for each residue, we divide the Ramachandran plot into four regions, referred to as "A," "B," "E," and "G" (Figure 2.2.a) roughly corresponding to clusters in the *PDB*. A fifth letter, "O," indicates a cis peptide bond and does not depend on ϕ or ψ. A protein with 70 amino acid residues has 70 torsion angle features, each with possible values A, B, E, G, and O.

2.2.2 Beta contact features

Of the two kinds of protein secondary structure, Rosetta predicts alpha helix structure somewhat more accurately than beta sheet structure. This is in large part because local contacts are easier to form during the Rosetta search process—in alpha helices, the hydrogen bonds are all local, whereas in beta sheets the bonds can be between residues that are quite distant along the chain. Beta contact features allow us to identify promising beta

contacts undersampled by Rosetta and hence to improve Rosetta's predictions of beta sheet structure.

A beta contact feature for residues i and j indicates the presence of two backbone hydrogen bonds between i and j. We use the same definition of beta pairing as the standard secondary structure assignment algorithm DSSP [Kabsch and Sander, 1983]. The bonding pattern can be either parallel (as between the red residues in Figure 2.2.b) or antiparallel (as between the blue residues). Furthermore, the pleating can have one of two different orientations. A beta pairing feature is defined for every triple (i, j, o) of residue numbers i and j and orientations $o \in \{\text{parallel}, \text{antiparallel}\}$. The possible values of a beta pairing feature are 0, indicating no pairing, and P1 or P2, indicating pleating of orientation 1 or 2, respectively.

2.3 Prediction of native features

Let $X_1, X_2, \ldots, X_n$ be all features, and let $x_i^1, x_i^2, \ldots, x_i^{m_i}$ represent the possible values of feature X_i. Let us consider the set $\{x_i^j\}$ of feature values for all i and j as a set of 0-1 valued functions, with $x_i^j(d)$ taking the value 1 to indicate that feature X_i assumes value x_i^j in conformation d. For modeling purposes, let us assume that each feature value x_i^j has an independent energetic effect; if present, it brings with it an average energy bonus b_i^j. Under these assumptions, the full energy of a conformation d is modeled as

$$E_0 + \sum_i \sum_j b_i^j \; x_i^j(d) \; + N,$$

where E_0 is a constant offset and N is Gaussian noise. This model is partially justified by the fact that the true energy is indeed a sum of energies from local interactions, and our features capture local structural information. Our hypothesis is that native feature values have lower energy on average even if other native feature values are not present. We are therefore only interested in finding feature values with weights below zero.

In order to identify a small set of potentially native feature values, we use L_1 regularization, or lasso regression [Tibshirani, 1996], to find a sparse model with only negative

weights. The minimization performed is

$$\underset{(b,E_0)}{\operatorname{argmin}} \sum_{d \in \mathcal{D}} \left(E(d) - E_0 - \sum_i \sum_j b_i^j \; x_i^j(d) \right)^2 + C \sum_i \sum_j |b_i^j|,$$

where $E(d)$ is the computed Rosetta energy of model d and C is a regularization constant. The small set of feature values that receive non-zero weights are those that best account for low energy in the initial population. These are the feature values we can most confidently predict to be native. The Least Angle Regression algorithm [Efron *et al.*, 2004] allows us to efficiently compute the trajectory of solutions for all values of C simultaneously. Experience with Rosetta has shown that constraining more than ten or fifteen torsion feature values can hamper search more than it helps; if there are very few fragments available for a given position that satisfy all torsion constraints, the lack of mobility at that position can be harmful. We typically take the point in the LARS trajectory that gives fifteen feature values.

2.3.1 Decision trees for beta contact features

Beta contact features are less suited to the lasso regression approach than torsion angle features, because independence assumptions are not as valid. For instance, contact $(i, j, \text{parallel})$ and contact $(i+1, j+1, \text{parallel})$ are redundant and will usually co-occur, whereas contact $(i, j, \text{parallel})$ and contact $(i-1, j+1, \text{parallel})$ are mutually exclusive and will never co-occur. However, these two pairings can give rise to otherwise very similar structures, and hence might both be energetically favorable. If LARS gives strong negative weights to each, then we may attempt to enforce both at once.

These considerations motivate a different approach to beta contact features. The proteins we are considering consist of no more than six beta strands; the precise pairings between these strands are therefore defined by at most five beta contact features. Using a decision tree, we divide the population of models into non-overlapping clusters defined by several beta contact feature values each. Lasso regression is then employed in each cluster separately to determine likely native torsion feature values.

We use decision trees of depth three. At each node, a beta contact feature is selected to use as a split point and a child node is created for each of the three possible values 0, P1, and P2. Our strategy is to choose split points which most reduce entropy in the features. The beta contact feature is therefore chosen whose mutual information with the other beta contact features is maximized, as approximated by the sum of the marginal mutual informations with each other feature. The score for feature X_i from the set $\{X_1, X_2, \ldots, X_n\}$ is

$$MI(X_i) = \sum_{j \neq i}^{n} I(X_j; X_i) \approx I(X_i; X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n),$$

where

$$I(X_i; X_j) = \sum_{x_i \in X_i} \sum_{x_j \in X_j} P(X_i = x_i, X_j = x_j) \log \left(\frac{P(X_i = x_i, X_j = x_j)}{P(X_i = x_i) P(X_j = x_j)} \right),$$

all probabilities being empirical probabilities observed within the subpopulation defined by the current decision tree node. The high-scoring feature is chosen as the split point.

Since some clusters are sampled more heavily than others, the lowest energy within a cluster is not a fair measure of its quality, even though, in principle, we care only about the lowest achievable energy. Instead, we use the 10^{th} percentile energy to evaluate clusters. Its advantage as a statistic is that its expectation is not dependent on sample size, but it often gives a reasonably tight upper bound on achievable energy. As a reasonable medium between including enough leaves to ensure the presence of the native topology among them and restricting sampling to few enough leaves that sampling of the native topology is not diluted too much, we restrict resampling to the three lowest-energy leaves. Ideally, we would concentrate our sampling entirely on the best leaf, but since we cannot generally identify which one it is, we have to hedge our bets. This tradeoff is characteristic of resampling methods.

2.4 Resampling

In the resampling round, we wish to search for new structures guided in some way by the predicted native feature values identified by the methods in the previous section. LARS

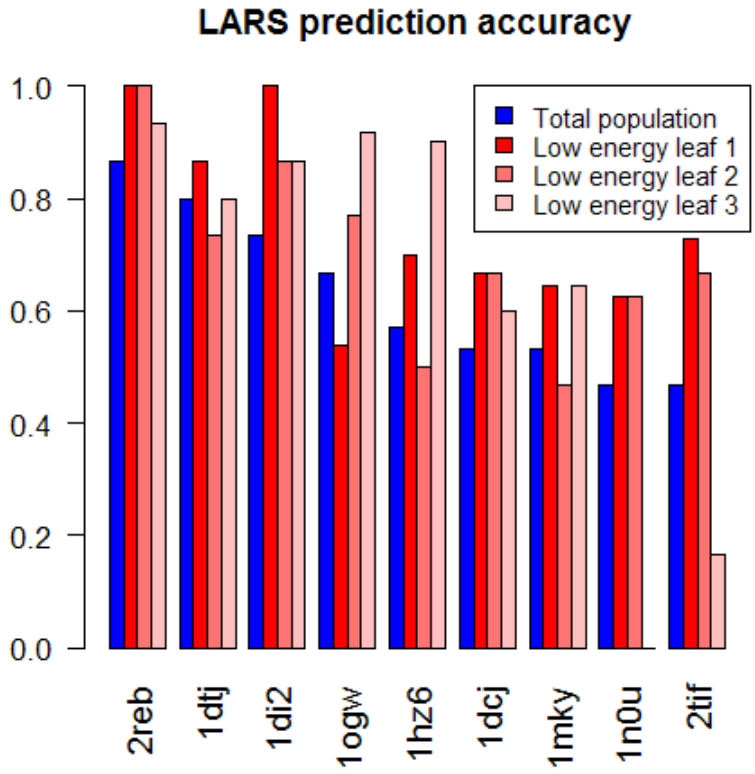


Figure 2.3: LARS prediction accuracy when fitted to total population and to the three decision-tree leaves with lowest 10th percentile energies, ordered here by average RMSD.

gives us a set of feature values that have a strong effect on energy. Our hypothesis is that feature values strongly associated with lower energies—namely, those selected by LARS and given negative weights—are more likely to be native, and that feature values given positive weights by LARS are more likely to be non-native. This hypothesis is born out by our experiments on a benchmark set of 9 small alpha/beta proteins. The LARS prediction accuracy is given in Figure 2.3. This chart shows, for each protein, the fraction of LARS-selected feature values correctly labeled as native or non-native by the sign of the LARS weight. Fifteen LARS feature values were requested per protein. The “low energy leaf” predictions were the result of running LARS only on models within the best three leaves of the beta contact decision tree, which were generally closer to the native than the population at large. Perhaps as a result, LARS generally achieved greater prediction accuracy when

restricted to their associated subpopulations. Leaves are sorted by average RMSD, so "low energy leaf 1," the "best" leaf, consists of models which are closest, on average, to the native conformation. The best leaf consisted of only native contacts for all proteins except 1n0u and 1ogw, but in both these cases it contained structures generally lower in RMSD than the population at large and resampling achieved improvements over plain Rosetta sampling. In general, LARS performed better on the leaves that were closer to the native structure, although there were a few notable exceptions.

It is clear from Figure 2.3 that LARS is informative about native feature values for most proteins. However, we cannot rely wholly on its predictions. If we were simply to constrain every LARS feature value, then Rosetta would never find the correct structure, since some incorrect feature values would be present in every model. Our resampling strategy is therefore to flip a coin at the beginning of the Rosetta run to decide whether or not to constrain a particular LARS feature value. Coins are flipped independently for each LARS feature value. Resampling improves on unbiased Rosetta sampling if the number of viable runs (runs in which no non-native feature values are enforced) is sufficiently high that the benefits from the enforcement of native feature values are visible. We have achieved some success by enforcing LARS feature values with probability 30% each, as demonstrated in the results section.

Greater LARS accuracy can be achieved by restricting attention to models within the clusters identified by the beta contact decision tree method. Our resampling strategy, given a decision tree, is to sample evenly from each of the top three leaves as ranked by 10^{th} percentile energy. Within the subpopulation of models defined by each leaf, we select torsion feature values using LARS.

It remains to describe the means by which we constrain features. Torsion features are easier to constrain than beta contact features; a torsion angle feature value can be constrained in Rosetta search simply by rejecting all proposed fragment replacement moves that place torsion angles outside the desired bins. Strings of torsion feature values are referred to as *barcodes* in Rosetta, and the apparatus for defining and constraining them was

	RMSD of low-energy model				Lowest RMSD of 25 low-energy models			
	Decision-tree		LARS-only		Decision-tree		LARS-only	
	Control	Resamp	Control	Resamp	Control	Resamp	Control	Resamp
1di2	2.35	2.14	2.76	0.97	1.78	1.34	1.82	0.73
1dtj	3.20	1.53	5.28	1.88	1.46	1.53	1.95	1.59
1dcj	2.35	3.31	2.34	2.11	2.19	1.86	1.71	1.88
1ogw	5.22	3.99	3.03	2.80	3.12	2.6	2.08	2.48
2reb	1.15	1.17	1.07	1.27	0.89	0.93	0.83	0.86
2tif	5.68	4.57	3.57	6.85	3.32	3.27	3.27	2.61
1n0u	11.89	11.60	11.93	3.54	9.78	3.19	3.54	2.84
1hz6A	2.52	1.06	3.36	4.68	2.38	1.06	1.97	1.19
1mkyA	10.39	8.21	4.60	4.58	3.43	3.25	3.33	4.23
Mean difference		-0.8		-1.03		-1.04		-0.23
Median difference		-1.11		-0.23		-0.33		-0.36
Number improved		7/9		6/9		7/9		5/9

Table 2.1: Results for the resampling rounds compared with control rounds of search for two resampling schemes: "LARS-only", in which only torsion features were constrained, and "decision-tree," in which torsion features and beta contacts were constrained. Results presented are the RMSD to native of the single lowest energy model generated during search and the lowest RMSD of the 25 lowest energy models generated during search.

developed in-house by Rosetta developers.

Beta contact features are enforced in Rosetta by means of a bridge in the *fold tree* [Bradley and Baker, 2006]. A pseudo-backbone-bond is introduced between the two residues to be glued together. This introduces a closed loop into the backbone topology of the protein. Torsion angles within the loop can no longer be altered without breaking the loop, so, in order to permit further fragment replacements, a cut (or "chainbreak") must be introduced somewhere else in the loop. The backbone now takes the form of a tree rather than a chain. After a Rosetta search trajectory terminates, an attempt is made to close the chainbreak with local search over several torsion angles on either side of it.

2.5 Results

We tested two Rosetta resampling schemes over a set of 9 alpha/beta proteins of between 59 and 81 residues. In the first scheme (referred to henceforth as "LARS-only"), 15 LARS-predicted torsion feature values were constrained at 30% frequency. In the second (referred to henceforth as "decision-tree"), three subpopulations were defined for each protein using

a decision tree, and within each subpopulation 15 LARS-predicted torsion feature values were constrained at frequencies heuristically determined on the basis of several meta-level "features of features," including the rate of the feature value's occurrence in the first round of Rosetta sampling and the magnitude of the regression weight for the feature value. This heuristic is a precursor of the nativeness predictors in the next chapter, but was not trained; the weights were set by hand. Each resampling scheme was compared against a control population generated at the same time. Exactly the same number of models were generated for the control and resampled populations. The control and resampled populations for the LARS-only scheme consist of about 200,000 models each. The populations for the decision-tree scheme consist of about 30,000 models each, due to limitations in available compute time. The difference in quality between the two control populations is partially explained by the different numbers of samples in each, and partially by changes in Rosetta in the time between the generation of the two datasets.

