Structured Probabilistic Models of Proteins across Spatial and Fitness Landscapes

Hetunandan Kamichetty

CMU-CS-11-116

March 2011

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Chris J. Langmead(co-chair)
Eric P. Xing (co-chair)
Jaime Carbonell
Chris Bailey-Kellogg

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: Graphical Models, Graphical Games, Protein Structure, Protein Design, Drug Design

Dedicated to the memory of my uncle, P.K. Narendra (1958–2007).

Abstract:

Proteins are dynamic molecules. They flex in space, adopting many different spatial configurations while performing their function and evolve over time, changing their amino acid composition in response to changing fitness landscapes. The thesis of this dissertation is that this *inherent* variability of proteins can be modeled by structurally *sparse* representations. These sparse models can then be used to efficiently reason about the properties of the protein by the means of algorithms that exploit their sparsity.

This dissertation develops the first Probabilistic Graphical Model that models the entire protein across spatial configurations (GOBLIN). By compactly encoding and manipulating a probability distribution over an exponentially large space, and using statistical inference algorithms that exploit structural sparsity, GOBLIN is able to compute experimentally measurable properties of protein interactions quickly and accurately.

We then develop a method of learning generative models of amino acid composition of evolutionarily related protein families (GREMLIN) that captures dependencies between sequential and long-range pairs of positions in the protein. GREMLIN is vastly more accurate than existing statistical models based on Hidden Markov Models; by effectively utilizing a distributed map-reduce framework, it also presents a scalable alternative to these extant approaches.

Building on these two contributions, this dissertation develops a game-theoretic approach to drug design (GAMUT). GAMUT determines the affects of a change in the fitness landscape on the composition of the protein. GAMUT can be used to design drug cocktails that remain effective against natural possible mutant variants of the target. Towards this, GAMUT develops a novel algorithm that bounds properties of the Correlated Equilibria of Graphical Games based on outer relaxations to the marginal polytope.

Acknowledgements

In Prof. Christopher Langmead and Prof. Eric Xing, I had the good fortune of having two great advisors. I thank them for their constant guidance and support. I also thank my thesis committee members, Prof. Jaime Carbonell and Prof. Chris Bailey-Kellogg for agreeing to be on my committee and providing their invaluable inputs. I'd be remiss if I did not thank Deborah Cavlovich for her help with all department matters.

Prof. Chris Bailey-Kellogg was also my advisor when I was an M.S student at Purdue. I learnt a lot about Computational Biology and science while in his lab. I owe a special thanks to him for his advice and guidance throughout my graduate life.

Much of what I learnt about leading my life, I learnt from my family. My parents taught me how to focus on the important things in life and how not to worry about the rest. My sister has an infectious enthusiasm for life and made sure I was always grounded in reality. For these lessons and their constant love and support, I remain indebted to them.

I've had the good fortune to have excellent friends over the years who have inspired me with their extraordinary abilities. There are too many to list here, but a few deserve special mention: Vyas Sekar, Varun Gupta, Swapnil Patil, Gaurav Veda, Avijit Kumar and Amitanand Aiyer were partners in crime in graduate school and earlier. They have inspired me with their extraordinary abilities and have suffered my company over the years. For this, I owe them my thanks.

Finally, I thank my wife, Ashwini, for her love and affection and for being both my strongest support and strictest critic. I've learnt that a necessary condition for giving a good

talk is to first practice it in front of my wife. It has worked every time!

Contents

Chapter 1

Introduction

Proteins are large polypeptide molecules that are workhorses of a cell. Despite their relatively limited compositional diversity (twenty letter alphabet) they are responsible for the birth, maintenance and death of cells and ultimately of life. They perform these functions by folding into diverse spatial configurations, by *varying* their configurations and in the long run, by varying their composition through the means of evolution.

This document proposes to model the protein probabilistically. The exponentially large degrees of freedom that have to be modeled entail the development of special purpose models that exploit the special properties of these molecules to efficiently encode their behavior. This document develops several structured models to perform these tasks.

