Computational Methods for Protein Structure Prediction and Energy Minimization

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Christopher Daniel Kauffman

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

George Karypis

July, 2013

Acknowledgements

A doctorate is a fruit born not just of individual effort, but of a whole ecosystem of love, support, nagging, coercion, and determination. Perhaps the only task more daunting than writing a dissertation is trying to attribute proper credit to those that were integral to its creation, but I will at least try here.

I first met George Karypis as a green-gilled undergraduate in Csci 4041 and he proceded to wring better code out of me than I previously thought possible. He has been successfully wringing good things out of me ever since and I am confident that this document is not the last of it. Whether it was working late together to meet paper deadlines or eating dinner with your jubilant kids, George, you have always supported me and for that I am forever improved.

The members of my doctoral committee were inspiring figures which greatly influenced the work presented here. Yousef Saad was the first person to encourage me towards research and my career has taken a distinctly numerical flavor due to his kind guidance. Arindam Banerjee broadened my perspective of the machine learning realm considerably. Chris Cramer made computational chemistry accessible to someone who almost abandoned all hope upon entering freshman chemistry. Tom Luo showed me that optimization theory is a lens which clarifies enormous swaths of the computing landscape. Thank you all.

Candace, Dan, Amy, and Beth: this work is as much yours as mine, thus the dedication. You may find that odd being that you will likely understand about four pages of more than a hundred, but trust me. Without you, I and this document simply wouldn't be here.

Michelle Vigen, the love of my life, you have kept me sane for the last two years as I struggled to finish while taking on other responsibilities. Every Doctor deserves a Companion, and I could not have done better had I searched all time and relative dimensions in space. My quality of life and work would be a shadow of its present abundance without your patience and support.

A huge cast of honorable characters deserves mention for their friendship and comradery throughout my days in graduate school: Andy Blair, Greg Williams, Jeff Hedman, Kyle and Chelsea Griffin, Brad Lemke, Scott Agster, Getiria Onsongo, Ebbing DeJong, Matt Hildebrand, John Stever, Seong Oh, Elita Poplavska, Ewa Papajak, Tyler Yin, Xia Ning, Shilad Sen, Navaratnasothie Selvakkumaran, Eliana Salvemini, Ikumi Suzuki, Zi Lin, Sigve Nakken, and lately Jeremy Iverson, Dominque LaSalle, David Anastasiu, and Evangelia Christakopoulou. Thank you for all your smiles, kindness, and forgiveness when I mispronounce your last names.

Finally, a special acknowledgement is in order for Kevin DeRonne, Huzefa Rangwala, and Nikil Wale, my great allies in the eternal phud struggle. Aside from helping with courses, codes, experiments, and editing, you have colored my life in so many other ways: dating and subsequently marrying my sister, battling on the squash court, getting me in the door for my first academic job, tolerating me as a flatmate for several years, and driving accross a distinctly boring part of the country to get to another boring part of the country (Connecticut). I will forever cherish the moments we shared together in DTC 464 and all the moments we have left to share.

Now, as I once saw in the comments above the `main` function in a certain well-loved graph partitioning software, “Let the games begin...”

Dedication

To Mom, Dad, Amy, and Beth, for making me exist, for tolerating my inherited "accuratist" tendencies, for the love and encouragement you bestowed upon me that made this work both possible and thoroughly enjoyable: Thank You.

Abstract

The importance of proteins in biological systems cannot be overstated: genetic defects manifest themselves in misfolded proteins with tremendous human cost, drugs in turn target proteins to cure diseases, and our ability to accurately predict the behavior of designed proteins has allowed us to manufacture biological materials from engineered micro-organisms. All of these areas stand to benefit from fundamental improvements in computer modeling of protein structures. Due to the richness and complexity of protein structure data, it is a fruitful area to demonstrate the power of machine learning. In this dissertation we address three areas of structural bioinformatics with machine learning tools. Where current approaches are limited, we derive new solution methods via optimization theory.

Identifying residues that interact with ligands is useful as a first step to understanding protein function and as an aid to designing small molecules that target the protein for interaction. Several studies have shown sequence features are very informative for this type of prediction while structure features have also been useful when structure is available. In the first major topic of this dissertation, we develop a sequence-based method, called LIBRUS, that combines homology-based transfer and direct prediction using machine learning. We compare it to previous sequence-based work and current structure-based methods. Our analysis shows that homology-based transfer is slightly more discriminating than a support vector machine learner using profiles and predicted secondary structure. We combine these two approaches in a method called LIBRUS. On a benchmark of 885 sequence independent proteins, it achieves an area under the ROC curve (ROC) of 0.83 with 45% precision at 50% recall, a significant improvement over previous sequence-based efforts. On an independent benchmark set, a current method, FINDSITE, based on structure features achieves a 0.81 ROC with 54% precision at 50% recall while LIBRUS achieves a ROC of 0.82 with 39% precision at 50% recall at a smaller computational cost. When LIBRUS and FINDSITE predictions are combined, performance is increased beyond either reaching an ROC of 0.86 and 59% precision at 50% recall.

Coarse-grained models for protein structure are increasingly utilized in simulations and structural bioinformatics to avoid the cost associated with including all atoms. Currently there is little consensus as to what accuracy is lost transitioning from all-atom to coarse-grained models or how best to select the level of coarseness. The second major thrust of this dissertation is employing machine learning tools to address these two issues. We first illustrate how binary classifiers and ranking methods can be used to evaluate coarse-, medium-, and fine-grained protein models for their ability to discriminate between correctly and incorrectly folded structures. Through regularization and feature selection, we are able to determine the trade-offs associated with coarse models and their associated energy functions. We also propose an optimization method capable of creating a mixed representation of the protein from multiple granularities. The method utilizes a hinge loss similar to support vector machines and a max/L1 group regularization term to perform feature selection. Solutions are found for the whole regularization path using subgradient optimization. We illustrate its behavior on decoy discrimination and discuss implications for data-driven protein model selection.

Finally, identifying the folded structure of a protein with a given sequence is often cast as a global optimization problem. One seeks the structural conformation that minimizes an energy function as it is believed the native states of naturally occurring proteins are at the global minimum of nature's energy function. In mathematical programming, convex optimization is the tool of choice for the speedy solution of global optimization problems. In the final section of this dissertation we introduce a framework, dubbed Marie, which formulates protein folding as a convex optimization problem. Protein structures are represented using convex constraints with a few well-defined nonconvexities that can be handled. Marie trades away the ability to observe the dynamics of the system but gains tremendous speed in searching for a single low-energy structure. Several convex energy functions that mirror standard energy functions are established so that Marie performs energy minimization by solving a series of semidefinite programs. Marie's speed allows us to study a wide range of parameters defining a Go-like potential where energy is based solely on native contacts. We also implement an energy function affecting hydrophobic collapse, thought to be a primary driving force in protein folding. We study several variants and find that they are insufficient to reproduce native structures due in part to native structures adopting non-spherical conformations.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Upon successfully discerning the first protein structure in 1958 using X-ray diffraction, Sir John Kendrew commented on the nature of the structure in his group's *Nature* article [1].

> Perhaps the most remarkable features of the molecule are its complexity and its lack of symmetry. The arrangement seems to be almost totally lacking in the kind of regularities which one instinctively anticipates, and it is more complicated than has been predicated by any theory of protein structure. Though the detailed principles of construction do not yet emerge, we may hope that they will do so at a later stage of the analysis.

Though structural biology has advanced by tremendous strides, we are still perhaps in Kendall's "later stage of the analysis." The reasons for proteins assuming their fascinating 3D shapes is still the source of speculation and controversy. Deriving the folded shape of a protein purely through computational means is far from straight-forward. Yet the importance of proteins in biological systems cannot be overstated: genetic defects manifest themselves in misfolded proteins with tremendous human cost, drugs in turn target proteins to cure diseases, and our ability to accurately predict the behavior of designed proteins has allowed us to manufacture biological materials from engineered micro-organisms. All of these areas stand to benefit from fundamental improvements in computer modeling of protein structures.

Central to protein modeling are the multiple ways in which proteins are abstractly represented. Their *primary sequence* is a chain of twenty types of amino acids connected by peptide bonds. The primary sequence is often treated in computer models simply as a string of characters. The sequence is encoded fairly directly in DNA and translated to a chain of amino acid residues by machinery in all living cells. In fact, many proteins are involved in their own manufacture, by reading DNA, translating it to messenger molecules, or transcribing those messages into a new chain of amino acids. Despite an alphabet of only 20 amino acid types, proteins fold to incredibly diverse shapes, called their *tertiary structure*. These shapes provide a wealth of functionality to cells. Physically, proteins are believed to fold in order to minimize the free energy of the system in which they reside with the folded conformation near the energy minimum. In natural proteins, the primary sequence largely determines the folded tertiary structure. The number of protein sequences with known structures has dramatically increased and many of them are publicly available, primarily via the Protein Data Bank [2].

The availability of sequence and structure data has led to a wealth of computational approaches to address questions about proteins. The work presented here addresses three such questions. They are

1. Given only a protein's amino acid sequence, can we identify the elements in the sequence which will interact with drug molecules? Once identified, can they be used to improve drug modeling?

2. What are the modeling trade-offs between a fine-grained representation of a protein structure where every atom is present and a coarse-grained representation where atoms are merged based on the amino acid sequence? Can we find coarse-grained models that are as accurate in simulations as the fine-grained models?

3. What means are available to effectively locate low-energy conformations for a given protein? If we are willing to utilize a coarse-grained model of structure and energy, can energy minimization be sped up?

Our work here employs machine learning and optimization to answer each of these questions.

1.1 Interactions of Proteins with Small Molecules

In the first three chapters we study the following task: given a protein's sequence, predict which residues in the sequence, if any, will interact with small molecules. Small molecules are alternatively referred to as ligands. Many protein functions are carried out by binding small molecules. Knowing the residues which bind ligands enables lab researchers to make intelligent choices about which sequence mutations to explore when studying the function of a protein.

In Chapter 3 we develop LIBRUS, a predictor that leverages only information on the primary sequence of a protein and show that it is as accurate as a competing method which requires the full structure of a protein to predict binding residues. With LIBRUS, one can estimate the binding residues of a protein lacking a known structure. The insight of LIBRUS is that intrinsic features of the target sequence and information on the binding residues of homologous sequences can be effectively combined in a support vector learning machine.

A frequent task in drug development is to study where and how a small drug molecule binds to a protein. This requires an accurate representation of the protein's binding site which can be difficult to obtain if the protein structure has not been solved. To overcome this difficulty, homology modeling is often employed in which a surrogate *template* structure is leveraged to establish a model for a target protein. In Chapter 4 we show LIBRUS to be useful in this setting as it can guide the construction of the homology model so that the resulting protein's binding site is more accurate.

1.2 Protein Representation

Any molecular system requires an *in silico* representation in computer modeling which in turn requires a set of parameters governing the forces between objects in the model. A traditional means of selecting parameters for such models is to use many variations of the parameters in simulations to determine the values which reproduce experimental data. While ultimately any model should be usable in a simulation, determining parameters this way is incredibly costly and is generally not possible to do using full proteins. For models of proteins, an oft-used approximation is *decoy discrimination*: find model parameters that best separate correctly folded native proteins from incorrectly folded

decoys with both types of structures fixed. This approach sacrifices the dynamic character of the system but enables much quicker evaluation of the model, particularly for large systems. To further mitigate the tremendous size and complexity of proteins, there is a trend of employing *coarse-grained models*. Rather than model each atom, bond, and interaction individually, atoms are grouped into *beads* and the forces between atoms are aggregated to forces between beads.

In Chapter 5 we evaluate several such coarse-grained structure and energy models. We show that machine learning tools are ideal for this task as they can be used to discern model parameters that best discriminate properly folded proteins from misfolded decoys. The discriminative power of combinations of energy functions with coarse- or fine-grained representation gives us insight into how accurate we can expect the combination to be in a simulation setting. We are able to show that though some accuracy is lost in by going to coarse-grained models, there is promise for representations that utilize beads for both the main-chain and side-chain of each amino acid. We also show that complex energy functions do not necessarily generalize well: the discriminative power of 2-body energy interactions is by and large as accurate as 2- and 3-body interactions.

In Chapter 6 we develop a novel machine learning algorithm that, given several choices of coarse- and fine-grained beads, selects a mixed representation that best discriminates native and decoy proteins. This is a first step towards automatic model selection for coarse-grained representations. Our bead selection method builds upon the regularization path techniques employed in a variety of machine learning tools so that we can report a complete picture of the accuracy over a variety of mixed models.

1.3 Effective Energy Minimization

In the final section of this dissertation, we address the fundamental problem of determining a protein's native structure based on its sequence, the problem of full tertiary protein structure prediction. As many other methods do, we model the native structure as the minimum energy conformation. In a departure from most other methods, we use

a completely abstract representation for structure and energy which is founded in distance geometry. This allows structure prediction to be solved using convex optimization techniques, in particular semidefinite programming. Our framework is dubbed Marie and is developed in Chapter 8.

Marie is used to study two types of energy functions commonly employed for protein modeling. We show how to implement a Go-like potential in Chapter 9 which is a very simple class of energy function dependent only on contacts in the folded protein structure. The speed of Marie allows a comprehensive study of parameters associated with our Go-like potential, something that is typically too expensive for a standard molecular dynamics simulation to accomplish.

Go-potentials are a theoretical tool and of little use for true structure prediction due to their dependence on the native structure being available. The hydrophobic collapse theory provides a more useful structure prediction mechanism by explaining protein folding as the process of burying hydrophobic amino acids to the interior of its structure in order to avoid energetically unfavorable interactions with water. In Chapter 9, we implement an energy function in Marie which models hydrophobic collapse. We find that predictions are fast but the minimum energy structures are not close to native structures. We identify several deficiencies with the hydrophobic collapse energy function and establish directions for further development of Marie for *ab initio* structure prediction.

1.4 Related Publications

The work in this dissertation has been published in the following venues.

- Chris Kauffman and George Karypis. LIBRUS: combined machine learning and homology information for sequence-based ligand-binding residue prediction. Bioinformatics, 25(23), 3099–3107, (2009).

- Chris Kauffman, Huzefa Rangwala, and George Karypis. Improving Homology Models for Protein-Ligand Binding Sites. LSS Computational Systems Bioinformatics Conference. Stanford, CA, 2008.

- Chris Kauffman and George Karypis. Ligand Binding Residue Prediction. In *Introduction to Protein Structure Prediction: Methods and Algorithms.* Wiley, Volume 14, Hoboken, N.J. 2010.

- Chris Kauffman and George Karypis. Coarse- and fine-grained models for proteins: Evaluation by decoy discrimination. Proteins: Structure, Function, and Bioinformatics, 81(5), 754–773, 2013.

- Chris Kauffman and George Karypis. Marie: A Framework for Convex Optimization of Protein Structures. In preparation.

Chapter 2

Ligand-binding Residue Prediction

In the next three chapters, we explore methods to predict protein residues that interact with small molecules. The present chapter motivates the problem by describing potential uses for such information and proceeds to discuss methods advanced for prediction. In Chapter 3, we propose and implement LIBRUS which is a sequence-based approach to binding residue prediction. LIBRUS is contrasted with another current method which relies on predicted protein structure to help identify ligand-binding residues. Finally, in Chapter 4 we employ sequence-based predictions in a homology modeling task which shows that the predictions are presently accurate enough to improve down-stream performance.

2.1 Background on Ligand-binding

Recent advances in high-throughput sequencing technologies have continued to increase the gap between the number of proteins whose function is well-characterized and the proteins for which there is no experimental functional data. As a result, life sciences researchers are becoming increasingly more dependent on computational methods to infer the function of proteins. To address this challenge, a number of novel and sophisticated methods have been developed within the field of computational biology which are designed to predict different aspects of a protein's function.

Our focus is on methods that predict, from a protein's primary sequence, the *ligand-binding residues* that bind to small molecules. Small molecules interact with proteins in regions that are accessible and that provide energetically favorable contacts. Geometrically, these binding sites are generally deep, concave shaped regions on the protein surface, referred to alternately as clefts or pockets.

Identifying ligand-binding residues reliably aids the overall understanding of the role and function of a protein by using them to subsequently predict the types of ligands to which they bind and, in the case of enzymes, the types of reactions that are catalyzed. Moreover, knowledge of the residues involved in protein-ligand interactions has broad applications in drug discovery and chemical genetics, as it may be used to better virtually screen large chemical compound libraries [3] and to aid the process of lead optimization [4, 5]. In addition, the ligand-binding residues of a protein can be used to influence target-template sequence alignment in comparative protein modeling which has been shown to improve the quality of the 3D models produced for the target's binding site [6]. These quality improvements in the binding site's 3D model are critical to docking-based approaches for virtual screening [7].

2.2 Overview of Existing Methods

Predicting ligand-binding site residues from sequence information is similar to several site interaction prediction problems involving DNA [8, 9, 10], RNA [11, 12], and other proteins [13, 14, 15]. Existing approaches for identifying ligand-binding residues can be broadly classified into two groups which alternately use machine learning and sequence homology to solve the problem.

2.2.1 Machine Learning Approaches

A number of groups have employed supervised machine learning techniques for binding residue prediction. This involves using some proteins to develop a model of what constitutes a binding residue and then testing the model on an independent set of proteins. A variety of features and techniques have been explored but the consensus seems to be that sequence profiles and conservation are the important features and support vector machines provide the best discrimination.

Fischer and coworkers presented a method for functional residue prediction based on sequence features [16]. They studied prediction for residues contacting ligands and also for the more restrictive catalytic site residues as defined in the Catalytic Site Atlas [17]. A Bayesian-type learner was trained to produce the probability of a residue being a binder with the primary feature of interest being residue conservation. The authors introduced a new conservation measure, FRcons, which proved the most effective in their benchmark but achieved a precision of less than 30% at sensitivity equal to 50%.

Petrova and Wu performed a fairly comprehensive evaluation of machine learning algorithms and features useful for direct prediction of catalytic residues in a small set of proteins [18]. They found that support vector machines were the most powerful method for this task. The features they found to be most important were residue conservation, amino acid identity, entropy, and characteristics of the nearest cleft to a residue. The first three of these are sequence features which may be utilized even when no structure is available for the target. Features of clefts necessitate the target structure to be either known or predicted.

Youn et al. also studied the use of various features with support vector machines for catalytic residue prediction [19]. Their evaluation encompassed a large set of 987 protein domains from SCOP [20, 21] which they analyzed at the family, superfamily, and fold levels. They achieved a *ROC* of 0.866 for feature-only predictions at the family level. However, catalytic residues are a more restricted set than general ligand binders: only 1.1% of the residues are in the positive class in their study while 8.6% of residues in our data were in the positive class. The precision and recall reported at the family level by [19] is quite low: 16.6% precision at 15.1% recall. Feature ranking was done and they found that PSSMs and the information per position (IPP) reported by PSI-BLAST were most useful for prediction. Structural conservation was found to be the next best feature.

2.2.2 Homology-Based Approaches

The transfer of sequence properties, such as ligand binding status, to the target based on its alignment to templates is a common method for prediction. These techniques are often referred to as as *homology transfer* (HT) as properties of the target sequence are predicted by transferring them from homologous templates. Homology transfer is

a close relative of nearest neighbor methods frequently employed for machine learning tasks. The primary difference is that nearest neighbor methods typically deal with individual objects with feature representations, while in homology transfer predictions are made on a per residue basis but similarity search is done using whole sequences with the sequence alignments determining individual residue relations.

The *firestar* algorithm of [22] utilizes homology transfer and conservation scores to make binding residue predictions from sequence. A profile is calculated for a target using PSI-BLAST and significant alignments are searched in their FireDB which is composed of ligand-binding proteins. The resulting multiple sequence alignment is used to estimate conservation of target residues which are then predicted to be binders if they align to template residues which are binders. In *firestar*, profiles are used to estimate the reliability of alignments between target and templates to determine when transfer should occur, but not to directly characterize ligand-binding residues.

Brylinski and Skolnick recently introduced FINDSITE as a method for making predictions about protein-ligand interactions [23, 24]. The method belongs to the homology transfer category but uses structural measures of similarity rather than sequence alignment. FINDSITE identifies templates by threading the target sequence through candidate template structures and retaining high-scoring templates. The accumulated templates are then structurally aligned to the target structure. If the target structure is not available, it is predicted using one of several methods. The binding status of template residues is then transferred to target residues based on this structural correspondence.

The drawback of FINDSITE is that the target structure is required for the alignment of templates. In cases where the target structure is available, FINDSITE can exploit it well to make binding site predictions. However, when it is not available, predicting the structure of the target protein can be a computationally expensive proposition with no guarantees on quality.

Chapter 3

Evaluation of Ligand Binding Prediction Methods

Here we discuss methods that predict ligand-binding residues directly from sequence. Two categories of methods are illustrated: those that treat the problem as a supervised machine learning task and those that use alignment and homology relationships to make predictions. These two approaches are prototyped and compared to one another. Their complementary nature suggests combining the two methods. This is done in LIBRUS which outperforms both base methods.

3.1 Methods

In this section we describe prototype algorithms which represent the basic ideas behind most sequence-based binding residue predictors. We begin by discussing relevant features to both types of algorithms. Sequence alignment plays a central role in the homology-based method and is described subsequently. With these tools laid out, two prototype prediction methods are described: homology-based transfer and machine learning with support vector machines. LIBRUS combines these two approaches and is described in the last section. We also briefly discuss FINDSITE which uses predicted structures to make its predictions rather than direct predictions from sequence.

3.1.1 Relevant Sequence Features

The primary source of information about proteins of unknown structure is their amino acid sequence. Evolutionary information may be inferred from the sequence using PSI-BLAST which computes a substitution profile for each residue in the protein sequence [25]. This profile has two parts: a position specific scoring matrix ($PSSM$) and a position specific frequency matrix ($PSFM$). The $PSSM$ is a real-valued matrix of dimension $n \times 20$ where n is the length of the protein. A row represents the log-odds probability of each of the twenty amino acids occurring at that sequence position. The row of a $PSSM$ may be used directly as a feature vector for a residue as is done in the machine learning case or may be utilized along with the $PSFM$ in alignment scoring schemes as will be done for the homology-based transfer method.

Secondary structure in proteins are locally recurring structures which are commonly divided into three major classes: α-helices (H), β-sheets (S), and unstructured coils (C). Each residue of the protein may be assigned one of these classes based on its tertiary structure, a feature referred to as secondary structure elements (SSE) and encoded as an $n \times 3$ matrix. A popular and long-standing means of assigning SSE is the DSSP program of Kabsch and Sander [26]. Many methods have been studied to predict secondary structure from protein sequence and some studies have shown that these methods can positively impact downstream prediction tasks [27, 28]. A relatively recent approach using support vector machines is YASSPP [29] which produces, for each residue of a protein, a likelihood of being helix, sheet, and coil. This predicted secondary structure, referred to as SSP, is used as a surrogate for SSE when the true secondary structure is unavailable.

3.1.2 Alignment Techniques

Given two protein sequences, a core problem is to construct a sequence alignment. The scoring mechanism used to construct this alignment can have drastic effects on the constructed alignment similarity score assigned to two sequences. The profile-based alignment scoring scheme that we used is derived from the work on PICASSO [30] which was shown to be very sensitive in subsequent studies [31, 32]. Our own work aligns sequences by computing an optimal alignment using an affine gap model with

aligned residues i and j in sequences X and Y, respectively, scored using a combination of profile-to-profile scoring and secondary structure matching. The score is given by

$$\begin{aligned} S(X_i, Y_j) = &\sum_{k=1}^{20} PSSM_X(i,k) \times PSFM_Y(j,k) \\ &+ \sum_{k=1}^{20} PSSM_Y(j,k) \times PSFM_X(i,k) \\ &+ w_{SSE} \sum_{k=1}^{3} SSE_X(i,k) \times SSE_Y(j,k), \end{aligned} \tag{3.1}$$

where $PSSM$, $PSFM$, and SSE are the profile matrices and secondary structure elements described in Section 3.1.1. We will frequently deal with the case of aligning a target of unknown structure with a template of known structure. In this situation, predicted secondary structure (SSP) is used in place of true secondary structure for the target. The parameter w_{SSE} is the relative weighting of the secondary structure score which is set to $w_{SSE} = 3$ based on our experience [6].

3.1.3 Homology-Based Transfer

Alignment of protein sequences is a powerful tool which allows characteristics of one to be inferred from the other. This is the crux of homology-based methods. Given a target protein, a database of template sequences with known binding information is searched for high scoring alignments to the target. Once good templates are identified, a score is assigned to each residue in the target based on the number of template residues which aligned against it and are known to be ligand binders. This score is referred to as the *homology-based transfer score* or HTS.

There are a number of dimensions along which alignment and prediction may be adjusted including the scoring mechanism and weighting of template contributions to the final prediction. The alignment scoring of Section 3.1.2 may be applied in many alignment frameworks (see Chapter 11 of [33]). Our experience has been that local alignments provide the best results due to reporting only the best matching target-template subsequences which can increase the reliability of prediction. The top 20

alignments should be used with weighting for each residue based on the alignment score in a window of seven residues. We tried a variety of alternatives to this which are detailed in [34].

3.1.4 Support Vector Machine Prediction

In this method, the prediction problem is treated as a supervised learning problem whose goal is to build a model that can predict whether a residue is ligand-binding or not, a binary classification problem. In supervised learning, each object of interest is encoded by a feature vector and a model is learned that can predict the class based on those features.

Recent research on building models for predicting various structural and functional properties of protein residues in [29] and [10] has suggested training SVMs [35] on sequence features of each residue to classify the residue as a ligand-binder or nonbinder. Effective features include position specific scoring matrices ($PSSM$) and predicted secondary structure (SSP) in a window around each residue. Sliding windows are an easy way to expand feature vectors. The results shown later use a window of nine residues centered on the residue of interest and concatenated the $PSSM$s and SSPs of adjacent residues for a total of 207 features per residue ($9 \times (20 + 3)$). Window features which extended beyond the first or last residue of the sequence were assigned zero values. This feature representation is closest to that of [19] where $PSSM$s in a sliding window of size 21 were employed in one of their methods for the related problem of predicting a protein's catalytic residues.

One important aspect of combining different types of features is providing proper weights on them as their numerical ranges may vary greatly. In the results reported later, we combined features by weighting them to have equal norm. Examples of the norms and weighting are given in Table 3.1. Properly weighting the combination of features significantly enhanced the performance of the final model.

Table 3.1: Average Norms of Residue Features.

Statistic	PSSM	SSP	HTS
Average	13.53	2.00	0.07
Std. Dev.	3.88	0.53	0.11
Weight	1.00	6.75	207.00

Columns are position specific scoring matrices (PSSM), predicted secondary structure vector (SSE), and homology transfer scores (HTS). The bottom row is the weight used on these features in SVMs for sequence-based predictions.

3.1.5 LIBRUS: Combining SVM and Homology-based transfer

Direct prediction by SVMs and prediction by homology-based transfer utilize training information in different ways to make their predictions. SVMs utilize intrinsic features of the residue represented as $PSSM$s and SSPs with little context for the residue within the whole protein nor any relation of the containing protein to other proteins in the training set. Conversely, homology-based transfer solely relies on the global context of the residue: where it is located in alignments of the containing protein against other proteins and how many ligand-binding residues align against it. The different characteristics of the information utilized by the two approaches suggests that their combination can lead to a better overall predictor.

A simple linear combination of SVM and homology transfer scores may be used. With proper weights set on the two scores, this approach works rather well as will be seen in the results.

Alternatively, an SVM may be trained on the $PSSM$s and SSEs of the direct prediction method and the homology-based transfer scores of the HT method. The resulting hybrid predictor utilizes both types of features. We have built such a predictor called LIBRUS [34] which uses a total of $9 \times (20 + 3 + 1) = 216$ features weighted according to Table 3.1.

3.1.6 FINDSITE

The methods mentioned in the previous sections solely utilize sequence information for targets of unknown structure to directly predict ligand binding residues. Alternatively, the target structure can be predicted and then utilized to identify binding residues.

This is the approach taken in FINDSITE which is a recent approach to binding site identification [23]. The results of this method on one dataset are provided later to contrast the direct predictions made by sequence-based methods.

FINDSITE identifies a number of predicted binding sites with associated binding residues for each target. The prediction values for these correspond to the fraction of template structure residues that were identified as ligand binding and aligned against the target residue. Up to the first five predicted binding sites are reported in the results section. Some residues appear as part of multiple binding sites in the FINDSITE predictions and have different scores associated with them in the different sites. In those cases, the score from the first binding site a residue occurred in was used as this was typically the largest and most well defined predicted binding site.

3.2 Experimental Setup

3.2.1 Sequence Data

The sequence-based methods were evaluated on a dataset referred to as DS1 which consists of 885 protein chains (268,699 residues) that were derived from the RCSB Protein Data Bank in October of 2008 (PDB, [2]). The set of proteins in DS1 were selected so that they satisfy the following constraints: (i) has better than 2.5 Å resolution, (ii) is longer than 100 residues, (iii) has an unbroken backbone, and (iv) has at least five residues in contact with a ligand. Finally, the dataset was culled so that no two sequences have above a 30% sequence identity according to NCBI's blastclust program.

Ligands in our datasets were small molecules in contact with proteins identified by scanning the PDB using the 'has ligand' search option. DNA, RNA, and other large proteins were excluded as candidate ligands as were ligands with fewer than eight heavy (non-hydrogen) atoms. We required proteins to have ligand-binding residues with a heavy atom within 5 Å of a ligand. By this definition, 8.6% of DS1 residues are ligand-binding residues (positive class). In-house software was developed to identify ligands and ligand-binding residues.

Protein sequences were derived directly from the structures using in-house software. When nonstandard amino acids appeared in the sequence, the three-letter to one-letter conversion table from ASTRAL [36] version 1.55 was used to generate the sequence[1]. When multiple chains occurred in a PDB file, the chains were treated separately from one another. Profiles for each sequence were generated using PSI-BLAST version 2.2.13 [25] and the NCBI NR database (version 2.2.12 with 2.87 million sequence, downloaded August 2005). PSI-BLAST produces a position specific scoring matrix ($PSSM$) and position specific frequency matrix ($PSFM$) for a query protein, both of which are employed for our sequenced-based prediction and alignment methods. Three iterations were used in PSI-BLAST with the default e-value threshold for inclusion in the profile and default expectation value (options `-j 3 -h 2e-3 -e 10`).

True secondary structure (SSE) for each protein of DS1 was obtained using the DSSP program [26] while predicted secondary structure (SSP) was obtained using YASSPP [29]. YASSPP predicted the correct secondary structure for 83% of the residues in DS1.

In the homology-based transfer method, template proteins are assumed to have known structure and therefore SSE is available for them while the targets must use SSP as they have unknown structure. Care must be taken so that the encoding of SSE is compatible with SSP. A straightforward means of defining the SSE is, for each residue, assign 1 to the dimension corresponding to its true state and 0 to the other dimensions: e.g. for a true helix, the encoding would be $[1, 0, 0]$, a true sheet $[0, 1, 0]$, and true coil $[0, 0, 1]$. Our experience has been that a better means of encoding true SSE to compare it to YASSPP's SSP is the following. The average YASSPP vector of all true helices was computed. For a true helix, the SSE is assigned this average vector. Similar averaging steps for sheets and coils were computed and used for true secondary structure. This ensures that SSE and SSP are scaled similarly.

A second dataset, referred to as DS2, was derived from the set of proteins used to evaluate FINDSITE in [23]. DS2 consists of 564 proteins (136,316 residues) after eliminating those sequences with 35% identity or better to any sequence in DS1 according

[1] `http://astral.berkeley.edu/seq.cgi?get=release-notes;ver=1.55`

to BLAST. This dataset was used to illustrate the relative performances of LIBRUS and FINDSITE with LIBRUS using DS1 as training data. Sequence features for the members of DS2 were derived as they are in Section 3.2.1.

3.2.2 Evaluation Metrics

Three-fold cross validation is used on DS1 to assess how well the methods generalize. In each step, two sets of the data were used to learn a model and predictions were made on the remaining set of targets. This generated a single prediction of binding/nonbinding for each residue which was subsequently used in evaluation.

To generate homology-based transfer scores, all targets in set one used sets two and three as the template database and similarly for sets two and three. This amounts to having two thirds of the data as templates for training with the remaining third as the test set. This allows us to directly compare the performance achieved by direct SVM predictions, homology-based transfer, and LIBRUS as all methods use identical training and testing data. The same cross-validation approach was also used to compute the predictions for linear combination of homology-based transfer and SVM scores (Section 3.1.5).

We evaluated the performance of the different methods using the receiver operating characteristic (ROC) curve [37]. This is obtained by varying the threshold at which residues are considered ligand-binding or not according to value provided by the predictor. In the case of the SVM predictions, a continuous prediction value is produced which is the distance from a hyperplane optimized to separate the positive and negative classes. This is the threshold which is varied to produce the ROC curve. For homology-based transfer scores, the threshold to be assigned a ligand-binding residue is varied to produce the ROC curve. The area under the ROC curve, abbreviated *ROC* (note italics), summarizes the predictor behaviour: a random predictor has $ROC = 0.5$ while a perfect predictor has $ROC = 1.0$ so that a larger *ROC* indicates better predictive power.

For any binary predictor, the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) determines standard classification statistics which we use later for comparison. These are

$$\text{Precision} = \frac{TP}{TP + FP}, \text{ and} \tag{3.2}$$

$$\text{Recall} = \text{Sensitivity} = \frac{TP}{TP + FN}. \tag{3.3}$$

Fischer et al. noted in their study of functional residue predictions that analyzing only an ROC curve can be misleading in terms of the performance of the predictor [16]. As an alternative, they present precision vs. recall plots (called precision-sensitivity plots in their work, referred to as PR curves here) as a means to compare performance. We provide this measure as well, both graphically and summarized by the area under the PR curve, abbreviated PR (note italics).

Performance differences between FINDSITE and LIBRUS on DS2 are illustrated using the Welch's t-test. This test assumes the populations are normally distributed with potentially unequal variance and calculates a p-value that the mean of one is higher than the other. In our case, this corresponds to one method outperforming another. Welch's t-test was used in favor of Student's t-test as the latter assumes equal variance of the populations which may not be the case for the methods under consideration. The populations we analyzed are the ROC and PR scores of each protein according to the predictions of LIBRUS and FINDSITE. The test allows us to determine whether, on average, one of the two methods outperforms the other on per-protein identification of ligand-binding residues.