Our two primary measures of success for a resampling run are both based on root-mean-square distance to the native structure. Root-mean-square distance (RMSD) is a standard measure of discrepancy between two structures. It is defined as the square root of the mean of the squared distances between pairs of corresponding backbone atoms in the two structures, under the alignment that minimizes this quantity. Our first measure of success is the RMSD between the native structure and the lowest scoring model. This measures Rosetta's performance if forced to make a single prediction. Our second measure of success is lowest RMSD among the twenty-five top-scoring models. This is a smoother measure of the quality of the lowest scoring Rosetta models, and gives some indication of the prediction quality if more sophisticated minima-selection methods are used than Rosetta energy ranking. Structures at 1Å from the native have atomic-level resolution—this is the goal. Structures at between 2Å and 4Å generally have several important structural details incorrect. In proteins the size of those in our benchmark, structures more than 5Å from the native are poor predictions.

Both resampling schemes achieved some success. The performance measures are shown

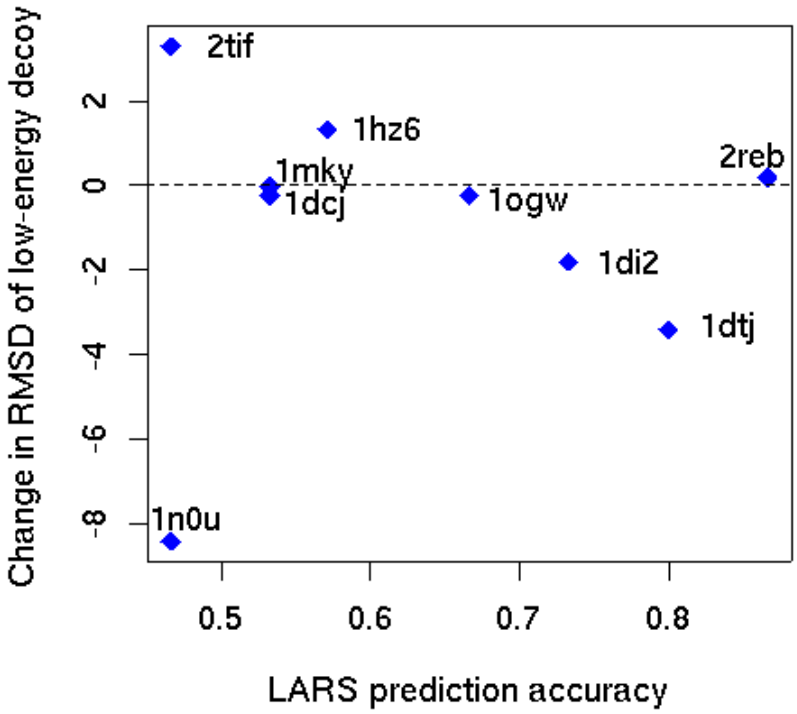


Figure 2.4: Relation of prediction accuracy to resampling improvement in LARS-only runs.

in Table 2.5. The decision-tree scheme performed more consistently and achieved larger improvements on average; it improved the low-energy RMSD in 7 of the 9 benchmark proteins, with a significant median improvement of 1.11Å. Particularly exciting are the atomic-resolution prediction for 1hz6 and the nearly atomic-resolution prediction for 1dtj. In both these cases, plain Rosetta sampling performed considerably worse. The LARS-only scheme was successful as well, providing improved lowest-energy predictions on 6 of the 9 benchmark proteins with a median improvement of 0.23Å. The LARS-only low-energy prediction for 1di2 is atomic-resolution at 0.97Å away from the native structure, as compared to 2.97Å for the control run. In general, improvements correlated with LARS accuracy (Figure 2.4). The two notable exceptions were 2reb, for which plain Rosetta search performs so well that constraints only hurt sampling, and 1n0u, for which plain

Rosetta search concentrates almost entirely on a cluster with incorrect topology at 10Å. Certain LARS-selected feature values, when enforced, switch sampling over to a cluster at around 3Å. Even when incorrect feature values are enforced within this cluster, sampling is much improved.

The cases in which the decision-tree scheme did not yield improved low-energy predictions are interesting in their own right. In the case of 1dcj, resampling does yield lower RMSD structures—the top 25 low rms prediction is superior, and the minimum RMSD from the set is 1.35, nearly atomic resolution, as compared to 1.95 for the control run—but the Rosetta energy function does not pick them out. This suggests that better techniques for selecting predictions from the model pool would improve our algorithms. In the case of 2reb, the unbiased rounds of Rosetta sampling were so successful that they would have been difficult to improve on. This emphasizes the point that resampling cannot hurt us too much. If a plain Rosetta sampling round of n models is followed by a resampling round of n models, then no matter how poor the resampled models are, sampling efficiency is decreased by at most a factor of 2 (since we could have generated n plain Rosetta samples in the same time). The danger is that resampling may overconverge to broad, false energy wells, achieving lower energies in the resampling round even though RMSD is higher. This appears to occur with 2tif, in which the LARS-only low-energy prediction has significantly lower energy than the control prediction despite being much farther from the native. Once more, better techniques for selecting a single prediction model might help.

2.6 Discussion and Conclusions

Our results demonstrate that the native feature selection technique improves structure prediction on a majority of the proteins in our benchmark set. The LARS-only method significantly improves Rosetta predictions in 3 of the 9 test cases, and marginally improves two more. The decision tree method expands the set of proteins on which we achieve improvements, including an additional atomic-level prediction. It is important to note that significant improvements over Rosetta on *any* proteins are hard to achieve; if our methods

achieved one or two significantly improved predictions, we would count them a success. Rosetta is the state of the art in protein structure prediction, and it has undergone years of incremental advances and optimizations. Surpassing its performance is very difficult. Furthermore, it doesn't hurt Rosetta too badly if a resampling scheme performs worse than unbiased sampling on some proteins, since models from the unbiased sampling round that precedes the resampling round can be used as predictions as well.

The decision tree and LARS allow us to pick out a few feature values that we can fairly confidently concentrate our attention on. However, in both cases, there are tradeoffs to be weighed when deciding how many feature values to select. If we select very few, they are more likely to be native; however, there is less to be gained by enriching them. If we select too many, the accuracy of LARS and the decision tree goes down, and the harm from all the incorrect feature values we are enriching outweighs the benefit from the native feature values. In this chapter, we have chosen compromises between these two extremes by hand: three decision tree leaves were resampled for each protein, and fifteen torsion feature values were enriched within each of these leaves. It would be more satisfying, however, to have this compromise chosen automatically. Even better, if our feature selection methods were able to give us some measure of the *confidence* of their predictions, perhaps we could avoid the penalty when large numbers of feature values are selected. Predictions with low confidence could be enriched very slightly, while predictions with high confidence could be constrained nearly all the time. In fact, if the confidences are reliable enough, we might be able to do away with the feature selection entirely and resample every single feature at once. This is the approach of the next chapter.

Chapter 3

Nativeness prediction

3.1 Overview

In Chapter 1, we discussed limitations of structure-based resampling methods, and in particular of response surface fitting—these methods amount to enrichment of regions of conformation space that have already been explored. We also discussed the limitations of existing feature-based resampling methods, particularly genetic algorithms—these methods blindly recombine features from successful *structures*, rather than seeking out successful *features*. In Chapter 2, a method was described to predict likely native feature values using feature selection methods. These features were then stochastically constrained in a subsequent resampling round, resulting in a few significant improvements over plain Rosetta sampling. However, there was no way to be certain which predictions were correct. The method was limited to enrichment of just a few native feature values in order to avoid enriching too many incorrect non-native feature values.

In this chapter we introduce a more sophisticated resampling method that makes use of a statistical model for prediction of nativeness probabilities to alter Rosetta's sampling distribution for *all* features, not just a selected few. We present evidence that Rosetta's initial sampling distribution represents Rosetta's initial beliefs about which feature values are native, beliefs based primarily on sequence information and on Rosetta's low-resolution energy potential. The statistical model updates these beliefs to take into account information from Rosetta's full-atom energy potential and other sources, or "meta-features."

Our specific choice of features is guided by the need to find important local determinants of global structure whose native values can be discerned by our predictor. But simply assessing how likely feature values are to be native does not solve our overall problem—we must also resample protein configurations from the featural representation. In this chapter we develop two new approaches to resampling from the target distribution provided by the nativeness predictor. The first is referred to as *fragment repicking*, and it involves changing the fragment pool available to Rosetta. For simple features such as torsion features this is relatively easy to do, but for more complex features such as beta pairing features it is difficult to adjust the fragment pool directly so as to match the target distribution. We have thus developed a second approach which extends the stochastic constraint methodology used in Chapter 2 to allow a richer distribution over constraints. This is necessary for stochastic enforcement of beta sheet features, since these features form a nested hierarchy–the draw of constraints at the bottom of the hierarchy depends on the draws from higher up.

Our resampling approach has three steps (Figure 3.1). The first and third step correspond closely to the similarly named steps in Chapter 2. In the first, or "discretization" step, we project an initial set of Rosetta models for the target protein from conformation space into a discretized feature space. Each model is represented as a string of discrete-valued features of three types: torsion features of the same kind as those used in Chapter 2, with values corresponding to bins in the Ramachandran plot; per-residue secondary structure features; and a three-level hierarchy of beta structure features with values corresponding to topologies, registers, and contacts. Rosetta's marginal sampling rate P_{samp} for each feature can be regarded as an initial sequence-based belief about which feature value is native; native feature values are, in general, sampled by Rosetta at higher rates than non-native feature values (Figure 3.2). In the second, or "prediction" step, we update Rosetta's initial beliefs using information about which feature values are associated with low energies in the initial set of Rosetta models to derive a new belief distribution P_{pred} in which many native feature values appear at significantly higher rates. In the third, or "resampling" step, we use Rosetta to sample a new set of models that match this updated distribution over features;

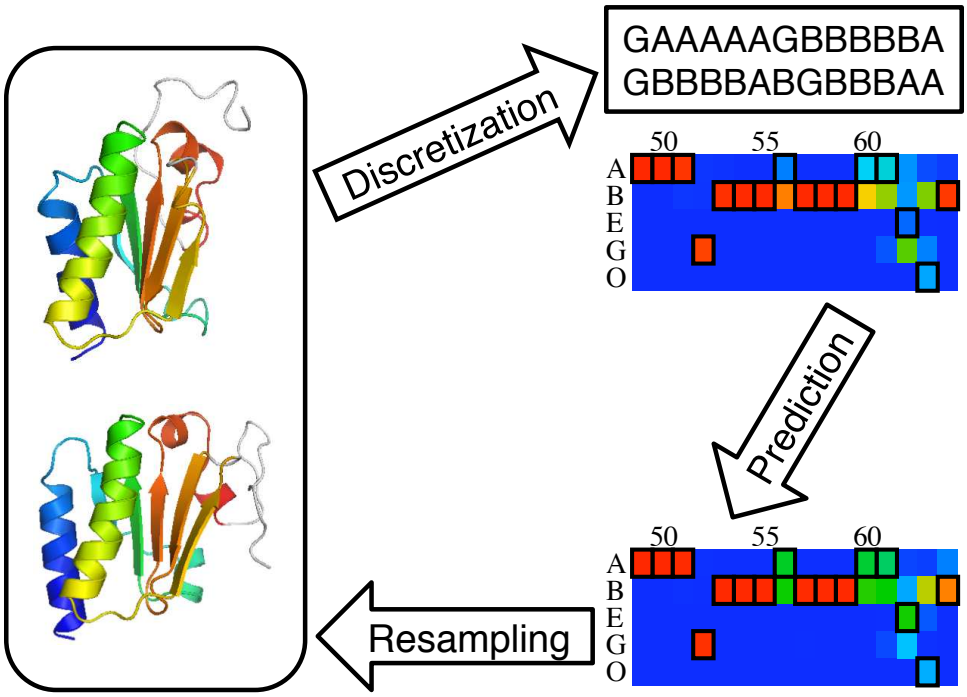


Figure 3.1: Flow-chart outline of the new resampling method. The process has three steps: discretization of initial models into feature strings, prediction of likely native feature values, and resampling with Rosetta to produce models with these feature values. The colored grids represent marginal sampling distributions over the torsion features associated with residues 49–64 of protein 1dcj, with possible values “A,” “B,” “E,” “G,” and “O,” corresponding to bins in the Ramachandran plot (Figure 2.2.a). The spectrum runs from blue, representing sampling rates near 0%, through green, representing rates near 50%, to red, representing rates near 100%. Each column represents the distribution over a single feature. A black outline indicates the native feature value. The top grid depicts the sampling distribution observed in the initial round of Rosetta search, and the bottom grid depicts the predicted native probabilities, which are used as targets in the resampling round of search.