First, we establish the basis for probabilistic modeling of proteins in Ch. 2 and the utility of graphical models in this task. This chapter shows the physical motivation behind such modeling and provides precedent for such models in the biophysics literature by describing them within the framework of graphical models. This chapter provides the motivation for the rest of the dissertation as well as the mapping from the output of subsequent models to

biophysically relevant properties of proteins.

Building on this, Ch. 3 describes a Probabilistic Graphical Model(PGM) to model proteins across spatial landscapes. This PGM models *both* backbone and side-chain configurational flexibility by modeling them as a multinomial distribution over one of (exponentially) many discrete configurations making it the first graphical model based approach to do so. The PGM compactly encodes the Boltzmann distribution over the spatial landscape around the native state, thereby allowing efficient (approximate) computations of its partition function. Ch. 4 and Ch. 5 describe the utility of this approach in two different applications: Bayesian model quality assessment in the context of structure prediction, and determining binding free-energies of Protein interactions. The former demonstrates that Bayesian estimates of model quality can out-perform traditional MAP estimates while the latter incorporates entropic contributions into computation of binding free energies of bio-molecular interactions thereby improving the accuracy by nearly 10%.

Ch. 6 proposes a new algorithm for learning the statistical patterns of protein sequences that utilizes recent advances in compressive sensing and sparse coding to accurately learn sparse graphical models of protein sequences. In Ch. 7 we study how this method could be used to learn generative models of protein families. We learn these models for very large protein families in a timely fashion by utilizing a distributed learning framework.

Finally, Ch. 8 develops an approach to approximate the properties of a Correlated Equilibrium in graphical games. Our approach, based on outer relaxations to the marginal polytope computes these approximations efficiently. We demonstrate that these relaxations are also remarkably accurate, often giving the exact solution. Using utilities computed using methods developed in Ch. 3 and Ch. 6, we then analyze the behavior of games arising in pathogen-drug interactions in Ch. 9. Our approach allows us to reason about possible future

mutations, the possible evolution of drug resistance, the efficacy of a set of drugs and design game-theoretically optimal drugs.

Chapter 2

Probabilistic models: A Biophysical interpretation

The thesis of this work is that proteins are efficiently and accurately modeled using structured probabilistic representations. At this juncture, it is natural to wonder if there is any particular reason for the emphasis on probabilistic representations. The aim of this chapter is to re-iterate the natural correspondence between the natural laws governing physical systems and certain probability distributions. Indeed, the hope is that by the end of this chapter, the reader would wonder if any representation *other than a probabilistic one* could be meaningful in this setting.

I start with a result that is the basis of statistical physics. It describes how the energies of particles are distributed under the laws of physics. The purpose of deriving this result is two-fold: (a) to remind the reader that there are fundamental reasons for probabilistic representations of proteins, and (b) to show that this result can be cast in more familiar terms. I then describe how graphical models allow for compact representation of such distributions.

Finally I will describe some previously published models that can be expressed as graphical models.

2.1 Boltzmann distribution from the laws of thermodynamics

Consider a closed classical system of N particles in thermal equilibrium (ie, the system is at constant temperature and the particles have been given time to "settle"). The first law of thermodynamics states that energy is conserved.

For simplicity of exposition, let us assume that there are a finite number of discrete (and distinct) energy levels $E_0, E_1, \ldots E_m$ in the system and an associated distribution $p(E) = [p_0, \ldots p_m]$ that describes the probability of a particle occupying a particular energy level. In such a system, the first law in conjuction with the conservation of mass can be stated as

$$\sum_i p_i E_i = E \tag{2.1}$$

The second law states the tendency of a system to attain the state of maximum entropy at equilibrium. This can be expressed as an objective function:

$$-\sum_i p_i \log(p_i) \tag{2.2}$$

Taken together, the two laws can be expressed as

$$\begin{aligned} \max \quad & -\textstyle\sum_i p_i \log(p_i) \\ \text{s.t.} \quad & \textstyle\sum_i p_i E_i = E \\ & \textstyle\sum_i p_i = 1 \\ & \forall i, p_i \geq 0 \end{aligned}$$

Theorem 1. ***The Boltzmann distribution*** *The solution to the above problem is of the form* $p_i = \frac{1}{Z}\exp(-\beta E_i)$ *where* Z *is a normalization constant and is commonly referred to as the partition function.*