3.3 Results of Binding Residue Prediction

3.3.1 Performance of Direct Sequence-based Predictors

The performance of the prototype methods described in Section 3.1 on dataset DS1 are shown in Table 3.2 and Figure 3.1. The methods are grouped into three classes: SVM prediction, homology transfer, and combined. Comparing the best performance achieved by each of the classes, we see that the combined methods achieve the best overall results. Among the two methods that fall in that category, LIBRUS, which

Table 3.2: Cross-Validation Results on the DS1 Dataset

	Overall		Per Protein			
Method	ROC	PR	μ_{ROC}	σ_{ROC}	μ_{PR}	σ_{PR}
SVM with PSSM	0.7545	0.2637	0.7487	0.1492	0.2930	0.1722
SVM with PSSM, SSE	0.7737	0.2942	0.7648	0.1532	0.3177	0.1886
Homology Transfer	0.7845	0.4516	0.7581	0.1811	0.4024	0.2971
Linear SVM and HTS	0.8259	0.4792	0.8030	0.1666	0.4342	0.2838
LIBRUS	0.8334	0.4807	0.8066	0.1686	0.4374	0.2809

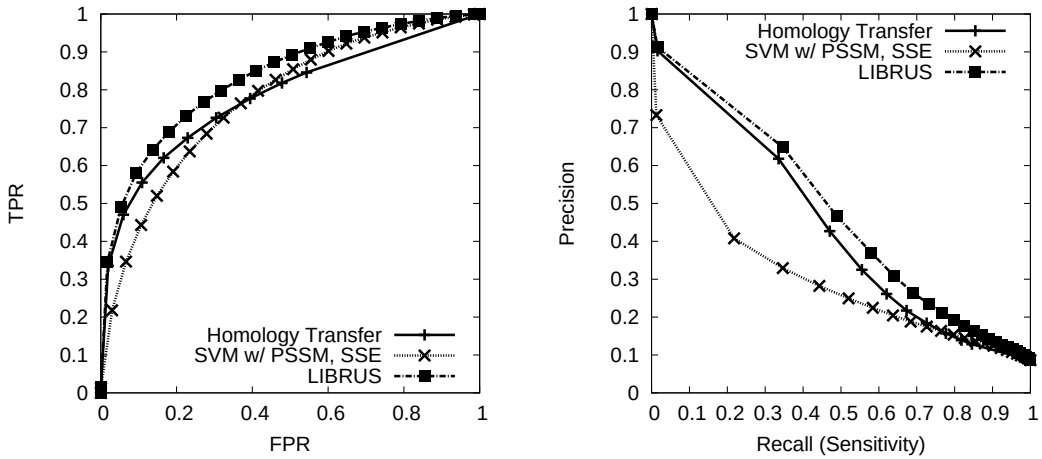
Three-way cross validation was used on the set of 885 proteins of the DS1 dataset. The overall area under curve is given for ROC and precision/recall (PR) curves in the first two columns. The per protein averages, μ, and standard deviation, σ, for these two statistics are also given.

uses SVM to combine this information, achieves the best overall results. Specifically, it achieves an overall $ROC = 0.8334$, which is better than the ROCs of 0.7737 and 0.7849 that were obtained by the SVM and homology-based transfer methods, respectively. Its performance in terms of the overall PR is also better, achieving a $PR = 0.4807$ compared to the PRs of 0.2942 and 0.4516 achieved by the other two classes of methods. These relative performance gains also hold when the experiments are evaluated in terms of the average per-protein ROC and PR. The performance of the simple linear combination of SVM and HTS scores also performs quite well, further re-enforcing the fact that coupling the two sources of information leads to a better overall predictor.

Comparing the other two classes of methods, we see that homology-based transfer outperforms the direct SVM-based approach that utilizes PSSM- and SSE-based features. The performance difference between these two schemes is more pronounced when the methods are evaluated in terms of their PR (both overall and per-protein).

Finally, the results of Table 3.2 show that when predicted secondary structure information is used to augment the PSSM-based features, the performance of the SVM-based method improves. This fact is in agreement with a number of studies that have shown that the inclusion of this type of information helps the performance of supervised learning methods [27, 28].

Figure 3.1: ROC and PR Curves of Some Sequence-based Predictors



(a) (b)

Curves are given for the overall performance on the DS1 dataset. (a) ROC curves and (b) Precision vs. Recall.

3.3.2 Performance of LIBRUS and FINDSITE

Performance measures for FINDSITE and LIBRUS predictions on the proteins in dataset DS2 are summarized in Table 3.3 while Figure 3.2 plots the ROC and PR curves obtained. Note that Tables 3.3–3.4 and Figure 3.2 also contain results for the scheme that combines the LIBRUS and FINDSITE predictions, which are discussed later in Section 3.3.3. Table 3.4 shows the results of a paired Welch's t-test comparing the methods. Comparisons on both ROC and PR are done in parts (a) and (b) of Table 3.4 respectively.

Examining the predictions of the various versions of FINDSITE and LIBRUS, in Table 3.3 we see that their overall prediction performance is quite close. The FINDSITE results using one site achieve the best PR (0.4955), whereas the FINDSITE results using three sites achieve the best ROC (0.8216). However, compared to the former method, LIBRUS achieves a better ROC (0.8169 vs 0.8088), whereas compared to the latter method, LIBRUS achieves a better PR (0.4565 vs 0.3760). The difference between FINDSITE and LIBRUS is somewhat more consistent when the per-protein results are

Table 3.3: Results on the DS2 Dataset.

	Overall		Per Protein			
Method	ROC	PR	μ_{ROC}	σ_{ROC}	μ_{PR}	σ_{PR}
FINDSITE 1 Site	0.8088	0.4955	0.7981	0.2040	0.4841	0.2978
FINDSITE 2 Sites	0.8187	0.4258	0.8043	0.1935	0.4360	0.2697
FINDSITE 3 Sites	0.8216	0.3760	0.8034	0.1852	0.3957	0.2436
FINDSITE 4 Sites	0.8182	0.3370	0.7970	0.1808	0.3620	0.2228
FINDSITE 5 Sites	0.8155	0.3074	0.7918	0.1716	0.3340	0.2055
LIBRUS	0.8169	0.4565	0.7982	0.1600	0.4165	0.2550
Combined	0.8617	0.5618	0.8410	0.1741	0.5324	0.2991

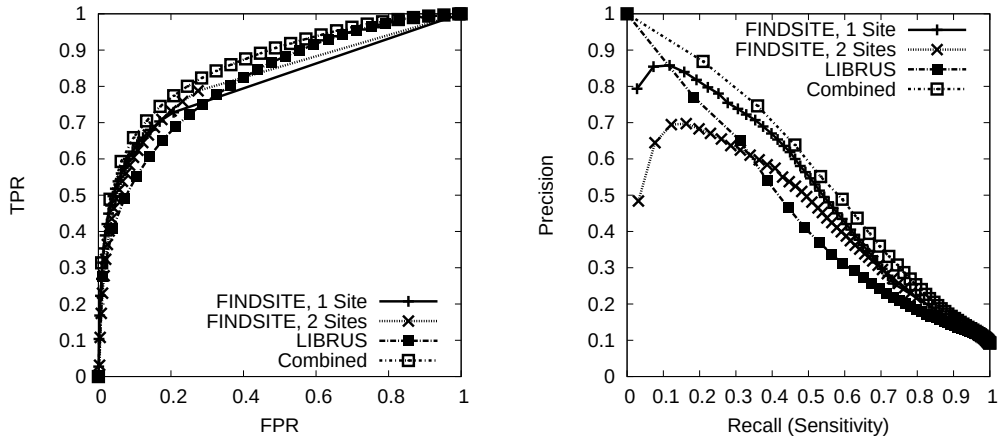
The performance of FINDSITE considering the first 5 binding sites and the best SVM method, LIBRUS, are shown. The dataset comprised 564 proteins from the FINDSITE benchmark that were sequence independent from the DS1 dataset that was used to train LIBRUS. The last row shows the results obtained by linearly combining the predictions produced by LIBRUS and FINDSITE 1 Site. For column descriptions, see Table 3.2.

considered, in which case the FINDSITE results using two sites lead to average ROC and PR (0.8043 and 0.4360) that are better than those produced by LIBRUS (0.7982 and 0.4165).

Figure 3.2 shows the ROC and PR plots graphically. According to part (a), the strength of LIBRUS is at higher false positive rates where it exceeds the TPR of FINDSITE. At low FPR, FINDSITE dominates LIBRUS with the crossing point at FPR=0.35 and FPR=0.40 for one and two sites respectively. In part (b), LIBRUS is seen to have better precision at very low recall, but falls below FINDSITE at 11% recall for one site and at 34% recall for two sites. At 50% recall, LIBRUS achieves 40% precision while FINDSITE achieves 55% and 49% precision for one and two sites respectively.

One aspect that we have not touched on empirically so far is the time required to make predictions. According to communications with the FINDSITE authors, running their program for a protein takes from 30 minutes to several hours. This is not surprising as FINDSITE needs to initially predict the structure of the protein and also identify good templates from their database. The amount of time required by LIBRUS to predict the ligand-binding residues of a protein is much lower. Based on the average

Figure 3.2: Comparison of FINDSITE and LIBRUS.



(a) (b)

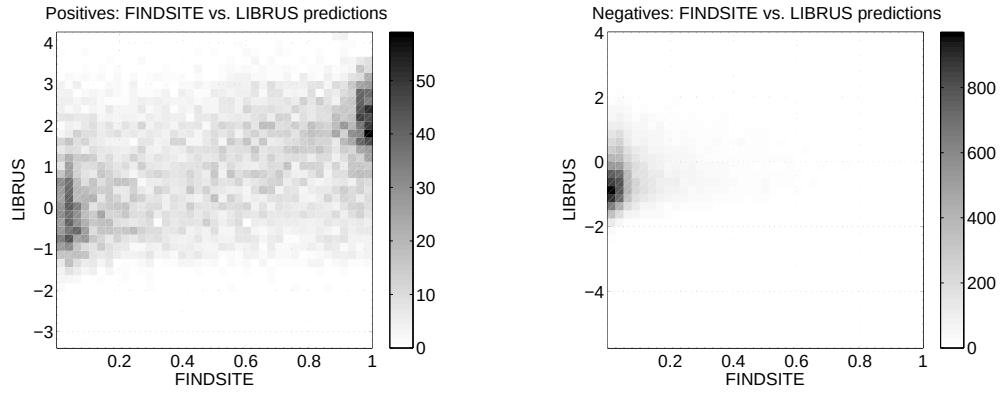
Overall comparison of FINDSITE to the sequence-only SVM learner developed in this work on the 564 independent proteins from the FINDSITE benchmark. (a) ROC curves of FINDSITE based on the top binding sites, the SVM approach, and the combined predictor. (b) Precision vs. Recall of the methods.

performance over many proteins, LIBRUS predictions can be made in under 10 minutes which encompasses profile generation, secondary structure prediction, alignment to the database, and final SVM prediction. A larger template database will lengthen this process somewhat, but we expect it to remain faster.

3.3.3 Complementary Nature of Sequence and Structure Predictions

While analyzing the nature of the predictions produced by FINDSITE and LIBRUS, we noticed that, though there is agreement on many of the residues they identified as being ligand-binding, there are enough differences to merit further inquiry. Figure 3.3 illustrates these differences by plotting the prediction scores produced by LIBRUS and FINDSITE (using one site) for the positive instances (ligand-binding residues) and the negative instances (nonbinding residues). In Figure 3.3(a) (positive class) we see that there are two clusters, one on the right and one on the left of the plot. The cluster on the right contains residues that FINDSITE predicts correctly, whereas the cluster on the left contains residues that FINDSITE mispredicts. The predictions produced by

Figure 3.3: Heatmap of FINDSITE and LIBRUS Prediction Values.



(a) (b)

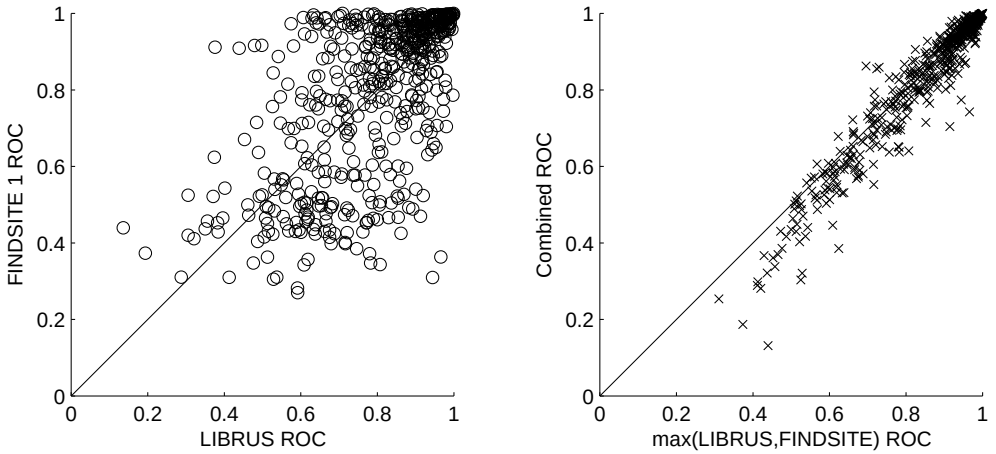
Heatmap illustrating FINDSITE and LIBRUS values on the positive class (a) and the negative class (b). The positive LIBRUS predictions on some mispredicted FINDSITE residues indicates LIBRUS may provide additional information in some cases. The correlations between FINDSITE and LIBRUS are 0.52 on the positive class, 0.27 on the negative class, and 0.48 overall. Note that residues which had FINDSITE predictions of zero were eliminated as they dominate the nonzero predictions.

LIBRUS are, to a large extent, in agreement for the right cluster (even though LIBRUS mispredicts some of these residues) but are split for the left cluster. LIBRUS predicts correctly (i.e., positive SVM score) a noticeable fraction of the residues that are falsely predicted as negative by FINDSITE. Overall, the Pearson correlation coefficient between FINDSITE predictions and LIBRUS predictions is 0.48.

Figure 3.4(a) illustrates how the above trend carries over to the whole protein. It plots the per-protein *ROC*s of LIBRUS and FINDSITE (one site) on DS2 against one another. The greatest density lies in the upper right corner where both methods achieve high *ROC*s. Points below the main diagonal indicate where LIBRUS outperforms FINDSITE while points above indicate the opposite. The large number of off-diagonal points shows that if information from both predictors can be exploited, overall predictions may be improved.

Motivated by the above differences, we linearly combined the prediction scores of LIBRUS and FINDSITE. The results of this combined predictor are reported at the bottom of Table 3.3, and in Figure 3.2. The combined predictor achieves higher overall *ROC* and *PR* than either approach on its own. Also notable is the superior per-protein

Figure 3.4: Complementary Nature of FINDSITE and LIBRUS Predictions.



(a) (b)

(a) LIBRUS vs. FINDSITE. The abundance of off-diagonal entries indicate LIBRUS and FINDSITE outperform one another on certain proteins and must be exploiting different signals for those proteins. (b) The *ROC* of the combined method is plotted against the maximum of LIBRUS and FINDSITE and achieves nearly the same performance.

prediction rate of both *ROC* and *PR* for the combined method which is statistically significant (Table 3.4, row/column Comb). This improvement is apparent in Figure 3.4 (b) in which the combined method achieves performance close to the maximum of both LIBRUS and FINDSITE.

3.3.4 Sequence and structure carry nearly the same amount of predictive information

Table 3.4 (a) shows that there is no statistical difference between LIBRUS and FINDSITE in terms of per-protein *ROC* performance. This is seen in the LIB row and column of the table in which no small p-values occur. This lack of significance is interesting as it shows sequence and predicted structure carry approximately equal amounts of information that may be used to identify ligand-binding residues. In terms of *PR* (Table 3.4

(b)), examining a single FINDSITE site outperforms LIBRUS at a statistically significant level ($p = 0.002$) while examining two FINDSITE sites is not significantly better than LIBRUS ($p = 0.106$). LIBRUS is nearly better than FINDSITE with three sites at a significant level ($p = 0.081$), and better than four and five sites ($p = 0.000$ for both).

Table 3.4: Statistical Comparison of Methods on the DS2 Dataset.

(a) Per Protein *ROC* *p*-values

	FS 1	FS 2	FS 3	FS 4	FS 5	LIB.	Comb.
FS 1	0.500	0.701	0.675	0.464	0.289	0.503	1.000
FS 2	0.299	0.500	0.466	0.257	0.126	0.281	1.000
FS 3	0.325	0.534	0.500	0.281	0.140	0.308	1.000
FS 4	0.536	0.743	0.719	0.500	0.310	0.545	1.000
FS 5	0.711	0.874	0.861	0.690	0.500	0.740	1.000
LIB.	0.496	0.719	0.692	0.455	0.260	0.500	1.000
Comb.	0.000	0.000	0.000	0.000	0.000	0.000	0.500

(b) Per Protein *PR* *p*-values

	FS 1	FS 2	FS 3	FS 4	FS 5	LIB.	Comb.
FS 1	0.500	0.002	0.000	0.000	0.000	0.000	0.997
FS 2	0.998	0.500	0.004	0.000	0.000	0.106	1.000
FS 3	1.000	0.996	0.500	0.008	0.000	0.919	1.000
FS 4	1.000	1.000	0.992	0.500	0.014	1.000	1.000
FS 5	1.000	1.000	1.000	0.986	0.500	1.000	1.000
LIB.	1.000	0.893	0.081	0.000	0.000	0.500	1.000
Comb.	0.003	0.000	0.000	0.000	0.000	0.000	0.500

Performance of the methods is compared via p-values on Welch's t-test. For the entry at row i, column j of the table, the alternate hypothesis that method i has a higher mean than method j is tested as an alternative to the methods having equal means. A low p-value indicates that method i has better performance than method j. Part (a) of the table shows performance comparisons in terms of per protein *ROC* while part (b) shows per protein *PR* comparisons. FINDSITE for various number of sites are reported in the FS row/columns, LIBRUS in LIB, and the combined FINDSITE/LIBRUS predictor in Comb.

Chapter 4

Guided Homology Modeling of Binding Sites

The preceding chapter showed that binding residues can be identified from sequence alone with reasonable accuracy. The sequence-based LIBRUS achieves close to the same accuracy as structure-based FINDSITE. The next logical step is to put those sequence-based predictions to use in some application.

In this chapter we explore such an application. Binding residue predictions are exploited to aid the development of a homology model of a protein. In drug discovery applications, the primary interest is in the binding site of the protein. By allowing predicted binding labels to influence the target-template alignment, the quality of the resulting predicted binding site structure is improved. This effect is most prevalent when the homology modelling problem is difficult, i.e. there is little relation between target and template.

4.1 Background on Homology Modeling

Accurate modeling of protein-ligand interactions is an important step to understanding many biological processes. For example, many drug discovery frameworks include steps where a small molecule is docked with a protein to measure binding affinity [7]. A frequent approximation is to keep the protein rigid, necessitating a high-quality model of the binding site. Such models can be onerous to obtain experimentally.

Computational techniques for protein structure prediction provide an attractive alternative for this modeling task [38]. Protein structure prediction accuracy is greatly improved when the task reduces to homology modeling [39]. These are cases in which the unknown structure, the target, has a strong sequence relationship to another protein of known structure, referred to as the template. Such a template can be located through structure database searches. Once obtained, the target sequence is mapped onto the template structure and then refined.

A number of authors have studied the use of homology modeling to predict the structure of clefts and pockets, the most common interaction site for ligand binding [40, 41, 42]. Their consensus observation is that modeling a target with a high sequence similarity template is ideal for model quality while a low sequence similarity template can produce a good model provided alignment is done correctly. This sensitivity calls for special treatment of the interaction site during sequence alignment assuming ligand-binding residues can be discerned a priori.

The factors involved in modeling protein interaction sites have received attention from a number of authors. These studies tend to focus on showing relationships between target-template sequence identity and the model quality of surface clefts/pockets.

DeWeese-Scott and Moult made a detailed study of CASP targets[1] that bind to ligands [40]. Their primary interest was in atom contacts between the model protein and its ligand. They measured deviations from true contact distances in the crystal structures of the protein-ligand complexes. Though the number of complexes they examined was small, they found that errors in the alignment of the functional region between target and template created problems in models, especially for low sequence identity pairs.

Chakravarty, Wang, and Sanchez did a broad study of various structural properties in a large number of homology models including surface pockets [41]. They noted that in the case of pockets, side-chain conformations had a high degree of variance between predicted and true structures. Due to this noise, we will measure binding-site similarity using the α-carbons of backbone residues. They also found that using structure-induced sequence alignments improved the number of identical pockets between model and true

[1] `http://predictioncenter.org`

structures over sequenced-only alignments. This point underscores the need for a good alignment which is sensitive to the functional region. It also suggests using structure alignments as the baseline to measure the limits of homology modeling.

Finally, Piedra, Lois, and Cruz executed an excellent large-scale study of protein clefts in homology models [42]. To assess the difficulty of targets, the true structure was used as the template in their homology models and performance using other templates was normalized against these baseline models. Though a good way to measure the individual target difficulty, this approach does not represent the best performance achievable for a given target-template pair. This led us to take a different approach for normalization. We follow their convention of assessing binding site quality using only the binding site residues rather than all residues in the predicted structure. As their predecessors noted, Piedra et al. point to the need for very good alignments between target and template when sequence identity is low.

The suggestion from these studies, that quality sequence alignments are essential, led us to employ sensitive alignment methods discussed in Section 4.4.

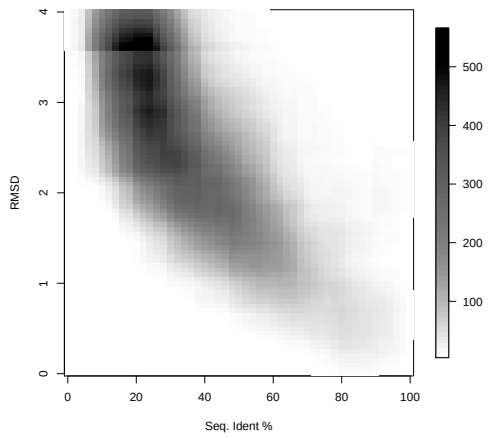
4.2 Homology Modeling with Binding Residue Predictions

Assuming that the ligand-binding residues of all template proteins are known, we illustrate a method to modify alignments of target and template. The modification influences ligand-binding residues to align to one another and discourages the alignment of binders to nonbinders. Once the target-template alignment is constructed, standard homology modeling techniques are employed to produce the target structure prediction. An analysis of the ligand binding site shows that these modified alignments improve the accuracy of this part of the model over standard alignment techniques.

4.3 Experimental Setup

In homology modeling experiments, target-template pairs are required. We used the set of 885 proteins in DS1 (Section 3.2.1) as the targets (structures to be predicted). We used the MAMMOTH structure alignment program to search the PDB for other proteins which had a significant structure alignment [43]. Of these, we kept templates which had

Figure 4.1: Distribution of Homology Pairs.



The heatmap varies in intensity based on the number of homology modeling pairs that have the sequence/structure relationship at the center pixel. A sliding window of 20% sequence identity and 0.8 Å is used to create the counts. Darker colors correspond to more pairs.

a bound ligand which would allow ligand binding residues to be used to influence the target-template alignment. We then proceeded to generate homology models for each target template pair using techniques described below. The final result included 2045 homology pairs and 862 individual target proteins. The distribution of these pairs in the sequence and structure relation space is given in Figure 4.1.

4.4 Alignment Modification by Binding Prediction

The basic framework for sequence alignment is identical to that of Section 3.1.2. As special attention needs to be given to the ligand binding residues, an additional term is incorporated into Equation 3.1 to reflect this goal.

Each residue is labelled either as ligand-binding or not. In the case of the targets, these labels the sequence-predicted labels obtained from LIBRUS. Templates always used true labels. Binding residue predictions that come from LIBRUS are a continuous valued numbers with positive values indicating stronger confidence that the residue is

a binding residue. To convert this into a discrete label, thresholding can be used. In the following results, a threshold of 0.7 was used so that residues above this value were labeled as predicted binders and those below were labeled nonbinders.

A a very simple approach to influence target-template alignments with predicted ligand binding labels is to add a constant m_{bb} whenever a predicted and binding residue in the target aligned with a true ligand binding residue in the template. Setting $m_{bb} = 0$ gives *standard alignments* which do not incorporate the predictions while setting $m_{bb} > 0$ gives a *modified alignment*. Setting $m_{bb} > 0$ encourages the alignment of binding residues and for the results reported below, $m_{bb} = 15$.

4.5 Homology Model Generation

Once a sequence alignment has been determined between target and template, homology modelling may be used to predict the target structure using a variety of standard tools described elsewhere. The results shown here employed version 9.2 of the MODELLER package which is freely available [44]. As input, MODELLER takes a target-template sequence alignment and the structure of the template. An optimization process ensues in which the predicted coordinates of the target are adjusted to violate, as little as possible, spatial constraints derived from the template.

MODELLER offers a high degree of flexibility and automation through a programmable interface. Modeling can be done using only a target sequence and a database of known structures. However, the comments by the software authors and numerous studies indicate that a crucial step in the problem is aligning target and template sequences. This is where predicted binding residues can be useful to influence the proper alignment of target and template.

4.6 Evaluation Metrics for Homology Modeling

The root mean squared deviation (RMSD) is a standard metric used to compare two protein structures. A low RMSD between target and template indicates similarity between two structures. Typically, only the α-carbon coordinates are used for the RMSD computation. Our interest is in the binding site and thus only a good measure of success

is to consider the RMSD between the ligand-binding residues in the true and predicted structures which follows the convention of Piedra et al. [42]. For brevity, this will be called the *lig*RMSD for ligand-binding residues RMSD.

Student's t-test is used on the *lig*RMSD of the standard alignment predictions paired with the corresponding *lig*RMSD of modified alignments to show when their performance differs significantly. The null hypothesis is that the two have equal mean while the alternative hypothesis is that the modified alignments produce models with a lower mean *lig*RMSD (a one-tailed test). We report p-values for the comparisons noting that a p-value smaller than 0.05 is typically considered statistically significant. We also report the mean improvement (gain) from using modified alignments. If the mean of all *lig*RMSD for the standard alignments is $\bar{R}_{stand}$ and that of a modified alignment is $\bar{R}_{mod}$, the percent gain is

$$\%\text{Gain} = \frac{\bar{R}_{stand} - \bar{R}_{mod}}{\bar{R}_{stand}}. \tag{4.1}$$

A positive gain indicates improvement through the use of the ligand-binding residue predictions while a negative gain indicates using predictions degrades the homology models.

Finally, a permutation test can be used to assure us that the observed gains are not tied to tightly to the particular data being used. For the sequence/structure subgroups of interest, the permutation test examines a random subsets one third the size of the subgroup and performs a paired Student's t-Test on the standard and modified *lig*RMSDs. The mean p-value over 100 random subsets are reported as μ_p and may be used as an indication of how well the parameters are expected to perform on future data. The standard deviation of the permutation test p-values is also given as σ_p.

4.7 Model Quality Improvements

We are interested in knowing when it is worth the extra effort to predict ligand-binding residues from sequence. For the homology modelling task, we would not expect the alignment of very similar target and template to benefit much from the additional knowledge

Table 4.1: Results of Homology Model Experiment

SeqID	RMSD	N	p-val	%Gain	μ_p	σ_p
$0 \leq 30$	$0 \leq 2$	27	0.8210	-3.61	0.5467	0.2679
$0 \leq 30$	$2 \leq 4$	1078	0.0000	2.61	0.0003	0.0009
$30 \leq 60$	$0 \leq 2$	347	0.9516	-1.44	0.8145	0.2070
$30 \leq 60$	$2 \leq 4$	438	0.0321	-0.83	0.2417	0.2011
$60 \leq 100$	$0 \leq 2$	166	0.9437	-8.14	0.7496	0.2157
$60 \leq 100$	$2 \leq 4$	35	0.7908	-0.33	0.6655	0.2564

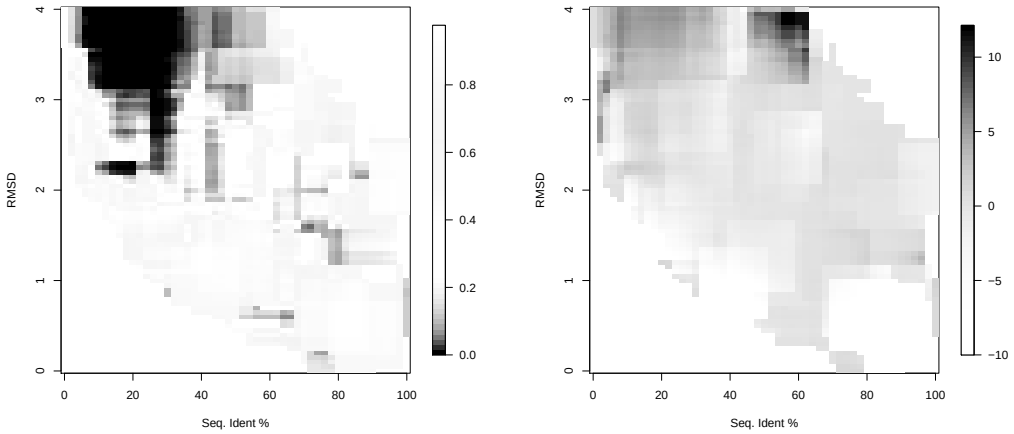
Results of the homology modeling experiment are divided into regions according to sequence identity and RMSD relations between the target and template. The p-value indicates whether gains from using predicted binding labels are statistically significant: smaller p-values correspond to greater significance. Gain is defined in Equation 4.1. N is the number of homology pairs satisfying the sequence/RMSD relationship and are used to compute the statistics. The final two columns are the mean (μ_p) and standard deviation (σ_p) of p-values in a permutation test which measures robustness of the results. A smaller μ_p indicates the results are robust.

of ligand-binding residues: as long as the alignment method is sensitive a good correspondence should be obtainable solely from sequence similarity. However, when the target and template are sufficiently different, ligand-binding residues have more potential to influence the proper alignment of binding residues.

Table 4.1 shows the results of homology modeling experiments restricted to different regions of target-template relationship. A t-test is conducted to determine if the average *lig*RMSD of models produced using LIBRUS-predicted binding labels is lower than for models produced using standard alignments. A small p-value indicates significant improvement in *lig*RMSD. The percentage improvement (gain as defined in Equation 4.1) is given for each subgroup along with the size of the subgroup. A negative gain indicates models using predicted labels were worse than those using standard alignments. The final two columns describe the mean and standard deviation of p-values for permutation tests on the subgroups.

Our intuition on the effectiveness of predicted binding labels is confirmed in Table 4.1. The regions with low sequence identity and high structure difference between target and template see the most improvement. For pairs with less than 30% sequence identity and more than 2Å RMSD, we can expect to get around 2.61% improvement in RMSD. These

Figure 4.2: Homology Model Improvements.



(a) p-values (b) Gains

(a) Statistical significance of homology model improvements. Pixels denote whether predicted ligand-binding residues improve homology models of the binding site. Pixel intensity corresponds to the p-value of a t-test measuring whether the mean *lig*RMSD of models which used predicted labels is lower than that of standard alignments. Dark pixels represent low p-values and statistical significance. Significant improvements are achieved when the target and template have low sequence identity and large RMSD (upper left corner). (b) Percentage of improvement (gain). The intensity of each pixel represents a lower *lig*RMSD using predicted labels in modified alignments versus using standard alignments. The gains are small but statistically significant in the region of low sequence identity and high RMSD between target and template. Greater gains occur in a few other regions but are not statistically significant.

results appear highly robust in the permutation test ($\mu_p = 0.0000$). For pairs with a close structure relationship (0 to 2Å RMSD), it does not appear predicted labels are useful as the gains are all negative in these cases (note, however, the small sample size for low sequence identity in the first line).

Figure 4.2 graphically represents the homology modeling results. In part (a), the intensity of each pixel of the figure corresponds to the p-value of a t-test on a subgroup of the dataset. The position along the Sequence Identity and RMSD axes indicates which pairs are used in the comparison. Subgroups are comprised of pairs in a window of 20% sequence identity and 0.8Å RMSD around the center. For example, at sequence identity of 20% and RMSD of 3.0Å , target-template pairs related by 10-30% sequence identity and 2.6-3.4Å RMSD are used to compute the p-value. The same approach is used in Figure 4.2 (b) which shows the subgroup percentage gain.

The pattern in Figure 4.2 follows that of Table 4.1: the region of low sequence and structure similarity (upper left corner) produces the significant results and positive gains. There are some large positive gains in a few other regions of the similarity space, particularly 50-60% sequence identity for high RMSD, but they are not statistically significant.

Practical lessons can be drawn from this experiment. When faced with generating a homology model of a ligand-binding site, one should consider the available templates carefully as this is the most critical step. Once selected, the template(s) should be aligned to the target sequence using the most sensitive alignment approach available. If it is found that the sequences are very similar, modeling can proceed as normal. If they are dissimilar, it is likely worth the effort to predict the ligand-binding residues of the target using a method such as LIBRUS and then recompute the alignment. Alternatively, the modeler may wish to first generate the usual homology model, use a structure-based method such as FINDSITE to predict the binding site, and then possibly re-align target and template to produce a better model. As mentioned in Section 3.3.4 it is not clear whether this latter approach will improve the binding-site predictions significantly. This is a matter which will require further study.

4.8 Discussion

The preceding three chapters have discussed the identification of protein residues involved in ligand binding. Identification may be done based solely on the protein sequence or by utilizing structure information when it is available. There are several downstream applications of this capability and we have illustrated that sequence-based predictions are presently accurate enough to impact homology modeling of the binding site in a positive fashion.

Though we have seen that the accuracy of binding site homology models increases by leveraging predicted binding residues, examining how these models actually affect docking experiments is unexplored territory. A simple benchmark would measure the docking scores of ligands using the true structure of the protein as the baseline and test whether homology models which use binding residues behave more or less closely to the baseline than models which do not use such predictions. An alternative approach is to

modify the scoring function or energy measure in docking experiments to favor locations with predicted residues. This may improve accuracy or intelligently bias the search space of docking locations. Success on any of these experiments would have a positive impact on docking-based virtual screening.

Another potential application of binding residues is to compare protein structures based on binding site and potential ligands. This is most applicable when structures are available and are thus appropriate for structure-based methods. Discovering proteins with a similar binding site to a particular target can help elucidate side-effects of introducing a small molecule. FINDSITE has already developed some methodologies to determine a ligand profile for a target protein and was utilized to examine function prediction of the protein based on the ligand profile. With the need for automated function assignment for proteins on the rise, it is likely that this trend will continue and develop additional sophistication.

Finally, recent work has used generic machine learning models which incorporate protein similarity to determine the structure-activity relationship of small molecules [45]. In this setting, the set of positive ligands for a target protein can be expanded by identifying other similar targets and adopting their positive ligands. Several methods of target similarity are developed from the standpoint of having no target structures. Sequence-based binding residue predictions may be leveraged in such cases to aid in determining the similarity of two protein targets. In cases where a structure is available, protein similarity for this application should likely be based upon binding sites which requires identification of binding residues by either sequence or structure means.

Chapter 5

Coarse- and Fine-grained Models for Proteins: Evaluation by Decoy Discrimination

In the next two chapters we explore the use of machine learning and optimization to evaluate the different representations of proteins and models of the energy of proteins. A primary difficulty in protein modeling is the large computational cost associated with all-atom molecular dynamics with the standard energy functions. A promising avenue to surmount this hurdle is to employ *coarse-grained models (CG)*. The central idea is very simple: to avoid the cost of modeling all atoms, merge atoms into groups with a single interaction center. The merged object is referred to here as a *bead*. Appropriate merging choices should preserve most aspects of the physical system reasonably in the CG model while reducing the the calculations required for simulations. Coarse-grained models are increasingly utilized in general molecular dynamics [46] while a wide variety of CG models specific to proteins have been proposed to overcome the tremendous number of variables in these systems (see the thorough review by Tozzini [47]). Researchers have merged all atoms in a residue into a single bead or limited number of main and side chain beads since the inception of protein modeling [48, 49]. Side chain interactions of proteins of particular importance leading some models such as SICHO use a single interaction center centered on the side-chain[50]. The popular and successful ROSETTA

approach to protein structure prediction relies on a model in which all heavy atoms of the backbone are used but sidechain atoms are merged into a bead [51]. Recent years have seen the advent of other models such as the two-center per residue UNRES force field [52, 53], the three-center CABS approach[54], and MARTINI force field which groups four heavy atoms together [55]. Kurkcuogu and co-workers showed that coarse-graining preserves the vibrational modes of two proteins even when reducing 5, 10, or 20 heavy atoms into a single interaction center [56]. They also explored using a fine-grained view of "interesting parts" of a protein while coarse-graining the remainder. The notion of a such a mixed representation is explored in the latter part of our work here.