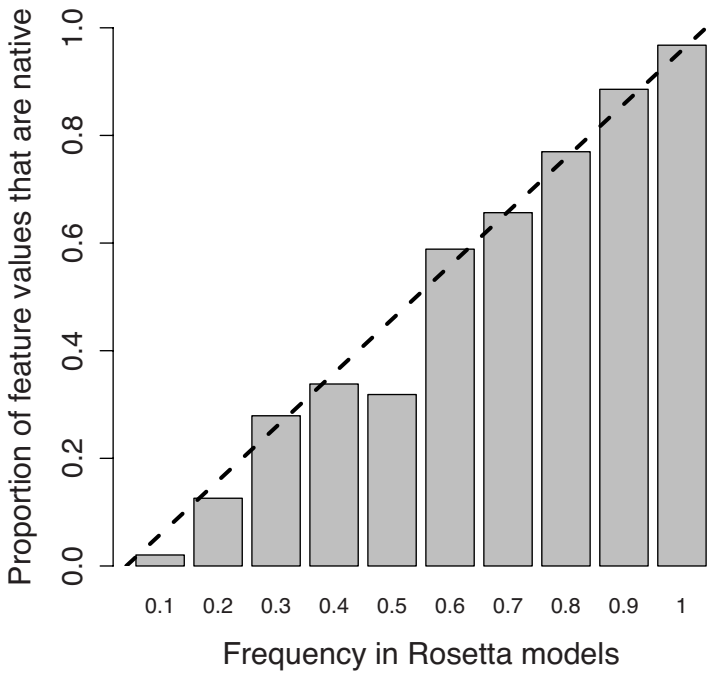


Figure 3.2: Torsion feature values pooled together from a benchmark set of 28 proteins and divided into bins according to P_{samp}. Bins are labeled by their upper bounds. The height of each bar indicates the proportion of the feature values within the bin that are native. The linear relationship suggests that the Rosetta sampling rate of a feature value is roughly equal to the chance that the feature value is native.

this is done by fragment repicking and by applying stochastic constraints during search. As the flow-chart suggests, the cycle can be iterated.

3.2 Discretization

The features used in this method fall into five classes: torsion features, per-residue secondary structure features, and three separate classes of beta sheet features. We will generally denote the i^{th} feature in a set by X_i, and its possible values by $x_i^1, x_i^2, \ldots, x_i^{m_i}$, with one of these, denoted by x_i^*, being the native one. We will occasionally be loose with our terminology and refer to a feature that assumes its native value as a native feature.

3.2.1 Torsion features

Torsion features are defined exactly as in Chapter 2. Each residue has an associated torsion feature, which takes values in the Ramachandran plot bins “A,” “B,” “E,” “G,” and “O.”

3.2.2 Secondary structure features

Secondary structure features are also associated with single residues. They take values in the standard alphabet “E,” “H,” and “L,” indicating whether the residue particiates in a sheet, helix, or loop.

3.2.3 Beta sheet features

The beta structure of a protein conformation can be parsed at several different levels, illustrated in Figure 3.3. At the topmost level is the *topology*, which identifies the beta strands that pair with each other. We describe a topology by the set of *pairings* that compose it. Each pairing has an associated orientation (either antiparallel or parallel). Each protein has a single topology feature, whose possible values are the possible sets of pairings.

At the second level, a pairing between two strands may be realized in several different ways. The alignment between residues in the two strands is called the *register*. A register may contain one or more *beta bulges*, in which one or more residues do not pair with any residues on the opposite strand. A register includes the position of all bulges along the pairing and the alignments in each bulge-free segment. Each pairing has an associated pairing feature, whose domain is the set of all registers for that pairing ever observed in the initial sampling round. Pairing features are defined using an agglomerative clustering of registers from the initial sampling round, with a distance metric that depends upon the locations of the centers of both paired strands in the register.

At the third and final level, a single register is consistent with a large set of possible beta contacts. A register prototype is built up from the initial sampling round by merging all beta contacts ever observed to participate in the register. The merging process generally results in a prototype that extends too far in each direction; the native conformation will

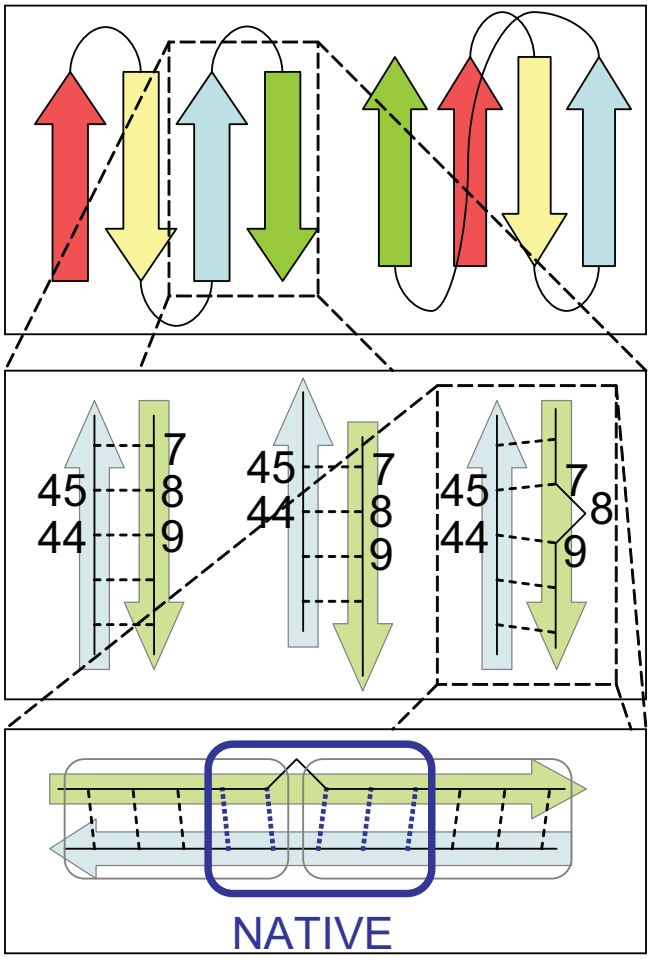


Figure 3.3: Beta topology, pairing, and contact features. At the top level is a single topology feature, with each value a possible topology. One such topology consists of several pairings, each of which has an associated pairing feature, shown in the middle level. The values of the pairing feature are all possible registers. Each register is associated with a set of contact features, shown in the bottom level. In this example, the register has two bulge-free regions, each associated with a contact feature circled in gray. The values of a contact feature are all possible contacts within the region. The contacts present in the native structure are circled in blue. To constrain the native register, one native constraint must be chosen from each contact feature.

have only a subset of beta contacts present in the native register prototype. In order to enforce a register, one constraint is required in each bulge-free region, so a contact feature is created for each such region whose possible values are all possible contacts from that region to enforce. This feature type differs from the others in that multiple different values may be native.

Example 1. *The native topology for 1di2 includes a beta sheet with three strands, strand A running from residue 19 to residue 25, strand B running from residue 33 to residue 39, and strand C running from residue 43 to residue 48. Strands A and B pair, as do strands B and C, so this topology has two associated pairing features, AB and BC.*

We will examine pairing feature AB in detail. The possible values for a pairing feature are registers, which we define as a set of beta contacts, each of which will be denoted by a pair (i, j) *of residue numbers. The possible registers for pairing AB include:*

$$
\begin{aligned}
R1: &\quad \{(18,40),(19,39),\ldots,(27,31)\} \\
R2: &\quad \{(18,40),(19,39),\ldots,(22,36),(24,35),\ldots,(27,32)\} \\
R3: &\quad \{(20,40),(21,39),\ldots,(27,33)\}
\end{aligned}
$$

Register $R2$ *differs from register* $R1$ *in having a beta bulge—residue 23 doesn't pair with any residue on strand B. Note that the beta contacts in these registers extend slightly outside the areas designated strand in the native structure, because they include all beta contacts ever observed in the initial sampling round.*

Each register brings with it one or more contact features, one for each bulge-free region in the register. The number of such features is therefore one greater than the number of bulges in the register. For example, register $R2$ *has two contact features, one with five possible values,* $\{(18,40),(19,39),(20,38),(21,37),(22,36)\}$*, and one with four possible values,* $\{(24,35),(25,34),(26,33),(27,32)\}$*. In order to constrain this register, two beta contact constraints must be chosen to enforce, one from each of these two contact features. Note that for beta contact features, multiple values might be native. Selecting any of the*

native contacts counts as a success.

Beta features are hierarchical; each pairing feature is associated with the topology from which it derives. If two different topologies both contain the same pairing, a copy of the pairing feature is created for each. This distinction is important for the prediction step, in which the predicted distribution over registers may depend on the topology. However, due to the partially independent energetic contributions of different features, models with a non-native topology that nonetheless includes a native strand pairing can in fact be informative about the correct register for that pairing; if a given register is energetically favorable even in models with incorrect global topology, it is more likely to be the native register. Therefore, in predicting which register is the native value for a pairing feature, we collect energy and distribution statistics both for models within the parent topology and for all models with the pairing. Beta contact features, too, are associated with a particular topology, so also give rise to these two classes of statistics.

3.3 Prediction

In the prediction step of our algorithm, we interpret Rosetta's feature sampling distribution $P_{samp}(X_i)$ as Rosetta's initial beliefs about which value for feature X_i is native. We update these beliefs using a *nativeness predictor* that incorporates both $P_{samp}(X_i)$ and various other features-of-features, or *meta-features*, to arrive at a new belief distribution $P_{pred}(X_i)$.

3.3.1 Form of the nativeness predictor

We build a set of five predictors, one for each class of features (torsion, secondary structure, topology, pairing, and contact). Each such nativeness predictor can be viewed as an ensemble of logistic regression models for each feature in the class, with the regression weights tied together.

Let $X_1, X_2, \ldots, X_n$ be all features from a single class (for example, all torsion features). Let $x_i^1, x_i^2, \ldots, x_i^{m_i}$ represent the possible values of feature X_i, and let x_i^* be the native value. Each feature value x_i^j—for instance, bin "B" of torsion angle 34 for protein 1dcj, or

the beta barrel topology for protein 1acf—has a corresponding numeric meta-feature vector $[P_{samp}(x_i^j), lowE(x_i^j), minE(x_i^j), \ldots]$ consisting of a mix of energy and distribution statistics from the initial round of Rosetta models. Brief descriptions of the meta-features we use are given in Table 3.1, along with the predictive power of each (as measured by the percentage of native feature values from all proteins in our benchmark that can be identified using this meta-feature alone). A justification for our choice of meta-features is given in the next section. Note that even though P_{samp} is in some sense "special," as it gives the prior belief distribution to be updated, it is treated just like the other meta-features for the purpose of the predictor.

Some meta-features are transformed, either to make their ranges comparable to one another or for reasons of mathematical convenience; for instance, the $P_{samp}(x_i^j)$ term is transformed to $\log(P_{samp}(x_i^j))$. Let $f_k(x_i^j)$ be the k^{th} transformed meta-feature of x_i^j and let $\Phi(x_i^j)$ be the vector of all single transformed meta-feature terms and their pairwise combinations. We compute the dot product between a weight vector β and $\Phi(x_i^j)$ via $\beta^t\Phi(x_i^j) = \sum_{k\neq k'} \beta_{k,k'} f_k(x_i^j) f_{k'}(x_i^j) + \sum_k \beta_k f_k(x_i^j)$. The presence of the pairwise combinations of features allows our model to take joint effects into account. Given a weight vector β, the predicted probability that x_i^j is native in our model is

$$P_{pred}(x_i^j) = \frac{e^{\beta^t\Phi(x_i^j)}}{\sum_{j'=1}^{m_i} e^{\beta^t\Phi(x_i^{j'})}}.$$

The form of the predictor allows it to output P_{samp} unmodified, given the proper setting of the weights (one for $\log(P_{samp})$ and zero for all others), so it is theoretically possible for the trained predictor to make no changes to the Rosetta sampling distribution. This is why we transform P_{samp} to $\log(P_{samp})$ in the meta-feature vector.

3.3.2 Choice of meta-features

We designed our meta-features to encompass both sequence information and all-atom energy information. The value of sequence information in native feature prediction has been established by the success of sequence-based secondary structure predictors like Psipred

Torsion meta-feature		Accuracy
P_{samp}	Rosetta sampling rate	88.9%
lowE	10^{th} percentile energy of models with the feature value	76.4%
minE	minimum energy of models with the feature value	87.7%
P_{frag}	rate of occurence of the feature value in the fragments	86.2%
loop	indicates either an E or O torsion feature value	
P_{pred}	output of nativeness predictor	91.1%

Secondary structure meta-feature		Accuracy
P_{samp}	Rosetta sampling rate	87.2%
lowE	10^{th} percentile energy of models with the feature value	72.8%
minE	minimum energy of models with the feature value	86.2%
$P_{psipred}$	secondary structure prediction from Psipred	87.7%
P_{jufo}	secondary structure prediction from JUFO	80.9%
P_{pred}	output of nativeness predictor	91.8%

Topology meta-feature		Accuracy
P_{samp}	Rosetta sampling rate	21.4%
lowE	10^{th} percentile energy of models with the feature value	21.4%
minE	minimum energy of models with the feature value	46.4%
co	approximate contact order of a structure with the given topology	
P_{pred}	output of nativeness predictor	60.7%

Register meta-feature		Accuracy
P_{samp}	Rosetta sampling rate	54.0%
lowE	10^{th} percentile energy of models with the feature value	44.7%
minE	minimum energy of models with the feature value	61.2%
bulge	indicates the presence of at least one beta bulge in the register	
P_{pred}	output of nativeness predictor	57.6%

Contact meta-feature		Accuracy
P_{samp}	Rosetta sampling rate	85.4%
lowE	10^{th} percentile energy of models with the feature value	68.9%
edgedist	distance (in residue numbers) of a contact from either end of a pairing	92.2%
oddpleat	indicates an anomaly in the pleating pattern	
P_{pred}	output of nativeness predictor	88.3%

Table 3.1: Meta-features by feature class. Accuracy indicates the percentage of native feature values from all proteins in our benchmark correctly identified by the meta-feature, i.e. those for which the meta-feature is highest (or lowest, in the case of energy-based meta-features) among all values for the associated feature. Accuracy values have been omitted for meta-features that are only informative in conjunction with other meta-features and so have no predictive value on their own. P_{pred}, the output of the nativeness predictor, is not a meta-feature; it is included here for comparison. The accuracy of P_{pred} is not always as high as the accuracy of each of its constituent meta-features because it optimizes sampling efficiency, a different metric than accuracy.