Proof. Adding lagrange multipliers β, μ for the constraints, we have

$$\begin{aligned} \min A \quad &= \textstyle\sum_i p_i \log(p_i) - \beta(\sum_i p_i E_i - E) - \mu(\sum_i p_i - 1) \\ \text{s.t.} \quad & \beta \geq 0, \mu \geq 0, \forall i, p_i \geq 0 \end{aligned}$$

Taking derivatives,

$$\begin{aligned} \tfrac{\partial A}{\partial p_i} = \quad & 1 + \log(p_i) - \beta(E_i) - \mu = 0 \\ \tfrac{\partial A}{\partial \beta} = \quad & 0 \Rightarrow \textstyle\sum_i p_i E_i = E \\ \tfrac{\partial A}{\partial \mu} = \quad & 0 \Rightarrow \textstyle\sum_i p_i = 1 \\ \text{s.t.} \beta \geq 0, \mu \geq 0, \forall i, p_i \geq 0 x & \end{aligned}$$

This gives,

$$\begin{aligned} \tfrac{\partial A}{\partial p_i} = 0 \quad & \Rightarrow p_i = \exp(-\beta E_i - \mu - 1) \\ \textstyle\sum_i p_i = 1 \quad & \Rightarrow \exp(-\mu - 1) \textstyle\sum_i \exp(-\beta E_i) = 1 \\ \Rightarrow \exp(-\mu - 1) = \quad & \tfrac{1}{\sum_i \exp(-\beta E_i)} \\ \Rightarrow p_i = \quad & \tfrac{1}{\sum_i \exp(-\beta E_i)} \exp(-\beta E_i) \end{aligned}$$

which is the required result.

□

Some readers will recognize this as the "max-entropy" problem. Interestingly, the original use of this principle in statistics was due to Jaynes [1957, 1963] who argued on its use based on this natural correspondence between statistical physics and statistics. Much of this thesis will exploit the same correspondence, but in the reverse direction: from statistics to statistical physics and specifically to biophysics. This correspondence will allow us to use statistical inference algorithms to compute biophysical quantities and use statistical learning algorithms to *learn* a boltzmann distribution from data.

In exploiting this correspondence in this fashion, we follow Kumar et al. [1992], Neal [2005] and other modern workhorses in biophysics that trace their origin to statistical methods. The interested reader is referred to Neal [1993] (Sections 2.4 and 6.2) and Kollman [1993] for an elaboration of this subject.

2.2 Markov Random Fields as a representation of the Boltzman distribution

$E(\mathbf{c})$, the total energy of the configuration can be be written as $E(\mathbf{c}) = \sum_{i=1}^{a} \sum_{j=i+1}^{a} E_{\mathbf{c}}(i,j)$, the sum of $\binom{a}{2}$ individual pair-wise contributions if there are a atoms in the system.

If we are dealing with a protein of n residues, it will be easier to rewrite the above expression by grouping contributions from atoms according to the residues they belong to

$$E(\mathbf{c}) = \sum_{k=1}^{n} \sum_{l=k+1}^{n} E_{\mathbf{c}}(k,l) \tag{2.3}$$

where $E_{\mathbf{c}}(k,l) = \sum_{i=1}^{a_k} \sum_{j=1}^{a_l} E_{\mathbf{c}}(i,j)$ and a_k, a_l are the number of atoms in residues k, l respectively group the contributions from all atoms between a pair of residues into a single term – $E_{\mathbf{c}}(k,l)$ between residues k and l for example.

In general, not all $E_{\mathbf{c}}(k,l)$ will be non-zero. In fact, most of the $\binom{a}{2}$ terms will be negligible and can be assumed to be zero. If we imagine a graph between residues with edges connecting residues only when $E_{\mathbf{c}}(k,l) \neq 0$, then, we can rewrite Eq. 2.3 as

$$E(\mathbf{c}) = \sum_{e=(k,l)\in\mathcal{E}} E_{\mathbf{c}}(e) \tag{2.4}$$

The graph encodes the zeros of the interaction energies *in this configuration* $\boldsymbol{c}$ – an absence of an edge between a pair of residues implies that the energy of interaction between them when the protein is in configuration $\mathbf{c}$, is zero.