Despite such attention, it is still not clear how much modeling accuracy is lost by switching to coarse representations. Part of the difficulty is that even coarse-grained models require heavy computation to perform molecular dynamics. Though such simulations are beginning to become tractable, it is still difficult to sample protein state space enough to evaluate a variety of models using dynamics. An example from a recent study of alpha-helical proteins using the coarse-grained UNRES force field found that for the 66-residue GCN4 protein, 4 of 10 simulation runs folded to near-native conformations for a total cost of around 99 hours of CPU time [57, Table 2]. Directly optimizing force field parameters using simulation is still largely out of reach.

An alternative vehicle for assessing protein models is through *decoy discrimination*. In this setting, one or more correctly folded proteins (natives) has associated with it incorrectly folded structures (decoys). The goal is to develop a scoring function that differentiates a native structure from its decoys. Since there are no dynamics involved, decoy discrimination is much cheaper as means to quickly evaluate models. One can also control the number of decoys and their characteristics directly if greater sampling of the state space is desired. Decoy discrimination has a long history in protein structure prediction and analysis. Scoring functions go by a variety of names including empirical force field, knowledge-based potential, statistical potential. The idea is always to assign an extreme score to natives and the opposite extreme to decoys. If low scores are assigned to natives, the score can be interpreted as a kind of energy function due to the widely held belief that native structures are at the protein's global potential energy minimum.

In this chapter we evaluate three levels of protein model granularity using decoy discrimination. At each granularity level, we assessed a variety of *feature types* including n-body interactions, solvent exposure, and dihedral angle bending. This gives insight into which features are informative at high versus low granularity and which may be discarded without affecting accuracy. For robustness, we used four different machine learning techniques to determine the model parameters. Comparing their relative performance illustrates aspects of linear versus nonlinear estimation and shows whether binary classification is a suitable means to determine model parameters. We adhered to a strict cross-validation methodology: models are assessed on a large dataset of 15 decoy sets and performance is measured only on structures that were not seen during parameter estimation. Two styles of cross-validation were employed: balancing decoys amongst folds and leaving whole decoy sets out. Both make for a strong test of whether the models generalize and allows us to identify difficult decoy sets.

In the next chapter, we propose a new method which can select bead types from a mixture of model granularities while maximizing the discrimination of native from decoys. This is a first step towards a data-driven method for protein model selection. We illustrate its behavior on the full set of decoys and explore how bead types from the different levels of granularity are combined.

5.1 Materials and Methods

We first establish a set of proteins on which to experiment. Each protein has a known native structure and multiple decoy structures which have an identical sequence and may share some structural features to the native but are in some way misfolded. The native structures can be thought of as a positive class and decoys a negative class. There are a variety of means to represent the protein and calculate its energy. These representations and energy functions are evaluated based on the ability of several machine learning tools to find parameters that separate the native and decoys sets. Two forms of cross-validation are used to establish charactersitics the protein representations, energy functions, and decoy sets.

Table 5.1: Protein Decoy Datasets.

Set	Nat.	Dec.	Total	Ref.	Source
fisa	4	200	204	[51]	http://dd.compbio.washington.edu/
fisa3	5	250	255	[51]	http://dd.compbio.washington.edu/
4state	7	300	307	[59]	http://dd.compbio.washington.edu/
lattice	8	400	408	[60]	http://dd.compbio.washington.edu/
lmds	9	450	459	[61]	http://dd.compbio.washington.edu/
casp5	17	267	284	[62]	http://www.fiserlab.org/potentials/casp_decoys/
moulder	20	1000	1020	[63, 64]	http://salilab.org/decoys/
casp6	24	447	471	[62]	http://www.fiserlab.org/potentials/casp_decoys/
tsai	30	1500	1530	[65]	http://depts.washington.edu/bakerpg/decoys/
casp7	34	755	789	[62]	http://www.fiserlab.org/potentials/casp_decoys/
rose	42	2100	2142	[51]	http://depts.washington.edu/bakerpg/decoys/
skol	47	2350	2397	[66]	http://cssb.biology.gatech.edu/amberff99
ro62	59	2950	3009	[67, 68]	http://depts.washington.edu/bakerpg/decoys/
casp8	68	1159	1227	[62]	http://www.fiserlab.org/potentials/casp_decoys/
lkf	115	5318	5433	[69]	http://titan.princeton.edu/2010-10-11/Decoys/
Combined	415	19446	19861		

The combined dataset of decoys used for all protein representation experiments. Proteins were drawn from 15 decoy sets generated by previous researchers. The columns are (Set) the decoy set, (Nat.) the number of distinct native proteins in the set, (Dec.) the number of decoys in each set limited to 50 per native, (Total) the total structures in the set, (Ref.) a citation describing the production of the decoy set, and (Source) the URL from which the decoy set was downloaded. Some native proteins belong to multiple decoy sets thus the Natives and Total column do not total to the the Combined row.

5.1.1 Dataset details

We combined decoys from 15 different sets of decoys that have been reported in literature, several of which are available from the Decoys R Us project [58]. The dataset is summarized in Table 5.1. The number of decoys associated with each native in different decoy sets varies. In order to keep the size of data manageable, we limited the number of decoys per native per decoy set to 50 structures. For example, though there are more decoys available in it, we used only 50 decoy structures for each of the 4 natives in the fisa set giving a total of 200 decoys and 204 total structures. We sampled the 50 decoys from those available to give each native both high-RMSD decoys which were badly misfolded and low-RMSD decoys which resemble the native structure closely. In several decoy sets, such as casp sets, each native had fewer than 50 decoys in which case we used all decoys.

Some native proteins appear in several decoy sets. This is why adding each entry in the Natives column of Table 5.1 gives more than the 415 natives in the Combined row. The combined set was pruned to ensure that no identical proteins were present. The 415 proteins share less than 90% sequence identity with one another. Some close relatives were kept to keep the set as large as possible but the majority of entries share little sequence similarity: there are 376 sequence clusters using `blastclust` at the 30% sequence identity threshold.

5.1.2 Cross-Validation

In cross-validation, the available data is divided into multiple folds. We used 4-fold cross-validation so that in the experiments of (4CV) Section 5.2, we trained models on 3/4 of the proteins and tested the learned model on 1/4 of the proteins. This process was done 4 times with a different quarter of the data left out each time. Performance statistics were collected for each fold and their mean and standard deviation are given in the experimental results. For the results in Section 5.2, each decoy set was evenly divided amongst the folds so that each fold had examples from every decoy set. We also performed cross-validation where whole decoy sets were left out for a total of 15 folds (DCV). At each step, one whole decoy set such as fisa or ro62, was left out and all remaining data was used to estimate model parameters.

As noted in Section 5.1.1, some natives are present in multiple decoy sets. During experiments, this handled in the following way. In the cross-validation experiments of Section 5.2, whenever a particular native was selected for training, all decoys from all sets associated with that native were also used for training. For whole decoy cross-validation experiments in Section 5.2, we divided the data on decoy sets. To test performance on a decoy set, the natives and decoys in it were removed from the training set. In addition, any decoys from different sets which were associated with a left out native were omitted from both training and testing. This prevents models from learning from any direct information on the test proteins.

5.1.3 Fine-, Medium-, and Coarse-grained Representations

The first key design choice of an empirical forcefield is the type of body which will be represented. This choice has the largest impact on how accuracy will be traded for efficiency. We use the generic term *bead* when referring to an object in a protein representations. In the fine-grained model, beads are physical atoms while at coarser levels of representation several atoms are merged into a single bead. We use three granularities of models.

Fine-grained: ***t32*** We adopted the model of Qiu and Elber which assigns all atoms to 32 types [70] and is referred to as the t32 representation. This set was chosen as the original study showed expanding to 46 types of atoms did not improve the discriminative power of the model and the t32 set prove quite robust on an evaluation of atomic and coarse-grained potentials to detect decoys using support vector machines by Zhang and Zhou [71]. An alternative would be the RAPDF/DFIRE set of 167 atom types which have been widely used [72, 73]. These proved less effective in Zhang and Zhou's evaluation potentially due to the large number of parameters which must be estimated.

Medium-grained: ***mc1*** The physical atoms of each residue were assigned to either the main-chain or side-chain giving each residue except glycine two beads. The barycenter (mean XYZ coordinate) of physical atoms in main-chain or side-chain groups determined the coordinates of each mc1 bead. This is an intermediate representation, more coarse than the atomic level but still allowing independent interaction centers for each residue. Each side-chain was assigned a type based on the amino acid. Glycine, alanine, and proline were treated specially: each was assigned a single interaction point specific to the residue. All other amino acids were assigned a specific side chain atom and a generic main chain atom. There are a total of 21 bead types in mc1. The mc1 model is similar to several prior models which use 2 beads per residue [59, 52, 74].

Coarse-grained: ***res*** All physical atoms in a residue were merged into a single bead at their barycenter. The res representation has 20 types of beads corresponding to each of the amino acids.

5.1.4 Types of Features

After choosing a level of representation, a variety of structural features may be calculated for a protein. We explored a range of generic features that represent common energy terms in empirical forcefields.

Two- and Three-body Interaction Features

Interactions between two bodies (2-body) is the most prevalent feature in empirical force fields, particularly for decoy detection. Most empirical force fields take a discretized approach to 2-body interactions: distances between each bead pair are assigned to a distance bin and a separate parameter is associated to each bin and pair-type.

For the fine-grained t32 representation, we adopted the same three distance bins as the original study by Qiu and Elber [70]: 2.0-3.5Å, 3.5-5.0Å, and 5.0-6.5Å. Atom pairs not in one of these distance ranges were ignored. There were $3 \times (32 \times (32+1))/2 = 1584$ features of this type. This forcefield appeared as t32S3 in the original and subsequent studies [70, 71]. For the medium-grained mc1 and coarse-grained res types, we included two additional bins for a total of five bins: 2.0-3.5Å, 3.5-5.0Å, 5.0-6.5Å, 6.5-8.0Å and 8.0-10.0Å. No attempt was made to optimize distance bins for predictive performance. The arbitrary nature of how bin cut-offs must be chosen is unsatisfactory and deserves further investigation into a more disciplined approach. These are referred to as the *2-body features*.

Examples of a potential energy functions which calculate interactions higher than two were first explored in by Munson and Singh [75]. They analyzed 2-, 3-, and 4-body potentials and found that 4-body potentials explain patterns of 4-body contacts in a statistically superior fashion to lower order interactions. However, 2-body potentials recognized native sequence-structure pairs equally as well as 3- and 4-body potentials in threading, the only difference being better Z-scores in the higher-body case. There has been some recent-work analyzing 4-body potentials mainly for their use in threading [76, 77, 78]. The number of parameters that must be estimated increases exponentially with n for n-body interactions. Estimating a large number of parameters given the limited number of native protein structures available can compromise the generality of such models. In order to assess whether this happens, we computed 3-body interactions

for the three representation levels. To avoid an explosion of parameters, 3-body features used only a single distance bin: 2.0-6.5Å for t32 (fine-grained) and 2.0-10.0Å for mc1 and res (medium- and fine-grained). In our experiments 3-body features were always included in addition to the 2-body features, the 2-body having the distance bins described above. These are referred to as *(2+3)-body* features.

Proteins represented by 2-body and (2+3)-body features are simply vectors of counts. However, longer proteins tend to have much larger total counts than their shorter counter-parts as they number of 2-body interactions increases quadratically with the length of the protein. Data with drastically different scales tends to degrade the performance of machine learners we used. We adopted a simple normalization: count vectors were normalized by the number of atoms (t32) or pseudoatoms (mc1,res) in the protein. A similar normalization procedure was used in previous decoy studies [71].

Contact (Single-Body) Features

Rather than distinguish interactions by the types of both bodies, a forcefield may instead limit consideration how densely individual bodies are packed. Typically this is done by counting the number of beads in a volume centered on a bead of one type. The count is attributed to the central bead type. This amounts to a sort of single-body energy as the types of the other beads are ignored. The density can correlate with a bead's placement at the surface of the protein (less crowded) or the interior (more crowded). Single-body potentials are referred to as "contact numbers" in some bioinformatics literature [79, 80] and we follow that convention referring to the feature as *contacts*. We calculate contacts using the same bins as are used for 2-body interactions above except that interactions count towards the total for both bodies (e.g. an alanine-arginine interaction counts towards both the contacts of alanine and arginine).

Solvent Exposure Features

Solvent accessibility was calculated for each bead by a sampling algorithm: 100 evenly spaced points were placed on the surface of each bead and were counted as buried if they were inside the radius of another bead and exposed if not. The fraction of exposed points was multiplied by the surface area of the bead to get the area.

The exposed areas were converted into discretized features using binning. Initially we experimented with fixed bin widths but determining appropriate cutoffs for each type of bead proved difficult. Beads which agglomerated several physical atoms do not have established radii. Their radii must be estimated from the data. Some decoy structure contain unrealistic bond lengths which can make the maximum surface area for an bead type abnormally large. In turn this large maximum distorts binning based on the proportion of an beads surface area to the maximum observed. A more robust strategy is required.

We discretized by calculating the empirical distribution of the surface areas of an bead type across the entire data set and used quantiles to determine bins. Beads are therefore evenly divided into the bins: if 4 bins are desired, each contains 25% of the beads. After examining the distribution for a number of bead types, it was not clear which number of bins was appropriate to use. Some bead types had complex distributions which would cause information loss if too few bins are used while other distributions were flat requiring only a few bins to represent. To avoid information loss, we included multiple overlapping bins and allowed the feature selection to determine which were important for identifying decoys. We included 5, 10, and 20 bins for each bead type. This mixed-quantile strategy gave better performance than a fixed number of bins according to initial tests with glmnet and we report it as the *exposure* feature subsequently.

Angle Features

Angles were generated by first examining all phi-psi angle pairs for each residue in the dataset (aside from N- and C-terminal residues). These were then clustered in two different ways. In the first, each residue was assigned to one of 8 clusters of phi-psi angle pairs which were determined according to K-means clustering as implemented by the `kmeans` function of the R package `stats` [81]. Counts of cluster membership were used as features for each protein giving $8 \times 20 = 160$ features. In addition, we counted transitions between 2 angle states as giving $160^2 = 25,600$ features. Since not every transition occurred in the data, there fewer angle features than the combined total of individual and transition clusters: on 25221 total features were observed rather than 25,760. These are referred to as the angle features with *20 groups* of amino acids.

We noted that most amino acids adopt similar phi-psi angle distributions and can be grouped together. On looking at the distribution of clusters, only proline and glycine had significantly different cluster arrangements. To reduce the number of angle features, 8 clusters of phi-psi anglese were computed for proline, 8 for glycine, and 8 for the combination of all other residue types. Transitions between these 24 clusters were also counted as for the 20 groups. A total of 204 single and transition features were used for the angle features with *3 groups* of amino acids.

Our treatment of angular features was inspired by the clustering of angle states and transitions used by Zhang and co-workers [82] and Bahar et. al [83]. Both used reduced alphabets of amino acids in determining angle states but favored the use of reduced model angles rather than phi-psi angles which require all atoms of the backbone to be present.

5.1.5 Discrimination Methods

Once the level of representation has been set and the structural features selected, a method must be selected to determine parameters (feature weights) for the final model. For decoy discrimination, the goal is to establish a set of model parameters that differentiate native proteins from decoys. We assessed four methods for discrimination and parameter estimation.

Support Vector Machines (SVM)

A linear support vector machine *(svm)* learns a vector of parameters w to representing a separating hyperplane between the positive (native protein) and negative (decoy) classes. A nonlinear SVM also separates the positive and native classes but uses a kernel, in our case the radial basis function (RBF) kernel (*svmrbf*). The kernel allows nonlinear boundaries to be learned at the cost of not being able to determine the parameter vector w for the structural features. We used a customized R [81] interface to LIBSVM [84] to train SVM models. We used a grid of values for the SVM cost parameter C and RBF kernel parameter gamma during cross-validation and report the best performing models.

The SVMRANK package was used to generate linear ranking SVM models *(svmrank)* [85, 86]. We did not explore nonlinear ranking as the linear and nonlinear results on the standard SVM were similar and the computational requirements for nonlinear ranking problems is prohibitive. We tuned the SVM cost parameter for svmrank over a grid of values and report the best result.

Penalized Regression Models

A penalized logistic regression model is learned by optimizing the following:

$$\max_{w} \sum_{i=1}^{N} \left[y_i w^T x_i - \log(1 + e^{w^T x_i}) \right] - \lambda \left[(1-\alpha)||w||_2 + \alpha||w||_1 \right]. \tag{5.1}$$

The left term represents the loss of the model and is the conditional log likelihood of observing the entire data set of size N with features x_i and classes y_i. The right term is regularizer. As in the SVM models, the end result of a logistic regression model is a vector w of feature weights. As the penalty parameter λ is increased, elements of w shrink to 0 which allows feature selection to be done. The α parameter controls the relative L1 and L2 penalty on the model. We set $\alpha = 0.9$ which introduces a small amount of L2-regularization on feature selection along with L1-regularization. This was found to improve overall performance. The `glmnet` R package was employed to train L1-penalized logistic regression models [87]. This package efficiently solves for all levels of the penalty parameter λ. We used 10-fold internal cross-validation with evaluation based on the area under the ROC curve to determine the optimal λ value for each model.

5.1.6 Performance Metrics

Decoy discrimination is an interesting problem from the machine learning standpoint as it is always unbalanced: for every positive instance which is the native protein structure there are potentially many negative instances which are misfolded. Performance is measured only on the ability to identify from amongst a pool of structure for a single protein the single native structure (or closest to native). For that reason, typical classification metrics such as ROC are unsuitable. We used several metrics commonly employed in other decoy discrimination literature.

Mean Native Rank (Rank) The native and associated decoy proteins are ranked by their prediction score and the rank of the native is taken. In cross validation, we report the mean of these ranks. A lower rank is better with mean native rank of 1 being the perfect prediction.

Top-1 Fraction In a given set of natives and decoys, we report the fraction of natives that are ranked higher than all their associated decoys (those that have native rank of 1). A higher Top-1 Fraction is better with 1.0 being the perfect.

Z-score The native protein structure is believed to have a lower free energy than misfolded decoys. Interpreting the prediction scores produced by an SVM or glmnet method as an energy, the Z-score is defined

$$Z = \frac{\mu_{decoy} - E_{native}}{\sigma_{decoy}} \tag{5.2}$$

where μ_{decoy} and σ_{decoy} are the mean and standard deviation of the decoy prediction scores and E_{native} is the prediction score for the native protein. A larger more negative Z-score corresponds to better separation of decoys from natives.

5.2 Four-fold Cross-Validation Experiment (4CV)

Proteins were represented at the coarse *res* level, medium *mc1* level, and fine-grained *t32* level to determine trade-offs associated with each representation. At each of these levels, features were calculated for each protein including *2-body* interactions, 2- and 3-body interactions (called *(2+3)-body*), *contact* counts (or 1-body interactions), and solvent *exposure*. Proteins were also represented using only their backbone *angle* data grouping residues into 3 or 20 groups for angle binning. Each representation/feature combination has a number of model parameters associated with it which may be set to discriminate native from decoy proteins. Section 5.1.3 describes the representation level and Section 5.1.4 describes structure features.

We considered four methods to fit model parameters: linear support vector machine training *(svm)*, nonlinear support vector machine training *(svmrbf)*, ranking support vector training *(svmrank)*, and penalized logistic regression *(glmnet)*. These are primarily classification methods which learn parameters to discriminate between two classes, in our case native and decoy structures. Section 5.1.5 gives details of these methods.

In our first experiment, 415 proteins with associated decoys (total 19,861 structures) were divided into four folds, each fold having a balanced number of proteins from each decoy set. We refer to this experiment as *four-fold cross-validation* (4CV). At each step, three folds were used for training and the remaining fold was used for evaluation. Performance is averaged over the four folds. The results are used to compare aspects of the parameter learning models and also evaluate the viability of each type of feature in each representation. The comparison is done based on the mean rank of the native structure *(Rank)*, the fraction of all natives ranked in the top position *(Top-1)*, and the *Z-score* which gives a normalized score (or energy) separation between natives and decoys. These performance measures are detailed in Section 5.1.6.

5.2.1 Linear vs. Nonlinear Classification

We first focus on linear and nonlinear SVMs *(svm and svmrbf)*. Table 5.2 compares svm and svmrbf in the 4-fold cross-validation experiment. The two classifiers have very similar performance. Of particular note are the 2-body results in the top section of Table 5.2 as they are most directly comparable to the results from Dong and Zhou [71]. With 2-body interaction features, we see a small benefit at the residue-level representation for using a nonlinear kernel, but at finer grained representations there is little to no benefit over the linear version of SVM. This trend is also present in the 2+3-body interactions and the contact/1-body interactions: some benefit is given at the coarsest representation level by using svmrbf but no such benefit is present at the finer mc1 and t32 levels. Solvent exposure features follow this trend but to a weaker extent with svmrbf only slightly out-performing svm at each level of granularity. Finally, angle data definitely benefits from the nonlinear SVM though it is comparatively a weak feature for identifying decoys.

Table 5.2: Comparison of Linear and Nonlinear SVM Learners.

Feature	Level	Method	Rank	Top-1	Z-Score	Params
2-body	res	svm	7.23(1.11)	0.499(0.029)	-2.09(0.11)	1073
	res	svmrbf	6.66(0.82)	0.549(0.036)	-2.25(0.07)	1073
	mc1	svm	3.50(0.62)	0.771(0.040)	-3.24(0.07)	1189
	mc1	svmrbf	3.46(0.71)	0.771(0.032)	-3.29(0.06)	1189
	t32	svm	2.62(0.52)	0.889(0.033)	-4.44(0.24)	1584
	t32	svmrbf	2.57(0.44)	0.896(0.029)	-5.11(0.34)	1584
(2+3)-body	res	svm	7.52(1.87)	0.410(0.041)	-1.92(0.08)	2682
	res	svmrbf	7.25(1.70)	0.456(0.033)	-2.01(0.06)	2682
	mc1	svm	4.36(0.50)	0.694(0.019)	-2.81(0.09)	3075
	mc1	svmrbf	5.16(0.55)	0.634(0.030)	-2.62(0.14)	3075
	t32	svm	1.97(0.36)	0.911(0.020)	-4.25(0.28)	7567
	t32	svmrbf	2.58(0.53)	0.870(0.025)	-4.15(0.28)	7567
contacts	res	svm	8.28(0.89)	0.417(0.050)	-1.74(0.09)	212
	res	svmrbf	7.36(0.80)	0.492(0.075)	-1.96(0.20)	212
	mc1	svm	4.17(0.54)	0.730(0.017)	-2.81(0.09)	222
	mc1	svmrbf	4.92(0.58)	0.655(0.057)	-2.73(0.13)	222
	t32	svm	3.69(0.76)	0.781(0.083)	-3.21(0.21)	204
	t32	svmrbf	4.20(1.04)	0.749(0.061)	-3.20(0.24)	204
exposure	res	svm	7.62(0.88)	0.409(0.052)	-1.75(0.11)	1266
	res	svmrbf	7.21(0.64)	0.496(0.049)	-2.31(0.19)	1266
	mc1	svm	5.17(0.82)	0.660(0.037)	-2.54(0.09)	1360
	mc1	svmrbf	4.82(0.57)	0.687(0.024)	-3.06(0.17)	1360
	t32	svm	2.92(0.85)	0.750(0.045)	-2.98(0.12)	1386
	t32	svmrbf	2.82(1.00)	0.769(0.025)	-4.95(0.42)	1386
angles	3 groups	svm	7.65(0.88)	0.407(0.030)	-1.62(0.06)	598
	3 groups	svmrbf	6.12(0.49)	0.492(0.050)	-1.86(0.08)	598
	20 groups	svm	7.76(0.97)	0.511(0.047)	-2.01(0.10)	25221
	20 groups	svmrbf	7.35(1.00)	0.518(0.054)	-2.06(0.11)	25221

Results of the 4CV experiment on protein representation are given divided by the type of feature used (Section 5.1.4), the level of representation (Section 5.1.3), and the discrimination method used to learn models (Section 5.1.5). The performance metrics Rank, Top-1, and Z-Score are described in Section 5.1.6. The mean across four cross-validation folds is given along with the standard deviation in parenthesis. The final column (Params) is the number of parameters in the row model.

The near equivalence of linear and nonlinear SVMs (svm and svmrbf) conflicts with earlier work which indicates linear SVMs or inferior to their nonlinear counterparts [71]. Our best explanation for this difference is that experiments in the previous work were restricted to single decoy sets for training and testing. For example, the two cross-validation experiments were done within the LKF and CASP7 datasets separately. Since decoy sets vary greatly in how the structures are generated, it is possible that characteristics of those datasets lent themselves to nonlinear separation. However, the model learned does not transfer to a decoy set with different characteristics. The experiment in which potentials were transferred to new decoy sets in [71, Table 4] indicated that the linear and nonlinear potentials behave similarly on truly new data. The issue of how well any potential can be applied to a truly new set of decoys is taken up in Section 5.3.

Despite their slightly superior performance on a few of the protein representations, there is a major disadvantage of nonlinear SVM models. Both linear and nonlinear SVMs tend to learn classification models based on support vectors which are simply specific training examples of some importance. In the linear case, through simple algebraic operations, the parameters for each feature can be recovered so we may know how each interaction affects the likelihood of being a decoy. This is not so for nonlinear SVMs: they learn a model that is implicitly embedded in a higher-dimensional space (infinite dimensional in the case of the svmrbf) which makes it very difficult to relate features in the original space to the likelihood of a protein being native or decoy. Due to this difficulty in interpretation and the fact that only marginal performance gains come from using a nonlinear kernel, we omit svmrbf from further discussion.

5.2.2 Regularized Logistic Regression vs. SVM Classification

With the number of features in representations ranging from 204 to 25,221, there is potential to over-fit parameters to training data which decreases the generalization of a model. A regularized method such as glmnet is designed to avoid this by charging a cost for the inclusion of any feature while learning model. Such method tend to generate sparse models with zero parameters associated to many features. SVMs do not encourage sparse models explicitly.

Table 5.3 and Figure 5.1 show a comparison of the performance of the linear SVM against the regularized logistic regression classifier glmnet. The svmrank classifier in this table is discussed later. Included in Table 5.3 are the number of parameters in the model *(Params)* which is also the number of structure features, how many parameters were nonzero, *(Selected)*, and the fraction of nonzero parameters *(Frac.)*. Also present are measures of model stability amongst the four cross-validation folds: the correlation of parameters learned and the overlap of nonzero parameters.

Table 5.3: Comparison of Methods/Representations/Features in 4CV Experiment.

Feature	Level		Method	Rank	Top-1	Z-Score	Params	Nonzero	Frac.	Correlation	Overlap
2-body	res	1	glmnet	6.45 (1.18)	0.532 (0.040)	-2.19 (0.10)	1073	485 (95)	0.452	0.790 (0.022)	0.820 (0.046)
		2	svm	7.23 (1.11)	0.499 (0.029)	-2.09 (0.11)	1073	1054 (12)	0.982	0.720 (0.010)	0.994 (0.004)
		3	svmrank	6.65 (0.99)	0.549 (0.039)	-2.24 (0.04)	1073	1064 (12)	0.992	0.777 (0.016)	1.000 (0.000)
	mc1	4	glmnet	2.96 (0.53)	0.771 (0.027)	-3.15 (0.08)	1189	650 (26)	0.547	0.811 (0.006)	0.822 (0.014)
		5	svm	3.50 (0.62)	0.771 (0.040)	-3.24 (0.07)	1189	1154 (14)	0.971	0.804 (0.009)	0.989 (0.002)
		6	svmrank	3.08 (0.24)	0.785 (0.035)	-3.26 (0.23)	1189	1176 (16)	0.989	0.786 (0.018)	0.997 (0.002)
	t32	7	glmnet	1.69 (0.50)	0.920 (0.017)	-4.34 (0.16)	1584	845 (113)	0.533	0.805 (0.021)	0.842 (0.029)
		8	svm	2.62 (0.52)	0.889 (0.033)	-4.44 (0.24)	1584	1577 (2)	0.996	0.937 (0.009)	0.999 (0.001)
		9	svmrank	1.93 (0.53)	0.920 (0.015)	-4.44 (0.13)	1584	1578 (2)	0.996	0.830 (0.036)	0.999 (0.001)
(2+3)-body	res	10	glmnet	6.92 (1.48)	0.470 (0.035)	-2.12 (0.16)	2682	774 (387)	0.289	0.680 (0.075)	0.738 (0.091)
		11	svm	7.52 (1.87)	0.410 (0.041)	-1.92 (0.24)	2682	2652 (48)	0.989	0.688 (0.011)	0.999 (0.000)
		12	svmrank	6.97 (0.86)	0.482 (0.021)	-2.11 (0.13)	2682	2657 (46)	0.991	0.742 (0.014)	0.999 (0.000)
	mc1	13	glmnet	3.59 (0.17)	0.713 (0.032)	-2.80 (0.11)	3075	1530 (328)	0.498	0.706 (0.024)	0.817 (0.036)
		14	svm	4.36 (0.50)	0.694 (0.019)	-2.81 (0.09)	3075	3031 (75)	0.986	0.747 (0.007)	0.999 (0.001)
		15	svmrank	4.02 (0.30)	0.682 (0.009)	-2.76 (0.18)	3075	3037 (74)	0.988	0.738 (0.019)	1.000 (0.000)
	t32	16	glmnet	1.56 (0.45)	0.911 (0.021)	-4.08 (0.19)	7567	2220 (125)	0.293	0.761 (0.025)	0.733 (0.022)
		17	svm	1.97 (0.36)	0.911 (0.020)	-4.25 (0.28)	7567	7538 (5)	0.996	0.827 (0.010)	0.998 (0.000)
		18	svmrank	2.17 (0.72)	0.908 (0.025)	-4.24 (0.24)	7567	7562 (4)	0.999	0.822 (0.033)	1.000 (0.000)
contacts	res	19	glmnet	7.13 (1.63)	0.472 (0.077)	-1.97 (0.23)	212	189 (17)	0.891	0.490 (0.328)	0.976 (0.028)
		20	svm	8.28 (0.89)	0.417 (0.050)	-1.74 (0.09)	212	212 (0)	1.000	0.581 (0.075)	1.000 (0.000)
		21	svmrank	6.64 (0.87)	0.499 (0.054)	-2.06 (0.19)	212	212 (0)	1.000	0.806 (0.032)	1.000 (0.000)
	mc1	22	glmnet	3.82 (0.37)	0.730 (0.049)	-2.83 (0.12)	222	192 (18)	0.865	0.737 (0.119)	0.973 (0.025)
		23	svm	4.17 (0.54)	0.730 (0.017)	-2.81 (0.09)	222	222 (0)	1.000	0.850 (0.018)	1.000 (0.000)
		24	svmrank	3.33 (0.61)	0.742 (0.054)	-2.80 (0.28)	222	222 (0)	1.000	0.752 (0.034)	1.000 (0.000)
	t32	25	glmnet	3.35 (0.56)	0.771 (0.068)	-3.15 (0.12)	204	160 (34)	0.784	0.456 (0.272)	0.970 (0.018)
		26	svm	3.70 (0.76)	0.781 (0.083)	-3.21 (0.21)	204	204 (0)	1.000	0.829 (0.012)	1.000 (0.000)
		27	svmrank	2.95 (0.45)	0.769 (0.040)	-3.14 (0.19)	204	204 (0)	1.000	0.796 (0.045)	1.000 (0.000)
exposure	res	28	glmnet	6.66 (1.03)	0.491 (0.063)	-2.03 (0.16)	1266	148 (16)	0.117	0.742 (0.029)	0.691 (0.040)
		29	svm	7.62 (0.88)	0.409 (0.052)	-1.75 (0.11)	1266	1266 (0)	1.000	0.606 (0.025)	1.000 (0.000)
		30	svmrank	6.71 (0.58)	0.499 (0.053)	-2.18 (0.17)	1266	1259 (1)	0.995	0.877 (0.009)	1.000 (0.001)
	mc1	31	glmnet	4.55 (1.11)	0.696 (0.043)	-2.74 (0.13)	1360	156 (18)	0.115	0.788 (0.024)	0.716 (0.038)
		32	svm	5.18 (0.82)	0.660 (0.037)	-2.54 (0.09)	1360	1360 (0)	1.000	0.754 (0.015)	1.000 (0.000)
		33	svmrank	4.12 (0.67)	0.641 (0.067)	-2.63 (0.18)	1360	1352 (1)	0.994	0.809 (0.016)	1.000 (0.001)
	t32	34	glmnet	2.56 (0.91)	0.778 (0.027)	-3.10 (0.12)	1386	183 (20)	0.132	0.740 (0.049)	0.712 (0.037)
		35	svm	2.92 (0.85)	0.750 (0.045)	-2.98 (0.12)	1386	1386 (0)	1.000	0.707 (0.020)	1.000 (0.000)
		36	svmrank	2.32 (0.70)	0.807 (0.028)	-3.26 (0.08)	1386	1370 (2)	0.989	0.755 (0.051)	1.000 (0.001)
angles	3 groups	40	glmnet	7.33 (0.49)	0.487 (0.033)	-1.86 (0.10)	598	461 (6)	0.771	0.795 (0.026)	0.923 (0.007)
		41	svm	7.65 (0.88)	0.407 (0.030)	-1.62 (0.06)	598	503 (3)	0.841	0.677 (0.020)	0.978 (0.006)
		42	svmrank	7.35 (0.41)	0.525 (0.027)	-2.08 (0.10)	598	542 (5)	0.906	0.766 (0.014)	0.988 (0.006)
	20 groups	37	glmnet	8.13 (0.82)	0.523 (0.029)	-1.97 (0.08)	25221	6916 (2833)	0.274	0.647 (0.027)	0.835 (0.049)
		38	svm	7.76 (0.97)	0.511 (0.047)	-2.01 (0.10)	25221	15530 (136)	0.616	0.671 (0.012)	0.873 (0.004)
		39	svmrank	7.85 (0.99)	0.561 (0.017)	-2.04 (0.12)	25221	20511 (188)	0.813	0.731 (0.019)	0.936 (0.005)

Notes on next page.

Table 5.3: Comparison of Methods/Representations/Features in 4CV experiment.
(Previous page) The various methods, representations, and features are compared for their decoy discrimination capability in 4-fold cross validation (4CV). The first series of columns are identical to Table 5.2. The rightmost columns give statics on the models learned. They are (Nonzero) the mean number of nonzero parameters in the row model, (Frac.) the fraction of nonzero parameters, (Correlation) the mean Pearson correlation coefficient between the parameter vectors of the four models, and (Overlap) the mean fraction of parameters which are nonzero in pairs of models. Standard deviations are given in parentheses.

In all representations, the effectiveness of regularization is apparent. The glmnet method performed equal to or better than svm while simultaneously selecting a relatively small number of important features. While svm tended to provide a slightly better Z-score than glmnet, glmnet dominated svm in providing a better mean native rank and top-1 fraction for natives.