[Jones, 1999]. We hypothesize that energy statistics from an initial round of sampling provide a valuable, non-redundant source of additional predictive power. Our hypothesis depends on global energy receiving approximately independent contributions from local structural features, so that native features will generally be associated with lower energies even when paired with non-native features. This hypothesis is motivated by the fact that the global energy is a sum of terms from physically local interactions. In order to capture these two sources of information, three meta-features are common to nearly all our predictors: P_{samp}, the Rosetta sampling distribution, *lowE*, the 10th percentile energy of models with the feature value, and *minE*, the minimum energy of models with the feature value.

P_{samp} contains primarily local sequence information. This can be deduced from the Rosetta search procedure: the principal Rosetta search move is to replace the torsion angles in a contiguous series of residues with a fragment selected at random from a fragment pool, which is itself derived by matching the local sequence and predicted secondary structure in the sequence being folded with the local sequence and actual secondary structure of fragments from structures in the PDB. The distribution of secondary structure features in the fragment pool, and hence in Rosetta models, is therefore closely related to sequence-based secondary structure predictions from Psipred [Jones, 1999], JUFO [Meiler *et al.*, 2001], and SAM [Karplus *et al.*, 1998]. The Monte Carlo search procedure selectively rejects some fragment replacement moves, but only on the basis of a low-resolution energy function that discriminates plausible tertiary structures from implausible ones. The high resolution refinement step in Rosetta, which does employ an all-atom energy function, does not generally modify conformations enough to alter the values of our structural features.

The meta-features *lowE* and *minE*, on the other hand, are direct measures of the lowest all-atom energies achievable in models with a given feature value. The very lowest energy models seen in search determine the value of *minE*, while *lowE*, whose expected value does not depend on the sample size, is a fairer measure of energy for promising feature values which are sampled very rarely and hence do not have a chance to appear in a low energy structure.

Each feature class also brings with it one or more additional class-specific meta-features. Many of these meta-features are designed to ameliorate common modeling pathologies. For example, Rosetta sampling is biased toward short-range pairings, as these are easier to form; the contact order [Plaxco *et al.*, 1998] of a topology is a useful meta-feature for correcting this bias, although not predictive of native topologies on its own.

P_{samp} for a pairing feature indicates the distribution of registers only among models with that pairing's parent topology; *lowE* and *minE* are similarly restricted to models with the parent topology. For pairing or contact features, energy or distribution statistics may also be computed over all models that have that pairing, not just those with the parent topology, in which case they will be marked by the superscript *all*.

3.3.3 Training

The free parameters in the predictor are the components of the weight vector β, which must be fitted by maximizing an objective function. Rather than fit some standard measure of belief accuracy, we aim to directly maximize the predictor's effectiveness as input to our resampling method. As we will outline later in Section 3.4, the resampling step of our algorithm attempts to modify Rosetta search to sample features according to the distribution P_{pred} instead of according to P_{samp}. Accordingly, we use as an objective function the *sampling efficiency* of P_{pred}, which we define as $\prod_{i=1}^{n} P_{pred}(x_i^*)$, the estimated probability of encountering a fully native structure in a single Rosetta run with feature distribution P_{pred}. The inverse of this quantity can be regarded as an approximation, ignoring correlations between features, of the expected number of Rosetta samples required to produce a native-like structure. In order to incorporate training data from multiple proteins into the objective function, the sampling efficiencies of each of the proteins in the training set are multiplied (in fact, since we work on a log scale for numerical stability, their logarithms are summed).

We fit β using the standard BFGS variant of Newton's method [Broyden, 1970]. The fitted weights for the various predictors (trained on a benchmark of 28 proteins) are shown

Torsion predictor

	P_{samp}	P_{frag}	*lowE*	*minE*	*loop*
P_{samp}	1.11	-0.16	-0.17	-0.43	-0.083
P_{frag}		0.51	0.35	-0.022	-0.16
lowE			-0.11	0.094	0.21
minE				-0.90	-0.24
loop					0.29

Secondary structure predictor

	P_{samp}	*minE*	*lowE*	$P_{psipred}$	P_{jufo}
P_{samp}	1.36	-0.35	-0.29	2.35	-0.24
minE		-0.27	-0.23	1.39	0.081
lowE			0.43	0.28	-0.38
$P_{psipred}$				1.10	-0.54
P_{jufo}					0.06

Topology predictor

	P_{samp}	*co*	*lowE*	$minE^{all}$
P_{samp}	1.49	-1.57	0.82	-1.37
co		-0.22	-0.043	-1.45
lowE			4.19	-2.11
$minE^{all}$				2.27

Register predictor

	P_{samp}^{all}	*minE*	$lowE^{all}$	*bulge*
P_{samp}^{all}	0.53	-0.042	0.13	1.06
minE		-0.86	0.29	0.50
$lowE^{all}$			0.55	0.074
bulge				0.50

Contact predictor

	P_{samp}^{all}	$lowE^{all}$	*edgedist*	*oddpleat*
P_{samp}^{all}	1.05	2.57	1.79	0.00038
$lowE^{all}$		-0.29	1.28	-0.48
edgedist			-0.14	-0.12
oddpleat				-0.45

Table 3.2: Predictor weights for the five feature classes. Weights for individual meta-features are on the diagonal, weights for pairwise terms are elsewhere.

in Table 3.2.

3.4 Resampling

Our results show that native features are generally more probable in P_{pred}, the output distribution of the nativeness predictor, than in Rosetta's initial sampling distribution P_{samp} (Section 3.5.1). This motivates using P_{pred} as a target distribution for Rosetta. In particular, letting $P_{resamp}(x_i^j)$ denote the sampling rate in Rosetta of feature value x_i^j after our fragment repicking and stochastic constraint protocols have been imposed, our approach aims to set $P_{resamp}(x_i^j) = P_{pred}(x_i^j)$. In this section we consider P_{pred} solely as a nativeness belief distribution from some unspecified source, putting aside for the moment the fact that we assumed $P_{resamp} = P_{pred}$ in training the predictors, and provide some justification for this particular relationship between P_{resamp} and P_{pred}.

There are many ways to use a nativeness belief distribution P_{pred} in defining a resam-

pling distribution, and the choice depends in general on how many samples we are permitted. If we are to sample only a single conformation, we should maximize our chance of sampling a native one. The probability of seeing the fully native feature string $\mathbf{x}^*$ in any single Rosetta trajectory is, under our independence model, $P_{resamp}(\mathbf{x}^*) = \prod_{i=1}^{k} P_{resamp}(X_i = x_i^*)$. But we are not sure, a priori, of which values are in fact the native ones. Under our belief distribution P_{pred}, the chance that $\mathbf{x}$ is the native feature string is (making the same independence assumption for beliefs as for sampling distributions) $P_{pred}(\mathbf{x}) = \prod_{i=1}^{n} P_{pred}(x_i)$. The *expected* chance of seeing the native in any given sample, with respect to our beliefs, is then $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) P_{resamp}(\mathbf{x})$. Maximization of this expectation with respect to P_{resamp}, subject to normalization constraints, can be solved in closed form using a standard Lagrangian multiplier argument. The maximum is given by $P_{resamp}(\hat{\mathbf{x}}) = 1$ for $\hat{\mathbf{x}} = \operatorname{argmax} P_{pred}(\mathbf{x})$ and 0 elsewhere. Given just a single sample, the optimal strategy is to try our single best guess for the native feature string.

At the other extreme, in the limit of infinite samples, the chance of sampling feature string $\mathbf{x}$ at least once is the step function $\mathcal{I}(P_{resamp}(\mathbf{x}) > 0)$, which takes the value 1 if $P_{resamp}(\mathbf{x}) > 0$ and 0 otherwise. The expected chance of seeing a native structure is then $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) \mathcal{I}(P_{resamp}(\mathbf{x}) > 0)$. This expectation reaches its optimum value of 1 whenever $P_{resamp}(\mathbf{x}) > 0$ for all $\mathbf{x}$ such that $P_{pred}(\mathbf{x}) > 0$. For very large numbers of samples, the optimum strategy is to spread sampling as evenly as possible. If, for instance, we are permitted as many samples as there are joint feature strings, the optimal strategy is to try each string exactly once.

For intermediate numbers of samples, a closed-form solution is more difficult to obtain. In addition, one is not typically sure, a priori, of how much sampling one is going to do. We choose to interpolate between the two extremes by minimizing the expected log number of samples required to sample a single native string, $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) \log(1/P_{resamp}(\mathbf{x}))$. This is equivalent to maximizing $\sum_{\mathbf{x}} P_{pred}(\mathbf{x}) \log(P_{resamp}(\mathbf{x}))$, in which the term $\log(P_{resamp}(\mathbf{x}))$ interpolates between the objective functions $P_{resamp}(\mathbf{x})$, which grows linearly in the value of P_{resamp}, and $\mathcal{I}(P_{resamp}(\mathbf{x}) \neq 0)$, which jumps immediately to 1. Solving this optimization

yields $P_{resamp}(\mathbf{x}) = P_{pred}(\mathbf{x})$, which is the strategy that we have adopted.

We force Rosetta to sample according to the updated P_{pred} feature distribution both by applying stochastic constraints and by picking new fragments. The former approach proves more effective for beta topology, pairing, and contact features, while the latter approach is the most successful for torsion features.

3.4.1 Stochastic constraints

A beta contact can be enforced by means of a bridge in the *fold tree* [Bradley and Baker, 2006] with an attendant chainbreak introduced in a nearby loop. A register can be enforced by means of one or more bridges. In general, b+1 bridges will be required to constrain a register with b bulges, one in each bulge-free segment.

In order to effect a desired feature distribution, models are generated using different sets of bridge constraints. Each Rosetta search trajectory begins with a random draw of constraints from a hierarchical data structure called a *constraint tree*. The leaves of a constraint tree are constraints. The non-leaf nodes are of two kinds: *aggregator nodes* and *selector nodes*. Selector nodes have their outgoing edges labeled with probabilities. Each selector node selects at most one of its children at random according to the distribution on its outgoing edges and passes to its parent the constraints passed up from this child. If the edge probabilities sum to one, exactly one child will always be selected; if they do not, it is possible that no child will be selected. An aggregator node aggregates constraints from each of its children and passes them up to its parent.

The constraint tree to enforce a distribution over beta structures has five levels. The root node is a selector node which selects among topologies according to the probabilities emerging from the topology predictor. Within each topology, an aggregator node adds in constraints for each of the pairings that compose the topology. Within each pairing, a selector node selects among registers according to the probabilities computed by the register predictor. Within a register, an aggregator node adds in one constraint from each contiguous bulge-free region. Finally, within a bulge-free region a selector node picks

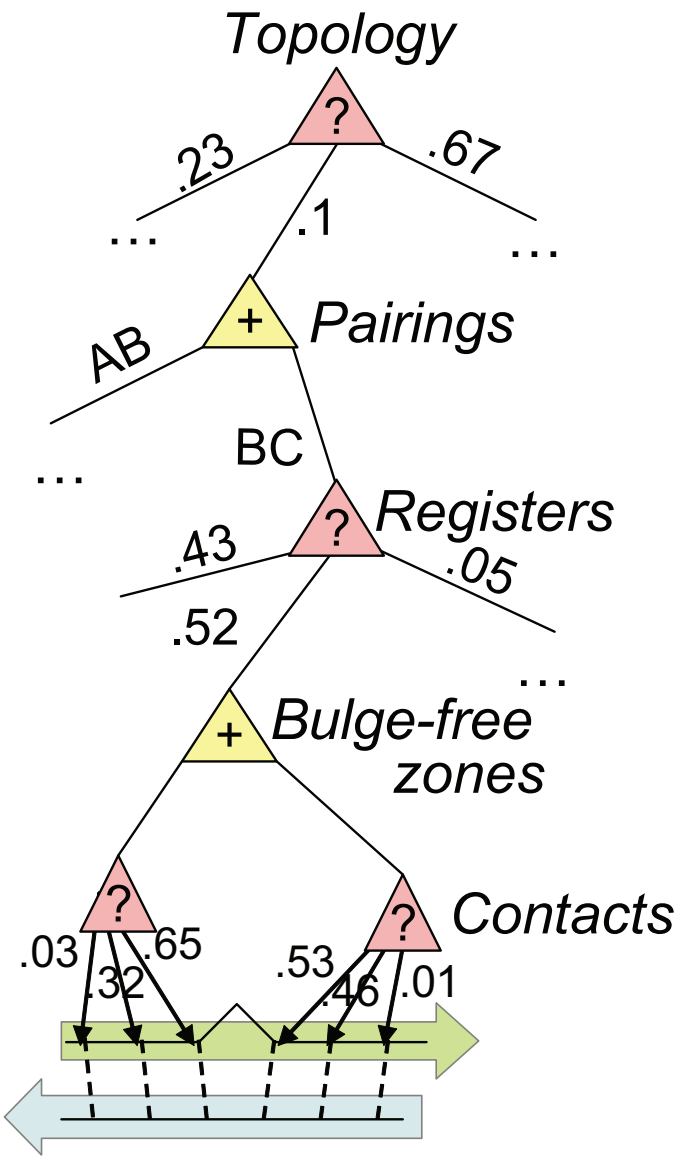


Figure 3.4: The constraint tree to effect a predicted distribution over all three levels of beta sheet features. Aggregator nodes are marked with a "+" and selector nodes are marked with a "?". The selection probabilities labeling the outgoing edges from the three levels of selector nodes going from top to bottom are chosen by the topology predictor, the register predictor, and the contact predictor, respectively.

a beta contact to enforce according to the probabilities from the contact predictor. An example constraint tree for a protein with three strands (A, B, and C) is shown in Figure 3.4. Only one path through the tree is shown, in which strands A and B pair and B and C pair, and a register is chosen for the BC pairing with a single beta bulge, but constraints are stochastically passed up from other paths as well.