While we defined the graph for a specific configuration, it should be intuitive to see that multiple "close" configurations can share the same graph structure. This shared graph encodes the zeros of the interaction energies that are common across a set of configurations. In particular, GOBLIN , a method presented in Ch. 3 determines the shared structure for all configurations of the protein that have the same backbone. Given this shared structure over the set of configurations, we can go back to the Boltzmann distribution and model it over this set. The structure of the graph tells us which energies are zero and do not have to be computed.

2.3 Biophysical models as examples of the Graphical Model framework

In this section, we will describe a few published biophysical models and describe how these can be described using the Graphical Models framework. The aim of this section is to illustrate that the framework is fairly general, and that this insight can automatically yield algorithms for various tasks using these biphysical models.

2.3.1 Gaussian and Anisotropic Network Models

Gaussian Network models (GNM) and their subsequent extension, the Anisotropic Network Model (ANM) are examples of simple coarse-grained models that model the structural flexibility of the protein around its native structure. Both models share the assumption of a harmonic potential (ie. a gaussian model) of structural flexibility. I will describe the GNM model here; the ANM is an extension of the GNM which can be treated analogously.

The GNM models the fluctuation of the protein structure around the native "mean" structure. It is a coarse-grained representation that treats the protein as a spring-mass system with each amino acid being represented by a single point mass and interactions between nearby positions of the protein with elastic springs (hence the name). According to the GNM, the probability of a fluctuation $\Delta X = [\Delta X_1, \ldots \Delta X_n]$ is

$$P(\Delta X) \propto \exp(\sum_{(i,j)\in E} \Delta X_i \gamma \Delta X_j)$$

where γ is model parameter that models the spring constant. By comparing this with Eq. 1, it is easy to see that this can be represented as a Markov Random Field with a harmonic

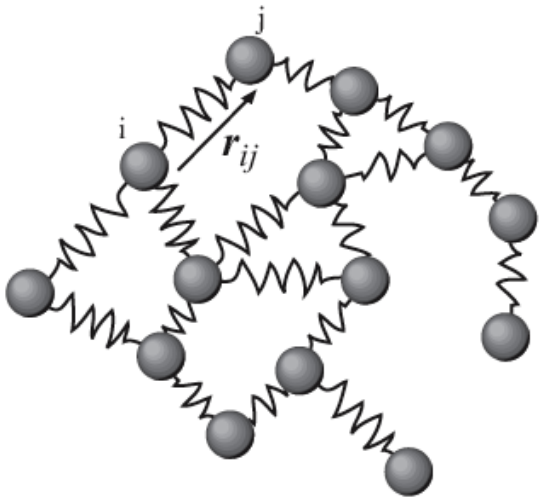


Figure 2.1: Elastic Network Models of proteins model the protein as a mass-spring system

potential. Edges E connect amino-acids between variables that are in contact with each other in the native structure of the protein. This corresponds to modeling the protein fluctuations as a multi-variate gaussian distribution with the precision matrix being defined by the contact matrix of the protein structure.

For a fluctuation $\Delta R = [\Delta X, \Delta Y, \Delta Z]$, the probability is given simply as $P(\Delta R) = P(\Delta X)P(\Delta Y)P(\Delta Z)$ (being "isotropic").

2.3.2 Computing pKa of proteins

The acid dissociation constant, pKa, of an acid is a quantitative measure of the strength of the acid. In proteins, the pKa of amino-acid side-chains play an important role in defining the pH dependent characteristics of a protein. Since the strength of an acidic (or basic) amino acid is affected by other neighboring acidic (or basic) amino acids nearby, the pKa of

the protein is dependent on how these interactions between residues affect the state of the protein. These interactions are naturally modelled as a graphical model. The probabilistic interpretation of the graphical model has a natural physical basis in terms of ionization free energies [Gilson, 1993, Bashford and Karplus, 1991].

The pKa of a protein is defined as follows:

$$pKa \propto -\beta \sum_{i=1}^{n} \sum_{x_i=0}^{1} x_i G_i + \sum_{i=1}^{n} \sum_{j>i}^{n} \sum_{x_j=0}^{1} x_i x_j G_{ij}$$

where x_i refers to the ionization state (neutral, or charged) and G_i, G_{ij} are contributions to the ionization energies.