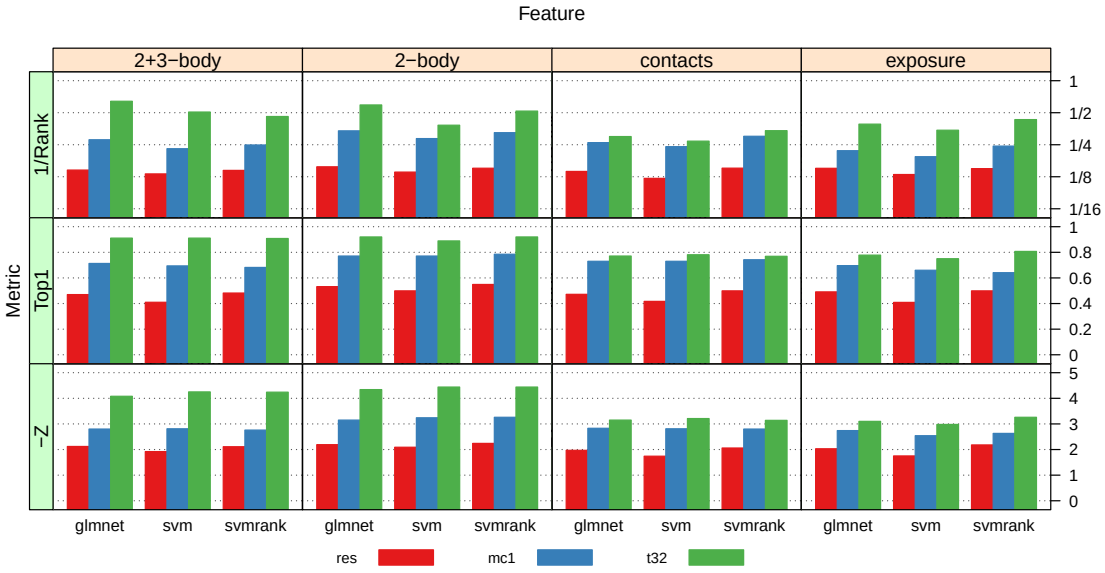
The Params, Nonzero, and Frac. columns of Table 5.3 give information on the size of the models learned in each case. The Nonzero column gives the average number of nonzero parameters in the model and Frac. relates this to the total possible number of nonzeros which is Params. The tendency of SVMs to produce dense models is apparent as in nearly all representations a large fraction of parameters are nonzero. Conversely, glmnet produced relatively sparse models everywhere except when the number of features was small (contacts and angles with 3 groups).

The two rightmost columns of Table 5.3 give information on the stability of the learned models by giving the mean correlation of parameters and the fraction of overlap of selected features amongst the four models learned during cross-validation. Both glmnet and SVM tend to produce fairly stable models and despite glmnet selecting a small fraction of features, there is a high degree of overlap of those selected different data is left out. This will be discussed further in the results of the cross-training experiment.

5.2.3 Binary Classification and Grouped Separation

The learning paradigm exercised by binary classifiers like SVMs and logistic regression is to distinguish all native proteins from all decoys. This is done by assigning model parameters that give a lower score to all native proteins than any decoy protein. Technically, this formulation is more restrictive than needed as in reality, we should only require

Figure 5.1: Visual Summary of Four Fold Cross-Validation (4CV).



This figure visually represents Table 5.3 The model features vary horizontally and the performance metric vertically. Within each cell, bars are grouped by the discrimination method used. Color indicates the representation level. The mean native rank statistic has been inverted to 1/Rank and Z-score to -Z so that larger bars indicate better performance.

a native structure to be lower in energy than its associated decoys, not the decoys of a different protein. For example, it may be difficult for a binary classifier to assign model parameters such that a very large native structure has a lower energy than a much smaller decoy that is close to its native structure. We have employed normalization on the sizes of proteins which may mitigate this to some extent. However it is still interesting to examine what happens when we relax the requirement that all natives are lower in energy than all decoys. We will refer to these two formulations as the *binary classification* formulation and the *grouped separation* formulation. The grouped separation approach has a longer history with many recent examples [69, 70, 88, 89] while the advent of machine learning in structural biology has led to the classification approach receiving some attention [71, 82, 90]. Grouped separation is typically solved using algorithms for linear programming while the binary classification problem is usually addressed with one of a plethora of machine learning tools.

To investigate the merits of the grouped separation model, we employed a ranking SVM (svmrank) in the same 4-fold cross-validation framework as the svm and glmnet (both binary classifiers). The ranking SVM learns a model in which data are grouped and parameters are sought to create a desired ranking within each group. In our case, the groups were the 415 proteins and the members of each group were a native along with all decoys associated with that native. In each group, the native was to be ranked lower in energy than the decoys, but there was no penalty for ranking a native higher in energy than a decoy in a different group. This was a relaxation over the svm and glmnet methods which did penalize ranking a native above a decoy in a different group.

The performance of the ranking SVM is reported in Table 5.3 as *svmrank* along with the svm and glmnet. In most cases svmrank improves slightly over the performance of svm and approaches the accuracy of glmnet. On the contact, exposure, and angle features svmrank produces a better mean native rank and top-1 fraction than svm and glmnet. The comparison illustrates an important point: standard binary classification restricts parameter estimation unnecessarily for decoy discrimination. This is important in situations where the protein is represented using a limited number of structure features employed as in the case of contacts (204-222 features) where the additional flexibility of svmrank led to improvement in the mean native rank statistic.

Table 5.4: Overall Best Method for Each Representation and Feature.

	Method	Level	Feature	Rank	Top-1	Z-score	Params	Selected
1	glmnet	t32	(2+3)-body	1.56 (0.45)	0.911 (0.021)	-4.08 (0.19)	7567	2220 (125)
2	glmnet	t32	2-body	1.69 (0.50)	0.920 (0.017)	-4.34 (0.16)	1584	845 (113)
3	svmrank	t32	exposure	2.32 (0.70)	0.807 (0.028)	-3.26 (0.08)	1386	1370 (2)
4	svmrank	t32	contacts	2.95 (0.45)	0.769 (0.040)	-3.14 (0.19)	204	204 (0)
5	glmnet	mc1	2-body	2.96 (0.53)	0.771 (0.027)	-3.15 (0.08)	1189	650 (26)
6	svmrank	mc1	contacts	3.33 (0.61)	0.742 (0.054)	-2.80 (0.28)	222	222 (0)
7	glmnet	mc1	(2+3)-body	3.59 (0.17)	0.713 (0.032)	-2.80 (0.11)	3075	1530 (328)
8	svmrank	mc1	exposure	4.12 (0.67)	0.641 (0.067)	-2.63 (0.18)	1360	1352 (1)
9	svmrbf	angles	3 groups	6.12 (0.50)	0.492 (0.050)	-1.86 (0.08)	598	- -
10	glmnet	res	2-body	6.45 (1.18)	0.532 (0.040)	-2.19 (0.10)	1073	485 (95)
11	svmrank	res	contacts	6.64 (0.87)	0.499 (0.054)	-2.06 (0.19)	212	212 (0)
12	glmnet	res	exposure	6.66 (1.03)	0.491 (0.063)	-2.03 (0.16)	1266	148 (16)
13	glmnet	res	(2+3)-body	6.92 (1.48)	0.470 (0.035)	-2.12 (0.11)	2682	774 (387)
14	svmrbf	angles	20 groups	7.35 (1.00)	0.518 (0.054)	-2.06 (0.11)	25221	- -

Columns are identical to those given in Table 5.3. The rows are ordered by the Rank performance statistic from best to worst.

The svm method was out-performed by both glmnet, a binary classification with regularization, and by svmrank, a group separation method with no regularization. Estimating parameters using both regularization to induce sparsity and the grouped separation formulation for flexibility could result in robust estimates. To our knowledge, there are no machine learning methods that specifically address this formulation. We incorporated grouped discrimination and regularization into our bead selection method which is discussed in Chapter 6.

5.2.4 Comparison of Representation Levels

A primary concern of this study is to examine how the granularity of a protein representations affects the accuracy it can achieve on some task, in our case decoy identification. Table 5.4 shows the best mean native rank achieved by any method in 4-fold cross-validation. The table is sorted by the mean native rank (the top-1 fraction and z-score follow nearly the same ordering).

As expected, accuracy strongly correlates with the granularity. The fine-grained t32 atomic features occupy the highest accuracy slots while the coarse-grained res and angle features are at the bottom of the table. There appears to be great promise in using the

mc1 representation or coarse-grained models akin to it. The t32 representation uses all atoms and at best identifies 92% of natives using 2-body interactions (line 2 of Table 5.4). Alternatively, mc1 uses a maximum of two beads per residue and gets 77% of natives correct using 2-body interactions (line 5). Between levels, this is a $\frac{1584-1189}{1584} = 25\%$ reduction in parameters for a $\frac{0.920-0.771}{0.920} = 16\%$ decrease in performance. Employing a single interaction point per residue in res representation gives a larger drop, down to a best top-1 fraction of 53% using 2-body interactions. This is a smaller step in parameter reduction ($\frac{1189-1073}{1189} = 10\%$) for a larger drop in accuracy ($\frac{0.771-0.532}{0.771} = 31\%$). The best mean native rank approximately doubles between representations: 1.56 at t32, 2.96 at mc1, and 6.45 at res. These together indicate that the models coarser than two beads per residue will be greatly handicapped in approximating the protein structure.

There appears to be little to no benefit from utilizing (2+3)-body interactions over 2-body interactions. Only at the t32 level is a slight benefit observed, while at coarser granularity no such benefit occurs. This casts a dim picture on the utility of considering higher-body interactions despite their use in recent studies [78, 76, 77]. However, there are many ways to construct higher-body features and our method, grouping all 3-body interactions into a single distance bin, may not be optimal for the task of decoy discrimination. Our choice was based on a desire to prevent the feature space from becoming intractably large while retaining informative interactions but we may have lost some key 3-body information with our binning procedure. It is essential that 3-body interactions show significant generalization in a test set. Table 5.5 shows the nonzero parameters in the (2+3)-body model selected by glmnet. At all three levels, 3-body features are selected with nonzero weights indicating that in the training set they appeared discriminative. The (2+3)-body models do badly on the test sets at the res and mc1 in cross-validation, badly at least compared to their 2-body counterparts. This indicates that many 3-body features do not generalize well and the training set sizes are not large enough to properly identify this fact. Further development of higher-body features will require careful validation to ensure that they do not suffer from over-fitting.

Coarse-grained, two-body potentials have long been used in protein structure analysis but recent work by Pokarowski and co-workers has shown that many published 2-body interaction potentials are essentially the sum of 1-body energies [91, 92]. Our use of contacts, which are 1-body potentials, serves as a performance validation of that work.

Table 5.5: N-body Feature Selection by Glmnet in 4CV.

Level	**2-body**	**(2+3)-body**	**Total**
res	239 / 1073	534 / 1609	773 / 2682
mc1	509 / 1189	1021 / 1886	1530 / 3075
t32	480 / 1584	1740 / 5983	2220 / 7567

The average number of nonzero parameters for (2+3)-body features determined by glmnet during 4CV is shown. Parameters are divided by type (2-body or 3-body) and the possible nonzero parameters is given.

Note that when creating the feature vector for a protein, observing beads for alanine and arginine between 2.0-3.5Å apart has the following effect: for a 2-body potential the count on feature A_R_2-3.5 is increased by 1; for 1-body potentials, the count on feature A_2-3.5 is increase by 1 as is the count on feature R_2-3.5. When the machine learner determines parameters for the 2-body potential, it assigns a single weight to A_R_2-3.5 count which is multiplied by the count and added to the total score. This weight is distinct from other 2-body features such as A_G_2-3.5. In the 1-body case, parameter weights are set for both the count of A_2-3.5 and the count of R_2-3.5 separately and their sum contributes to overall score. This additivity is like a constraint that any A-R interactions are the sum of two 1-body paramters associated with A and R. For that reason, the "contacts" feature is equivalent to the Pokarowski's reduction of 2-body terms to sums of 1-body terms. Their results indicate that 1-body terms should do equally well to 2-body terms in prediction tasks. Pokarowski et al. examined coarse-grained potentials (our res level) and used only a single distance bin for the potentials. Our results in Table 5.3 show at the res level that contacts (1-body interactions) have nearly the same performance (mean rank 6.64) as 2-body features (mean rank 6.45). This is in good agreement with the notion that the coarse interaction of two residues is essentially the sum of two 1-body terms. Results at the mc1 level are similar: 2-body features achieved mean rank 2.96 while 1-body features were close at mean rank 3.33. These findings expand on Pokarowski and co-workers studies in that they are a true illustration of the predictive power of 1- versus 2-body potentials and they are not restricted to a single distance bin (res and mc1 models used 5 distance bins). However,

at the atomic level (t32), the reduction from 2- to 1-body terms gave a larger drop in prediction performance (mean rank 1.56 for 2-body versus 2.95 for 1-body). For a fine-grained interactions and energy, the 1-body approximation apparently breaks down.

Both the contact features and exposure features did surprisingly well at each level of representation. At the coarse res level, they provided nearly the same amount of information as 2-body interactions. Contacts were quite effective at the mc1 level, exposure was less so. Both contacts and exposure were more distant from 2-body interactions at the t32 level, though they still provide a higher degree of discriminatory power than the two coarse-grained representations. Along with the failure of (2+3)-body features at the res and mc1 levels, this suggests coarse-grained models may benefit from pursuing simpler features such as solvent exposure and the density of bead packing.

The angle representation was and oddity in that svmrbf method proved most effective at fitting its parameters, though other methods came close in terms of the top-1 fraction. The svmrbf model for angles clustered into 3 groups surprising achieves a better mean native rank than any res level features. For all methods, clustering the angles into 3 groups provided better performance than dividing into 20 groups based on the amino acid type. This may be in part explained due to model additivity. Interactions between beads can be considered somewhat independently in that two good contacts are more energetically favorable than two bad contacts with one good and one bad somewhere in between. This property lends itself reasonable well to linear models (svm, svmrank, glmnet). Angle bending is not quite so independent: a two locally favorable bends may be globally unfavorable if they create clashes or near clashes in the protein chain. The additivity property is no longer a good approximation and linear estimation methods will miss such relations. Nonlinear learning methods, such as svmrbf, are better at deriving models which incorporate nonadditivity.

5.3 Whole Decoy Set Cross-Validation Experiment (DCV)

In this experiment, a whole decoy set was left out during training and then used to evaluate the learned model. We refer to this methodology as *decoy set cross-validation (DCV)*. DCV is more challenging than four-fold cross-validation (4CV) as decoys from different sets are generated using different methodologies. Decoys used for training may

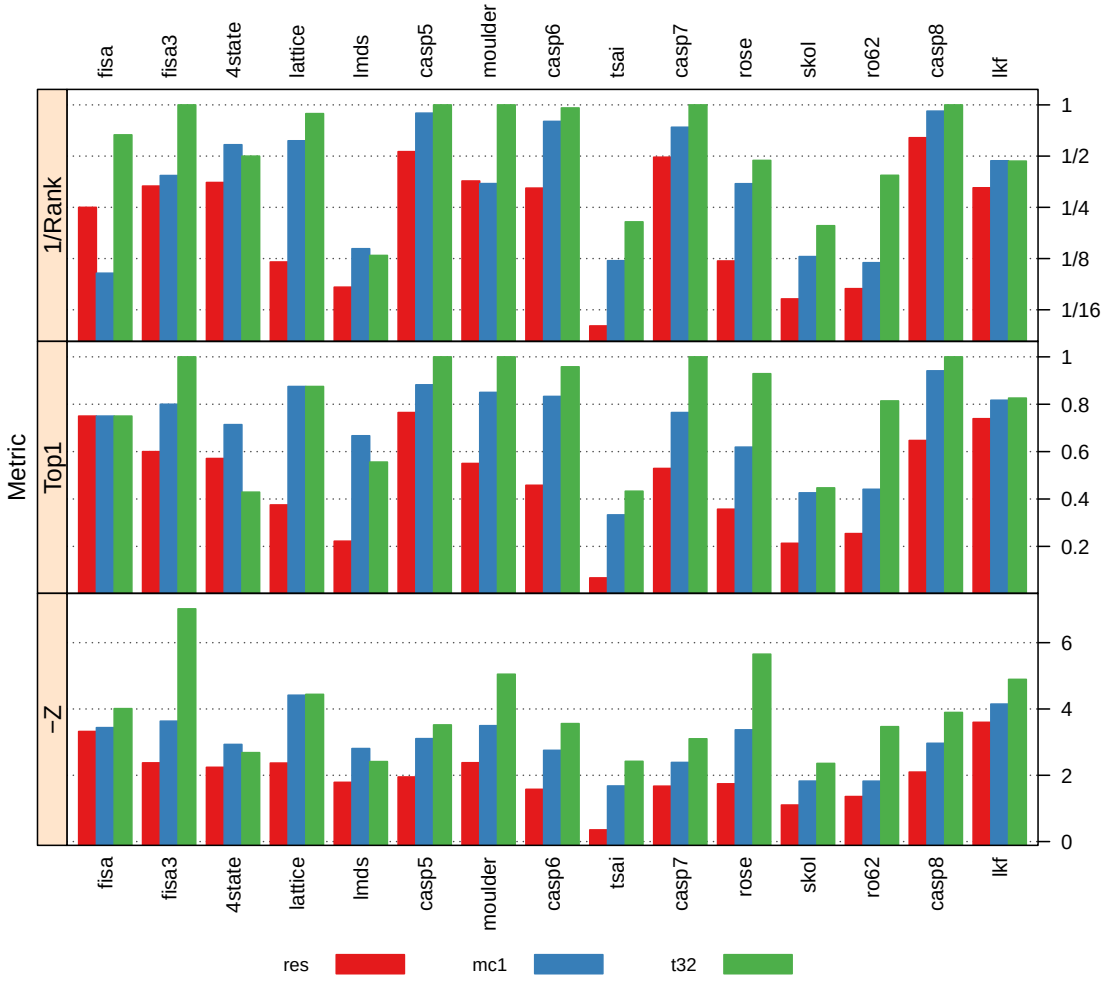
have different characteristics than those that appear in testing. DCV identifies the difficult decoy sets and tests whether patterns learned on decoy sets generalize to truly new data.

In the DCV experiment, we limited ourselves to 2-body interactions at the res, mc1, and t32 representation levels. We used only the glmnet method for parameter estimation. This combination (glmnet with 2-body features) was representative of the best performance according to Section 5.2 and should prove representative of varying structural features and parameterization method.

Table 5.6 presents numerical results for the cross-training evaluation while Figure 5.2 gives a visual summary of the results. As the representation varies from coarse-grained res to fine-grained t32, performance generally improves on all decoy sets. A few exceptions are the 4state and lmds sets in which the atomic detail of t32 performs worse than the 2-interaction point model of mc1. The 4state decoy set was originally created using a reduced representation [59] which may explain why mc1 and res perform favorably on it compared to t32. Though lmds decoys were created using an all-atom model [61], global functional forms were used to explicitly smooth out local energy minima in the decoys. Without unfavorable atomic interactions, the t32 features are not as informative explaining why the mc1 representation, which does not rely on atomic clashes, transfers from other decoy sets to lmds more readily.

Performance across decoy sets varied drastically. The sets lmds, tsai, and skol proved very challenging for all levels. When left out, the best representation for each data set achieved 66% (lmds/mc1), 43% (tsai/t32), and 45% (skol/t32) Top-1 recognition of native proteins over decoys. This is compared to rates in the 80-95% range for most other large sets. Decoys in these sets were all subjected to some energy minimization or structural relaxation to remove many obvious atomic clashes, procedure that is known to substantially increase difficulty [66, 93]. Future work on decoy should focus on making improvements on this kind of decoy set. The ro62 set also provided a challenge for the coarse-grained representations but was handled readily by the t32 level. This set was produced using the ROSETTA software but incorporated a feedback loop to increase the number of decoys near the native structure ([67, 68], Rhiju Das personal communication). Studies of coarse-grained models would benefit from analyzing this set.

Figure 5.2: Visual Summary of Leaving Whole Decoy Sets Out (DCV).



The performance of glmnet is summarized when leaving whole decoy sets out during cross-validation (DCV). Only 2-body interactions trained by glmnet were used. The decoy set left out varies horizontally across the cells, performance metric varies vertically. Color indicates the representation level. The mean native rank (Rank) and Z-score (Z) have been inverted to 1/Rank and -Z so that larger bars indicate better performance.

Table 5.6: Results of Leaving Whole Decoy Sets Out (DCV).

	Rank			Top-1			Z-score					Cor_{all}		
Data	**res**	**mc1**	**t32**	**res**	**mc1**	**t32**	**res**	**mc1**	**t32**	**N**	**Mam**	**res**	**mc1**	**t32**
fisa	4.000	9.750	1.500	0.750	0.750	0.750	-3.322	-3.438	-4.011	204	8.39	0.992	0.990	0.989
fisa3	3.000	2.600	1.000	0.600	0.800	1.000	-2.375	-3.633	-7.021	255	7.86	0.996	0.997	0.994
4state	2.857	1.714	2.000	0.571	0.714	0.429	-2.241	-2.931	-2.684	307	9.03	0.989	0.988	0.976
lattice	8.375	1.625	1.125	0.375	0.875	0.875	-2.369	-4.416	-4.442	408	7.13	0.985	0.988	0.972
lmds	11.778	7.000	7.667	0.222	0.667	0.556	-1.788	-2.807	-2.414	459	7.36	0.977	0.981	0.962
casp5	1.882	1.118	1.000	0.765	0.882	1.000	-1.950	-3.106	-3.520	284	11.09	0.994	0.994	0.989
moulder	2.800	2.900	1.000	0.550	0.850	1.000	-2.379	-3.497	-5.049	1020	10.96	0.978	0.989	0.996
casp6	3.083	1.250	1.042	0.458	0.833	0.958	-1.577	-2.753	-3.560	471	9.19	0.990	0.991	0.986
tsai	19.900	8.233	4.867	0.067	0.333	0.433	-0.353	-1.678	-2.421	1530	7.30	0.924	0.943	0.922
casp7	2.029	1.353	1.000	0.529	0.765	1.000	-1.672	-2.388	-3.101	789	13.74	0.986	0.983	0.983
rose	8.262	2.905	2.119	0.357	0.619	0.929	-1.743	-3.371	-5.654	2142	9.03	0.930	0.934	0.936
skol	13.809	7.787	5.128	0.213	0.426	0.447	-1.103	-1.825	-2.358	2397	8.24	0.904	0.913	0.916
ro62	12.017	8.458	2.593	0.254	0.441	0.814	-1.359	-1.823	-3.467	3009	8.77	0.914	0.898	0.929
casp8	1.559	1.088	1.000	0.647	0.941	1.000	-2.097	-2.965	-3.896	1227	10.31	0.968	0.976	0.987
lkf	3.070	2.130	2.139	0.739	0.817	0.826	-3.598	-4.149	-4.894	5433	8.50	0.884	0.843	0.804
Cor_N	0.205	0.127	0.148	-0.076	-0.300	-0.086	-0.133	0.027	-0.081	1.000	-0.118	-0.919	-0.979	-0.933
Cor_{Mam}	-0.581	-0.474	-0.498	0.434	0.376	0.535	-0.020	0.111	0.039	-0.118	1.000	0.277	0.226	0.312
Mean DCV	6.561	3.994	2.345	0.473	0.714	0.801	-1.995	-2.985	-3.899			0.961	0.961	0.956
Mean 4CV	6.450	2.960	1.690	0.532	0.771	0.920	-2.190	-3.150	-4.340					
SD DCV	5.558	3.210	1.994	0.218	0.184	0.225	0.807	0.815	1.330			0.038	0.045	0.050
SD 4CV	1.180	0.530	0.500	0.040	0.027	0.017	0.097	0.079	0.164					

Only 2-body interactions were used as structure features and only the glmnet method was for parameter estimation. The decoy set *left out* during training and used as the test is listed in the first column. Performance statistics by representation level are listed in subsequent columns. The N column gives the number of structures in each decoy set. The Mam column gives the average Mammoth structure alignment score between natives in the row decoy set the best structure in a different decoy set. The Cor_{all} columns give the Pearson correlation coefficient of the row model with the model trained on all decoy sets. The middle row, Cor_N, gives the correlation coefficient of each column with the N (decoy set size) column. The lower part of the table compares the overall mean and standard deviation of statistics when leaving one decoy set out at a time (DCV, this table) versus leaving one balanced fold out as was done in the previous experiment (4CV, Table 5.3).

The columns for Cor_{all} of Table 5.6 indicates how much a decoy set affects learned parameters. It gives the Pearson correlation coefficient between the parameter vector of the model learned when the row's decoy set is left and the model learned when all decoy sets are used. An important point is that *training on all decoy sets together leads to a perfect model with Rank and Top-1 of 1.0 on all sets at all levels of granularity.* This is clearly an over-fit of the data that will not generalize to new types of decoys. However, analyzing the influence each decoy set has on the all-decoy-set parameters paints an interesting picture.

When correlation is low, it indicates the decoy set is exerting influence on the all-set model as the left-out model parameters differ from the all-set parameters. Confounding this reasoning is the variance in size of decoy sets. The center row of Table 5.6 labeled Cor_N indicates that the influence decoy sets exert on the overall model correlates very well with their size. The performance statistics (Rank, Top-1, Z-score) do not correlate well with the decoy set size (row Cor_N) but the model stability measure Cor_{all} exhibits high negative correlation to decoy set size (rightmost columns of row Cor_N). When larger decoy sets are left out, parameter estimates drift farther from the estimates based on all decoy sets. However, for a small but difficult decoy set like lmds, parameters are similar whether the decoy set is used or left out ($\text{Cor}_{all} = 0.962$ for t32). Clearly some small changes in the parameters have a big impact on performance: training with all sets including lmds gives a mean native rank of 1.0 on lmds, while training with all sets except lmds gives a mean native rank of 11.8 on lmds. A simple correlation coefficient between model parameters does not seem an adequate measure of the stability of those models nor how they will generalize to new data.

As an alternative to simple correlations, we examined structural relations between proteins in different decoy sets. We aligned all native structures in a decoy set against all other natives using the Mammoth structure alignment program [43]. The best structure alignment score for each native in a decoy set was recorded and the average over all natives in a decoy set is given in the Mam column of Table 5.6. A low Mam value indicates the natives in a decoy set share few structural characteristics with representatives in other decoy sets. The row Cor_{Mam} gives the correlation Mam with performance statistics and model stability. It has moderate correlation to Rank and Top-1 and weak correlation to model stability (Cor_{all}). The correlation adds to the explanation of why

decoy sets like lmds, skol, and tsai are difficult: they contain so distinct structures with few similar structures in other sets from which to learn. Counter examples are fisa3 and lattice which have low Mam scores but good performance in terms of Rank and Top-1. However, these sets are small. The combination of structural distinctness and aforementioned energy minimization is likely the full reason why lmds, tsai, and skol are so difficult.

The difficulty of leaving whole decoy sets out is further illustrated by comparing performance on this experiment (Mean DCV) and the results obtained from 4-fold cross-validation (4CV Mean) in which decoy sets were balanced across the four folds. Mean performance statistics are shown near the bottom of Table 5.6. The 4CV experiment has generally better performance statistics than DCV and the standard deviation of DCV folds is much wider than for than in 4CV. This underscores the fact that testing a model on a new decoy set is a true out-of-sample estimate where the decoys may be drawn from an entirely different distribution than the training data.

There is a large difference between predicting a completely new protein structure and predicting the structure of a protein with an identified structural template. This fact is employed in some decoy data sets in that the decoy generation mechanism is influenced by knowledge of the native structure or a template. To assess how much this affects our own study, we looked at the results on DCV aggregated over decoy sets which used knowledge of the native or a template to generate decoys versus those that did not. Ostensibly the use of a native or good template should produce decoys with more native-like characteristics which should conversely make decoy discrimination harder: there are fewer difference between a native and template-influenced decoys. Aggregated results are shown in Table 5.7 along with a listing of which decoy sets fell were influenced in some way by a template. The reasons for a decoy set qualifying as template-influenced (Templ=yes) are describedin detail in supplementary material; use of a close structural relative or the native protein itself or fixing native secondary structure confers template influence. The means for template-free generation ("No" rows in Table 5.7) are indeed better indicating that these decoys are easier to identify than those based on templates. However, none of the performance measures exhibits a statistical difference between template-based and template-free decoy generation sets. This is due to the large variance of performance between the datasets in each group.

Table 5.7: Difficulty Discriminating Decoys Generated with and without Templates.

			Rank			Top-1			Z-score		
Templ?	**#Sets**	**Stat**	**res**	**mc1**	**t32**	**res**	**mc1**	**t32**	**res**	**mc1**	**t32**
No	6	Mean	4.763	2.927	2.442	0.522	0.766	0.802	-2.045	-2.936	-3.497
Yes	9	Mean	9.259	5.595	2.201	0.401	0.636	0.800	-1.920	-3.060	-4.503
No	6	SD	6.155	3.589	1.439	0.244	0.213	0.200	1.017	1.081	1.632
Yes	9	SD	4.613	2.604	2.374	0.198	0.153	0.252	0.697	0.652	0.988
Yes v. No		p-value	0.163	0.154	0.811	0.337	0.232	0.989	0.800	0.808	0.216

Columns are (Templ) whether templates influenced the decoy generation, (#Sets) the number of data sets in the group, (Stat) mean or standard deviation, and (remaining columns) the aggregate statistic for each performance measure. The upper portion of the table shows the mean and standard deviation of performance measures on DCV (rows of Table 5.6). Data sets which used templates in decoy generation are 4state, lmds, moulder, skol, casp5-8, lkf. Those that did not use tempaltes are fisa, fisa3, lattice, rose, tsai, ro62. The bottom row shows p-values for a two-tailed T-test on whether the means of each stastistic are different from one another. High p-values indicate the means are not likely to be different.

5.4 Discussion

Two over-arching observations emerge from our comparison of three granularities of protein representation and variety of energy terms in them. First, the atomic-level detail (t32 model) gives the best performance definitively, but great improvement over single bead per residue (res model) can be gained by differentiating side and main chain interactions (mc1 model). The best mean native rank over 415 native proteins in 4-fold cross-validation are 6.45 for res and 2.96 for mc1. This improvement comes at a very low cost in terms of the number of parameters associated with the mc1 model: for 2-body interactions there are 1073 parameters to learn in res versus 1189 in mc1, an increase of only 116 parameters. Going to atomic detail in t32 requires 1584 parameters for 2-body interactions.

The second broad message is that low-resolution features, (contact counts and solvent accessible surface area), provide a surprisingly large amount of discriminatory power regardless of their representation level. This is compared to 2-body and (2+3)-body interactions. The decrease in performance for using contact counts or solvent exposure rather than 2-body interactions at the t32 level gives a drop in mean native rank about 1.1; the average drop is only 0.77 at mc1 level and 0.20 at the res level. This is a sign that using lower resolution energy terms when low resolution models are employed

does not compromise accuracy. At least, the exclusive use of contacts or exposure does not compromise accuracy much more than the initial choice of a coarse-grained representation.

In terms of parameter estimation methodology, three additional technical results come from our analysis. There is little benefit from using nonlinear parameter estimation techniques for the protein representations and features examined. The nonlinear svm models performed little better than linear versions and have several drawbacks (two hyper-parameters versus one for linear, longer training times, and a lack of explicit parameter representations, Section 5.2.1). It also seems that training models through grouped separation rather than binary classification, as was done with svmrank, deserves additional exploration. Combining this training approach with the sparsity-inducing regularization of glmnet could produce more robust parameters. We are currently testing methods to do this. Finally, performance on a single decoy set, even within cross-validation, is not indicative how well a model will generalize to new decoy sets. It is difficult to assess how stable any of these models might be as small changes in parameter can drastically alter performance on difficult decoy sets.

Our intention with this study is not to suggest a particular scheme by which to do structure prediction, but instead to get at whether coarse-grained models limit the accuracy of prediction methods. An oft-employed protein prediction strategy is the following. We want to formulate the best structure prediction within time T. The first step is to search for a structural template using one of many good methods. Should a template or templates be found, the amount of conformational sampling required is reduced by many orders of magnitude by searching near the template. The remaining time up to T is probably best spent using a fine-grained model like t32 as the representation does not limit the accuracy of predictions much. If no template is found, then we must do a large-scale sampling of the protein's conformational space to get a sense of low-energy shapes. Certainly employing a coarse-grained model will limit the accuracy of predictions, but much more conformational ground can be covered using a coarse-grained model due to the smaller number of beads in the model. Our results indicate coarse-grained models provide enough fidelity to guide sampling to reasonably close approximations of native structures. So lacking a good template, most of the prediction time up to T should

be spent on coarse-grained sampling, perhaps with some subsequent fine-grained refinement. To our knowledge, most successful prediction schemes work roughly in this way with possible iteration between coarse and fine modes. From the stand-point of template utilization, our work confirms that this strategy is quite reasonable. Further inquiry is required to determine whether analysis of template-based or template-free decoys can yield insight into specific prediction tasks. For example, restricting parameter estimation to template-based decoys only may increase our understanding model refinement while the template-free setting may be more useful to derive parameters for new fold predictions.

Chapter 6

Mixed Model Selection

In this chapter, we propose a new method which can select bead types from a mixture of model granularities while maximizing the discrimination of native from decoys. This is a first step towards a data-driven method for protein model selection. We illustrate behavior of this bead selection method on the full set of decoys from the last chapter and explore how bead types from the different levels of granularity are combined.

To date, efforts to derive reduced protein representations have primarily focused on choosing the model according to physical intuition. After choosing a representation and functional form, force field parameters a determined in order to reproduce experimental results or discriminate structural decoys. Representations such as t32, mc1, and res, are derived from a priori knowledge of what seems sensible for modeling purposes and other features are discarded to avoid computational costs.

A pure data-mining viewpoint takes the opposite approach of including all potentially useful features in an unbiased way. The useful features are selected during parameterization to maximize accuracy. It is not clear whether this philosophy can be directly incorporated in molecular dynamics. However, for the decoy discrimination setting, it is readily employable to select beads from different representations to create a mixed model.

We conduct an experiment in which each protein from the full decoy set of Section 5.1.1 was simultaneously represented using res, mc1, and t32, and all 2-body interactions between beads were counted (see Section 5.1.3 and Section 5.1.4). This included cross-level interactions such as those between the mc1 bead R.sc (arginine sidechain) and

the t32 atom type OX1. We developed an optimization procedure to do decoy discrimination in this mixed representation. It is a *bead selection method* thus we abbreviate it as *BSM*. Details of BSM are given in Section 6.1. Briefly it uses a regularization path approach similar to glmnet while doing grouped decoy separation as svmrank does (see Section 5.1.5 and Section 5.1.5 for descriptions of these two machine learning methods from the last chapter). At high regularization levels, few beads are in the model while progressively lowering regularization allows more beads to enter the model and lets us observe the trade-off between model complexity and performance. Beads are selected from any of the three granularity levels giving us insight into which parts of the protein may be approximated using coarse-grained beads. Parameters were fit using all proteins in the 15 decoy sets so there is no cross-validation in this experiment.

6.1 Decoy Separation with Bead Selection (BSM)

The glmnet method performs feature selection but it has the following limitation. For 2-body features, individual pair-wise parameters such as R.res-A.res (a res level interaction) may be driven to zero. However, in another parameter associated with R.res, such as R.res-C.res is nonzero, C.res still plays a part in the model. Rarely does glmnet drive all parameters associated with a bead type to zero simultaneously. As long as some pairwise interactions are nonzero for a bead type, it cannot be dropped.

We surmounted this limitation by designing a method which discriminates natives from decoys while doing bead selection. As it is a *bead selection method* we refer to as *BSM*. BSM is designed to simultaneously drive all parameters associated to a bead type to zero together thereby allowing the bead to be eliminated.

As an optimization problem, BSM takes the following form. The vector of parameters or feature weights w must be chosen so that the decoys have higher energy than natives. Formally this is

$$w^T x_{decoy_i} - w^T x_{native} > 0 \tag{6.1}$$

where x are the feature vectors for a decoy and its associated native structure. We constructed an $n \times f$ decoy matrix D which is the difference of feature vectors between each decoy and its corresponding native protein. The rows of D are the term on the left-hand side of Equation 6.1; columns correspond specific feature differences. The

matrix vector product Dw gives a vector of the energy differences between decoys and natives. In this formulation we only compare natives to their associated decoys as in svmrank. Also as in support vector machine approaches, we used the hinge loss to encourage a large energy gap between decoys and associated natives. The hinge loss is $h(z) = \max\{0, 1 - z\}$ and when z is a vector, it produces a positive vector. It reaches a minimum of zero when input z is 1 or greater. The loss function is denoted $h(Dx)$: any decoy not exceeding the native in energy by 1 unit has nonzero loss.