3.4.2 Stochastic constraints for torsion angles

It is possible in principle to use the constraint tree approach for torsion constraints as well, although the approach has serious drawbacks; we describe it here mainly to highlight the strengths of the fragment repicking method.

The constraint tree to enforce a torsion distribution is simple in structure. The root is an aggregator node with a selector node child for each torsion feature. The selector node for torsion feature X_i has five leaf children, one for each possible Ramachandran bin. Constraints are therefore chosen independently for each torsion angle X_i. It remains to determine the selection probabilities $\{b_i^j\}_{j=1}^5$. b_i^j is the chance that value x_i^j will be enforced. The selection probabilities must sum to at most 1; $1 - \sum_{j=1}^5 b_i^j$ is the probability that no constraint will be activated for feature X_i. The torsion predictor gives us a target marginal sampling distribution $P_{pred}(X_i)$. We could certainly arrive at this distribution by setting $b_i^j = P_{pred}(X_i = x_i^j)$, but then we would be constraining every single torsion angle on every run. With this strategy, the chance of constraining all native features on a given run is vanishingly small (the chance of a fully native feature string is higher without constraints because native feature values are positively correlated). Furthermore, even if all constraints happen to be correct, constraining every single torsion angle doesn't give Rosetta the mobility it requires to form strand pairings and other global structural features. In order to maximize Rosetta's search mobility and to allow beneficial feature correlations, we apply as *few* constraints as possible in order to hit our target distribution P_{pred}. If, for instance, $P_{pred}(X_i) = P_{samp}(X_i)$, we do not enforce any constraints for X_i at all.

The desired selection probabilities can be computed by means of a simple constrained

optimization. If $P_{samp}(x_i^j)$ is the sampling rate of x_i^j, then the new sampling rate under selection distribution $\{b_i^j\}_{j=1}^5$ will be $b_i^j + (1 - \sum_{j=1}^5 b_i^j) P_{samp}(x_i^j)$. We wish to minimize $\sum_{j=1}^5 b_i^j$, the chance of applying a constraint for feature X_i, subject to the conditions $P_{pred}(x_i^j) = b_i^j + (1 - \sum_{j=1}^{m_i} b_i^j) P_{samp}(x_i^j)$ for all j, $\sum_{j=1}^{m_i} b_i^j \leq 1$, and $b_i^j \geq 0$ for all j. These conditions permit a one-manifold of solutions: $b_i^j = P_{pred}(x_i^j) - \alpha P_{samp}(x_i^j)$ for any α. It can easily be verified that when all b_i^j are assigned using this formula, $P_{pred}(x_i^j) = b_i^j + (1 - \sum_{j=1}^{m_i} b_i^j) P_{samp}(x_i^j)$ for all j. The higher the value of α, the less constrained X_i will be; in fact, α is exactly the probability that no constraint is enforced at X_i. But α is limited by the non-negativity constraints on b_i^j. Its maximum permitted value is $\min_j P_{pred}(x_i^j)/P_{samp}(x_i^j)$. This value of α yields the selection probabilities that we employ.

Unfortunately, this torsion constraint enforcement scheme does not grant perfect control over the torsion distribution. The relation between our target probabilities and the observed sampling rates in a resampling round with torsion constraints is shown in Figure 3.5.a. In this plot, we first begin to see the limitations of the independence model of sampling. In some cases we are quite far from hitting our target distribution. The primary reason is the presence of correlations between features. For instance, constraining a residue to have its torsion angles in the helical region of the Ramachandran plot (region A) often has the side effect of forcing adjacent residues to be helical as well. This effect motivates the move to a Markovian model of the feature distribution. In a Markovian model, random variables in a set are assumed to be independent of all other random variables when conditioned on their immediate neighbors in a chain. For variables $\{X_1, X_2, \ldots, X_n\}$, a joint distribution $P(X_1, X_2, \ldots, X_n)$ can be decomposed as the product of local conditional probabilities $P(X_1)P(X_2|X_1)P(X_3|X_2)\cdots P(X_n|X_{n-1})$. In order to capture effects from both the left and the right, we represent the joint distribution as a mixture of chains flowing left and right:

$$P(X_1, X_2, \ldots, X_n) = \frac{1}{2}\left[P(X_1)\cdots P(X_n|X_{n-1}) + P(X_1|X_2)\cdots P(X_n)\right].$$

The marginal distribution $P(X_i)$ can now be calculated by

$$\frac{1}{2}\left[\sum_{X_{i-1}} P(X_{i-1})P(X_i|X_{i-1}) + \sum_{X_{i+1}} P(X_{i+1})P(X_i|X_{i+1})\right].$$

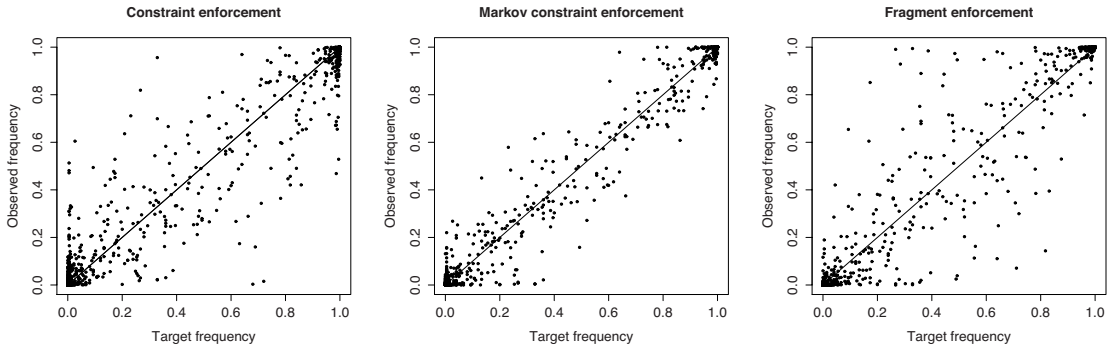
This model is a reasonable approximation of the distribution of torsion features, since the conformation of a residue is most immediately affected by the conformations of its predecessor and successor in the chain. We wish to determine the selection probabilities $\{b_i^j\}_{j=1}^{m_i}$ to make the marginal distribution of X_i match the target $P_{pred}(X_i)$ under the conditional distributions given by P_{samp} .

The key is to assume that we have already applied stochastic constraints to X_i's neighbors, X_{i-1} and X_{i+1}, to make their marginal distributions match $P_{pred}(X_{i-1})$ and $P_{pred}(X_{i+1})$. We also assume that the conditional distributions are unaffected by constraints, in the sense that if X_{i-1} is constrained to be x_{i-1}^j, and X_i and X_{i+1} are unconstrained, the distribution we'd expect to observe of X_i is $P_{samp}(X_i|X_{i-1} = x_{i-1}^j)$. By our simplistic mixture of left- and right-flowing chains, we mix the contributions of X_{i-1} and X_{i+1} with equal weights. Then the distribution we expect to observe of X_i, given whatever stochastic constraints have been applied to its neighbors but before any have been applied to X_i, is

$$P'_{pred}(X_i) = \frac{1}{2}\left[\sum_{X_{i-1}} P_{pred}(X_{i-1})P_{samp}(X_i|X_{i-1}) + \sum_{X_{i+1}} P_{pred}(X_{i+1})P_{samp}(X_i|X_{i-1})\right].$$

Now we can simply choose selection probabilities for X_i to bring $P'_{pred}(X_i)$ up to our target $P_{pred}(X_i)$, using exactly the same method that we used to bring $P_{samp}(X_i)$ up to $P_{pred}(X_i)$ in the previous section. By taking pairwise conditional effects into account in applying stochastic constraints, we achieve sampling rates significantly closer to their targets (Figure 3.5.b).

Unfortunately, stochastic enforcement of torsion constraints introduces a host of problems. Enforcing too many constraints in a given run can deny Rosetta the search mobility it requires to minimize energy. Constraint of a residue to a rare torsion bin limits the number of fragments from the fragment pool available as moves for that residue. Most seriously, the independent enforcement of torsion constraints means that contradictory constraints



(a) (b) (c)

Figure 3.5: Observed sampling rates versus targets for torsion feature values under (a) constraint enforcement, (b) Markovian constraint enforcement, and (c) fragment repicking. Target rates are given by P_{pred}.

are sometimes constrained simultaneously. For instance, adjacent residues might be constrained to be helix and strand. In the previous chapter, we avoided this difficulty by constraining only a few torsion features, generally distant in sequence. In this chapter, we present a new approach: fragment repicking.

3.4.3 Fragment repicking

Rosetta sampling rates for torsion features are closely correlated with rates of occurrence of those features in the set of fragments used for Rosetta sampling. We can therefore change Rosetta sampling rates significantly by repicking fragments. If P_{pred} is our target distribution, with marginal distribution $P_{pred}(X_i)$ for each torsion feature X_i, then we repick fragment files in such a way that the rate of occurrence of each value for feature X_i in the fragment file closely matches the rate given by $P_{pred}(X_i)$. The fragment files are picked using a simple greedy quota-satisfaction method.

The fragment-picking method of distribution enforcement has several important advantages over stochastic constraints. First, it provides more fragments for rare native features, increasing the likelihood that one of them will be near the native geometry. Second, and most significantly, it sidesteps some of the inadequacies of the independence model. When the marginal distributions in P_{pred} are matched, correlations between nearby torsion fea-

tures come along for free within the fragments. Rather than a combination of helical and strand residues, fragments will generally consist of all helical or all strand residues.

The fragment-picking method has one major disadvantage: correlation between fragment probabilities and sampling rates is only loose, so tight control over feature frequencies cannot be achieved by this method alone (Figure 3.5.c). However, tests including corrective constraints on under-sampled feature values (using the techniques described in Section 3.4.2) have not resulted in improvements to our method. Allowing Rosetta to under-sample features from the repicked fragments may in fact be a valuable check on our predictions if they are in strong disagreement with Rosetta's low resolution energy function. Fragment repicking remains the most effective method we have found for modification of torsion feature distributions.

3.5 Results and Discussion

We present results demonstrating both the accuracy of our nativeness predictors and the success of our resampling method in predicting protein structures. All results are from a benchmark set of 28 proteins ranging in size from 51 to 128 residues. For each test protein, nativeness predictors are trained only on the other proteins in the benchmark, so no testing is performed on training data.

3.5.1 Nativeness predictor accuracy

Structures generated by Rosetta do contain native feature values at higher rates, on average, than non-native values, but by updating Rosetta beliefs using energy information, we significantly increase the number of native feature values sampled at higher rates. Across our benchmark, most native torsion values were more likely in the updated distribution P_{pred} output by our nativeness predictors than in P_{samp} (Figure 3.6).

The nativeness predictor generally identifies native features more accurately than any single meta-feature—the feature value for which P_{pred} is highest is more likely to be native than the feature value for which other meta-features are highest (or lowest, in the case of

energy-based meta-features). This is shown in Figure 3.7, where we display the number of native torsion values correctly identified by P_{pred} and by two different meta-features for two different proteins, overlaid on a histogram of the number of native torsion values present in models generated in the initial sampling round. The yellow arrow indicates the native structure, with all torsion angles correct. The red arrow indicates the number of native torsion values correctly identified by P_{samp}. The blue arrow indicates the number of native torsion values correctly identified in the lowest energy model (*minE*). The purple arrow indicates the number of native torsion values correctly identified by P_{pred}. By incorporating information from both sources using fitted weights, the nativeness predictor P_{pred} performs better than either one alone. Results are similar for most other proteins in our benchmark. Figure 3.8.a shows the number of native torsion feature values missed by P_{samp} (Rosetta's prior beliefs), versus by P_{pred} (the updated beliefs from our nativeness predictor). For 24 out of 28 proteins in the benchmark set, P_{pred} performs as well or better.