This is an example of a binary-valued graphical model where the pKa is equal to the log partition function of this graphical model. Classical approaches to this problem include a naive mean field approach [Bashford and Karplus, 1990], brute-force summation with some pruning[Bashford and Karplus, 1991] and even Structured Mean Field techniques [Gilson, 1993].

The examples described in the previous two subsections develop graphical models that model proteins using a simplified force-field in a coarse grained fashion, and another that models a subset of the positions of the protein and only model their electrostatic interactions between them. In the following chapter we will aim to considerably generalize this approach by constructing graphical models that model all the atoms of proteins using a realistic force-field.

Chapter 3

Graphical Models of Protein Structures across Spatial Landscapes

A protein consists of some number of atoms across one or more polypeptide *chains*. A *configuration* of the protein corresponds to the geometry of each of its constituent atoms. While a protein is commonly represented as a single configuration (usually the crystalline form), at room temperature, a more accurate representation of the protein would be as an ensemble of configurations. We will adopt such a representation, by treating the configuration of a protein as a random variable $\mathbf{C}$ in some configurational space $\mathcal{C}$.

Boltzmann's law describes the probability distribution over $\mathcal{C}$ of a physical system at equilibrium; according to it, the probability of a configuration $\mathbf{c} \in \mathcal{C}$, $P(\mathbf{C} = \mathbf{c})$, with *internal energy* $E_{\mathbf{c}}$ is

$$P(\mathbf{C} = \mathbf{c}) = \frac{1}{Z} \exp\left(\frac{-E_{\mathbf{c}}}{k_B T}\right) \tag{3.1}$$

where $Z = \sum_{\mathbf{c} \in \mathcal{C}} \exp(-E_{\mathbf{c}}/k_B T)$ is the *partition function*, k_B is Boltzmann's constant, and

T is the absolute temperature in Kelvin.

Much of statistical physics is devoted to the study of the properties of this distribution since many physical properties of the protein can be expressed as functions of this distribution. The rate of a chemical reaction, for example, depends on the logarithm of the ratios of the corresponding partition functions.

Note that this distribution is over the entire configurational space $\mathcal{C}$, a space that is exponentially large in the size of the protein making this distribution too complicated to explicitly manipulate for proteins with a realistic size. In what follows, we will attempt to simplify this distribution by exploiting its properties around near-native equilibrium while reducing as little of the predictive power as possible.

We first partition the entire set of atoms into two disjoint sets – *backbone* and *side-chain* – since the nature of the assumptions we'll make in each set are very different. *Backbone atoms* (denoted by **b** here) refer to those that are common to all 20 amino acid types, while *side-chain* atoms (denoted by **r**) are those that differ among the different kinds of amino acids. $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots\}$, is a set of variables, one for each chain in the protein, representing the conformation of the backbone atoms, $\mathbf{r} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots\}$ where $\mathbf{r}_i$ represents the conformation of the side-chain atoms of residue i, and $E_{\mathbf{b}}, E_{\mathbf{b}}(\mathbf{r})$ represent energies of backbone **b** and side-chain **r** in backbone **b** respectively.

Using the fact that $\mathbf{c} = \{\mathbf{b}, \mathbf{r}\}$, $E_{\mathbf{c}} = E(_{\mathbf{b}}) + E_{\mathbf{b}}(\mathbf{r})$, we can now rewrite the joint distribution and the partition function as:

$$P(\mathbf{C} = \mathbf{c}) = P(\mathbf{B} = \mathbf{b})P(\mathbf{R} = \mathbf{r}|\mathbf{B} = \mathbf{b}) \tag{3.2}$$

$$Z = \sum_{\mathbf{b}} \exp(-\frac{E_{\mathbf{b}}}{k_B T}) Z_b \tag{3.3}$$

where $Z_b = \sum_{\mathbf{r}} \exp(\frac{-E_{\mathbf{r}}}{k_B T})$ is the partition function over the side-chain conformational space with a fixed backbone.