To balance the loss function, we applied regularization to the parameters w. This took a special form where the penalty applied to groups of variables associated with a single bead type. In the case of 2-body interactions, each feature was associated with two bead types such as the interaction of $w_{CAH-R.sc}$ where bead types CAH and R.sc are from the t32 and mc1 representations respectively. For a particular bead type A, we compute the maximum absolute value, $\max_X |w_{A-X}|$, where X can be any bead type. This coefficient is penalized during parameter estimation. As in glmnet, the absolute value or L1-penalty induces sparsity in parameters driving some of w to zero. The max or L-∞ norm has also been used in literature for regularization, and their combination has come under some scrutiny recently [94].

The final form of our optimization problem is then

$$\min_w h(Dw) + \lambda \sum_A \max_X |w_{A-X}| \tag{6.2}$$

where the fixed parameter λ governs the trade-off between loss and regularization. The problem is convex so that it has a global minimum but nonsmooth due the hinge, L1, and L-∞ loss.

We explored several methods to solve the optimization problem for BSM. Equation 6.2 can easily be cast as a linear program, but standard LP solvers have memory requirements that scale quadratically with the problem size. In our situation we are using a mixture of all 2-body features from the res, mc1, and t32 representations so that D is large and dense, around 20K by 10K with 45% nonzero entries. This proved to much for standard solvers. Coordinate descent is another reasonable choice as it is used to great effect in approaches such as glmnet [87]. However, careful analysis of Equation 6.2 reveals that the regularization term is nonseparable. In such cases, coordinate

descent is not guaranteed to converge [95]. Instead we employed the subgradient descent method which is very general but suffers from limited accuracy and speed [96]. In our case, tractability and solvabilty out-weight speed concerns.

In Equation 6.2, we started λ at a very large value which drives the entire parameter vector w to zero and gradually reduced the magnitude of λ. This is identical to the regularization path approach of glmnet in that bead types will enter the model by becoming nonzero at different points along the path. We used 2500 subgradient steps at each value of λ. Step sizes were reduced were reduced in the subgradient method using $1/\sqrt{k}$ where k was the subgradient iteration.

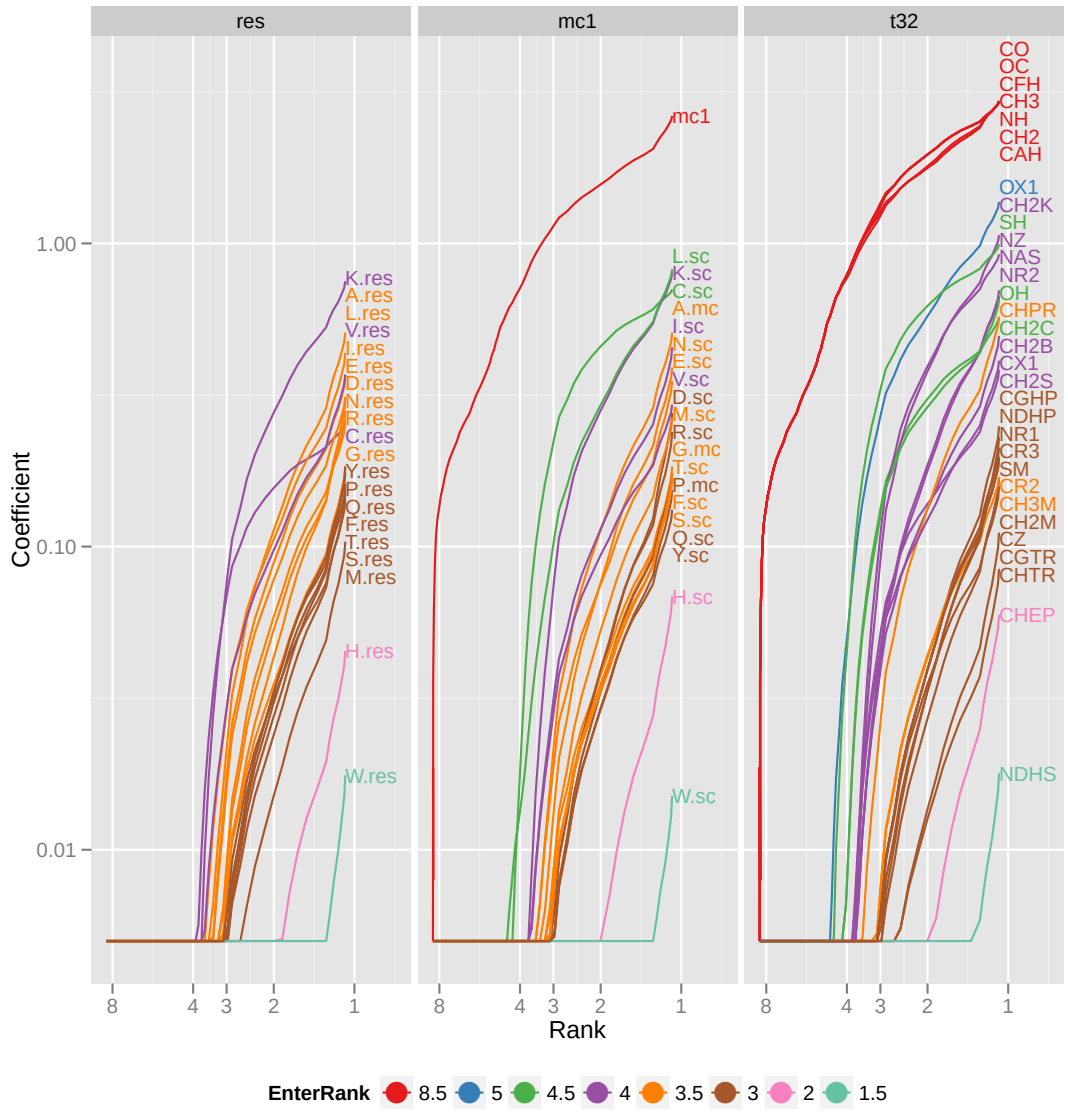
6.2 Analysis of Selected Beads

The main results of this experiment are shown in Figure 6.1. Our BSM procedure varies a regularization penalty which gradually allows more bead types to enter the model. Allowing more beads into the model tends to give better performance so that each regularization point corresponds to a performance statistic. The plot in Figure 6.1 contrasts the mean native rank statistic against the coefficients associated with each type of bead in the three representation levels. Color indicates when the bead types enter the model by going from a zero to a nonzero coefficient. The bead types are listed on the right side in positions roughly corresponding to their weight when the mean native rank reaches 1 (e.g. all native structures have rank 1 for perfect performance).

The first beads to enter the model were related to the protein backbone. From t32 the types NH, CAH, CO, and OC are backbone atoms as is the mc1 bead mc1 (the bead type and representation level are both named mc1). These types overlap in that they represent the same parts of the protein, but according to the model's behavior, including both is beneficial. Additional t32 atom types that entered immediately are CH2, CH3, and CFH, the beta and gamma carbons of most residues. Using only these 8 bead types, a mean native rank of 4.6 was achieved. This is interesting in that it indicates a large number of decoys must contain obvious backbone defects.

Next to enter was the t32 atom OX1: it represents the charged oxygens in the side chain of aspartic and glutamic acid. It was followed shortly afterward by t32 atoms SH and CH2C, the sulfur and beta carbon in cysteine, and OH, the oxygen in serine,

Figure 6.1: Results of mixed bead selection by BSM.



The x-axis shows the mean native rank of the model generated. Rank improves from left to right. The y-axis shows the coefficient associated with each bead type. A zero coefficient means a bead type may be eliminated from the model. Color indicates the approximate rank at which beads enter the model as the regularization penalty varies.

threonine, and tyrosine. Additionally, the mc1 beads L.sc (leucine sidechain) and C.sc (cysteine sidechain) entered around this rank. It is interesting to again see overlapping elements, cysteine sidechain beads from both t32 and mc1, enter at approximately the same time. At this regularization level, 14 bead types were utilized which gave 3.9 mean native rank.

The bead types then entered in larger groups with a variety of t32 and mc1 beads activating. Beads from the res representation also entered. Several coherent groups representing charged side chains entered at approximately the same rank including lysine (K.res, K.sc, CH2K, NZ), aspartic/glutamic acid (CH2B, CX1, D.sc), and arginine (NR2). The cysteine res bead, C.res, also entered at this regularization level, well after the t32 and mc1 representations.

The remaining beads entered in large groups except for six outliers which entered very late. These were related to histitdine (H.res, H.sc, CHEP) and tryptophan (W.res, W.sc, NDHS). Their late inclusion indicates they do not factor into decoy discrimination heavily.

It seems a great deal of discriminatory power resides in only a few bead types. Modeling the backbone properly gives the initial and largest performance boost to achieve a mean native rank of 4.6. Adding a few select bead types that model charged groups and cysteine brings the rank down to 3.9. After that, a wider variety of bead types is required to get better rank.

6.3 Behavior of Bead Selection

The behavior shown in Figure 6.1 illustrates several deficiencies of BSM. Bead types representing the same part of the protein at different granularities seem to enter at the same time. Ideally we would like BSM to prevent such redundancies. Figure 6.1 does not illustrate the maximum performance achievable by models using a subset of bead types. It may be that using only the first 14 bead types selected, a lower rank can be achieved, but this would require parameterizing only on these types. The BSM allows other bead types to enter as the regularization level changes giving only a rough idea of how effective each group of beads will be in isolation.

While it is tempting to compare the ranks achieved by the mixed selection of BSM to those presented in Section 5.2 and Section 5.3, those experiments used a cross-validation framework that give more robust estimates of performance on future data. BSM was evaluated on all data and may over-estimate the achievable rank by the selected models. Our purpose was to explore the potential of automated model selection. Testing the mixed models produced by BSM will be the subject of future work.

6.4 Discussion

Our data-driven approach to selecting proteins mixed representations (BSM) led to modest insight into mixing beads from different granularities Section 6. Rather than select one coarse representations for low performance and gradually shift to a finer-resolution as regularization is eased, BSM seemed to select similar beads from multiple representation levels at the same regularization points. Backbone beads are selected initially, then a few important side chain beads, particularly cysteine, then equivalent res/mc1/t32 beads at similar levels of regularization. While this gives some indication of the relative importance of different bead types, in most modeling situations we would not use redundant representations such as the mc1 backbone bead along with t32 backbone atoms like CAH, NH, and CO. In general, enforcing mutual exclusion in training would destroy the convexity of the optimization problem making it much less tractable to solve numerically. While difficult, it is worth additional work to determine if alternative formulations exist which produce less redundant models as this would have much greater impact in the modeling community.

Chapter 7

Energy Minimization for Protein Structure Prediction

This chapter surveys work related to our energy minimization framework, Marie, which is developed in the next chapter. We review the current foundational theories of what causes proteins to fold, approximations for this process, some methods that have been used to speed the location of low-energy conformations, and some subjects related to our optimization approach.

Most naturally occurring proteins assume their native shape relatively quickly, implying a fast search of conformational space [97]. The current working theory of why this happens is that natural proteins have an overall funnel-shaped energy landscape: there are some energy peaks and valleys, but the unique native structure is situated at the bottom of a relatively deep and broad energy trough where conformational search tends to lead [98].

A grand goal of computational structural biology is to efficiently simulate proteins folding in silico thereby allowing insight into their behavior and function in the living cell. As such dynamic simulations are incredibly difficult and computationally intense, a frequent zeroth-order approximation is simply to determine only the folded structure rather than simulate the whole folding process. Despite great strides by the protein structure prediction community, this is still a difficult task for proteins with new structure. Most *ab initio* approaches exploit some notion of an energy function and perform

a conformational search in an attempt to identify low-energy candidates for the native structure. To speed the conformational search, many such structure prediction methods approximate the physical dynamics of a system at high levels of abstraction such as the swapping of whole backbone fragments.

Traditionally, the conformation of a protein structure is represented by model variables such as coordinates or dihedral angles. An energy function is chosen which calculates the potential energy of the protein based on these model variables. Potential energy functions, or force fields as they are often called, are typically highly nonconvex so that the energy surface is extremely bumpy and finding a global minimum is difficult. However, it is believed that if the energy function is reasonably accurate, then the global energy minimum represents the conformation a naturally occurring protein will adopt. Thus it is of interest to locate the global minimum for a given protein and a variety of methods have been proposed to expedite finding it. Here we discuss some of the work that is relevant to our own approach.

7.1 Energy landscapes and Hydrophobic Collapse

A very general and plausible explanation for protein folding mechanics is given in the theory of energy funnel landscapes [99, 100, 98, 101]. It describes conformations of a protein as existing on a landscape of free energy with peaks at conformations which have unfavorable free energy properties and valleys at conformations where free energy is minimized. Every flexible molecule has such a landscape and though such molecules exist as statistical ensembles due to random fluctuations, the conformations in free energy valleys occur more frequently. Proteins are interesting in that there is evolutionary pressure to select amino acid sequences that adopt certain shapes which are beneficial to the survival of the organism. The fundamental dogma of structural biology, that a DNA sequence encodes an amino acid chain which gives rise to a specific protein shape, is in part a result of the specific energy surface induced by the selected amino acid chain. Presumably, the observed natural amino acid sequences have been selected for the stability, folding speed, and utility of the folded protein shapes they encode. The most obvious mechanism for this, advocated by the energy landscape theory, is that the

energy surface of naturally selected proteins is a broad and deep funnel with the desired folded or *native* conformation at the bottom of the funnel. This accounts for the speed and stability observed in proteins.

Energy landscape theory explains protein folding as an energy minimization process but stops short of specifying which forces drive the process. The free energy minima must be deep enough to overcome the loss of entropy of going from a flexible chain to a relatively inflexible, folded protein. However, the contributions of various energetic components that create the funnel are the subject of continued debate. It has long been agreed that hydrophobic collapse plays a significant role in folding[102, 103, 104]. Amino acids with hydrophobic side-chains do not have favorable interactions with water and tend to reside on the interior of the protein while polar side chains can participate in hydrogen bonding and often appear at the surface. Some accounts cite this as *the* primary driving force and claim other observed phenomena in proteins, such as the hydrogen bonding in secondary structure helices and sheets, are merely local optimizations that enable more efficient packing and burial. Alternative hypotheses place much greater emphasis on intra-protein hydrogen bonding and tout it as a significant factor if not the fundamental driving force in folding[105, 106, 107, 108]. While it will likely require continued advances in experimental techniques to put the debate over the primary driving force to rest, the two alternatives are under continued scrutiny through simulation. We show how hydrophobic collapse can be approximated in our Marie framework in Section 8.4 and show results starting in Section 9.4.

7.2 Go-Potentials

Go-potentials are an idealization of the energetics involved in protein folding. They were proposed to reduce the "frustration" or roughness of the energy surface of a protein while still preserving most of the essential features of protein energy surfaces. The original Go-potential used an extremely simplified protein representation of beads on a 2D lattice and the only energetic contributions came from pairs of beads deemed native contacts [109, 110, 111]. Since then, Go-like potentials have been found to induce a funnel shape for most naturally occurring proteins [112], replicate the multi-state folding dynamics of small proteins, and correlate well in simulations with experimentally observed folding

rates [113]. Go-like potentials have been employed with coarse-grained representations of proteins and full atomic models, in lattice simulations [114], in standard molecular dynamics simulations [115], and with other conformation search methods such as Monte Carlo search [116], many of which are mentioned in the review by Onuchich and Wolynes [113].

The feature that relates all Go-like potentials is that two beads in contact in the native conformation attract one another while beads not in contact in the native conformation do not attract one another. The attractive forces can be either uniform for all native contacts or vary [117], but are treated as far outweighing the energetic contributions of nonnative contacts, so that nonnative contacts may be ignored. This creates a much smoother energy surface and has led widely to near-native conformations in folding simulations. However, the computational cost of locating low-energy conformations even on the smoothed energy funnel remains, a problem we address here.

A key issue is determining what exactly constitutes a native contact. Several approaches have been employed but the most common is to select a cutoff distance, later referred to as *cut*. Any pair of beads or heavy atoms closer than *cut* are considered in contact and are assigned attractive forces. Some literature has reported experimentation with different values of *cut* [118] and there is a general belief that a range of *cut* values will induce Go-potentials where the native structure is at the energy minimum. In this work we will be able to efficiently verify this fact using our energy minimization framework. A Go-like potential is established in Marie in Section 8.3 and experimental results using it are discussed starting in Section 9.2.

7.3 Convex global underestimator by Dill

A conformation search approach designed to exploit the funnel shape of a protein energy landscape was proposed by Dill, Phillips, and Rosen in [119, 120] and is referred to as the convex global underestimation (CGU) technique. Conformations of a protein are randomly sampled and then the structure is relaxed to a local energy minimum according to an energy function of choice. The original energy function used in CGU was a simple combination of excluded volume to disallow clashing atoms, hydrophillic/hydrophobic contact energy, and torsion angles. Ostensibly any energy function will fit into the

framework so long as the configuration variables associated with the conformation can be easily extracted. Once a variety of conformations have been sampled, a convex function is fit which maps the configuration variables to the measured energies at each sample. The convexity is guaranteed in the original paper by fitting a sum of convex quadratic functions. The minimum of this convex fit is quickly obtained. The process then repeats but this time with sampling restricted to conformations near the global minimum of the fit.

This iterative process has much in common with trust-region methods from optimization theory. [121]. They repeatedly build a model of the objective based on a region of interest, in CGU's case the sampled conformations, and directs further attention to promising areas near the model minimum.

Dill, Phillips, and Rosen found that for small proteins, up to 36 residues, CGU predicted structures that were very close to native structures. This lent evidence that the proteins examined possessed a funnel-shaped energy function. They noted that the approach is embarrassingly parallel in that once the sampling region was determined, any number of processors may independently select from the region and determine relaxed local minima. The coordination step of fitting a convex model to the local energy minima takes a relatively short amount of time so that on a 32-node super-computer in 2001, a the global minimum of a 50-residue protein was located in 9 hours. They noted that the compute time of CGU scales empirically at the rate of n^4 where n is the number of degrees of freedom.

CGU is different from the present work in the following ways. The CGU approach approximates an arbitrary energy function in a region using convex functions. It then refines the approximation in regions of interest. Our method, Marie, directly uses a convex energy function during the entire prediction. The Marie energy function is somewhat more costly to minimize as it requires solving a semidefinite program rather than a quadratic program.

Marie entails iteration as we attempt to overcome the nonconvex dimensionality problem by gradually biasing search towards low-dimensional answers. Solving this series of SDPs in some ways parallels solving the sequence of quadratic subproblems of CGU, however in our approach we are addressing a nonconvex constraint while CGU is narrowing its trust-region to likely low-energy regions (see above).

7.4 aBB global energy minimization by Floudas

Similar to the CGU method described above, the alpha branch-and-bound (aBB) method applied to proteins in [122] uses an atomically detailed energy function. Whereas CGU uses convex approximations of the function in order to speed the search, the aBB method guarantees a global minimum will be found by using a branch and bound strategy. Dihedral angles are used as internal coordinates for the protein conformations. Angle regions are partitioned into regions. Performing local minimization of the energy function U starting from any point in the angle region gives an upper bound on the region's minimum energy. The authors show a lower bounding function L can be constructed for the region by adding to U a separable quadratic function on the deviation of angles from the boundary of the allowed region. These terms "overpower" the nonconvexity of U to make L convex so that its minimum can be quickly found. If the lower bound determined by L exceeds the best upper bound so far seen, the region can safely be discarded as not containing the global minimum. If not, the region is bisected and the process is repeated on the smaller regions. The authors prove this method is guaranteed to explore all regions in angle space that may contain the global minimum and when the correct region is found, the upper an lower bounds coincide yielding the minimum-energy conformation.

This procedure is computationally expensive. In order to speed it up, the authors partition dihedral angles into 3 groups: those that vary and allow branching, those that vary in minimization but are not used for branching, and those that are fixed based on experimental data. The original study illustrated experiments on all-atom models of single amino acids which took 10 minutes or less at the time. They observed an empirical scaling of under n^3 where n is the number of variables allowing branching.

7.5 Fragment assembly: Rosetta and TASSER

Primary examples of fragment assembly methods include Rosetta [51] and TASSER [123, 124]. Central to these methods is the that the recurring local structures in proteins can be exploited to speed up conformational search. To that effect, rather than continuously optimize conformation parameters, fragment assembly methods make discrete changes by altering the structure of whole segments of a protein. These segments are three or

nine residues in length in the case of Rosetta. Selection of the library of fragments to be used is done a priori. The probability that an amino acid sequence will adopt a fragment structure is calculated by analyzing the known protein structures in a database and frequent fragments are retained and may potentially be substituted into a closely matching sequence.

Fragment assembly is usually performed in a Monte Carlo framework so that particular fragment substitutions are accepted or rejected probabilistically. The substitution is attempted by switching the shape of a protein segment with that of the selected library fragment. The chance for acceptance is based the overall change in the global potential energy of a protein due to its new conformation with reductions increasing the likelihood of acceptance.

Changing the geometry of whole segments of a protein a biases local structure towards known energetically favorable conformations. Many local minima are skipped as the overall shape of the protein hops between conformations based on the fragment substitutions. The discrete fragment substitutions are interspersed with continuous, all-atom coordinate minimization steps.

Fragment assembly methods usually perform many structure optimizations with varied initial conformations in an attempt to sample as much of the potential energy landscape as possible, a requirement shared by traditional molecular dynamics approaches. The runs are largely independent and can be done in parallel which has led to distributed efforts such as Rosetta@Home [125] that mirror previous traditional molecular mechanics efforts such as Folding@Home[126]. However, the approximation used by fragment assembly, swapping in and out whole local geometries, has proven to increase the likelihood near-native protein conformations are sampled making these some of the most successful methods according to the CASP experiments[127].

7.6 Distance Matrix Embedding

The primary object of concern in our optimization problems is a variable equivalent to the distance matrix of a protein. Historically, a variety of methods have used similar techniques when distance information is available and coordinate information is desired.

Our model functions in the space of distance matrices making it natural to examine how other distance geometry methods fare in comparison. Distance geometry approaches use a pairwise distance matrix as input and produce d-dimensional coordinates which adhere as closely as possible to the available proximity information. In our case $d = 3$ which corresponds to three-dimensional molecules.

A number of disciplines use distance geometry methods. In statistics and psychology, multidimensional scaling has been studied as a way to visualize data using only similarity or dissimilarity information [128]. There has been renewed interest in distance geometry through semidefinite programming due to its applications to wireless sensor location [129] and to graph embedding [130].

In structural biology, nuclear magnetic resonance spectroscopy (NMR) can generate distance-based information about molecules in solution, particularly proteins. Converting this distance information into atom coordinates allows the structure of the protein to be ascertained [131, 132]. Recently SDP methods have been applied to resolve coordinates based on incomplete distance information [133]. Using all atoms in a protein leads to very large SDPs, thus the approach in [133] uses a series of smaller problems on limited subsets of all atoms and then employs a stitching procedure to combine the results. We are interested in preserving some of the global energy properties of proteins so we adopt a reduced model of the protein to keep the SDP tractable (see Section 8.1).

7.7 Distance Geometry and Direct Protein Structure Prediction

Distance geometry methods have also been explored in their own right for ab initio protein structure prediction [134]. The general idea is to start with a distance matrix in which all atoms of a protein were at optimal distance from one another according to a simple energy function. For a simple Lennard-Jones type energy function, this can be done analytically as there is a single distance at which two atoms are at a minimum and the overall minimum for the sum of these puts all pairs of atoms at those distance. Such an arrangement is impossible in low-dimensional space: imagine the optimal distance is 1 unit but we are trying to place three atoms on a one-dimensional line, each 1 unit apart. It is impossible and instead requires a two-dimensional plane if all are to be at

the optimal distance from the others. So the arrangement of many atoms at optimal distance resides in a high-dimensional coordinate space. The problem is convex in the relaxed dimensionality but nonconvex if lower-dimensionality is enforced. In [134], from the initial point, a penalty function is used which directly penalizes the embedding dimension of the protein. The weight on this penalty function starts at zero and is gradually increased to force lower dimensional embeddings of the protein. Tests on small proteins (5 residues) showed some promise for the approach but it has received little attention since its original proposal.

Our work contends with similar difficulties in that we propose a very simple energy function based on burial which is convex in relaxed dimensionality. In order to force low-dimensional solutions, we use convex iteration Section 8.10 which solves a series of SDPs that gradually increase the weight on an objective term biasing the solution towards low dimensional coordinate space.

Chapter 8

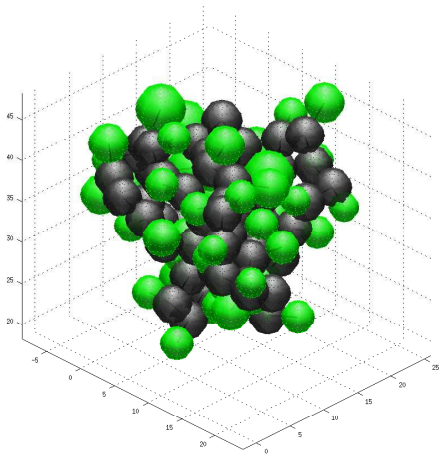
The Marie Model

In this chapter we give an overview of our energy minimization framework, Marie. We first describe the reduced model of proteins we employ in experiments. We then describe the two main types of energy functional forms we explore with Marie: a Go-like potential and a hydrophobic collapse energy function. The latter comes in several variants. Next the constraints used to represent bonds and avoid steric clashes are described. Finally, we describe how the functional forms and constraints can be represented as a semidefinite program and thus solved using convex optimization methods.

8.1 A Reduced Protein Model

We work with a reduced model of protein structure in which each residue is represented by one or two hard spheres, referred to as *beads*. The first bead is centered on a residue's α-carbon and contains all heavy atom centers on the main chain and the β-carbon if it is present. A second bead is at the barycenter of any remaining side chain atoms and contains their centers. Proline is treated specially and is represented with only a single bead which is centered on the α-carbon. Two sets of parameters are required for this model: the radii of each type of bead and the distances between bonded beads of each type (adjacent main chain beads and bonded main chain/side chain beads). The mean value for these parameters is calculated over the entire dataset and the true structures of proteins are adjusted to take on these idealized values. The fit of idealized structures to natives is quite good with an average RMSD of 0.285 ± 0.045Å. In addition, each bead

Figure 8.1: Bead Model for Protein 1tuc used in Marie Experiments.



Main chain beads are colored dark gray and side chain beads are colored light green. Each bead encloses atoms belonging to a particular residue on either the main or side chain.

is given a fixed size with the radius of the bead taken as the average for that kind in the database. Bead sizes are calculated by finding the mean coordinate of the atoms to be represented and extending the radius of the bead to enclose the center of the heavy atoms along with 20% of their van der Waals radii.

We studies this reduced model in Chapter 5 (and reference [135]) where it is called *mc1*; it is an intermediate between single-bead per residues and all-atom protein models. Reduced models of proteins are frequently used in structure prediction methods as reviewed in [136]. Figure 8.1 shows an example of the native structure of a protein in its mc1 representation.

8.2 Overview of Marie

The central idea of Marie is to predict protein structure by solving an optimization problem with the following features.

Bonds Distances between bonded beads are maintained exactly (Section 8.6.1).

Clashes Clashes between non-bonded beads are prevented by maintaining a minimum distance between them (Section 8.6.2).

Secondary Structure Any known secondary structure (SS) is maintained by maintaining exact distances between main chain beads in the same secondary structure helix or sheet (Section 8.6.3).

Minimize Energy Locate the conformation which minimize global energy according to the selected energy function. The two energy functions we explore are a Go-like potential and a function affecting hydrophobic collapse.

Go-like potential Bead pairs in contact in the native conformation are attracted to one another and should have their distance minimized while avoiding clashes of all beads (Section 8.3). Native contacts are assigned an attractive force of $obj_{ij} = 1$ associated with the squared distance d_{ij}^2 between natively contacting beads. The cutoff distance *cut* defining native contacts was varied in experiments to explore its effects on the minimum energy structures (methods in Section 8.3, results in Section 9.2 and Section 9.3).

Hydrophobic Collapse Find a conformation that balances three components: (1) compactness by minimizing the burial chamber *radius*, (2) minimizes exposure *exp* of hydrophobic beads, and (3) minimizes burial *bur* of hydrophillic beads (Section 8.4). Structures are made compact by setting $objradius = 0.5$. Several strategies were explored for for assigning specific constants to *objbur* and *objexp* (methods in Section 8.5, results in Section 9.4 to Section 9.8).

The energy functions and constraints implemented in Marie are summarized in Table 8.1 while the specifics of each are described in subsequent sections.

8.3 Go-like Potential Energy Function

In Go-potentials, the primary forces governing folding are attractive forces between beads which are in contact in the protein's native structure. Typically, a distance cutoff *cut* is chosen and any pair of unbonded beads that are at or closer than *cut* in the native structure are assigned an attractive force. Standard approaches assign a Lennard-Jones potential to the pair: they achieve an energy minimum at a distance equal to the distance in the native structure with energy increasing sharply as the pair gets too

Table 8.1: Summary of Energy Functions and Constraints Implemented in Marie.

Variant	Description	SS	$objradius$	$objexp_i > 0$?	$objbur > 0$?
Go	Native contacts attract	Free	0	0	0
Compact	All beads equally hydrophobic	Free	0.5	1	0
AvgRSA	Use average RSA of bead types	Free	0.5	$avgrsa_i < \overline{avgrsa}$	$avgrsa_i > \overline{avgrsa}$
RealRSA	Use real RSA of each bead	Free	0.5	$realrsa_i < \overline{realrsa}$	$realrsa_i > \overline{realrsa}$
Compact-SS	All beads equally hydrophobic	Fixed	0.5	1	0
AvgRSA-SS	Use average RSA of bead types	Fixed	0.5	$avgrsa_i < \overline{avgrsa}$	$avgrsa_i > \overline{avgrsa}$
RealRSA-SS	Use real RSA of each bead	Fixed	0.5	$realrsa_i < \overline{realrsa}$	$realrsa_i > \overline{realrsa}$

Each row represents one type of energy function implemented in Marie. The first row is the Go-like potential and remaining rows are hydrophobic collapse energy functions. The SS column indicates whether secondary structure is free to vary or fixed. The $objradius$ column gives the weight in the energy function on minimizing the $radius$ of the burial sphere. The $objexp_i > 0$? column indicates when bead i has a positive term in $objexp$ which causes the solver to try to bury it. The $objbur_i > 0$ is similar and indicates the goal is to expose the bead. See Section 8.5.1, Section 8.5.3, and Section 8.5.4 for details of the energy functions and Section 8.6.3 for how secondary structure is fixed.

close. Energy increases gradually and flatlines at zero as the pair becomes more distant. Lennard-Jones potentials are nonconvex which makes locating global minima difficult when energy functions are used.

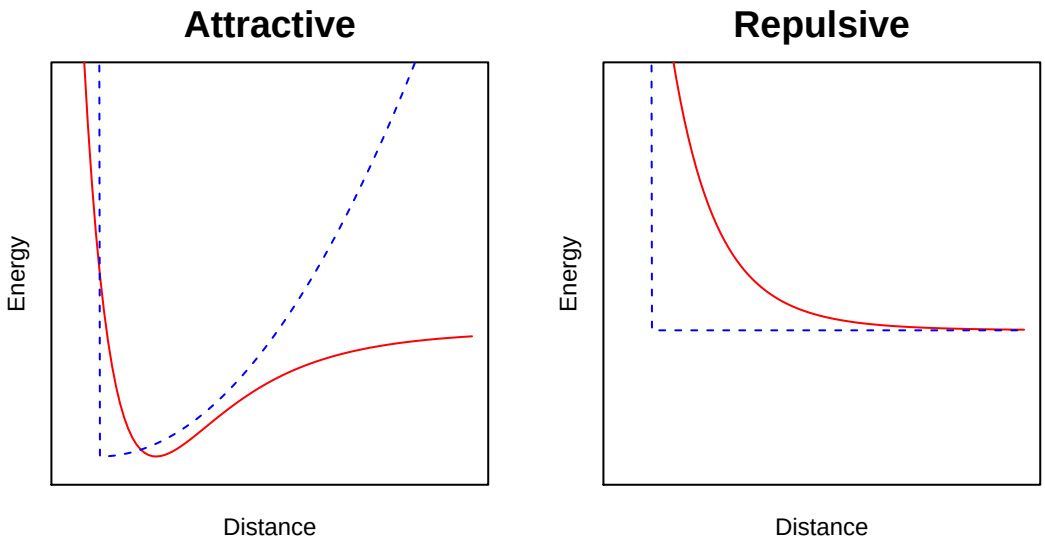
We approximate a Go-like potential in Marie with a simple squared potential with a barrier at a minimum distance. This approximation diverges from the behavior of the standard Lennard-Jones as beads grow more distant but when the pair are proximal, the squared approximation and true Lennard-Jones potential behave similarly. Both of these are illustrated in Figure 8.2.

Importantly, the Marie approximation is a convex function which will facilitate fast location of low-energy conformations. Machine learning algorithms often use a similar approach: the ideal loss function is nonconvex but to preserve the convexity of the problem a convex loss function is chosen as an approximation.

Go-potentials require specification of the distance cutoff *cut* below which native beads are considered in contact. Beads which are not in contact have a general repulsive force assigned to between them which prevents beads from clashing by rising energy costs for closely spaced beads. Our approximation of this in Marie is a simple barrier that prevents clashes. Both the standard and convex approximation are shown in Figure 8.2.

Figure 8.2: Comparison of Standard Go-potential and Marie's Approximation

.



(Left) The solid line represents the shape of a standard Lennard-Jones attractive force used in many a Go-potentials. The dashed line is a squared-approximation of the Lennard-Jones potential used in Marie for its Go-like potential. Only beads below a cutoff distance *cut* in the native structure are subject to attractive forces. **(Right)** A typical repulsive force assigned to bead pairs above the native contact distance threshold *cut*. The solid line is the standard force while Marie's approximation is the dashed line.

To implement the attractive and repulsive forces in Marie, the squared distances between bead pairs is tracked. Each bead is a hard sphere and the squared distance between the surfaces of spheres i and j is tracked in a variable $dslack_{ij}$. Each $dslack_{ij}$ variable has an associated objective weight $objdslack_{ij}$. If beads i and j are in contact in the native structure, Marie minimizes $dslack_{ij}$ by setting $objdslack_{ij}$ to 1 which creates an attraction between the beads during energy minimization. Beads not in contact in the native structure have $objdslack_{ij}$ set to 0 which means no attractive force exists between them. A minimum distance is established based on the van der Waals radii of each bead. All beads are constrained not get closer than this minimum distance as described later (Section 8.6.2).

The central idea for both the Lennard-Jones and the squared approximation used by Marie is the same: the only contributions to energy are the attractive force between natively contacting beads.

8.4 Hydrophobic Collapse Energy Function

While of theoretical importance, Go-potentials are based on contacts derived from native structures and thus are of little utility for de novo structure prediction. Here we establish the ability of Marie to employ an energy function which is of more use for predicting new protein structures.

According to the hydrophobic collapse theory of folding, native structures have a hydrophobic core comprised mainly of residues averse to interactions with solvent. Surrounding this core are hydrophillic residues which have more favorable interactions with solvent. In addition, most proteins are reasonably compact with tightly packed interiors. These three features suggest using a *burial chamber*: hydrophobic residues are to be positioned inside the chamber, hydrophillic residues are to be outside, and the overall volume of the chamber is to be minimized.

The simplest model for the burial chamber is a *burial sphere* which has a position and squared radius, the variable *radius*. We will show in later sections how using the squared radius and squared distances rather than the unsquared radius and distances will

enable the problem to be represented as a semidefinite program which enables efficient optimization. As the native state of proteins is compact, we will minimize $radius$ so that the weight on the model variable $radius$, referred to as $objradius$, will be positive.