In order to compare the accuracy of our nativeness predictor methodology against a standard benchmark, we specialized to secondary-structure prediction and trained a per-residue secondary structure predictor for comparison against Psipred [Jones, 1999], a standard sequence-based predictor, with accuracy defined as the fraction of residues for which the native value was given the highest probability. Psipred's prediction was used as a meta-feature in this predictor, so training could have recapitulated Psipred by placing all weight on this meta-feature to the exclusion of all others. Instead, it distributed weight between Psipred, P_{samp}, and various energy terms. Figure 3.8.b shows that our joint predictor is more accurate on our benchmark set, echoing previous results indicating that low-resolution tertiary structure prediction can inform secondary structure prediction [Meiler and Baker, 2003]. Mean prediction accuracy is 88.4% for our predictor, as compared to 84.5% for Psipred. Accuracy increases in 22 of 28 targets. The computation time required for our method may make it impractical for use as a secondary structure predictor—our predictions were performed using 20000 Rosetta samples for each target—but this test does give some indication of the power of energy information in native feature prediction. This infor-

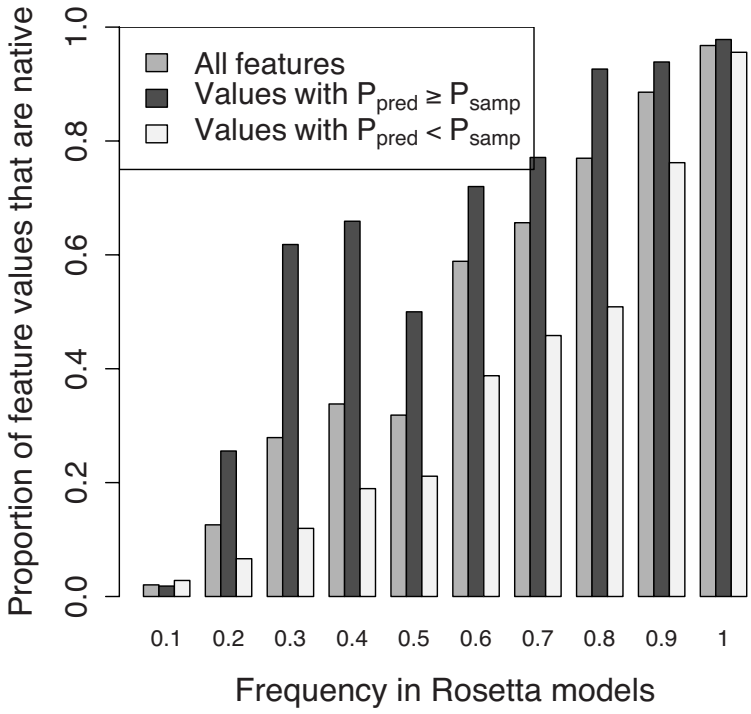
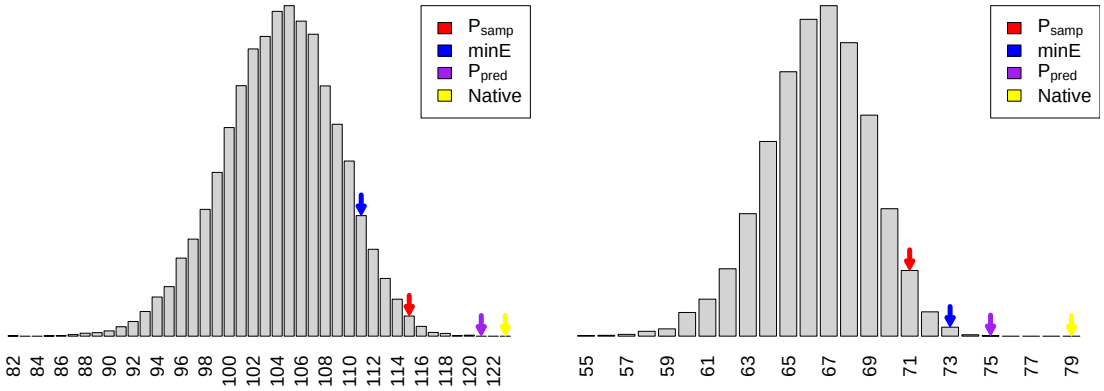
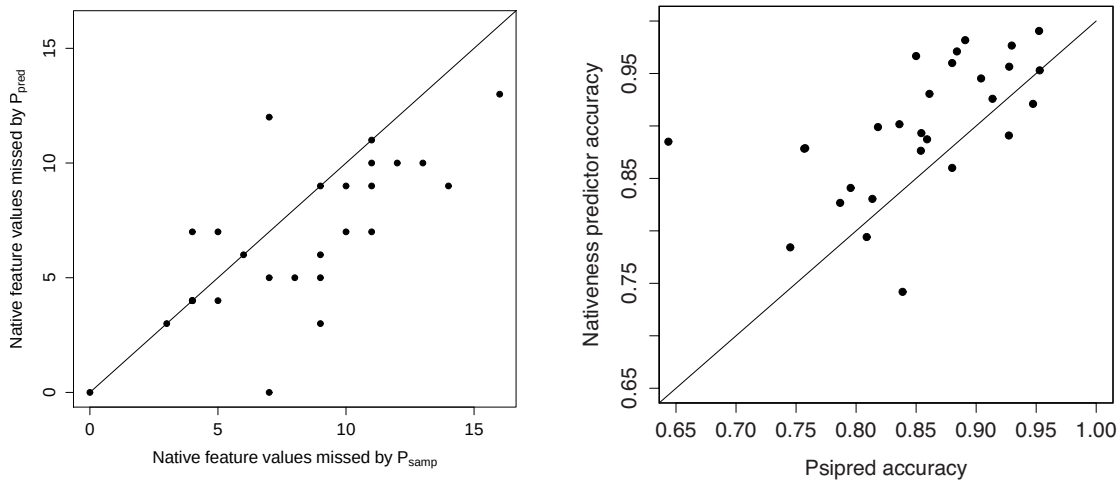


Figure 3.6: Subclassification of torsion feature values from Figure 3.2 by P_{pred}. Feature values given higher probability by P_{pred} than by P_{samp} are significantly more likely to be native. The central gray bars are the same as in Figure 3.2.



(a) (b)

Figure 3.7: (a) Number of native feature values for 1acf identified by several different meta-features. Red arrow: number of native feature values identified by P_{samp}. Blue arrow: *minE*. Purple arrow: P_{pred}. Yellow arrow: native. Each column of the histogram shows the number of 1acf models from a pool of 20000 generated by Rosetta that had the indicated number of native torsion feature values. (b) Native feature value counts for 1mky.



(a) (b)

Figure 3.8: (a) Number of native feature values misidentified by P_{samp} and P_{pred}. (b) Secondary structure predictor accuracy on 28-protein benchmark.

mation is clearly not available to a predictor like Psipred which only takes sequence data within a local window into account.

Our resampling strategy is to substitute P_{pred} for P_{samp} as Rosetta's sampling distribution, so the most important measure of the accuracy of P_{pred} is its effectiveness in this capacity. This can be estimated in advance of performing the actual resampling. In section Section 3.3.3, we defined sampling efficiency to be the chance of sampling an all-native feature string in a single Rosetta search trajectory. Assuming that features are sampled at least partially independently, the sampling efficiency of P_{pred} can be estimated as $\prod_{i=1}^{n} P_{pred}(x_i^*)$ for all native feature values x_i^*. The log ratio (to base ten) between this sampling efficiency and the sampling efficiency of P_{samp} is shown for torsion features in Figure 3.9 and for topology and pairing features in Figure 3.10. The expected efficiency gains for torsion angle features are rough estimates, since some native torsion feature values are in fact highly correlated. The efficiency increases for beta topology features are more realistic, since there is only one topology feature per protein and hence no correlation effect. The hashed bars in Figure 3.10 indicate the additional expected efficiency gain from resampling

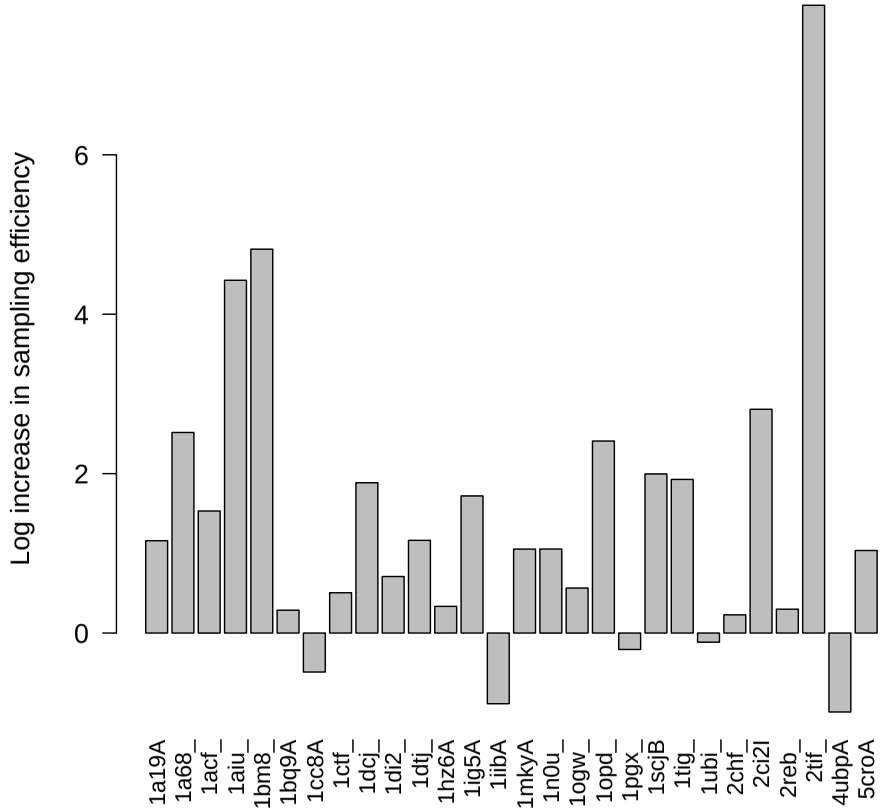


Figure 3.9: Predicted increase in sampling efficiency by protein for torsion features.

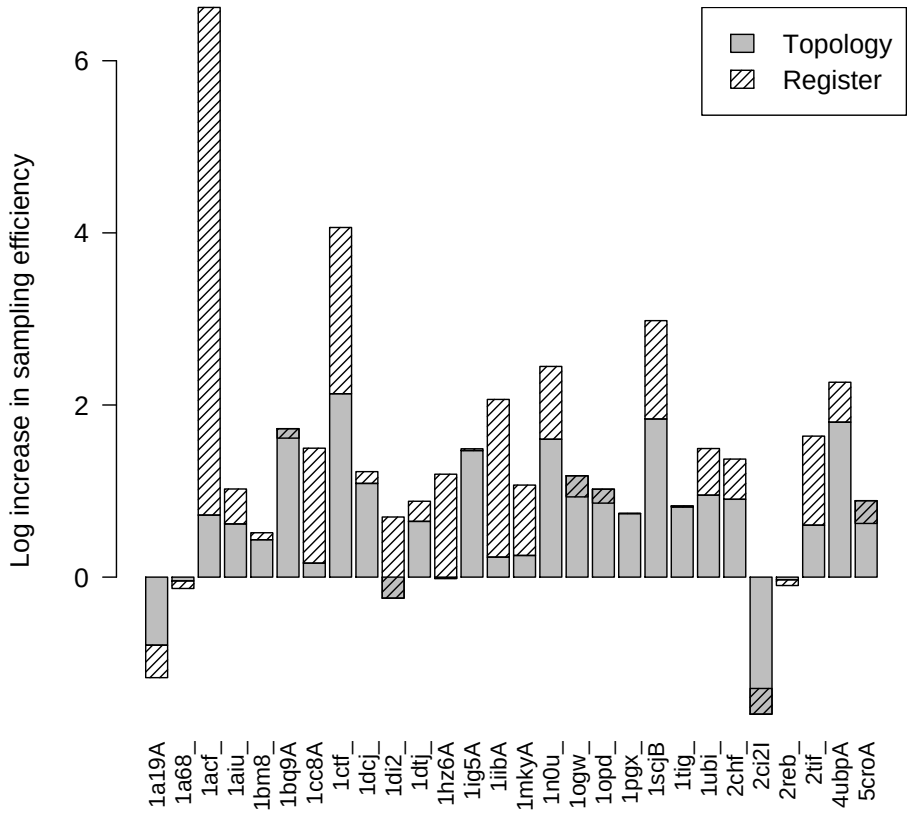


Figure 3.10: Predicted increase in sampling efficiency by protein for beta pairing features. Gray bars indicate efficiency changes due to topology resampling and clear hashed bars indicate efficiency changes due to register resampling. The register bars begin where the topology bars end and occasionally go in the opposite direction, in which case gray and hashed overlap. For instance, in 1di2 the sampling rate of the native topology decreases slightly, but this decrease is more than made up for by the increase in sampling of the native registers within that topology.

of pairing features. P_{pred} placed a median 5.3-fold higher probability on the native topology than P_{samp} (the median sampling rate for native topologies was 0.07). In many cases, the increase was much greater. P_{pred} further placed a median 2.25-fold higher joint probability on the co-occurence of all the native registers within the native topology (the median sampling rate for individual native registers within native topologies was 0.44).

3.5.2 Resampling

We employ two techniques to guide Rosetta search toward the target feature distribution P_{pred}: fragment repicking to enforce torsion feature distributions, and stochastic beta bridge constraints to enforce topologies, registers, and contacts. We found per-residue secondary structure features difficult to constrain by these methods so these features were not used in the resampling round.

For each of the 28 benchmark proteins, ranging in size from 51 to 128 residues, we generated 20000 models as input to our resampling methods. Topology, pairing, contact, and torsion predictors were trained from these data sets. To prevent training on the test set, a different predictor was trained for each protein in the benchmark from the model sets for the other 27 proteins. We repicked fragments for each protein based on the distribution produced by the torsion predictor. Three resampled sets were then generated: one with the repicked fragment files (*frag*), one with stochastically constrained beta pairing features (*beta*), and one with both (*frag+beta*). In each case, 10000 new models were sampled for each protein.