Given a specific backbone trace **b**, due to the nature of the physical forces in action, pairs of residues distally located according to trace are expected to exert very little direct influence on one another. In statistical terms, we say that such residues are independent of each other when conditioned on the event $\mathbf{B} = \mathbf{b}$. We will exploit these conditional independencies present in $P(\mathbf{R} = \mathbf{r}|\mathbf{B} = \mathbf{b})$ to compactly encode it as a Markov Random Field(MRF).

An MRF $\mathcal{G}$ is a probability distribution over a graph, and can be represented as a tuple $(\mathbf{X}, \mathcal{E}, \Phi)$, where the set of random variables in the multivariate probability distribution are the set of vertices – **S** and **B** in this case – while edges $e \in \mathcal{E}$ join residues that are directly dependent on each other and $\Phi = \{\phi_1, \phi_2, ..., \phi_m\}$ is a set of functions (popularly called factors) over random variables. *

The PDB structure is assumed to be one of the micro-states of the system at equilibrium. Atoms are allowed to deviate from their crystal structure coordinates. This is done by discretizing each degree of freedom and grouping atoms together according to side-chain rotamer libraries (e.g., Lovell et al. [2000]). Additional simplifying assumptions include fixing bond lengths and bond angles to idealized values or to the observed in the crystal structure. When discrete approximations are used, the MRF efficiently encodes an ensemble of micro-states of size $O(k^n)$, in $O(kn)$ space, where k is the average number of conformations per residue, and n is the number of residues in the protein.

In our model, the functions in $\mathcal{G}$ (i.e., ϕ_i) are defined in terms of a Boltzmann factor. That

*Since we use $\mathcal{G}$ to represent a conditional probability distribution, this is also referred as a Conditional Random Field(CRF). Since commonly used CRFs [Lafferty et al., 2001] are usually chain graphs, we use the more general term, MRF, to avoid confusion.

is, $\phi_i(\mathbf{r}_{\phi_i}) = \exp\left(-\frac{E(x_{\phi_i})}{k_B T}\right)$, where x_{ϕ_i} is the set of atoms that serve as arguments to ϕ_i, and $E(x_{\phi_i})$ is the potential energy of those atoms as defined by a molecular force field. In theory, any molecular force field can be used. We specifically use the ROSETTA potential $E_{Rosetta}$ that ROSETTA uses in computing $\Delta\Delta G$[Kortemme and Baker, 2002] which is composed of the following terms:

- E_{ljatr}, E_{ljrep}, the attractive and repulsive parts of a $6-12$ Lennard-Jones potential used to model van der Waals interactions.
- E_{sol}, the Lazardus-Karplus solvation energy that approximates the solvation energy by using an implicit solvent Lazaridis and Karplus [1999].
- E_{hb}, is the Hydrogen bond energy as computed by Kortemme et al. [2003]

$E_{Rosetta}$ is a linear combination $\mathbf{w}^T\mathcal{E} = w_{ljatr}E_{ljatr} + w_{ljrep}E_{ljrep} + w_{sol}E_{sol} + w_{hb}E_{hb}$. The vector $\mathbf{w}$ that defines the linear combination is typically learnt by fitting the energy terms to physical observations like $\Delta\Delta G$[Kortemme and Baker, 2002].

Fig. 3.1 illustrates the construction of $\mathcal{G}$ using a protein complex: Chymotrypsin complexed with the third domain of turkey ovomucoid(OMTKY). Fig. 3.1-A shows a single configuration $\mathbf{c}$ of the protein complex, with the residues in the Chymotrypsin chain and the OMTKY chain shown in different colors (red,blue respectively) for visual clarity. In contrast, the MRF $\mathcal{G}$ shown in Fig. 3.1-B models a *distribution* over all possible $\mathbf{r}$ for a given backbone trace. The construction of $\mathcal{G}$ is identical irrespective of whether the protein is a single chain, or multiple chains as in the case of a protein complex. What does change is the nature of the physical interactions being captured by the ϕ in $\mathcal{G}$ in each case. The potential terms in $E_{Rosetta}$ capture both the intra-molecular interactions (shown as solid lines), and the inter-molecular interactions(shown as dashed lines).