Whether a particular bead number i has its center inside or outside the burial sphere can be determined by whether the squared distance d_i^2 between bead i and the burial sphere center is less than squared radius of the burial sphere $radius$:

$$d_i^2 \leq radius. \tag{8.1}$$

However, since bead i is a hard sphere with radius r_i, whether any of the surface of i is exposed or buried is a more complex calculation. We want to allow hydrophobic beads to be outside the burial chamber and hydrophillic inside if it is energetically advantageous to the global structure of the protein. We approximate exposure and burial by the following:

$$exp_i = \max\{d_i^2 - radius + r_i, 0\} \tag{8.2}$$

$$bur_i = \max\{radius - d_i^2 + r_i, 0\} \tag{8.3}$$

$$0 \leq d_i^2, r_i, exp_i, bur_i, radius \tag{8.4}$$

Figure 8.3 gives an intuitive sense of what is represented by Equation 8.2 and Equation 8.3. When the distance to the burial sphere center d_i is equal to the burial radius $radius$, the bead is both exposed and buried by an amount proportional to the bead radius r_i. Smaller distances from the burial chamber center result in more burial and less exposure. Conversely moving away from the burial chamber center results in more exposure.

The variables exp_i and bur_i are inexact approximations of how much of a bead is exposed or buried. Ideally we would use the exposed surface area of each bead. Unfortunately this is notoriously difficult to approximate well even for a fixed structures though some recent approaches may put it in reach for structure optimization procedures [137]. Our approach approximates how much of a bead is inside or outside the burial

chamber. A closer approximation for exposure would be

$$exp_i' = \max\{\min\{d_i - radius, 2r_i\}, 0\}. \tag{8.5}$$

Notice that distance d_i is present instead of d_i^2 and that a minimum is present: a completely exposed bead does not become more exposed by moving farther from the burial chamber. Both the unsquared distance and the minimum in Equation 8.5 are not convex relations so we restrict ourselves to the approximate but efficient forms give in Equation 8.2 and Equation 8.3.

The model beads will have different propensities to be either buried or exposed which are reflected in the objective values associated with *exp*, called *objexp*, and those associated with *bur*, called *objbur*. Since we are treating structure prediction as an energy minimization problem, when bead i is hydrophobic we minimize its exp_i so that $objexp_i > 0$. Conversely, when bead i is hydrophillic, we minimize bur_i so that $objbur_i > 0$.

The over-arching idea of Marie's approximation of hydrophobic collapse is shown in Figure 8.4. The goal to minimize the *radius* is balanced against the values of *exp*, the exposure of beads, and *bur*, the burial of beads. The specific balance is based on the relative magnitudes of $objradius, objexp$, and $objbur$.

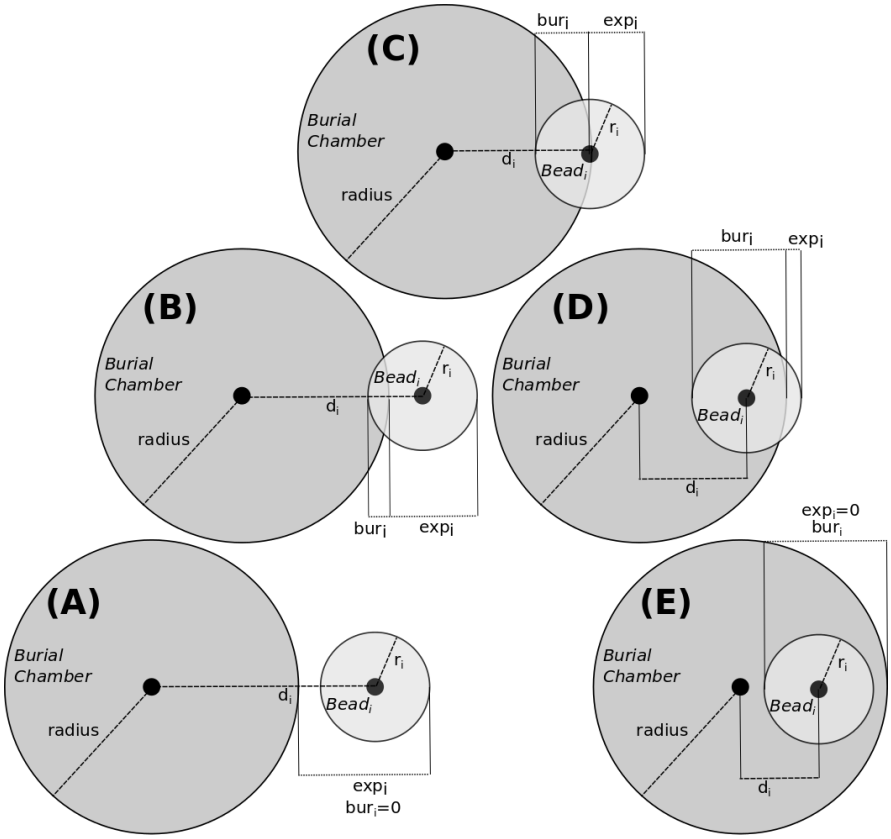
8.5 Variants of Hydrophobic Collapse

Several variants of hydrophobic collapse were examined, all of which vary how the objectives on burial and exposure (*objbur*/*objexp*) are set for specific beads. These are described below.

8.5.1 Compact: Compaction Only

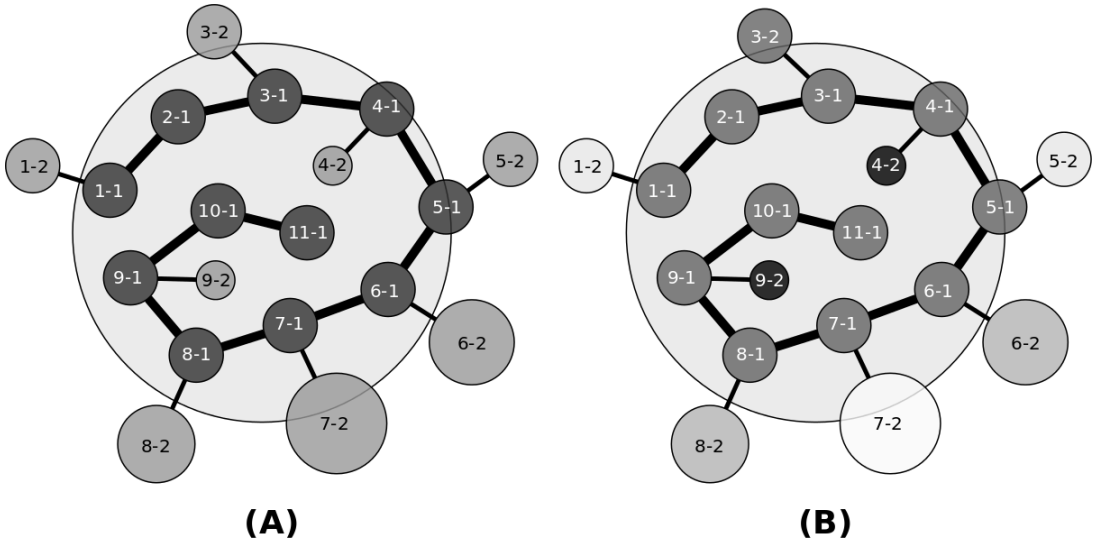
The Compact energy function simply enforces a compact structure by minimizing the radius of the burial chamber and treating all beads as hydrophobic. Each bead i has a variable exp_i associated with its exposure (how far it is placed outside the burial

Figure 8.3: Illustration of how burial and exposure are represented.



The figure is arranged from (A) most exposed to (E) most buried.

Figure 8.4: Hydrophobic Collapse Energy Function and Constraints in 2D.



(A) A conformation in the reduced model. Dark circles are the main chain beads and lighter circles are side chain beads. The burial chamber is shown as a large lightly colored circle. Main chain bonds are a very thick line while main/side chain bonds are a thinner line. Some main chain beads do not have side chain beads (2-1, 10-1, 11-1). **(B)** The same conformation but the coloring of each circle corresponds to the hydrophobicity of a bead. Darker beads are more hydrophobic and want to avoid exposure ($objexp > 0, objbur = 0$) while lighter beads are hydrophillic and want to avoid burial ($objexp = 0, objbur > 0$). The conformation shown has some hydrophobic beads partially exposed (3-2, 5-1) and a hydrophillic bead which is partially buried (7-2). Further optimization may change the size and position of the burial sphere and change the positions of beads. The optimal conformation is the one with a minimum radius for the burial sphere and best partition of hydrophobic beads to the interior and hydrophillic beads to the exterior.

chamber). The objective term $objexp_i$ corresponding with exp_i is set to 1 for each bead so that exposure minimized. The objective term $objbur_i$ associated with bur_i, how deeply bead i is buried, is set to 0.

The Compact variant serves as a baseline to see if there is value in the subsequent energy functions which place beads according to their propensity to be buried or exposed.

8.5.2 Relative Solvent Accessible Surface Area (RSA)

The relative solvent accessible surface area (RSA) of a bead is the fraction of its total surface area which is exposed to solvent. The measure is relative in that it is normalized by the total surface area of a bead so that large beads are on the same scale as small. RSA is a real number from 0.0 to 1.0 with higher values indicating more exposure. The side-chains of some amino acids are much more likely to be exposed or buried based on their specific chemical properties. These tendencies are manifested by differences in the mean RSA of each type of amino acid over known protein structures.

We calculate solvent accessibility for each bead in our data set by a sampling algorithm: 100 evenly spaced points are placed on the surface of each bead. A probe bead with radius 1.4Å is placed in contact with each surface point of the target bead. If the probe clashes with any other bead in the model, that surface point of the target bead is counted as buried; the surface point is considered exposed otherwise. The bead's RSA is the fraction of exposed surface points.

For a particular protein, the real RSA of bead i is referred to as $realrsa_i$. We also calculate the average RSA of each bead type in our dataset. There are 21 bead types. For a particular protein, the notation $avgrsa_i$ refers to average RSA of beads with the same kind as bead i.

8.5.3 AvgRSA: Average RSA determines objective

To account for the average propensity for beads to be exposed or buried, the AvgRSA energy function uses the average RSA, $avgrsa_i$, of each bead i in a protein to compute weights on its exp_i and bur_i variables. We tried several schemes on two small proteins

(data not shown) and settled on the following transformation.

$$\overline{rsa} = \frac{1}{nbead} \sum_{i=1}^{nbead} avgrsa_i \tag{8.6}$$

$$h_i = 2 \times (avgrsa_i - \overline{rsa}) \tag{8.7}$$

$$objbur_i = \max\{h_i, 0\} \tag{8.8}$$

$$objexp_i = \max\{-h_i, 0\} \tag{8.9}$$

This transformation accounts for the varying numbers of hydrophobic and hydrophillic beads in each protein by first centering each bead's hydrophobicity by subtracting off the mean $\overline{rsa}$ of RSAs in that protein in Equation 8.6 and Equation 8.7. The pseudo-hydrophillicity h_i is positive when the bead tends to be exposed and negative when it tends to buried. The penalty on burying a bead in Equation 8.8 is positive when the the bead tends to have higher than average RSA for that protein. Similarly, Equation 8.8 sets up a penalty for exposing beads which have a lower than average RSA for the protein. In each case, one of either $objexp_i$ or $objbur_i$ will be zero and the other will be positive.

The AvgRSA energy function uses the *objexp* and *objbur* from the above scheme along with a weight of $objradius = 0.5$. AvgRSA is the closest to what could be used in a true protein structure prediction scheme as it does not assume any special knowledge of the true protein structure properties.

8.5.4 RealRSA: Real RSA determines objective

The RealRSA energy function uses the true RSA of each bead to determine burial and exposure objectives. We use the following transformation of $realrsa_i$ to calculate *objexp* and *objbur*. It is identical to Equation 8.6-8.13 except that $avgrsa_i$ is replaced

with $realrsa_i$.

$$\overline{rsa} = \frac{1}{nbead} \sum_{i=1}^{nbead} realrsa_i \tag{8.10}$$

$$h_i = 2 \times (realrsa_i - \overline{rsa}) \tag{8.11}$$

$$objbur_i = \max\{h_i, 0\} \tag{8.12}$$

$$objexp_i = \max\{-h_i, 0\} \tag{8.13}$$

The RealRSA energy function is not a practical structure prediction scheme as it utilizes the true RSA of each bead which is not known without first known the structure. In principle, methods exist to predict RSA from sequence alone and may be useful here. Intuitively, we expect that RealRSA would produce better predictions due to its use of privileged information about the true structure.

8.6 Model Constraints

In physics simulations, atoms are subject to bonded and unbonded interactions. When bonded, the potential energy contribution of the two atoms is minimized when they are a prescribed distance apart. The bond is usually treated like a spring so that energy rises sharply when the bond length is either stretched or compressed. Unbonded interactions vary but the most common is the Lennard-Jones interaction in which potential energy contributions go to zero as the distance between atoms increases to infinity. There is an energy minimum when the two atoms are a prescribed distance apart and energy again rises sharply when the two atoms come too close together.

We take a much simplified approach in Marie: bonded and unbonded interactions are represented as fixed constraints rather than part of the energy function. Distance geometry allows us to enforce exact distances between beads or specify inequalities on the distances. This capability is used in several ways described in subsequent sections.

8.6.1 Bonded Beads

Bonded beads are kept at a constant distance from one another. This is done by fixing their squared distance d_{ij}^2 to one another. For beads i and j which are bonded, we determine the mean distance between bonded beads of types i and j (see Section 8.1; let it be called $\bar{d}_{t(ij)}$. During structure optimization, we include the constraint

$$d_{ij}^2 = \bar{d}_{t(ij)}^2, \text{ for all } i,j \text{ that are bonded} \tag{8.14}$$

so that the bond never stretches or compresses. This is done for the bonds between adjacent main chain beads and for bonded main/side chain beads. Note that this equality is a hard constraint: it is not allowed to be violated during optimization.

8.6.2 Unbonded Beads

In our Go-like potential, favorable native contacts between unbonded beads are modeled with a squared potential. For all other unbonded beads we do not attempt to model favorable unbonded energy contributions. That includes all beads when using the hydrophobic collapse energy function. However, in order to avoid clashes between the hard-sphere beads, we enforce minimum distance constraints between unbonded beads. The mean radius of each bead type is determined from the dataset (Section 8.1). Call these $\bar{r}_{t(i)}$ for bead i and $\bar{r}_{t(j)}$ for bead j. The distance between beads i and j is required to be large enough so that their radii do not overlap. This is done again with their squared distance and a nonnegative slack variable $dslack_{ij}$ associated with the pair. The constraint is

$$d_{ij}^2 - dslack_{ij} = (\bar{r}_{t(i)} + \bar{r}_{t(j)})^2, \text{ for all } i,j \text{ that are unbonded} \tag{8.15}$$

$$dslack_{ij} \geq 0, \text{ for all } i,j \tag{8.16}$$

Since $dslack$ is nonnegative, d_{ij}^2 will always be larger than the sum of the two bead radii squared so that the beads will not overlap.

8.6.3 Fixed Secondary Structure

Several locally repeated structural patterns occur in proteins, primarily alpha helices and beta sheets. Knowledge of which parts of the protein sequence adopt these shapes is a huge advantage in full structure prediction. To that end, we use DSSP [26] to calculate secondary structure assignments in the true structure and in some method variants used these to constrain the shape of the protein.

We can enforce known secondary structure in the protein the same way bonds are enforced, through maintaining exact distances. This is done between main chain beads within a sequence window of a particular bead. In our case we use a window of size 11 so that main chain bead i had its distance to main chain beads $i-5, \cdots, i-2, i-1, i+1, i+2, \cdots, i+5$ fixed at the true distance. Very long secondary structure elements are thus locally rigid but have some ability to bend over their entire length. An alternative to fixing secondary structure is to penalize the deviation of bead distances from specified ranges which represent secondary structure. This could be handled equally as well in the distance geometry framework.

Marie variants Compact, AvgRSA, and RealRSA do not use any explicit knowledge of secondary structure while variants Compact-SS, AvgRSA-SS, and RealRSA-SS do use true secondary structure to constrain the protein conformations. The true secondary structure of the protein is privileged information so that the X-SS variants could not be used for true structure prediction. Instead, one of a variety of secondary structure prediction methods would be required. We leave that avenue to be explored in future work as our primary interest is simply how secondary structure affects the quality structures generated in the simple model.

8.6.4 Minimum Sum of All Squared Distances

Both the Go-like and hydrophobic collapse energy functions have the quality of trying to achieve compactness. Any model involving forces that compact a protein runs the risk of forcing it to an unrealistically small conformation. This has two effects. Native structures are not necessarily as packed as they might be so the energy function may bias

conformations found by Marie away from native structures. It also seems to decrease optimization time in our experiments with Go-like potentials in Section 9.2 (see the description there for why).

The constraint is simply implemented by summing all squared distances between all beads and using a slack variable to ensure they are above some constant value. During the Go-potential experiments, we utilize a *DistLevel* for this constant amount. It is the fraction of the total sum of squared distances in the native protein taken for the minimum. For example, a protein with a sum of squared distances of 1.0×10^7 might be modeled in Marie using $DistLevel = 0.5$ which means the sum of square distances is constrained to be above 0.5×10^7 while a $Distlevel = 1.0$ would use a minimum distance of 1.0×10^7, the same as the native. *DistLevel* constraints incorporate knowledge of the native structure for a protein making them unsuitable for ab initio prediction, but it is not difficult to estimate a good lower bound on the sum of squared distances for a protein of a given size and amino acid composition which could be used in ab initio prediction.

8.6.5 Fixing All Distances: Native Structures

Arbitrary pairs of beads can be held at fixed distances in addition to bonded beads and secondary structure. Taken to the extreme, the whole protein structure can specified by fixing all pair-wise distances. This is useful in the case that we want to calculate the lowest energy burial sphere for a specific structure such as the known native conformation. We do this in order to establish the energy (objective value) of the native structure according to different energy functions (Compact, AvgRSA, RealRSA). The energy of native structures can then be compared to the energy of conformations found using Marie.

8.7 Distance Geometry

The methods described above model proteins and their energy using squared distances, between bead centers and between the burial chamber center and bead centers. The implicit representation of the protein and burial chamber is thus a squared distance matrix. In this section we show how a squared distance matrix is related to a certain

positive semidefinite matrix called a Gram matrix which is also related to the coordinates of the points represented in the distance matrix. This section bridges the gap between the model of proteins used by Marie and the mechanism to find minimum energy conformations which will be semidefinite programming.

A set of n beads in d dimensions is a matrix $R \in \mathbb{R}^{d \times n}$. In all cases we assume that the beads are centered so that the sum of each row of R is zero. We are most concerned with $d = 3$ as beads in three-dimensional space are sought. The squared distance matrix for R is referred to as D with $D(i, j)$ equal to the squared distance between the ith and jth bead.

$$D(i, j) = ||R(:, i) - R(:, j)||_2^2 \tag{8.17}$$

Unless otherwise noted, we will refer to a squared distance matrix simply as a distance matrix. Distance matrices are square and symmetric with a zero diagonal.

For a collection of beads $R \in \mathbb{R}^{d \times n}$, the associated *Gram matrix* is defined $G = R^T R$. Gram matrices are square and symmetric. Assuming the beads R are centered, they have a have a unique distance matrix D and corresponding unique Gram matrix G. By definition, Gram matrices are positive semidefinite (PSD): they have d positive eigenvalues and the remainder of the eigenvalues are zero. The PSD property is denoted $G \succeq 0$.

Given a Gram matrix, corresponding bead coordinates may be recovered from it via an eigenvalue decomposition. Let $G = Q \operatorname{diag}(\lambda) Q^T$ where Q is the matrix of eigenvectors and λ the vector of associated eigenvalues, sorted in decreasing order. Then

$$R = \operatorname{diag}(\sqrt{\lambda}) Q^T. \tag{8.18}$$

If G has d nonzero eigenvalues, then its beads exist in d-dimensional space. When three-dimensional coordinates are desired and there are more than three nonzero eigenvalues, the remaining eigenvalues are set to zero.

Distance and Gram matrices are related by the entry-wise equation

$$D(i, j) = G(i, i) + G(j, j) - G(i, j) - G(j, i). \tag{8.19}$$

This comes from the definition of squared distance matrices and Gram matrices.

$$\begin{aligned} D(i,j) &= ||(R(:,i) - R(:,j))||^2 && (8.20)\\ &= (R(:,i) - R(:,j))^T (R(:,i) - R(:,j)) && (8.21)\\ &= R(:,i)^T R(:,i) + R(:,j)^T R(:,j) - 2R(:,i)^T R(:,j) && (8.22)\\ &= G(i,i) + G(j,j) - 2G(i,j) && (8.23) \end{aligned}$$

A distance matrix of size n may be converted into a Gram matrix. Let $V = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ (the geometric centering matrix in Section 5.4.2.2 of [138]); then

$$G = -\frac{1}{2} V D V. \tag{8.24}$$

Note that if D is not a valid distance matrix, the resulting Gram matrix will not be positive semidefinite (i.e. it may have some negative eigenvalues). If the coordinates R are centered, the row and column sums of the Gram matrix G will also be zero which allows us to center coordinates using constraints on G.

8.8 Semidefinite Programming (SDP)

We established in Section 8.7 that distance matrices correspond to Gram matrices and Gram matrices are positive semidefinite. Any constraints on a squared distance matrices can thus be represented as linear constraints on its corresponding Gram matrix. This allows us to express structure prediction problems in the Marie model as semidefinite programs (SDPs). We begin with a brief overview of semidefinite programming.

Mathematical programming deals with problems in which an objective function is to be minimized according to constraints on the optimization variables. In an SDP, the optimization variable is required to be a positive semidefinite matrix and the constraints and objective are linear in this variable. The standard form of SDPs is typically written:

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \succeq 0. \end{aligned} \tag{8.25}$$

In this program, the problem data are a vectorized symmetric matrix $c \in \mathbb{R}^{n^2}$, referred to as the objective, and matrix $A \in \mathbb{R}^{m \times n^2}$ with vector $b \in \mathbb{R}^m$ which are the constraints. The optimization variable $x \in \mathbb{R}^{n^2}$ is required to be a vectorized symmetric PSD matrix, indicated by $x \succeq 0$. The equation $Ax - b = 0$ constitutes a set of linear constraints on x.

A key feature of SDPs is that they are convex and thus polynomial time algorithms exist to locate a globally optimal solution or indicate that the problem is infeasible or unbounded. [139] offers a good introduction to convex programming, a wide variety of applications of SDP are discussed in [140], and [138] discusses relation use of SDPs to distance geometry extensively.

Program 8.25 can be modified to accommodate problems involving inequalities. Nonnegative slack variables are introduced for any inequality constraints so that the solver deals only with equality constraints. The new form of the program is as follows.

$$\begin{aligned} \min_{x_l, x_s} \quad & c_l^T x_l + c_s^T x_s \\ \text{s.t.} \quad & A_l x_l + A_s x_s = b \\ & x_l \geq 0,\ x_s \succeq 0. \end{aligned} \tag{8.26}$$

Every variable in this program is either a vectorized positive semidefinite matrix which is part of x_s or a nonnegative linear variable which is part of x_l. The result of solving Program 8.26 is the optimal point (x_l^*, x_s^*). The vectorized PSD variable x_s^* may be converted into a PSD matrix simply by reshaping it into a square matrix.

8.9 Marie as an SDP

In Marie, the linearized Gram matrix G is the entirety of x_s in Program 8.26. It represents the positions of each bead and the center of the burial sphere and allows squared distances between them to be expressed as linear constraints on x_s. The remainder of the model comprises nonnegative linear variables that are in x_l. These include the *dslack* variables which prevent bead clashes, the burial sphere *radius*, and the exposure exp_i and burial bur_i variables for each bead. Additional nonnegative slack variables are also part of x_l.

The constraint matrices A_s and A_l encode the constraints described in the previous sections by multiplying combinations of Gram matrix entries and linear variables. This is how equality distance constraints, inequality distance constraints, and position relative to the burial chamber are enforced. As a concrete example, suppose constraint 55 is that beads 12 and 34 are unbonded and must not clash.

$$A_s(55,:)x_s + A_l(55,:) = b(55) \tag{8.27}$$

$$G(12,12) + G(34,34) - 2G(12,34) - dslack(12,34) = (r_{t(12)} + r_{t(34)})^2 \tag{8.28}$$

$$D(12,34) - dslack(12,34) = (r_{t(12)} + r_{t(34)})^2 \tag{8.29}$$

$$d_{12,34}^2 \geq (r_{t(12)} + r_{t(34)})^2 \tag{8.30}$$

The last relation is due to the nonnegativity of each *dslack* variable. The constraints on exposure and burial can be similarly expressed as linear constraints on G, exp, bur, $radius$, and some additional slack variables.

The objective weights c_l and c_s of Program 8.26 contain terms associated with each variable in x_l and x_s. They are mostly zero except for the following: (1) in Go-like potentials *objdslack* is set to 1 for pairs of beads in contact in the native structure, (2) for hydrophobic collapse *objradius* which is associated with the burial chamber $radius$, (3) and for hydrophobic collapse *objexp*/*objbur* which are associated with the bur/exp of each bead are set to nonnegative values according to several schemes. See Section 8.3 and Section 8.4 for details of how *objdslack*, *objradius*, *objexp*, and *objbur* are specifically calculated for each energy function.

8.10 Rank Reduction via Convex Iteration

At this point, we have established that the Marie model is a semidefinite program and can be optimized by solving Program 8.26. Typically the solution x_s^* returned by SDP solvers has a high rank, i.e. many nonzero eigenvalues. This means that the beads represented in x_s^* exist in a high-dimensional space. Unfortunately constraining rank in an SDP destroys its convexity resulting in an NP-hard problem. The optimization

community is greatly interested in heuristics that reduce the rank of SDP problems. A number of these heuristics are explored in [138] and we have found that the convex iteration technique there to be particularly useful for our distance geometry problems.

In convex iteration, one first solves the usual SDP and then modifies the original objective by a search direction designed to favor a low-rank solution. Repeating this process converges to some solution that is typically lower rank than the original. Though there are no guarantees that a desired rank will be achieved, we have found convex iteration works well to produce rank-three solutions for our problems.

In [138], the search direction is found by solving another SDP. In this work, we use the spectral decomposition to determine the search direction. This is done to avoid the computational cost of solving another SDP and, in preliminary tests, did not seem to affect the overall accuracy of the coordinates which were ultimately produced.

Let c_s be the original SDP objective and $x_s^{(1)}$ be the solution found the first step during convex iteration with $x_s^{(1)} = Q\,\mathrm{diag}(\lambda)Q^T$. Assume that the eigenvalues are arranged in descending order. We are seeking three-dimensional solutions so we determine the first search direction by allowing the leading eigenvalues to grow and attempting to minimize the remaining eigenvalues. Thus the search direction w is determined by

$$\lambda' = (0, 0, 0, 1 + \lambda_4, \cdots, 1 + \lambda_n), \tag{8.31}$$

$$w = Q\,\mathrm{diag}(\lambda')Q^T. \tag{8.32}$$

For step $i > 1$ of convex iteration with solution $x^{(i)} = Q\,\mathrm{diag}(\lambda)Q^T$ we use a slightly different search direction.

$$\lambda'' = (0, 0, 0, 1, \cdots, 1), \tag{8.33}$$

$$w = Q\,\mathrm{diag}(\lambda'')Q^T. \tag{8.34}$$

Empirical performance in initial testing was the only guide for these decisions.

In the second and subsequent rounds of convex iteration the full objective is a weighted sum of the original objective and the search direction, e.g. $c_s^{(i)} = (1 - \alpha)c_s/||c_S||_F + \alpha w/||w||_F$. The weight α is increased over interactions from 0 to 1 to put more emphasis on a low-dimensional solution.

In practice, we have found that nearly all structures we tested converged to low-rank solutions after at most 25 iterations of the above process. The final solution is not guaranteed to be the global minimum but the hope is that starting with a global answer to the dimensionality-relaxed problem improves the quality of the final solution.

Chapter 9

Experiments with Marie

In this chapter show experimental results of using the Marie model. We start with the experimental setup utilized for both experiments. We then show results for two experiments which employ Marie to study what constitutes a native contact in Go-potential models. We study the *cut* parameter that defines native contacts in detail for four small proteins then expand to a larger set of 108 proteins and test fewer values for *cut*. Finally, we turn to the hydrophobic collapse energy function which is also handled in Marie. We study the variants described in Section 8.5 to determine whether they produce native-like structures.

9.1 Experimental Setup

9.1.1 Optimization Software and Hardware

The backbone of our methods is a SDP solver for which we employ CSDP [141]. Matlab^{TM} is used to implement convex iteration approach and perform other high-level computations, graphics, and data analysis [142]. Computations are run on 6-core Intel Xeon 2.9 GHz processors of the Koronis system and on 8-core Intel Xeon 2.8 GHz processors of the Itasca system, both maintained by the Minnesota Supercomputer Institute.

9.1.2 Data Sets

We conduct experiments using 108 proteins also employed to benchmark the structure prediction method Rosetta[39]. This allows us to compare results to one of the leading methods in protein structure prediction. We choose from the medium and large proteins described in Table 1 of [39]. They range in size from 54 to 99 residues with an average of 70 residues. When converted to beads in our reduced model, the proteins ranged from 93 to 174 beads averaging 132 beads.

9.1.3 Evaluation Metrics

A standard measure of quality for a model protein structure is to compute its root mean squared deviation (RMSD) to the true protein. After optimally superimposing the coordinates of the two structures, RMSD measures how far the model deviates from the truth. All of the predictions our methods produce may be reflected coordinates as we are dealing solely with distance-based predictions. For that reason, we measure the RMSD of both a given prediction and its reflection to the true structure and report the better RMSD. Under mild conditions, such as the presence of an α-helix in the predicted structure, it is possible to determine which orientation of the predicted protein is appropriate but we leave automatic structure determination to future work.

9.1.4 Analysis

We collect the RMSDs of each prediction and use Welch's t-test on the measurements to determine the overall quality of each method. This test compares the average performance of one method versus another. In each test, we report the p-value which is typically considered significant if $p \leq 0.05$. Welch's t-test is preferred as it does not assume the two methods being evaluated have equal variance in their predictions. We used the used `t.test` function from the R `stats` package [81]. Plots of the results were produced using the using R package `ggplot2` [143].

9.2 Native Contact Cutoff Distance in Go-like Models for Four Small Proteins

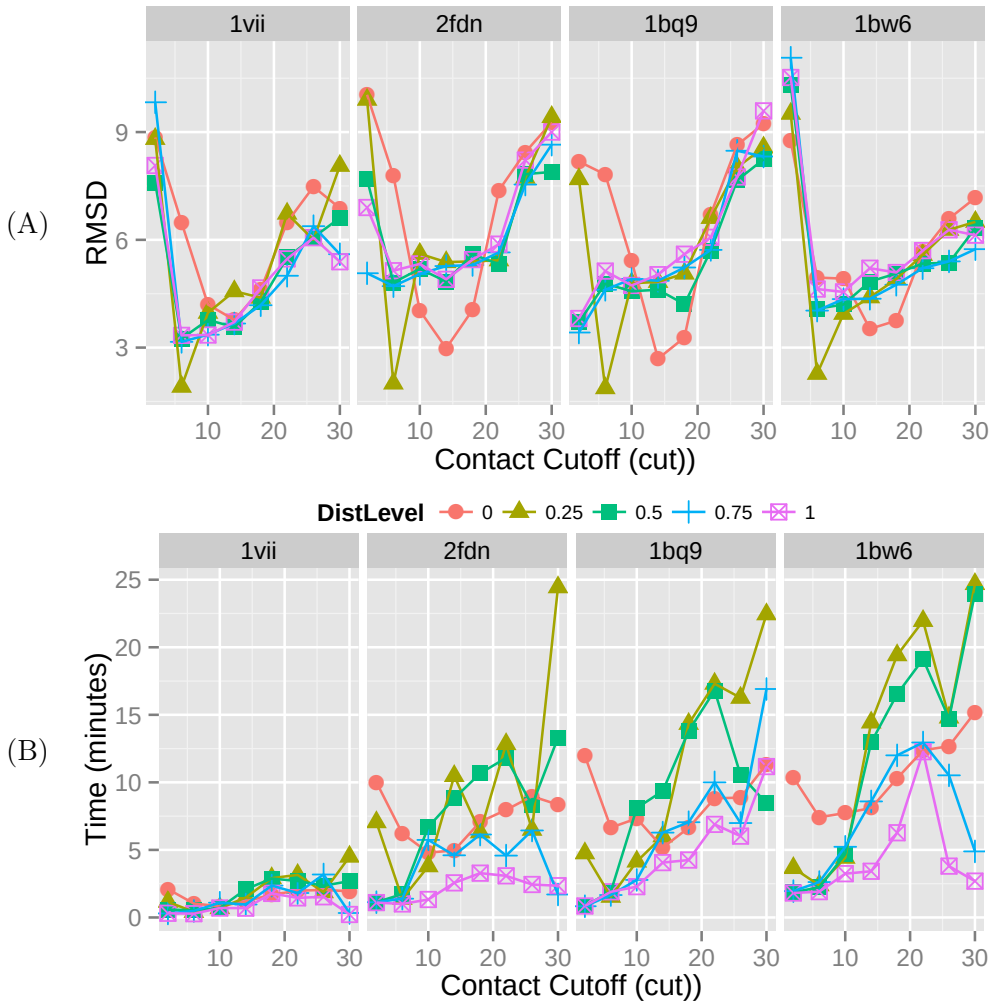
As described in Section 8.3, the only forces in Go-models are attractive forces between beads which are below a cutoff distance *cut* in the native conformation of the protein. In this set of experiments we used Marie to explore appropriate choices for *cut*. Normally with standard molecular dynamics or lattice-based simulations, exploring the *cut* parameter would be a computationally costly endeavor. The approximations used in Marie allow us to quickly find low-energy conformations for any value of *cut* enabling a thorough exploration of the parameters. Appropriate choices for *cut* should produce conformations that have low RMSD to the native structure. Higher RMSD indicates that either too few or too many beads are attracted to another so and that the native conformation does not minimize the energy function induced that choice of *cut*.

Figure 9.1 shows examples of varying *cut* for four small proteins. The *cut* value was varied for each protein in increments of 4Å starting at 2Å and ending at 30Å. Native beads below *cut* distance were subjected to an attractive force in the the Marie model for Go-like potentials. Additionally, the sum of all squared distances between bead pairs was constrained to a minimum value. This tests whether limiting the compactness of predictions affects the quality of the prediction. The constraints were at fractions 0.0, 0.25, 0.50, 0.75, and 1.00 of the native conformation's sum of squared pairwise distances and it is referred to as $DistLevel$.

For three of the four proteins in Figure 9.1, a *cut* value of 2Å does not lead to native-like conformations. Using values of *cut* starting at 6Å gives native-like structures under most conditions. Increasing *cut* to larger values gradually degrades the quality of the predicted model. Several exceptions to this trend are worth mentioning.

For $DistLevel$ values of 0.0 and 0.25, *cut* gives different prediction qualities. The minimum for $DistLevel = 0.0$ was minimal at 14Å on all four proteins. For $DistLevel = 0.25$ the best RMSDS overall for all proteins was at *cut* =6Å with sharp jumps up for smaller or larger *cut* values. As this is a distinct trend and appears somewhat less regular, we chose in the second, larger experiment with Go-like potentials to use a $DistLevel = 0.75$ but a finer-grained exploration of $DistLevel$ is currently being pursued.

Figure 9.1: Varying Cutoff Distance in Go-like Potential on Four Small Proteins.