Figure 3.11 shows a histogram of the 1^{st} percentile RMSD for the resampled populations, and Figure 3.12 shows a similar histogram of the median RMSD of the best 1% of models by energy. We compared to a control population with standard fragments and no constraints and to a positive control population in which purely native beta pairing feature values were enforced—the best possible beta pairing feature distribution—and torsion features were sampled based on the repicked fragments. 1% RMSD is a sample-size independent measure of the lower limit of RMSDs achieved in a round of search. The median

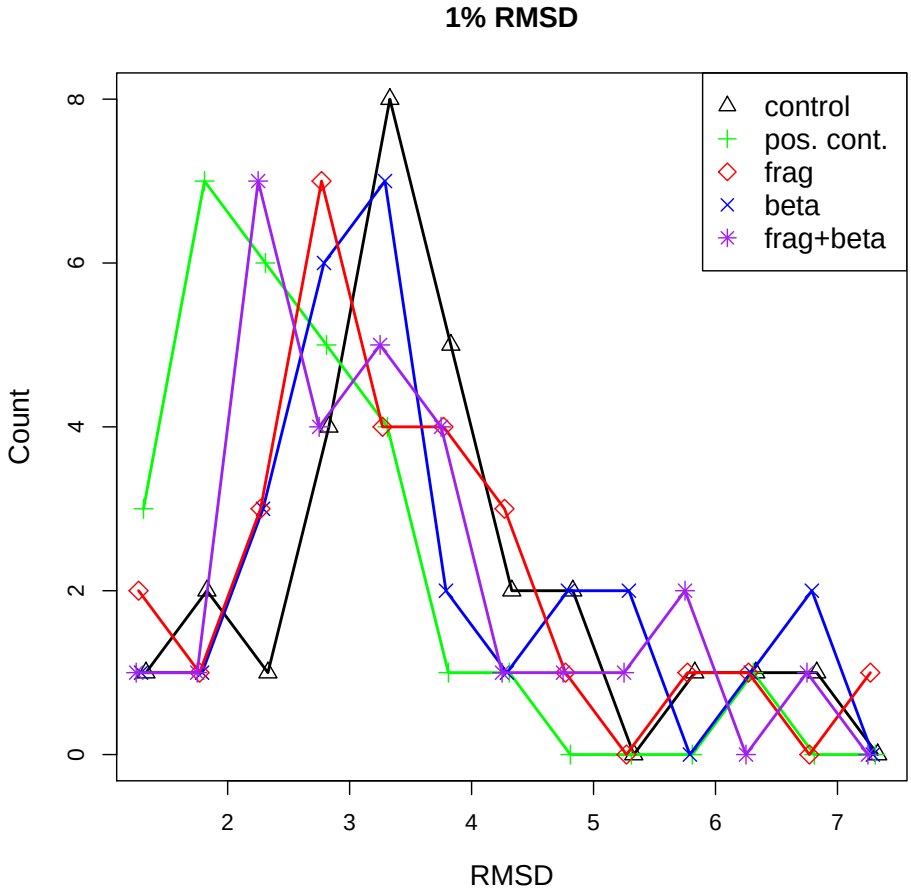


Figure 3.11: Histogram of 1^{st} percentile RMSD for a benchmark set of 28 alpha/beta proteins among models generated by fragment repicking ("frag"), beta topology resampling ("beta") and both ("frag+beta"), compared with a control set with no constraints and a positive control set in which the native beta pairings were enforced.

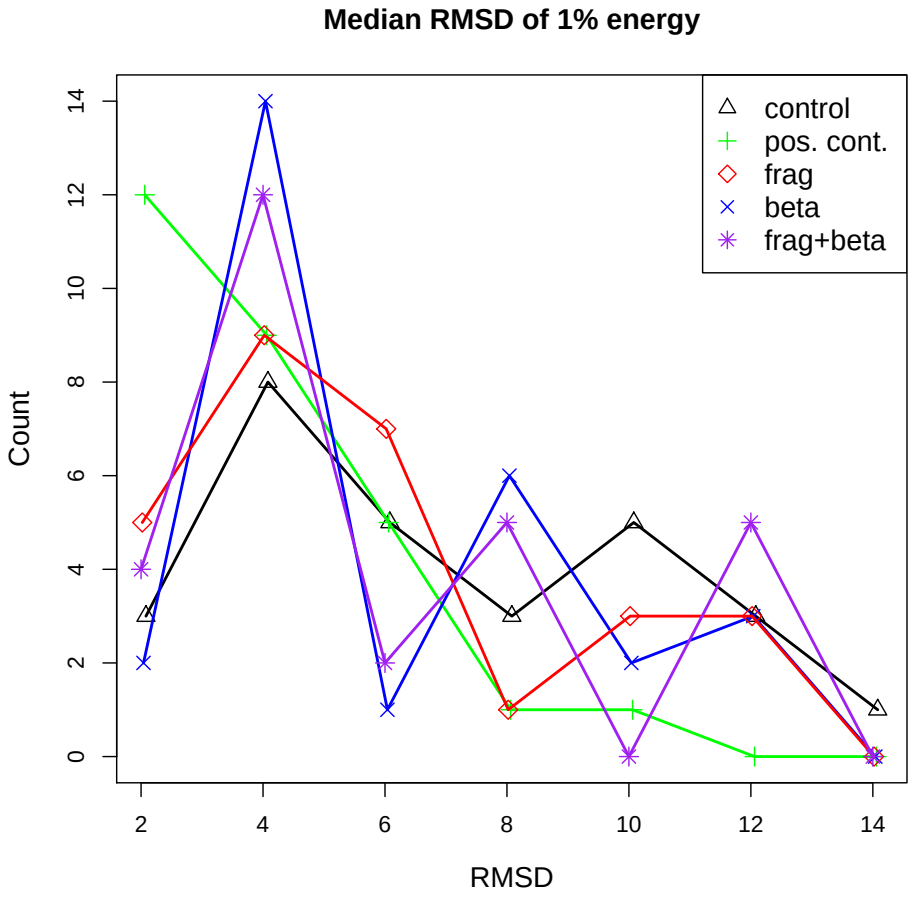


Figure 3.12: Histogram of the median RMSD of models in the 1^{st} percentile by energy among models generated by fragment repicking ("frag"), beta topology resampling ("beta") and both ("frag+beta"), compared with a control set with no constraints and a positive control set in which the native beta pairings were enforced.

	1% RMSD					Median RMSD of 1% energy				
	Control	Frag	Beta	Frag+beta	Pos.	Control	Frag	Beta	Frag+beta	Pos.
1di2	2.68	2.31	3.25	2.20	1.96	4.80	3.69	4.59	4.17	2.40
1dtj	2.89	2.16	2.52	2.11	2.55	4.54	2.81	3.04	2.89	3.00
1dcj	3.90	3.34	2.46	2.45	1.99	6.03	5.02	3.99	3.84	2.46
1ogw	3.12	3.10	3.23	3.13	1.99	4.52	4.64	4.12	3.95	2.35
2reb	1.23	1.10	1.23	2.06	1.26	1.36	1.32	1.43	2.84	1.28
2tif	3.16	2.84	4.02	3.77	2.04	4.75	5.60	4.78	4.95	2.66
1n0u	3.73	7.32	3.09	3.07	2.16	7.60	11.66	3.82	3.92	2.74
1hz6A	2.39	2.63	2.17	2.09	1.61	3.32	3.12	3.25	3.22	2.67
1mkyA	3.74	3.61	3.97	3.99	3.24	6.45	5.38	5.52	5.62	4.62
1a19A	3.50	3.21	6.93	5.85	2.97	6.58	6.17	12.61	11.69	7.11
1a68	6.44	5.86	6.85	6.80	6.23	11.33	11.28	8.89	8.40	9.37
1acf	6.91	4.66	5.11	4.64	3.56	13.65	9.95	12.12	11.19	5.82
1aiu	1.72	1.50	2.15	1.69	1.88	2.08	1.70	3.67	2.02	3.50
1bm8	5.63	4.32	6.21	5.52	4.47	11.64	8.77	11.15	11.33	6.33
1cc8A	2.72	2.56	2.73	2.59	1.63	3.66	3.55	4.45	4.99	1.98
1bq9A	4.73	4.02	4.51	3.79	3.50	8.16	6.80	7.93	6.97	6.49
1ctf	4.28	3.56	3.42	3.07	3.19	9.86	5.11	8.04	4.13	5.13
1ig5A	3.02	2.63	2.73	2.33	2.76	4.57	3.65	4.82	3.55	4.39
1iibA	3.13	3.81	3.35	3.70	2.55	9.74	11.24	8.53	11.08	4.17
2ci2I	4.54	6.22	5.18	5.36	2.43	9.57	6.81	7.28	8.76	3.69
2chf	3.46	2.98	3.31	2.98	2.56	6.43	3.29	10.72	8.23	3.80
1opd	3.62	2.81	2.90	2.38	2.01	4.57	3.85	4.26	3.78	3.49
1pgx	1.55	1.56	1.56	1.42	1.29	2.84	2.81	2.06	2.25	1.57
1scjB	2.86	2.50	3.62	3.48	1.88	6.87	2.93	7.39	7.14	2.62
1tig	3.93	3.48	3.14	3.01	2.05	11.42	4.66	4.13	4.00	3.10
1ubi	3.03	2.71	2.96	2.73	1.33	9.06	3.33	4.07	3.47	1.53
5croA	3.21	3.77	2.87	2.91	2.36	8.26	9.34	4.62	7.52	3.96
4ubpA	4.30	4.32	4.58	4.43	3.47	9.53	10.48	10.59	11.50	5.17

Table 3.3: Results from a 28 protein benchmark. The results in the first five columns show the 1^{st} percentile RMSD for resampled populations in which fragments were repicked according to the output of the torsion predictor ("*frag*"), beta topology, registers, and contacts were stochastically constrained according to the output of the beta sheet feature predictors ("*beta*"), or both ("*frag+beta*"). Positive controls were also generated using repicked fragments and all native beta pairing constraints. The results in the rightmost five columns show the median RMSD of models in the 1^{st} percentile by energy.

RMSD of the best 1% of models by energy is a sample-size independent measure of the quality of the lowest-energy Rosetta models.

All three of the resampled distributions were shifted toward lower RMSD when compared to the controls (in black). Fragment repicking alone ("*frag*") performed quite well, decreasing the 1% RMSD by a median 0.32Å, but the modes of the distributions suggest that while both *frag* and *beta* improve on Rosetta, their combination *frag+beta* yields the greatest gains. Several of the improvements of the *frag+beta* results over the controls were particularly striking. 1acf improved by 2.27Å to 4.64Å, 1dcj improved by 1.45Å to 2.20Å, and 1opd improved by 1.24Å to 2.38Å. However, beta topology resampling also led to worse predictions for some targets; in fact, the median improvement in 1% RMSD was smaller (by 0.22Å) than for fragment repicking alone due to significant losses on a few targets. These difficult cases can be observed in the right tail of the *frag+beta* histogram. In some of these cases, the positive control results, which employ repicked fragments along with all-native fold tree constraints, were worse than results with repicked fragments alone. This suggests that these failures may result from limitations in Rosetta's fold-tree protocol, possibly from difficulties in closing the chainbreaks that must be introduced when fold tree constraints are added.

The largest gains were observed in the median RMSD of the low energy models. The combination of fragment repicking and beta topology resampling yielded a median improvement of 0.68Å over the controls and a mean improvement of 0.92Å. This reflects the fact that for targets in which Rosetta samples several different competing topologies, beta feature resampling increases sampling of the native topology so that lower energies for this topology are achieved.

Numeric results data for all the proteins in our benchmark are given in Table 3.3.

3.6 Conclusion

In this chapter we have introduced a simple statistical model for incorporating energy information in the prediction of native structural features. The model can be adapted to any

class of discrete-valued features. When applied to per-residue secondary structure features, it yields greater prediction accuracy on our test set than Psipred. When applied to per-residue torsion features, it identifies native values more often than Rosetta sampling. It also yields notably accurate results in the prediction of native beta strand topologies, where the predicted distribution, P_{pred}, places a median 13.6-fold higher probability on the native topology with all native registers than the Rosetta sampling distribution.

The primary application for our nativeness predictors is a resampling method that improves on Rosetta. In a number of the proteins in our benchmark, the native register for a pairing was present in the model population but was never present in the sub-population with the native topology. Our nativeness predictors were often able to identify these two native feature values separately so that they appeared together in the resampling round. Our resampling methods, which constrain Rosetta search according to the output of the nativeness predictors, significantly outperform plain Rosetta ab initio search. The benefits from resampling of torsion features and of beta topology and pairing features appear to be cumulative, although topology resampling did run into several problem cases. Resampling using the combination of feature types achieved lower RMSDs for most targets and lower RMSDs among the low-energy models typically used as Rosetta predictions. Improvements were significant within the 1–5Å range of the best Rosetta models for most of these targets—improvement of the median RMSD of low energy models averaged 0.92Å. Our experiments indicate that fragment repicking along with beta topology resampling leads to significant gains over plain Rosetta results. However, the Rosetta methodology for enforcing pairings—modifications of the fold tree—introduces new sampling challenges, most notably the closure of chainbreaks. This places a limit on the gains that can currently be achieved by beta topology resampling. With advances in the Rosetta fold tree protocol, the accuracy of our beta topology, pairing, and contact predictors has the potential to translate into even more significant improvements over fragment repicking alone.

The nativeness predictor methodology extends naturally to any new class of discrete-valued structural features, as long as the sampling distribution of these features can be

controlled in some way in the resampling round. Our methods might be extended to include any number of other structural features, such as helix packing or rotamer features, or more global topological motifs.

Although in this chapter we concentrate our efforts on ab initio modeling, the application of our resampling method to comparative modeling would be straightforward. The principle is very much the same—from an initial pool of candidate conformations, perhaps derived from a set of different templates, native-like feature values would be identified using nativeness predictors and enriched in a subsequent resampling round. Nativeness predictors for comparative modeling might take into account meta-features relating to template information, for instance the proportion of templates which have the feature value. New feature types specific to comparative modeling might also be developed. One particularly promising possibility is to create a set of local alignment features, one for each residue (or gap-free block of residues). The alignment feature for a residue would take values in the possible template residues to which the target residue might be aligned. An initial sampling round in which models are generated for many possible alignments would give energy information that could be used in a nativeness predictor to identify the correct alignment.