Each panel represents results on one of 4 small proteins. The X-axis varies *cut*, the distance which defines native contacts. Color and shape indicate constraints on the minimum compactness in the estimated conformations (See Section 8.3 and the accompanying text for details). (A) The Y-axis indicates the RMSD of the low-energy conformation found by Marie versus the native conformation. (B) The Y-axis shows the compute time in minutes for Marie to locate the low-energy conformation. Lower RMSD and time is better.

Protein 1bqn exhibited somewhat different behavior from the other three tested as it had low RMSD predictions even for $cut = 2$Å. The secondary structure classes may matter here: 1bqn is composed primarily of beta sheets while the other three primarily feature alpha helices (1vii and 1bw6) or a mixture of both secondary structure types (2fdn).

Figure 9.1 also gives timing information for the simulations. The general trend is that time increases as the *cut* goes up. There is also a trend that higher $DistLevel$ leads to shorter computation time. These two factors are related. Recall that Marie locates low-energy conformations by solving a series of semidefinite programs according to the convex iteration scheme described in Section 8.10. This is necessary as the solution to a given SDP may represent a high-dimensional structure and the iteration gradually forces solutions to resolve to 3D space. When many beads are attracted to one another as is the case for high *cut* values, the initial SDP solutions tend to exit in very high dimensions as more beads can be in contact without violating the other constraints of the system. This requires additional iterations to resolve to 3D structures increasing the computation time required. Countering this are the effects of $DistLevel$ which constrains the protein from becoming too compact thus precluding too many beads from crowding together. Higher $DistLevel$ biases conformational search towards lower-dimensional solutions and reduces the number of convex iterations.

The closest comparison for computation times we have located from literature are from [116] in which the 46-residue alpha/beta protein crambin was studied using a Go-like potential. A major pursuit of the paper was to study the folding dynamics of crambin for which a Monte Carlo simulation was employed. Marie trades the ability to observer dynamics of the system for speed. Our results above on proteins that are 36-residues (1vii) and 53-56 residues (1b26, 2fdn, 1bq9) in Figure 9.1 range from between 1 and 25 minutes. Shimada et al. report that high temperature simulation took on average 16 hours on a 550 Mhz computer and adjusting for processor speed increase (2.8 Ghz in our case) would likely run on the order of 3 hours. Low temperature simulations took 40 hours then and would likely take 7-8 hours today. A caveat to this is that the simulations of Shimada et al. involved all heavy atoms while we use a coarse-grained model here, but for a parameter study of *cut* the time-savings is apparent.

Table 9.1: Correlations of Size with RMSD and Run Time for Go-like Potential.

	cut=5	cut=10	cut=20	cut=30	All
RMSD	0.166	0.265	0.254	-0.006	0.126
Time	0.885	0.707	0.729	0.659	0.624

The correlations correspond to the plotted points shown in Figure 9.2 on 108 proteins. The correlation with The differing values of *cut* give different correlation levels while the column All averages over all *cut* values. Time is with the logarithm of the time in minutes as this correlation was stronger than the untransformed time.

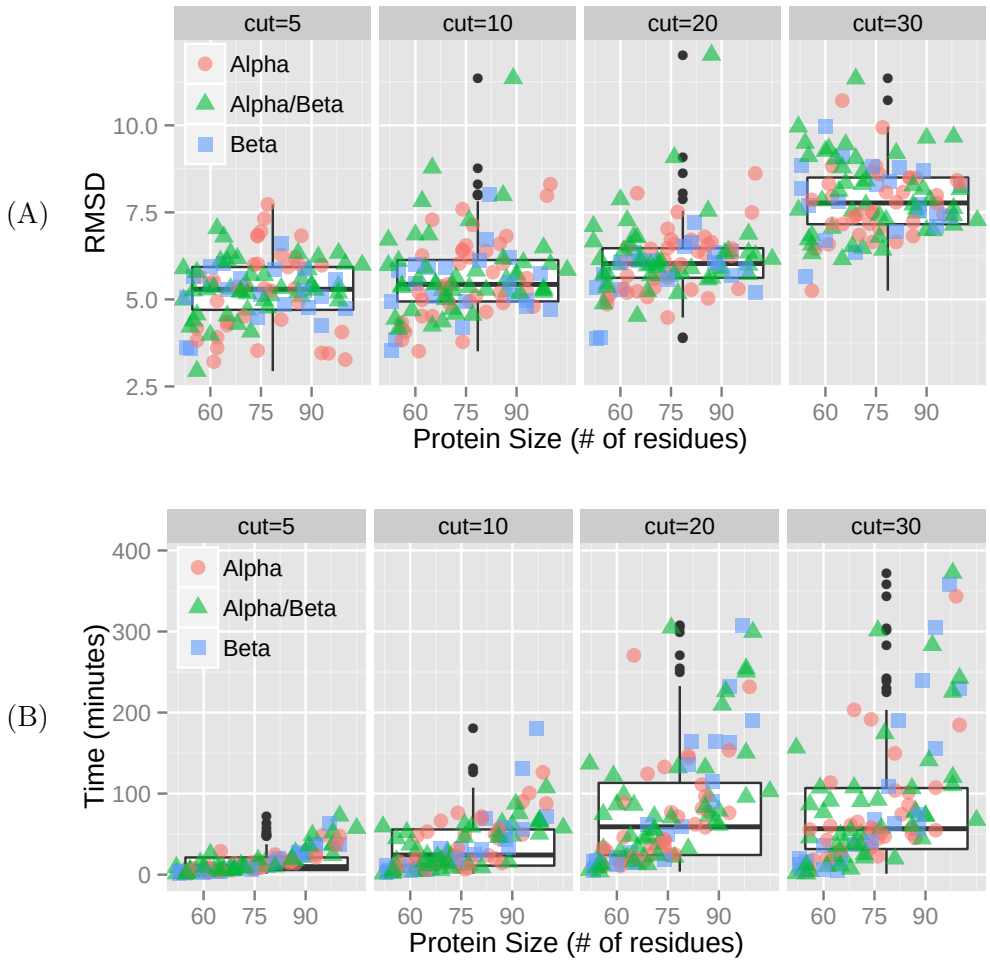
9.3 Large-Scale Evaluation of Native Cutoff Distance

We expanded our evaluation of the influence of *cut* to a larger set of 108 proteins. For these tests we fixed $DistLevel = 0.75$ and used *cut* values of 5, 10, 20, and 30 Å. Figure 9.2 shows both the RMSD of the conformations found by Marie and the time taken to find the conformation. The mean RMSD is approximately equal for $cut = 5$ and $cut = 10$ but there is strong statistical evidence that the former gives better RMSD results: Welch's t-test of the mean RMSD for $cut = 5$ being less than for $cut = 10$ has a p-value of 0.001. Section 9.2 indicated a tendency for larger values of *cut* to degrade the RMSD of the predicted conformation and that trend continues here with larger values of *cut* giving poorer RMSDs.

In [118], the authors experimented with *cut* values of 5.5Å and 6.5Å on 18 small proteins in a molecular dynamics simulation with Go-potentials. They found no qualitative difference between the structures produced using for either value. Our results here are in agreement and expand them to encompass a much wider range of values for *cut*.

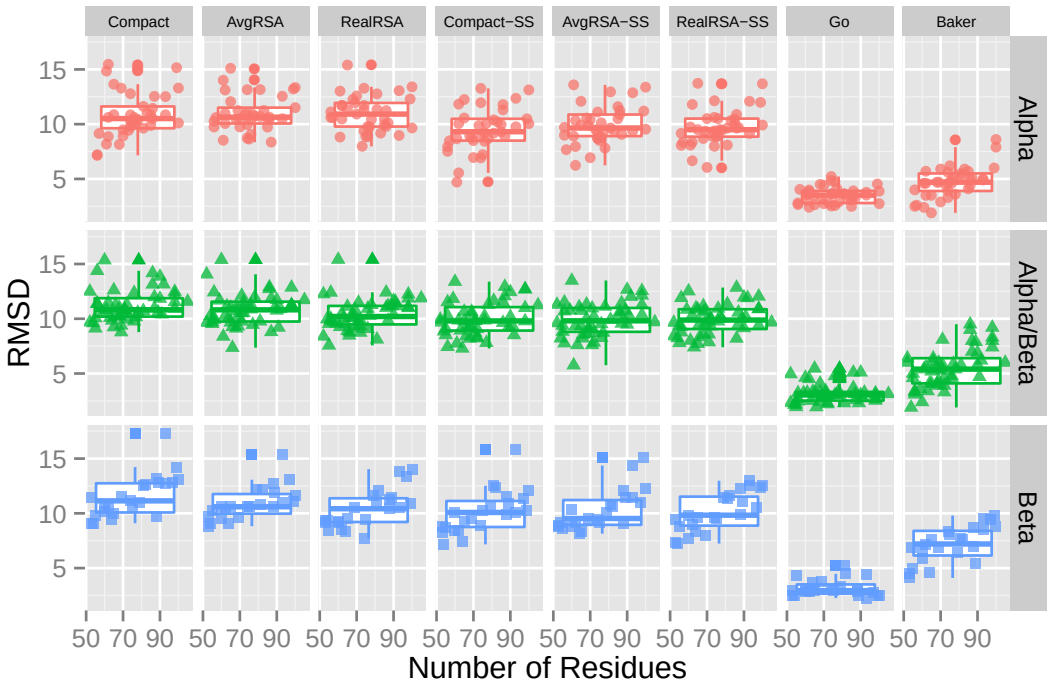
The time taken by Marie to produce structures is clearly dependent on the size of the protein according to Figure 9.2 and Table 9.1 while the two extreme values of $cut = 5$ and $cut = 30$ exhibit virtually no correlation with RMSD and the intermediate values of *cut* exhibit weak correlation.

Figure 9.2: Large-scale Evaluation of Go-like potential



Each panel represents results for a different value of *cut* (native contact distance) on a set 108 proteins. The aggregated results as a box and whisker plot are overlaid with the individual results for each protein. For both (A) and (B) the X-axis shows the number of residues in the protein. The Y-axis in (A) shows the RMSD of the conformation located by Marie and in (B) the computation time to find that conformation in log scale. Color and shape indicate the secondary structure of the proteins. In all cases $DistLevel = 0.75$ Lower RMSD and time is better.

Figure 9.3: Marie Prediction Quality Measured by RMSD.



Within each panel, X-position indicates the size of the protein and Y-position indicates the RMSD of the predicted structure (lower is better). Performance is given by the box and whisker plot overlaid with the points that they summarize. Panel rows represent different secondary structure classes. Panel columns represent different method variants: (Compact) minimizing only the overall compactness of the protein, (AvgRSA) minimize compactness while trying to properly place beads on the interior/exterior based on their average RSA over the dataset, (RealRSA) identical except use the actual RSA of the native structure, (Go) Go-like potential with $cut = 5$ and no lower bound on compactness. The X-SS columns represent variants in which secondary structure is partially fixed to that found in the native protein structure. The final column contains the results for Rosetta reported in [39].

Table 9.2: Marie Prediction Quality and Comparisons.

Method	Mean	SD	Compact	AvgRSA	RealRSA	Compact-SS	AvgRSA-SS	RealRSA-SS	Go	Baker
Compact	11.12	1.771		0.964	1.000	1.000	1.000	1.000	1.000	1.000
AvgRSA	10.87	1.501	0.036		0.991	1.000	1.000	1.000	1.000	1.000
RealRSA	10.59	1.563	0.000	0.009		1.000	1.000	1.000	1.000	1.000
Compact-SS	9.85	1.773	0.000	0.000	0.000		0.183	0.366	1.000	1.000
AvgRSA-SS	9.99	1.655	0.000	0.000	0.000	0.817		0.791	1.000	1.000
RealRSA-SS	9.89	1.549	0.000	0.000	0.000	0.634	0.209		1.000	1.000
Go	3.29	0.815	0.000	0.000	0.000	0.000	0.000	0.000		0.000
Baker	5.58	1.861	0.000	0.000	0.000	0.000	0.000	0.000	1.000	

The left section gives the mean (Mean) and standard deviation (SD) of the RMSD over all 108 proteins in the dataset. The right section gives p-values from a paired Welch's T-test of whether the row method has a lower mean than the column method. The first six rows are hydrophobic collapse energy functions, the Go row is the Go-like potential, and Baker are the results of the Rosetta fragment assembly program from [39].

9.4 Comparison of Hydrophobic Collapse Energy Functions

Table 9.2 and Figure 9.3 compare our hydrophobic collapse energy functions with our Go-like potential and a result from a fragment assembly program. The first three rows of Table 9.2 show the performance of compaction (Compact), using the average RSA of each bead type to determine its objective (AvgRSA), and using the actual RSA to determine the objective (RealRSA). Performance is measured in terms of the RMSD of the predicted conformation to the known native conformation. The following rows show variants in which secondary structure is fixed. The final two rows show the performance of the Go-like potential (Section 8.3) of the Rosetta method on the benchmark reported by the Baker lab [39]. Incremental improvements are made going from Compact to AvgRSA to RealRSA. Statistically, AvgRSA is not better than Compact (p-value 0.2670 in 4th column) indicating there is little benefit to RMSD quality from trying to use the average burial/exposure to guide their placement during optimization.

Moving down the first column, the X-SS methods are variants in which the true secondary structure of the protein was fixed. This led to significant improvement over in predictions in terms of RMSD to native. When secondary structure is fixed, the minor difference between Compact, AvgRSA, and RealRSA disappear. This is interesting as it indicates as before that there is little improvement gained even from using the true RSA of each bead.

Only a limited amount of effort has been devoted to tuning the weight of compaction versus hydrophobic collapse and it is likely that further attention is required here. We have subsequently inspected the effects of weighting the size of the burial chamber differently and it seems this parameter needs to be chosen carefully to avoid completely dominating the separation of hydrophobic and hydrophillic beads across the boundary. If it is too large, burial and exposure are ignored and all three variants will behave as Compact does.

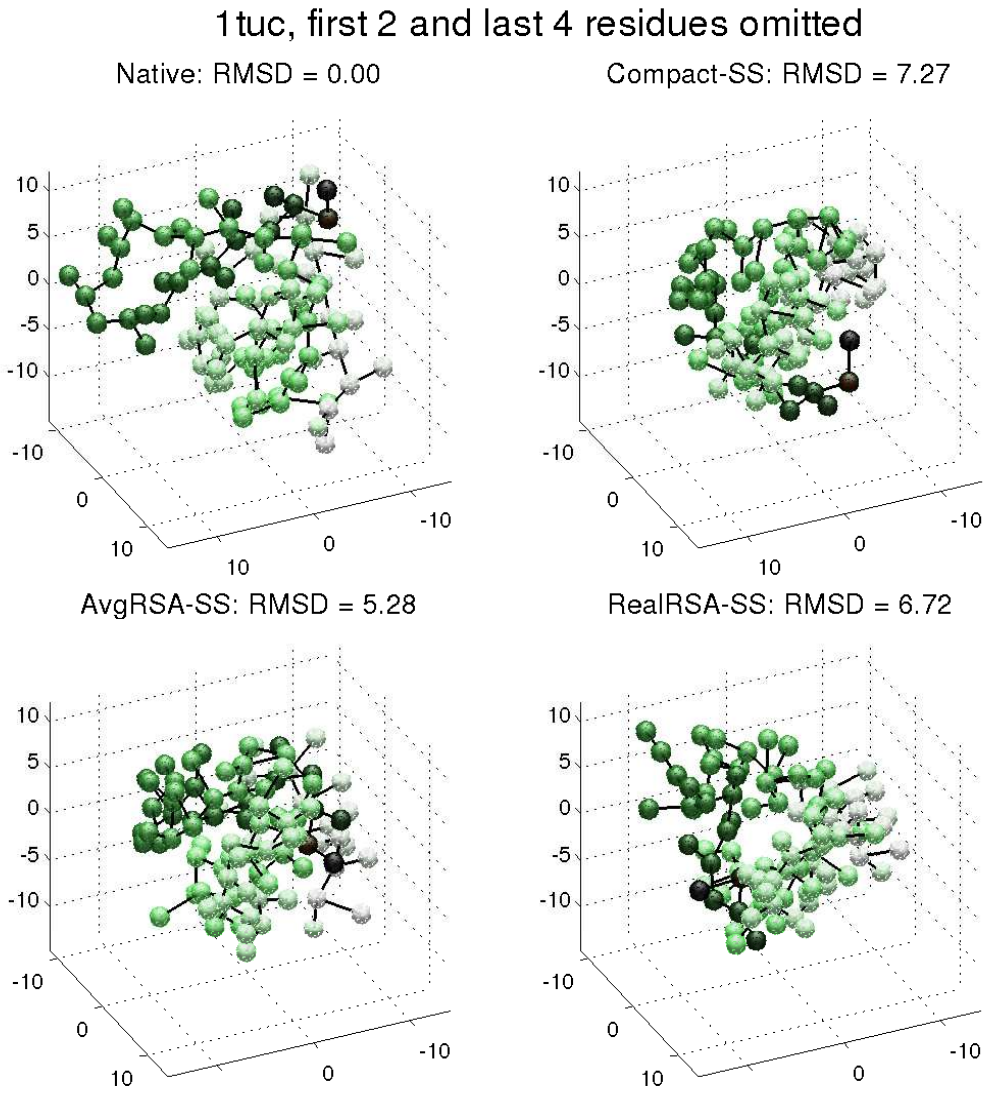
In some cases there are clear differences between how the energy functions behave. Figure 9.4 shows one such instance. In all three of the cases, Compact, AvgRSA, and RealRSA, the general position of the protein chain is correct. The tendency of the upper right Compact energy function to produce an overall spherical shape is clearly visible. AvgRSA and RealRSA produce somewhat more subtle variations on this with some beads protruding form a spherical core. Figure 9.5 illustrates these subtleties by coloring beads according to their exposure in the native structure. The Compact method simply produces a sphere with no distinction between placement of buried and exposed residues. AvgRSA and RealRSA clearly create a hydrophobic core and relegate exposed residues to the surface.

9.5 Hydrophobic Energy of Predicted and Native Conformations

To gain insight into how the variants of hydrophobic collapse behave, we compare the energy for the conformations produced in the Marie variants to the energy of the known native structure of the protein. As described in Section 8.6.5, this is done by fixing all pairwise distances of the structure to their native conformation values and then finding the minimum energy burial sphere position and size. The burial chamber is then placed and sized to minimize the energy function, one of Compact, AvgRSA, or RealRSA.

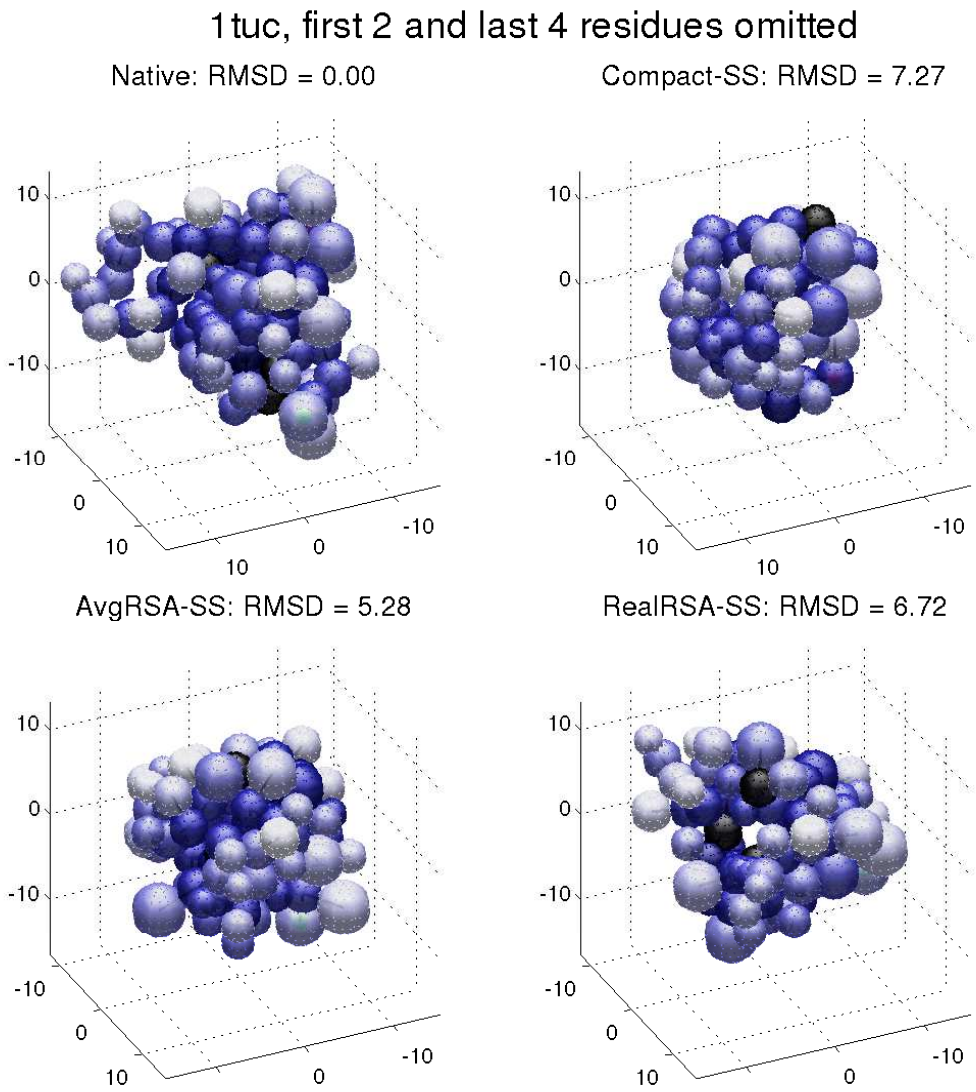
Figure 9.6 shows the energy values for conformations found by Marie under three sets of structure constraints: no constraints, fixed secondary structure, and fixed to native structure. Allowing complete flexibility of the structure, shown in the first panel column of Figure 9.6, yields the lowest energy values for the three hydrophobic collapse energy function variants (Compact, AvgRSA, and RealRSA). When secondary structure

Figure 9.4: Prediction Results for 1tuc Using Fixed Secondary Structure.



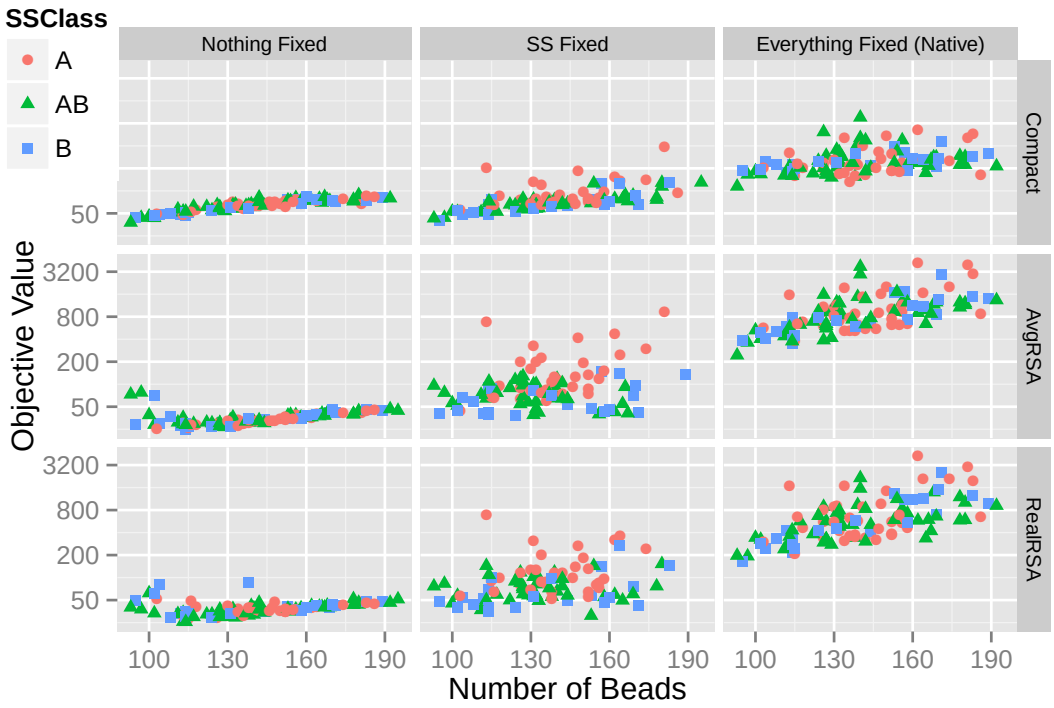
Each structure is shown in the rotation and translation that minimizes the main chain RMSD to the native conformation. As the N- and C-terminus of the native protein are somewhat extended, they are removed to compute this alignment (this was not done in the aggregate statistics). The coloring corresponds to the position of beads along the main chain: light the N-terminus and dark the C-terminus with all beads equal in size.. This coloring demonstrates the overall architecture of the beads in the Marie variants matches that of the native structure in the upper left. The three prediction methods, compaction only (Compact, upper right), optimized burial based on average RSA (AvgRSA, lower left), and based on real RSA (RealRSA, lower right), each give somewhat different structure predictions..

Figure 9.5: Alternative Prediction Results for 1tuc Using Fixed Secondary structure.



Display is identical to Figure 9.4 except that (1) the sizes of beads reflect their actual size and (2) the coloring of beads indicates their RSA in the native structure. Dark blue beads are buried while light beads are exposed.

Figure 9.6: Hydrophobic Collapse Energy of Conformations found by Marie



The size of proteins in beads varies along the X-axis of each panel and objective value of the final conformation varies along the Y-axis (log scale). Proteins in different secondary structure classes are distinguished by the marker color and shape. The column panels vary how much of the structure is held fixed while row panels vary which objective function is used to minimize. Panels in column one correspond to the Compact, AvgRSA, and RealRSA methods: no parts of the structure are fixed except adjacent backbone beads. Panels in column two correspond to the Compact-SS, AvgRSA-SS, and RealRS-SS methods where secondary structure is fixed. Panels in column three correspond to holding the whole structure fixed at its native conformation.

is fixed, shown in the second panel column, energy values generally increase. Fixing the whole structure to its native conformation and only allowing the burial chamber to change size and position results in the highest energy values. This trend is the inverse of the RMSD relationships observed in Figure 9.3 where fixing secondary structure on average lowered the RMSD of Marie conformations to native.

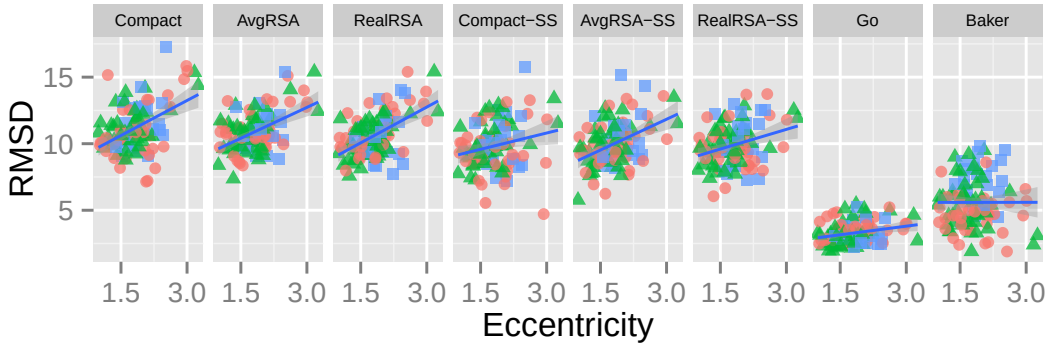
According to the Compact, AvgRSA, and RealRSA energy functions, native structures average much higher in energy than the conformations produced by the Marie optimization procedure. This means that native conformations do not reside near the global minimum of the the Compact, AvgRSA, and RealRSA energy functions which explains why the RMSDs for Marie conformations are quite high. We will need to do a thorough exploration of the parameters in *objradius*, *objexp*, and *objbur* in order to determine whether values exist which give native conformations low energy or whether there is an inherent limitation in the hydrophobic collapse model. It may be the case that the form of our energy functions, which accounts only for forces related to compactness and hydrophobicity, are simply insufficient to accurately model proteins. Should hydrogen bonding and van der Waals forces be required, it will be an indication that folding is not fully accounted for by hydrophobic collapse.

Finally, treating conformational search as a search for a global minimum of the energy function misses a key part of true protein folding: chains are subject to entropic forces and will naturally not adopt certain conformations such as knots. We have observed knotting in the conformations found by Marie (data not shown). Though such conformations have low energy according to Compact, AvgRSA, and RealRSA, they are not likely to be adopted by real proteins. Incorporating entropy or at least some notion of inaccessible conformational spaces into Mare will be a key area of further work.

9.6 Eccentricity of Native Structures

In the general theory of hydrophobic collapse, compactness is defined mainly in terms of isolating hydrophobic residues from solvent. The net effect is a structure that is not elongated. In the present implementation of hdyrophobic collapse in Marie, this compactness is described by a sphere. However, native proteins are observed in a variety of different conformations that bury hydrophobic residues but do not produce an overall spherical

Figure 9.7: Effects of Eccentricity of Native Structure on Prediction Quality



The X-axis shows the eccentricity of each native protein described by Equation 9.1. The Y-axis shows the RMSD of the predictions using using different energy functions. Color and shape of each point corresponds to the secondary structure of the protein. The trend line is a linear fit to the data.

shape to the protein. We suspect that the spherical assumption is a contributing factor to why Marie does not find native-like structures under hydrophobic collapse. Evidence of this appears in Figure 9.7 and Table 9.3 which show strong correlation of RMSD to how sphere-like the native protein is when hydrophobic collapse energy functions are employed.

We measured eccentricity as follows. We first computed the minimum volume bounding ellipsoid of the native conformation of each protein. This ellipsoid has three orthogonal axes with lengths A, B, C and we computed the eccentricity as the maximum ratio between the any two of the three.

$$Eccentricity = \max\{A/B, A/C, B/C\} \tag{9.1}$$

As the native protein becomes less spherical, the Compact AvgRSA, and RealRSA energy functions produce structures that are less native-like. The Go-like potential and Rosetta do not exhibit this trend. For comparison, Table 9.3 shows the correlation of RMSD with the size of the protein being predicted in terms of the number of beads in the protein. Nearly all energy functions produce worse RMSDs on larger proteins, the sole exception being the Go-like potential. The results from [39] also show correlation

Table 9.3: Correlation of RMSD with Size and Eccentricity

Method	Size	Eccentricity
Compact	0.525	0.490
AvgRSA	0.461	0.513
RealRSA	0.501	0.544
Compact-SS	0.545	0.213
AvgRSA-SS	0.511	0.486
RealRSA-SS	0.557	0.398
Go	0.171	0.247
Baker	0.526	-0.032

The two columns show the Pearson correlation coefficient between RMSD of low-energy structures found by Marie using various energy functions and two properties of the native conformation: its size measured by the number of beads in the model and the eccentricity of its bounding ellipsoid (see Section 9.6). Correlation with eccentricity reflects the smoothing lines drawn in Figure 9.7. For comparison, the correlation with the results from Baker's group are also shown [39].

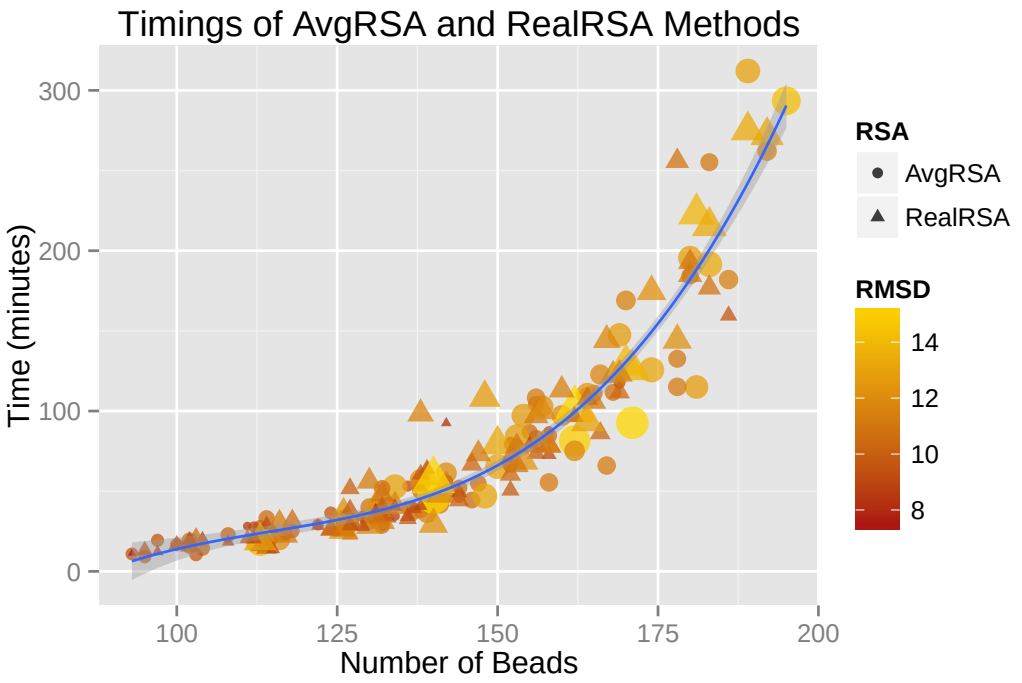
between protein size and RMSD of the best model. While all of the hydrophobic collapse energy functions produce worse structures as eccentricity increases, the effects of eccentricity are less pronounced for the Go-potential and Baker's results are essentially uncorrelated with eccentricity. Rosetta does not use a spherical measure of compactness so does not bias conformational search in the same way that Marie's hydrophobic collapse implementation currently does.

9.7 Computation Time for Hydrophobic Collapse

The running time of a subset of experiments is shown in the Figure 9.8. Only the AvgRSA and RealRSA methods were shown. They were run on the same system (the Itasca system, Section 9.1.1) for consistency. The plot includes a degree-3 polynomial fit of the number of beads to the time required for solution. The fit is tight with $R^2 = 0.91$, however a degree-4 polynomial gives a slightly better fit with slightly lower statistical significance based on the additional parameter according to an analysis of variance.

The run-time performance of SDP solvers is largely governed by the need to factor a system matrix whose size is proportional to the number of constraints in the SDP. In our case this is n^6 for n beads (see Section 8.6 for an explanation). It may be that we

Figure 9.8: Computation Time for Two Hydrophobic Collapse Energy Functions



Run-times are shown for AvgRSA (circles) and RealRSA (triangles). Both methods use no secondary structure information). X-position plots the size of the proteins and Y-position the amount of time taken to make the prediction (in minutes). The size and color of the plot symbols indicates their rough RMSD (smaller/darker is better). The blue line represents a linear fit of run-time by a degree 3 polynomial of the number of beads. The shaded area is the 95% confidence interval and goodness of fit is $R^2 = 0.91$.

have not tested large enough problems to see the increase in complexity associated with solving large SDPs. For larger proteins or models which include more beads per residue, this will likely become and issue.

9.8 Comparisons to Rosetta

The best structures predicted by Rosetta are better than those the hydrophobic collapse functions as indicated in the Baker row of Table 9.2 in the last panel column of Figure 9.3. This is not surprising for several reasons. First and foremost, the RMSDs reported in [39] per structure are for the best structure found out 1000 generated. In vivo, proteins are constantly flexing rather than rigid conformations so an ensemble of conformations around the native state is appropriate. However, in a predictive setting it may be difficult to select the best structure or cluster of structure from amongst the many that are generated. Second, the energy function and model used in Rosetta is far more detailed than those of Marie. Finally, Rosetta leverages a large database of background information in the form of its fragment libraries and secondary structure predictions which bias its overall predictions. While some of the variants used here employ secondary structure information and real RSA, there is no fragment library in Marie.

Even with those caveats, the quality of structures generated by hydrophobic collapse is currently not competitive with fragment assembly methods. It is clear from the results using the Go-like potential that given the right information, Marie is capable of locating near-native structures: the Go predictions are closer to native than Rosetta but exploit information on the native structure to achieve that end. It is unlikely our approximation of hydrophobic collapse would be competitive with the detailed and well-tuned energy functions of fragment assembly methods. However, in the future Marie may provide a computationally cheap starting point for more detailed conformational sampling.

The results reported in [39] cite an average run time of 12 hours to generate 1000 structure for a protein on a 450 MHz workstation. An average workstation today is in the 1.8 GHz range, roughly 4 times as fast for an approximate run time of 3 hours. That makes the present effort reasonably competitive in terms of speed.

9.9 Discussion of Marie Results

In all, we have demonstrated that there are distinct advantages to using Marie to model protein energy minimization as a convex optimization problem. We have shown how two classes of energy functions, Go-like potentials and hydrophobic collapse, can be handled by Marie thus enabling fast energy minimization. This capability is gained at the expense of being able to observe intermediate conformations of proteins as they fold: semidefinite programming only produces physically meaningful protein conformations on convergence to a low-dimensional solution. For this reason, Marie is at present incapable of allowing folding rate studies. However, from a pure structure prediction standpoint, this is valuable trade assuming we are able to determine an energy function which yields near-native protein structures without a prior knowledge of those structures. Many protein structure prediction methods already sacrifice physical accuracy to speed conformation search such as the swapping of backbone whole fragment shapes performed by Rosetta. Marie approaches the structure prediction standpoint from a pure optimization approach and speedily produces near-native structures when a Go-like potential is employed.

We observed several deficiencies in our hydrophobic collapse energy function which uses a burial sphere to separate hydrophobic residues to the interior and hydrophobic residues to the exterior. Currently, Marie finds low-energy conformations that are more compact and spherical than native structures. We anticipate that careful parameter tuning may improve this situation, but there are many questions left to answer on the viability of this approach to hydrophobic collapse. It is very possible for hydrophobic beads to be inside the burial sphere but still exposed to solvent. This happens when no hydrophobic bead is on the exterior to separate the hydrophobic beads from solvent. We also found that many native proteins adopt an overall prolate spheroid or cigar-like shape rather than a strict sphere. A spherical burial chamber is inappropriate to accurately model such proteins and we are currently exploring more general models for the burial chamber that can still be handled in an semidefinite program.

Tuning parameters in Marie can benefit from several areas. The cutting planes method for training machine learners may be useful for parameter tuning [85, 144, 86]. The procedure iterates between two modes: (1) estimate parameters so that a collection

of decoy objects are all higher scoring than a target object and (2) generate the minimum scoring object according to the current parameters; if it is far from the target object, add it to the set of decoys. The references mentioned use this paradigm for protein sequence alignment as finding the best "alignment object" for two sequences can be done relatively quickly given the alignment scoring parameters. Marie may not be fast enough to execute this cycle quickly enough to make cutting planes tractable. Inverse optimization [145] may provide a viable alternative methodology. Given an optimal point, in our case the native protein structure, and an optimization model, inverse optimization seeks the objective function which gives rise to the optimal point. This is precisely the problem we have at hand.

However, a second possibility also explains the fact that low-energy structures under our simple hydrophobic collapse energy functions are not close to native structures. It may be that the hydrophobic collapse theory itself is insufficient to explain the shape native proteins adopt. We can verify this by conducting a thorough parameter search of the Marie energy functions as described above to determine whether more native-like structures can be produced within hydrophobic collapse energy functions. We can also add in distance constraints to simulate hydrogen bonding, an alternative proposal for the driving force in protein folding. Such constraints are readily handled in Marie. Comparing these two alternatives will give insight into the fundamental properties of proteins folding.

Chapter 10

Conclusion and Future Directions

In the preceding chapters we have advanced computational methods for protein structure prediction and energy minimization is three separate areas. These are reviewed along with future directions in the following three summaries.

10.1 Interactions of Proteins with Small Molecules

In Chapter 3 we described LIBRUS which predicts which parts of a protein sequence will interact with small molecules. By combining both homology information and support vector learning, LIBRUS is capable of making binding-residue predictions nearly as accurately as a competing method which requires the full protein structure in order to make predictions. This capability enables lab practitioners to make intelligent choices about which sequence mutations to explore when studying the function of a protein. We utilized LIBRUS predictions in Chapter 4 and found that they are accurate enough to bolster the geometric accuracy of the 3D binding site for homology models of binding proteins. This was particularly helpful in cases where the protein being modeled does not have a close relative with known structure to use for a template. LIBRUS guides the alignment of the target's binding residues to those of the distantly related homolog so that the resulting binding site models the true structure's geometry closely.

As mentioned, the next phase in evaluation will be to examine how these models actually affect docking simulations between the protein model and ligand. LIBRUS predictions may also be used to intelligently bias the search space of docking locations

towards residues that are predicted to be binders. Finally, it is often the case that one is interested in proteins that are functionally similar to a target. Since function is in many cases tied to the binding site, LIBRUS predictions may be useful in schemes that identify functionally-related structures.

10.2 Protein Representation

In Chapter 5 we established a framework in which various protein representations and energy models can be evaluated. Coarse-grained and fine-grained models for structures along with associated energy functions were evaluated for their accuracy at discriminating misfolded protein decoys from the correctly folded native structures. The results of these experiments indicate that fine-grained atomic models are superior in accuracy but that intermediate representations that at least distinguish between backbone and sidechain portions of each residue provide a substantial performance boost over single-bead per residue models. The additional complexity of including 3-body interactions does not seem to improve much over 2-body interactions and there is a surprising amount of discriminative power in 1-body energies such as contact counts and solvent exposed surface area.

There are many more types of energy functions that can be explored within the machine learning framework established in Chapter 5. We restricted our attention to the oldest and most frequently used energy functions but the menagerie of energy functions is much broader. Of recent interest are force-fields that utilize the orientation of beads such as in [146]. Utilizing our framework, we can judge whether the complexity introduced by the orientation-dependent terms brings any additional benefits in terms of model accuracy.

Our bead selection method, BSM, in Chapter 6 allowed a mixture of fine- and coarse-grained representations to be assessed. The method selects beads using a regularization framework to establish exactly which parts of the representation should be introduced to maximize the accuracy at the minimum modeling cost.

Due to the size and density of the matrices inovolved with BSM, we employed subgradient methods to solve the resulting optimization problem. While adequate for our initial experiments, subgradient methods suffer have only mediocre speed and lack formal

criteria for detecting convergence to an optimal point. As mentioned, it is not possible to use several other optimization tools such as general purpose cone solvers (the problem is too large) or block coordinate descent (the BSM objective function is not separable). Problems of this type have received a great deal of attention recently and new methods have been proposed to solve them more efficiently than generic subgradient methods. The smoothing proximal gradient [147] is one such method which may be applied to bead selection and we are currently investigating its suitability.

10.3 Effective Energy Minimization

In Chapters 8 to 9 we developed a method to find low-energy protein conformations using convex optimizations, the Marie framework. Marie synthesized traditional energy functions with convex optimization and distance geometry. This novel approach completely relinquished the ability to observe folding dynamics in order to achieve very fast prediction of low-energy conformations. We showed how to implement a Go-like potential in Marie and that for a wide range of native contact distances, Marie finds near-native structures. There appears to be great deal of flexibility in how one defines a native contact as such a large range of cut-offs yield near-native structures at their energy minimum. We also showed that Marie can optimize a hydrophobic collapse energy function corresponding to an established driving force in protein folding. Marie effectively minimizes energy, but the resulting structures are not native-like in all the variants we explored. Our study revealed several deficiencies with the current approach to hydrophobic collapse, particularly that it favors spherical proteins too much. Many native structures are more cigar-shaped and are thus not approximated well by our current burial sphere approach.

Future work on Marie will focus on generalizing the burial sphere to an arbitrary ellipse so that a wider range of structures can be handled in hydrophobic collapse. As we mentioned in Chapter 9, a variety of parameters must be set in the hydrophobic collapse energy function and we are looking at using inverse optimization in order set them. This would fully synthesize Marie with the machine learning paradigm as it would be a closed loop learning and prediction scheme based on the theory of convex optimization.

References

[1] JC Kendrew, G Bodo, HM Dinitzis, RG Parrish, H Wyckoff, and DC Phillips. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, 181(4610):662–6, 1958.

[2] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, T N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The Protein Data Bank. *Nucl. Acids Res.*, 28(1):235–242, 2000.

[3] Joel R Bock and David A Gough. Virtual Screen for Ligands of Orphan G Protein-Coupled Receptors. *Journal of Chemical Information and Modeling*, 45(5):1402–1414, 2005.

[4] Konrad H Bleicher, Hans-Joachim Bohm, Klaus Muller, and Alexander I Alanine. Hit and lead generation: beyond high-throughput screening. *Nat Rev Drug Discov*, 2(5):369–378, May 2003.

[5] Dirk Weber, Claudia Berger, Timo Heinrich, Peter Eickelmann, Jochen Antel, and Horst Kessler. Systematic optimization of a lead-structure identities for a selective short peptide agonist for the human orphan receptor BRS-3. *J Pept Sci*, 8(8):461–475, August 2002.

[6] Chris Kauffman, Huzefa Rangwala, and George Karypis. Improving Homology Models for Protein-Ligand Binding Sites. In *LSS Computational Systems Bioinformatics Conference*, Stanford, CA, 2008.

[7] N Moitessier, P Englebienne, D Lee, J Lawandi, and C R Corbeil. Towards the development of universal, fast and highly accurate docking//scoring methods: a long way to go. *Br J Pharmacol*, 153(S1):S7—-S26, November 2007.

[8] Yanay Ofran, Venkatesh Mysore, and Burkhard Rost. Prediction of {DNA}-binding residues from sequence. *Bioinformatics*, 23(13):i347—-i353, July 2007.

[9] Shandar Ahmad and Akinori Sarai. {PSSM}-based prediction of {DNA} binding sites in proteins. *BMC Bioinformatics*, 6:33, 2005.

[10] Huzefa Rangwala, Christopher Kauffman, and George Karypis. A Generalized Framework for Protein Sequence Annotation. In *Proceedings of the NIPS Workshop on Machine Learning in Computational Biology*, Vancouver, B.C., Canada., 2007.

[11] Michael Terribilini, Jae-Hyung Lee, Changhui Yan, Robert L Jernigan, Vasant Honavar, and Drena Dobbs. Prediction of RNA binding sites in proteins from amino acid sequence. *RNA*, 12(8):1450–1462, 2006.

[12] Manish Kumar, M Michael Gromiha, and G P S Raghava. Prediction of RNA binding sites in a protein using SVM and PSSM profile. *Proteins*, 71(1):189–194, April 2008.

[13] Yanay Ofran and Burkhard Rost. Predicted protein-protein interaction sites from local sequence information. *FEBS Lett*, 544(1-3):236–239, June 2003.

[14] Ming-Hui Li, Lei Lin, Xiao-Long Wang, and Tao Liu. Protein protein interaction site prediction based on conditional random fields. *Bioinformatics*, 23(5):597–604, 2007.

[15] Asako Koike and Toshihisa Takagi. Prediction of protein-protein interaction sites using support vector machines. *Protein Engineering, Design and Selection*, 17(2):165–173, 2004.

[16] JD Fischer, CE Mayer, and J Soding. Prediction of protein functional residues from sequence by probability density estimation. *Bioinformatics*, 24(5):613–620, March 2008.

[17] Craig T Porter, Gail J Bartlett, and Janet M Thornton. The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res*, 32(Database issue):D129—-D133, January 2004.

[18] Natalia V Petrova and Cathy H Wu. Prediction of catalytic residues using Support Vector Machine with selected protein sequence and structural properties. *BMC Bioinformatics*, 7:312, 2006.

[19] Eunseog Youn, Brandon Peters, Predrag Radivojac, and Sean D Mooney. Evaluation of features for catalytic residue prediction in novel folds. *Protein Sci*, 16(2):216–226, February 2007.

[20] A G Murzin, S E Brenner, T Hubbard, and C Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247(4):536–540, April 1995.

[21] Antonina Andreeva, Dave Howorth, John-Marc Chandonia, Steven E Brenner, Tim J P Hubbard, Cyrus Chothia, and Alexey G Murzin. Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res*, 36(Database issue):D419—-D425, January 2008.

[22] Gonzalo López, Alfonso Valencia, and Michael L Tress. firestar–prediction of functionally important residues using structural templates and alignment reliability. *Nucleic Acids Res*, 35(Web Server issue):W573—-W577, July 2007.

[23] Michal Brylinski and Jeffrey Skolnick. A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. *Proc Natl Acad Sci U S A*, 105(1):129–134, January 2008.

[24] Jeffrey Skolnick and Michal Brylinski. FINDSITE: a combined evolution/structure-based approach to protein function prediction. *Briefings in Bioinformatics*, 10(4):378–391, July 2009.

[25] S F Altschul, T L Madden, A A Schaffer, J Zhang, Z Zhang, W Miller, and D J Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucl. Acids Res.*, 25(17):3389–3402, 1997.

[26] Wolfgang Kabsch and Chris Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.

[27] Ke Chen and Lukasz Kurgan. PFRES: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics*, 23(21):2843–2850, 2007.

[28] Krzysztof Ginalski, Jakub Pas, Lucjan S Wyrwicz, Marcin von Grotthuss, Janusz M Bujnicki, and Leszek Rychlewski. ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. *Nucl. Acids Res.*, 31(13):3804–3807, 2003.

[29] George Karypis. {YASSPP}: Better Kernels and Coding Schemes Lead to Improvements in SVM-based Secondary Structure Prediction. *Proteins: Structure, Function and Bioinformatics*, 64(3):575–586, 2006.

[30] David Mittelman, Ruslan Sadreyev, and Nick Grishin. Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments. *Bioinformatics*, 19(12):1531–1539, August 2003.

[31] A Heger and L Holm. Picasso: generating a covering set of protein family profiles. *Bioinformatics*, 17(3):272–279, March 2001.

[32] Huzefa Rangwala and George Karypis. fRMSDPred: predicting local RMSD between structural fragments using sequence information. *Comput Syst Bioinformatics Conf*, 6:311–322, 2007.

[33] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press, New York, 1997.

[34] Chris Kauffman and George Karypis. LIBRUS: combined machine learning and homology information for sequence-based ligand-binding residue prediction. *Bioinformatics*, 25(23):3099–3107, 2009.

[35] Vladimir N Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[36] John-Marc Chandonia, Nigel S Walker, Loredana Lo Conte, Patrice Koehl, Michael Levitt, and Steven E Brenner. ASTRAL compendium enhancements. *Nucleic Acids Res*, 30(1):260–263, January 2002.

[37] Tom Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006.

[38] Philippe Ferrara and Edgar Jacoby. Evaluation of the utility of homology models in high throughput docking. *Journal of Molecular Modeling*, 13(8):897–905, August 2007.

[39] D Baker and A Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–96, October 2001.

[40] Carol DeWeese-Scott and John Moult. Molecular modeling of protein function regions. *Proteins*, 55(4):942–961, June 2004.

[41] Suvobrata Chakravarty, Lei Wang, and Roberto Sanchez. Accuracy of structure-derived properties in simple comparative models of protein structures. *Nucleic Acids Res*, 33(1):244–259, 2005.

[42] David Piedra, Sergi Lois, and Xavier de la Cruz. Preservation of protein clefts in comparative models. *BMC Struct Biol*, 8(1):2, January 2008.

[43] Angel R Ortiz, Charlie E M Strauss, and Osvaldo Olmea. MAMMOTH (Matching molecular models obtained from theory): An automated method for model comparison. *Protein Sci*, 11(11):2606–2621, 2002.

[44] A Sali and T L Blundell. Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol*, 234(3):779–815, December 1993.

[45] Xia Ning, Huzefa Rangwala, and George Karypis. Multi-Assay-Based Structure-Activity Relationship Models: Improving Structure-Activity Relationship Models by Incorporating Activity Information from Related Targets. *Journal of Chemical Information and Modeling*, 49(11):2444–2456, 2009.

[46] Volker Baschnagel, Jorg; Binder, Kurt; Doruker, Pemra; Gusev, Andrei A.; Hahn, Oliver; Kremer, Kurt; Mattice, Wayne L.; Muller-Plathe, Florian; Murat, Michael; Paul, Wolfgang; Santos, Serge; Suter, Ulrich W.; Tries. Bridging the Gap between atomistic and coarse-grained models of polymers: Status and perspectives. In *Viscoelasticity, atomistic models, statistical chemistry; Advances in Polymer Sciences*, volume 152 of *Advances in Polymer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, July 2000.

[47] V Tozzini. Coarse-grained models for proteins. *Curr Opin Struct Biol*, 15(2):144–150, 2005.

[48] Michael Levitt. A simplified representation of protein conformations for rapid simulation of protein folding. *Journal of Molecular Biology*, 104(1):59–107, June 1976.

[49] Yuzo Ueda, Hiroshi Taketomi, and Nobuhiro Gō. Studies on protein folding, unfolding, and fluctuations by computer simulation. II. A. Three-dimensional lattice model of lysozyme. *Biopolymers*, 17(6):1531–1548, June 1978.

[50] A Kolinski and J Skolnick. Assembly of protein structure from sparse experimental data: an efficient Monte Carlo model. *Proteins*, 32(4):475–94, September 1998.

[51] K T Simons, C Kooperberg, E Huang, and D Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of molecular biology*, 268(1):209–25, April 1997.

[52] A Liwo, S Odziej, M R Pincus, R J Wawak, S Rackovsky, and H A Scheraga. A united-residue force field for off-lattice protein-structure simulations. I. Functional forms and parameters of long-range side-chain interaction potentials from protein crystal data. *Journal of Computational Chemistry*, 18(7):849–873, 1997.

[53] Adam Liwo, Piotr Arłukowicz, Cezary Czaplewski, Stanisław Ołdziej, Jarosław Pillardy, and Harold A Scheraga. A method for optimizing potential-energy functions by a hierarchical design of the potential-energy landscape: Application to the UNRES force field. *Proceedings of the National Academy of Sciences*, 99(4):1937–1942, 2002.

[54] Yang Zhang, Andrzej Kolinski, and Jeffrey Skolnick. TOUCHSTONE II: a new approach to ab initio protein structure prediction. *Biophysical journal*, 85(2):1145–64, August 2003.

[55] Siewert J Marrink, H Jelger Risselada, Serge Yefimov, D Peter Tieleman, and Alex H de Vries. The MARTINI force field: coarse grained model for biomolecular simulations. *The journal of physical chemistry. B*, 111(27):7812–24, July 2007.

[56] Ozge Kurkcuoglu, Robert L. Jernigan, and Pemra Doruker. Mixed levels of coarse-graining of large proteins using elastic network model succeeds in extracting the slowest motions. *Polymer*, 45(2):649–657, January 2004.

[57] Ana V Rojas, Adam Liwo, and Harold A Scheraga. Molecular dynamics with the United-residue force field: ab initio folding simulations of multichain proteins. *The journal of physical chemistry. B*, 111(1):293–309, January 2007.

[58] R Samudrala and M Levitt. Decoys 'R' Us: a database of incorrect conformations to improve protein structure prediction. *Protein science : a publication of the Protein Society*, 9(7):1399–401, July 2000.

[59] B Park and M Levitt. Energy functions that discriminate X-ray and near native folds from well-constructed decoys. *Journal of molecular biology*, 258(2):367–92, May 1996.

[60] R Samudrala, Y Xia, M Levitt, and E S Huang. A combined approach for ab initio construction of low resolution protein tertiary structures from sequence. *Pacific Symposium On Biocomputing*, 516:505–516, 1999.

[61] Chen Keasar and Michael Levitt. A Novel Approach to Decoy Set Generation: Designing a Physical Energy Function Having Local Minima with Native Structure Characteristics. *Journal of Molecular Biology*, 329(1):159–174, May 2003.

[62] Dmitry Rykunov and Andras Fiser. New statistical potential for quality assessment of protein models and a survey of energy functions. *BMC Bioinformatics*, 11(1):128, January 2010.

[63] Bino John and Andrej Sali. Comparative protein structure modeling by iterative alignment, model building and model assessment. *Nucleic Acids Research*, 31(14):3982–3992, 2003.

[64] David Eramian, Min-yi Shen, Damien Devos, Francisco Melo, Andrej Sali, and Marc A Marti-Renom. A composite score for predicting errors in protein structure models. *Protein science : a publication of the Protein Society*, 15(7):1653–66, July 2006.

[65] Jerry Tsai, Richard Bonneau, Alexandre V Morozov, Brian Kuhlman, Carol A Rohl, and David Baker. An improved protein decoy set for testing energy functions for protein structure prediction. *Proteins*, 53(1):76–87, October 2003.

[66] Liliana Wroblewska and Jeffrey Skolnick. Can a physics-based, all-atom potential find a protein's native structure among misfolded structures? I. Large scale AMBER benchmarking. *J Comput Chem*, 28(12):2059–2066, September 2007.

[67] Rhiju Das, Bin Qian, Srivatsan Raman, Robert Vernon, James Thompson, Philip Bradley, Sagar Khare, Michael D Tyka, Divya Bhat, Dylan Chivian, David E Kim, William H Sheffler, Lars Malmström, Andrew M Wollacott, Chu Wang, Ingemar Andre, and David Baker. Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home. *Proteins*, 69 Suppl 8:118–28, January 2007.

[68] Philip Bradley, Kira M S Misura, and David Baker. Toward high-resolution de novo structure prediction for small proteins. *Science (New York, N.Y.)*, 309(5742):1868–71, September 2005.

[69] C Loose, J L Klepeis, and C A Floudas. A new pairwise folding potential based on improved decoy generation and side-chain packing. *Proteins*, 54(2):303–14, February 2004.

[70] Jian Qiu and Ron Elber. Atomically detailed potentials to recognize native and approximate protein structures. *Proteins*, 61(1):44–55, October 2005.

[71] Qiwen Dong and Shuigeng Zhou. Novel Nonlinear Knowledge-Based Mean Force Potentials Based on Machine Learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8:476–486, 2011.

[72] R Samudrala and J Moult. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *Journal of molecular biology*, 275(5):895–916, February 1998.

[73] Chi Zhang, Song Liu, Hongyi Zhou, and Yaoqi Zhou. An accurate, residue-level, pair potential of mean force for folding and binding based on the distance-scaled, ideal-gas reference state. *Protein science : a publication of the Protein Society*, 13(2):400–11, February 2004.

[74] Adam Liwo, Piotr Arłukowicz, Stanisław Ołdziej, Cezary Czaplewski, Mariusz Makowski, and Harold A Scheraga. Optimization of the UNRES Force Field by Hierarchical Design of the Potential-Energy Landscape. 1. Tests of the Approach Using Simple Lattice Protein Models. *The Journal of Physical Chemistry B*, 108(43):16918–16933, 2004.

[75] P J Munson and R K Singh. Statistical significance of hierarchical multi-body potentials based on Delaunay tessellation and their application in sequence-structure alignment. *Protein science : a publication of the Protein Society*, 6(7):1467–81, July 1997.

[76] Yaping Feng, Andrzej Kloczkowski, and Robert L Jernigan. Four-body contact potentials derived from two protein datasets to discriminate native structures from decoys. *Proteins*, 68(1):57–66, July 2007.

[77] B. Krishnamoorthy and A. Tropsha. Development of a four-body statistical pseudo-potential to discriminate native from non-native protein conformations. *Bioinformatics*, 19(12):1540–1548, August 2003.

[78] Pawel Gniewek, Sumudu P Leelananda, Andrzej Kolinski, Robert L Jernigan, and Andrzej Kloczkowski. Multibody coarse-grained potentials for native structure recognition and quality assessment of protein models. *Proteins: Structure, Function and Bioinformatics*, 79(6):1923–1929, June 2011.

[79] Akira R Kinjo, Katsuhisa Horimoto, and Ken Nishikawa. Predicting absolute contact numbers of native protein structure from amino acid sequence. *Proteins*, 58(1):158–65, January 2005.

[80] Zheng Yuan. Better prediction of protein contact number using a support vector regression analysis of amino acid sequence. *BMC bioinformatics*, 6(1):248, January 2005.

[81] R Development Core Team. *R: A Language and Environment for Statistical Computing*, volume 1 of *R Foundation for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011.

[82] Jinfeng Zhang, Rong Chen, and Jie Liang. Empirical potential function for simplified protein models: combining contact and local sequence-structure descriptors. *Proteins*, 63(4):949–60, June 2006.

[83] I Bahar, M Kaplan, and R L Jernigan. Short-range conformational energies, secondary structure propensities, and recognition of correct sequence-structure matches. *Proteins*, 29(3):292–308, November 1997.

[84] Chih-chung Chang and Chih-jen Lin. LIBSVM: a library for support vector machines. *Science*, 2(3):1–39, 2011.

[85] Thorsten Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 06*, pages 217–226, Philadelphia, PA, USA, 2006. ACM, ACM Press, New York, NY, USA.

[86] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

[87] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[88] R Rajgaria, S R McAllister, and C A Floudas. A novel high resolution C-alpha C-alpha distance dependent force field based on a high quality decoy set. *Proteins: Structure, Function, and Bioinformatics*, 65(3):726–741, 2006.

[89] R Rajgaria, S R McAllister, and C A Floudas. Distance dependent centroid to centroid force fields using high resolution decoys. *Proteins*, 70(3):950–70, February 2008.

[90] Ching-Wai Tan and David T Jones. Using neural networks and evolutionary information in decoy discrimination for protein tertiary structure prediction. *BMC bioinformatics*, 9(1):94, January 2008.

[91] Piotr Pokarowski, Andrzej Kloczkowski, Robert L Jernigan, Neha S Kothari, Maria Pokarowska, and Andrzej Kolinski. Inferring ideal amino acid interaction forms from statistical protein contact potentials. *Proteins: Structure, Function, and Bioinformatics*, 59(1):49–57, 2005.

[92] Piotr Pokarowski, Andrzej Kloczkowski, Szymon Nowakowski, Maria Pokarowska, Robert L Jernigan, and Andrzej Kolinski. Ideal amino acid exchange forms for approximating substitution matrices. *Proteins*, 69(2):379–93, November 2007.

[93] Julia Handl, Joshua Knowles, and Simon C Lovell. Artefacts and biases affecting the evaluation of scoring functions on decoy sets for protein structure prediction. *Bioinformatics*, 25(10):1271–1279, May 2009.

[94] Ariadna Quattoni, Xavier Carreras, Michael Collins, and Trevor Darrell. An Efficient Projection for L1,Infinity Regularization. In *Proceedings of the 26th Annual International Conference on Machine Learning ICML 09*, pages 1–8, Montreal, QC, Canada, June 14 - 18, 2009, 2009. ACM Press, New York, NY, USA.

[95] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2007.

[96] Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, USA., 2nd edition, September 1999.

[97] Ken A Dill, S Banu Ozkan, Thomas R Weikl, John D Chodera, and Vincent A Voelz. The protein folding problem: when will it be solved? *Current opinion in structural biology*, 17(3):342–6, June 2007.

[98] Ken A Dill and Hue Sun Chan. From Levinthal to pathways to funnels. *Nat Struct Mol Biol*, 4(1):10–19, January 1997.

[99] H Frauenfelder, S G Sligar, and P G Wolynes. The energy landscapes and motions of proteins. *Science*, 254(5038):1598–1603, 1991.

[100] J D Bryngelson, J N Onuchic, N D Socci, and P G Wolynes. Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins*, 21(3):167–95, March 1995.

[101] K A Dill. Polymer principles and protein folding. *Protein science : a publication of the Protein Society*, 8(6):1166–80, June 1999.

[102] K A Dill. Dominant forces in protein folding. *Biochemistry*, 29(31):7133–7155, 1990.

[103] H S Chan and K A Dill. Origins of structure in globular proteins. *Proceedings of the National Academy of Sciences*, 87(16):6388–6392, 1990.

[104] Michal Brylinski, Leszek Konieczny, and Irena Roterman. Hydrophobic collapse in (in silico) protein folding. *Computational biology and chemistry*, 30(4):255–67, August 2006.

[105] J K Myers and C N Pace. Hydrogen bonding stabilizes globular proteins. *Biophysical journal*, 71(4):2033–9, October 1996.

[106] George D Rose, Patrick J Fleming, Jayanth R Banavar, and Amos Maritan. A backbone-based theory of protein folding. *Proceedings of the National Academy of Sciences of the United States of America*, 103(45):16623–33, November 2006.

[107] Jianmin Gao, Daryl A Bosco, Evan T Powers, and Jeffery W Kelly. Localized thermodynamic coupling between hydrogen bonding and microenvironment polarity substantially stabilizes proteins. *Nat Struct Mol Biol*, 16(7):684–690, July 2009.

[108] C Nick Pace. Energetics of protein hydrogen bonds. *Nat Struct Mol Biol*, 16(7):681–682, July 2009.

[109] Haruo Abe and Nobuhiro Go. Noninteracting local-structure model of folding and unfolding transition in globular proteins. II. Application to two-dimensional lattice proteins. *Biopolymers*, 20(5):1013–1031, 1981.

[110] Nobuhiro Go and Haruo Abe. Noninteracting local-structure model of folding and unfolding transition in globular proteins. I. Formulation. *Biopolymers*, 20(5):991–1011, 1981.

[111] Nobuhiro Go. Theoretical Studies of Protein Folding. *Annual Review of Biophysics and Bioengineering*, 12(1):183–210, 1983.

[112] Shoji Takada. Go-ing for the prediction of protein folding mechanisms. *Proceedings of the National Academy of Sciences*, 96(21):11698–11700, 1999.

[113] José Nelson Onuchic and Peter G Wolynes. Theory of protein folding. *Current Opinion in Structural Biology*, 14(1):70–75, 2004.

[114] S Banu Ozkan, Ivet Bahar, and Ken A Dill. Transition states and the meaning of [Phi]-values in protein folding kinetics. *Nat Struct Mol Biol*, 8(9):765–769, September 2001.

[115] E M Boczko and C L Brooks. First-principles calculation of the folding free energy of a three-helix bundle protein. *Science*, 269(5222):393–396, 1995.

[116] Jun Shimada, Edo L Kussell, and Eugene I Shakhnovich. The folding thermodynamics and kinetics of crambin using an all-atom monte carlo simulation. *Journal of Molecular Biology*, 308(1):79–95, 2001.

[117] Victor Muñoz and William A Eaton. A simple model for calculating the kinetics of protein folding from three-dimensional structures. *Proceedings of the National Academy of Sciences*, 96(20):11311–11316, 1999.

[118] Nobuyasu Koga and Shoji Takada. Roles of native topology and chain-length scaling in protein folding: A simulation study with a Gō-like model. *Journal of Molecular Biology*, 313(1):171–180, 2001.

[119] K A Dill, A T Phillips, and J B Rosen. Molecular structure prediction by global optimization. In I.M. Bomze, T. Csendes, R. Horst, and P.M. Pardalos, editors, *Developments in Global Optimization*, pages 217–234, Dordrecht, Netherlands, 1997. Kluwer Academic.

[120] A T Phillips, J B Rosen, and K A Dill. Convex global underestimation for molecular stucture prediction. In Athanasios Migdalas, P M Pardalos, and Peter Varbrand, editors, *From Local to Global Optimization*, pages 1–18, Boston, Mass., 2001. Kluwer Academic Publishers.

[121] Andrew R Conn, Nicholas I M Gould, and Philippe L Toint. *Trust-region methods*, volume MPS/SIAM S of *MPS/SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 2000.

[122] Maranas Androulakis, C D Maranas, I P Androulakis, and C A Floudas. A Deterministic Global Optimization Approach for the Protein Folding Problem. *J. Chem. Phys*, 100:133–150, 1996.

[123] Yang Zhang and Jeffrey Skolnick. Automated structure prediction of weakly homologous proteins on a genomic scale. *Proceedings of the National Academy of Sciences of the United States of America*, 101(20):7594–7599, 2004.

[124] Sitao Wu, Jeffrey Skolnick, and Yang Zhang. Ab initio modeling of small proteins by iterative TASSER simulations. *BMC biology*, 5(1):17, January 2007.

[125] David E Kim, Ben Blum, Philip Bradley, and David Baker. Sampling bottlenecks in de novo protein structure prediction. *Journal of molecular biology*, 393(1):249–60, October 2009.

[126] A L Beberg, D L Ensign, G Jayachandran, S Khaliq, and V S Pande. Folding@home: Lessons from eight years of volunteer distributed computing, 2009.

[127] Lisa Kinch, Shuo Yong Shi, Qian Cong, Hua Cheng, Yuxing Liao, and Nick V Grishin. CASP9 assessment of free modeling target predictions. *Proteins: Structure, Function, and Bioinformatics*, 79(S10):59–73, 2011.

[128] Ingwer Borg and Patrick Groenen. *Modern Multidimensional Scaling: Theory and Applications, 2nd Edition*. Springer, 2005.

[129] Michael W Carter, Holly H Jin, Michael A Saunders, and Yinyu Ye. Spaseloc: An adaptable subproblem algorithm for scalable wireless network localization. *SIAM J. on Optimization*, 2005.

[130] Blake Shaw and Tony Jebara. Structure Preserving Embedding. In Léon Bottou and Michael Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 937–944, Montreal, June 2009. Omnipress.

[131] K Wurhrich. Protein structure determination in solution by NMR spectroscopy. *Journal of Biological Chemistry*, 265(36):22059–22062, 1990.

[132] G M Crippen and T F Havel. *Distance Geometry and Molecular Conformation.* Wiley, 1988.

[133] Pratik Biswas, Kim-Chuan Toh, and Yinyu Ye. A Distributed SDP Approach for Large-Scale Noisy Anchor-Free Graph Realization with Applications to Molecular Conformation. *SIAM Journal on Scientific Computing*, 30(3):1251–1277, 2008.

[134] Enrico O Purisima and Harold A Scheraga. An approach to the multiple-minima problem in protein folding by relaxing dimensionality : Tests on enkephalin. *Journal of Molecular Biology*, 196(3):697–709, August 1987.

[135] Chris Kauffman and George Karypis. Coarse- and fine-grained models for proteins: Evaluation by decoy discrimination. *Proteins: Structure, Function, and Bioinformatics*, 81(5):754—-773, 2013.

[136] Andrzej Kolinski and Jeffrey Skolnick. Reduced models of proteins and their applications. *Polymer*, 45(2):511–524, 2004.

[137] Elizabeth Durham, Brent Dorr, Nils Woetzel, René Staritzbichler, and Jens Meiler. Solvent accessible surface area approximations for rapid and accurate protein structure prediction. *Journal of Molecular Modeling*, 15(9):1093–1108, 2009.

[138] Jon Dattorro. *Convex Optimization and Euclidean Distance Geometry.* Meboo Publishing, 2009.

[139] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge University Press Cambridge, 2006.

[140] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[141] Brian Borchers. CSDP, A C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

[142] The MathWorks Inc. Matlab Version 7.2. Natick, Massachusetts, 2006.

[143] Hadley Wickham. *ggplot2: elegant graphics for data analysis.* Springer New York, New York, NY, USA, 2009.

[144] John Kececioglu and Eagu Kim. Simple and fast inverse alignment. In *Annual International Conference on Research in Computational Molecular Biology (RECOMB). Volume 3909 of Lecture Notes in Computer Science*, pages 441–455. Springer, 2006.

[145] Clemens Heuberger. Inverse Optimization: A Survey on Problems, Methods, and Results. *J. Combin. Opt*, 8:329–361, 2004.

[146] Hongyi Zhou and Jeffrey Skolnick. GOAP: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction. *Biophysical journal*, 101(8):2043–52, October 2011.

[147] Xi Chen, Qihang Lin, Seyoung Kim, Jaime G. Carbonell, and Eric P. Xing. Smoothing proximal gradient method for general structured sparse regression. *Annals of Applied Statistics*, 6(2):719–52, 2012.