Chapter 4

Thesis Conclusion

We have defined a new framework for protein structure prediction, which we call "resampling," that encompasses a variety of previous algorithms. A resampling method aims to extract information from previously generated models to guide further rounds of search. In this thesis we have drawn a distinction between "structure-based" and "feature-based" resampling methods. Methods in the former category concentrate future search around the most successful models seen in earlier stages of search; methods in the latter category recombine *features* of models seen in earlier search. A number of successful algorithms have been introduced which fall into the structure-based category, but structure-based resampling methods do suffer from certain pitfalls. They are generally limited to exploration of areas of conformation space already seen in the initial round of search; to explore new areas, they must take small, incremental steps away from areas known to them, and the direction in which to explore is not always clear. Several feature-based resampling algorithms exist, but by and large they are genetic algorithms, which recombine features blindly from the best structures in the previous round. Furthermore, the feature space representations seldom encapsulate global structural properties.

In this thesis, we have introduced two new feature-based resampling algorithms. As is inherently the case for feature-based methods, they avoid the pitfalls of structure-based methods by explicitly recombining features, hence exploring regions of conformation space never encountered in the initial round. But rather than randomly recombining features of the most successful *structures*, as in genetic algorithms, they commit wholly to the

idea of *feature*-based resampling and explicitly set out to identify successful features. Our first method, which we call "native feature selection," uses feature selection techniques to extract a few likely native features from models generated in the first round of search. These features are then enriched in the resampling round. The method produces a few notable successes, but also several notable failures. Our second method, which we call "nativeness prediction," uses a statistical "nativeness predictor" trained to predict the chance that each feature value ever observed in the initial sampling round is native. The method generalizes naturally to many different classes of structural features; we apply it to torsion features, secondary structure features, and a three-tiered hierarchy of beta topology features in order to address both local and global structure. Resampling using the output of the nativeness predictors yields gains over plain Rosetta search both larger and more consistent than the gains achieved with our first method. In significantly improving a state-of-the-art ab initio structure prediction algorithm, our method itself achieves state-of-the-art performance.

The work in this thesis opens a number of promising avenues for future research. Most significantly, our methods could very easily be extended to comparative modeling—structure prediction using a template protein whose structure has been experimentally determined. As more and more proteins have their structures experimentally determined, more and more targets have close sequence homologs. As a result, the structure prediction field seems to be moving increasingly toward comparative modeling. For our methods to remain relevant, generalization to comparative modeling, as outlined in Section 3.6, is the natural next step.

The core principle of our resampling work—that statistics derived from an initial sampling round are informative about local structural features—has the potential to be a powerful and broadly applicable tool in protein structure prediction. The high dimensionality and multiple minima that make high resolution protein structure prediction difficult to solve using traditional methods provide an excellent application for modern machine learning methods. The intersection between the two fields is just beginning, and we are excited to see further developments.

References

S. F. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment tool. *Journal of Molecular Biology*, 215:403–410, 1990.

S. F. Altschul, T. L. Madden, A. A. Schäffer J. Zhang, Z. Zhang, W. Miller, and D-J Miller. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

C. B. Anfinsen, E. Haber, M. Sela, and Jr. F. W. White. The kinetics of the formation of native ribonuclease during oxidation of the reduced polypeptide domain. *Proceedings of the National Academy of Sciences, U.S.A.*, 47:1309–1313, 1961.

C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.

D. Baker and D. A. Agard. Kinetics versus thermodynamics in protein folding. *Biochemistry*, 33:7507–7509, 1994.

D. Baker and A. Šali. Protein structure prediction and structural genomics. *Science*, 294:93–96, 2001.

T. L. Blundell, B. L. Sibanda, M. J. Sternberg, and J. M. Thornton. Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*, 326:347–352, 1987.

R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. M. Strauss, and D. Baker. Rosetta in CASP4: Progress in ab initio structure prediction. *Proteins*, 45:119–126, 2001.

M. J. Bower, F. E. Cohen, and Jr. R. L. Dunbrack. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *Journal of Molecular Biology*, 267:1268–1282, 1997.

J. U. Bowie and D. Eisenberg. An evolutionary approach to folding small alpha-helical pro-

teins that uses sequence information and an empirical guiding fitness function. *Proceedings of the National Academy of Sciences, U.S.A.*, 91:4436–4440, 1994.

G. E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions (with discussion). *Journal of the Royal Statistical Society Series B*, 13(1):1–45, 1951.

J. Boyan and A. W. Moore. Learning evaluation functions to improve optimization by local search. *The Journal of Machine Learning Research*, 1:77–112, 2001.

P. Bradley and D. Baker. Improved beta-protein structure prediction by multilevel optimization of nonlocal strand pairings and local backbone conformation. *Proteins*, 65:922–929, 2006.

C. Branden and J. Tooze. *Introduction to protein structure*. Garland Publishing, Inc., 19 Union Square West, New York, NY 10003-3382, 2nd edition, 1999.

W. J. Browne, A. C. T. North, D. C. Phillips, K. Brew, T. C. Vanaman, and R. L. Hill. A possible three-dimensional structure of bovine α-lactalbumin based on that of hen's egg-white lysozyme. *Journal of Molecular Biology*, 42(1):65–70, 1969.

C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6(1):76–90, 1970.

T. J. Brunette and O. Brock. Improving protein structure prediction with model-based search. *Bioinformatics*, 21 (Suppl. 1):66–74, 2005.

Y. Cui, R. S. Chen, and W. H. Wong. Protein folding simulation with genetic algorithm and supersecondary structure constraints. *Proteins*, 31:247–257, 1998.

T. Dandekar and P. Argos. Potential of genetic algorithms in protein folding and protein engineering simulations. *Protein Engineering*, 5(7):637–645, 1992.

R. David, M. J. Korenberg, and I. W. Hunter. 3D-1D threading methods for detecting remote protein homologies. *Pharmacogenomics*, 1:445–455, 2000.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics (with discussion)*, 32(2):407–499, 2004.

D. Fischer. 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. *Proteins*, 51:434–441, 2003.

A. Fiser, R. K. Do, and A. Šali. Modeling of loops in protein structures. *Protein Science*, 9:1753–1773, 2000.

K. Ginalski, A. Elofsson, D. Fischer, and L. Rychlewski. 3D-jury: a simple approach to improve protein structure predictions. *Bioinformatics*, 19:1015–1018, 2003.

A. Godzik. Fold recognition methods. *Methods of Biochemical Analysis*, 44:525–546, 2003.

E. S. Huang, S. Subbiah, J. Tsai, and M. Levitt. Using a hydrophobic contact potential to evaluate native and near-native folds generated by molecular dynamics simulations. *Journal of Molecular Biology*, 257:716–725, 1996.

L. Jaroszewski, L. Rychlewski, and A. Godzik. Improving the quality of twilight-zone alignment. *Protein Science*, 9:1487–1496, 2000.

L. Jaroszewski, L. Rychlewski, W. Li, and A. Godzik. Comparison of sequence profiles. Strategies for structural predictions using sequence information. *Protein Science*, 9:232–241, 2000.

D. T. Jones, W. R. Taylor, and J.M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.

D. T. Jones. Successful ab initio prediction of the tertiary structure of NK-lysin using multiple sequences and recognized supersecondary structural motifs. *Proteins*, 29:185–191, 1997.

D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292(2):195–202, 1999.

R. S. Judson, E. P. Jaeger, A. M. Treasurywala, and M. L. Peterson. Conformation searching methods for small molecules II: a genetic algorithm approach. *Journal of Computational Chemistry*, 14:1407, 1993.

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

K. Karplus, C. Barrett, and R. Hughey. Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.

J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips. A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature*, 181:662–666, 1958.

W. A. Koppensteiner and M. J. Sippl. Knowledge-based potentials—back to the roots. *Biochemistry*, 63:247–252, 1998.

T. Lazaridis and M. Karplus. Discrimination of the native from misfolded protein models with an energy function including implicit solvation. *Journal of Molecular Biology*, 288:477–487, 1999.

J. Lee, H. A. Scheraga, and S. Rackovsky. New optimization method for conformational energy calculations on polypeptides: Conformational space annealing. *Journal of Computational Chemistry*, 18:1222–1232, 1997.

M. R. Lee, J. Tsai, D. Baker, and P. A. Kollman. Molecular dynamics in the endgame of protein structure prediction. *Journal of Molecular Biology*, 313:417–430, 2001.

M. Levitt. Accurate modelling of protein conformation by automatic segment matching. *Journal of Molecular Biology*, 226:507–533, 1992.

M. A. Martí-Renom, A. C. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Šali. Comparative protein structure modeling of genes and genomes. *Annual Review of Biophysics and Biomolecular Structure*, 29:291–325, 2000.

J. Meiler and D. Baker. Coupled prediction of protein secondary and tertiary structure. *Proceedings of the National Academy of Sciences, U.S.A.*, 100(21):12105–12110, 2003.

J. Meiler, M. Müller, A. Zeidler, and F. Schmäschke. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *Journal of Molecular Modeling*, 7(9):360–369, 2001.

K. M. Misura, D. Chivian, C. A. Rohl, D. E. Kim, and D. Baker. Physically realistic homology models built with ROSETTA can be more accurate than their templates. *Proceedings of the National Academy of Sciences, U.S.A.*, 103:5361–6366, 2004.

J. Moult, J. T. Pedersen, R. Judson, and K. Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins*, 23(3):ii–iv, 1995.

J. Moult, K. Fidelis, A. Kryshtafovych, B. Rost, T. Hubbard, and A. Tramontano. Critical assessment of methods of protein structure prediction—Round VII. *Proteins*, 69(S8):3–9, 2007.

B. H. Park and M. Levitt. The complexity and accuracy of discrete state models of protein structure. *Journal of Molecular Biology*, 249:493–507, 1995.

B. H. Park and M. Levitt. Energy functions that discriminate X-ray and near native folds from well-constructed decoys. *Journal of Molecular Biology*, 258:367–392, 1996.

J. T. Pedersen and J. Moult. Ab initio structure prediction for small polypeptides and protein fragments using genetic algorithms. *Proteins*, 23:454–460, 1995.

K. Petersen and W. R. Taylor. Modelling zinc-binding proteins with GADGET: Genetic Algorithm and Distance Geometry for Exploring Topology. *Journal of Molecular Biology*, 325:1039–1059, 2003.

D. Petrey and B. Honig. Free energy determinants of tertiary structure and the evaluation of protein models. *Protein Science*, 9:2181–2191, 2000.

K. W. Plaxco, K. T. Simons, and D. Baker. Contact order, transition state placement and the

refolding rates of single domain proteins. *Journal of Molecular Biology*, 277(4):985–994, 1998.

B. Qian, A. R. Ortiz, and D. Baker. Improvement of comparative model accuracy by free-energy optimization along principal components of natural structural variation. *Proceedings of the National Academy of Sciences, U.S.A.*, 101(43):15346–15351, 2004.

B. Qian, S. Raman, R. Das, P. Bradley, A. McCoy, R. Read, and D. Baker. High-resolution structure prediction and the crystallographic phase problem. *Nature*, 450:259–264, November 2007.

Jr. R. L. Dunbrack and M. Karplus. Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. *Nature Structural Biology*, 1:334–340, 1994.

A. A. Rabow and H. A. Scheraga. Improved genetic algorithm for the protein folding problem by use of a Cartesian combination operator. *Protein Science*, 5(9):1800–1815, 1996.

C. S. Rapp and R. A. Friesner. Prediction of loop geometries using a generalized Born model of solvation effects. *Proteins*, 35:173–183, 1999.

RCSB. Annual report. `http://www.pdb.org`, July 2008.

R. Samudrala and J. Moult. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *Journal of Molecular Biology*, 275:895–916, 1998.

K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from frgments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of Molecular Biology*, 268:209–225, 1997.

M. J. Sippl, M. Hendlich, and P. Lackner. Assembly of polypeptide and protein backbone conformations from low energy ensembles of short fragments: development of strategies

and construction of models for myoglobin, lysozyme, and thymosin beta 4. *Protein Science*, 1:625–640, 1992.

M. J. Sippl. Knowledge-based potentials for proteins. *Current Opinion in Structural Biology*, 5:229–235, 1995.

J. Skolnick, D. Kihara, and Y. Zhang. Development and large scale benchmark testing of the PROSPECTOR 3.0 threading algorithm. *Proteins*, 56:502–518, 2004.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, 58(1):267–288, 1996.

UniProt Consortium. The Universal Protein Resource (UniProt). *Nucleic Acids Research*, 36:D190–D195, 2008.

A. Šali and T. L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, 234:779–815, 1993.

M. Vingron and M. S. Waterman. Sequence alignment and penalty choice. Review of concepts, case studies and implications. *Journal of Molecular Biology*, 235:1–12, 1994.

Y. N. Vorobjev, J. C. Almagro, and J. Hermans. Discrimination between native and intentionally misfolded conformations of proteins: ES/IS, a new method for calculating conformational free energy that uses both dynamics simulations with an explicit solvent and an implicit solvent continuum model. *Proteins*, 32:399–413, 1998.

B. Wallner, H. Fang, and A. Elosson. Automatic consensus-based fold recognition using Pcons, Pro Q, and Pmodeller. *Proteins: Structure, Function, and Genetics*, 6:534–541, 2003.

K. Wüthrich. Protein structure determination in solution by NMR spectroscopy. *Journal of Biological Chemistry*, 25(36):22059–22062, 1990.

L. Xu, R. Snchez, A. Sali, and N. Heintz. Ligand specificity of brain lipid-binding protein. *Journal of Biological Chemistry*, 271:24711–24719, 1996.

Y. Zhang. Template-based modeling and free modeling by I-TASSER in CASP7. *Proteins*, pages 108–117, 2007.

H. Zhou and Y. Zhou. Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. *Proteins*, 58:321–328, 2005.