Given the structure of $\mathcal{G}$ and the potentials Φ as described above, we can rewrite the conditional distribution $P(\mathbf{R} = \mathbf{r}|\mathbf{B} = \mathbf{b})$ by using the Hammersley-Clifford theorem Clifford [1990] in the following manner.

$$P(\mathbf{R} = \mathbf{r}|\mathbf{B} = \mathbf{b}) = \frac{1}{Z_{\mathbf{b}}} \prod_{\phi_i \in \Phi} \phi_i(\mathbf{r}_{\phi_i}) \tag{3.4}$$

where once again we have the partition function $Z_{\mathbf{b}}$ associated with a specific backbone:

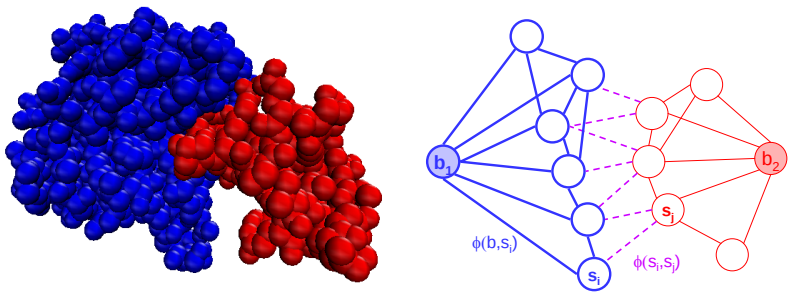
$$Z_{\mathbf{b}} = \sum_{\mathbf{R}} \prod_{\phi_i \in \Phi} \mathbf{r}_{\phi_i} \tag{3.5}$$

Notice that the due to the choice of the Boltzmann factor for Φs, this distribution is consistent with the Boltzmann distribution of Eq. 3.1. To obtain the joint distribution, one needs to simply multiply Eq. 3.4 with the probability of the particularly backbone conformation **b** according to Eq. 3.2.

Thus, the probability of a given state is simply the product of the functions, suitably normalized.

3.1 Probabilistic Inference and Free Energy Calculations

The *free energy* of a protein is a measure on the ensemble and is defined as: $G = H - TS$ where H is the *enthalpy*, or the expected internal energy, S is the entropy of the ensemble and T is the temperature of the system. The rates of biochemical reactions are determined by *changes* in free energies. For example, ΔG_{bind}, the *binding free energy*, is the change in free energy when two proteins, A and B, bind: $G_{AB} - (G_A + G_B)$. It determines the rate of association of A and B. Often, the quantity of interest in tasks such as protein design is



(A) (B)

Figure 3.1: (A) Chymotrypsin complexed with the third domain of turkey ovomucoid (OMTKY). While protein structures are often shown as a single conformation, in reality they occupy ensembles of conformations; our method models both side-chain and backbone ensembles. (B) Part of an MRF encoding the conditional distribution over the ensembles of conformations. Blue nodes with thick lines correspond to Chymotrypsin, red nodes with thin lines correspond to OMTKY. Solid lines refer to intra-molecular interactions, and dashed lines refer to inter-molecular interactions. The nodes labeled $\mathbf{b}_1, \mathbf{b}_2$ represent the conformation of the backbone atoms in the two chains, while the nodes labeled $\mathbf{r}_i$ and $\mathbf{r}_j$ corresponds to conformations of side-chain atoms. (Since all the variables in the graph represent conformations, for visual clarity, we omit the X in their labels.)

$\Delta\Delta G$, the change in the ΔG value upon mutation from wild-type: $\Delta G_{mutant} - \Delta G_{wild-type}$. For example, a beneficial mutation results in a better ΔG for the variant than for the wild-type, and thus a negative $\Delta\Delta G$. Computing G therefore is extremely useful in determining the properties of the protein.

The free energy of a physical system is related to the Boltzmann distribution by way of the partition function: G is simply $-k_B T \log Z$. The MRF model from the previous section provides a compact representation of the Boltzmann distribution that enables us to compute a quick and good approximation to the binding free energy by solving a statistical inference problem to compute Z.

Evaluating the product in Eq. 3.4 is straightforward for any given configuration of the random variables. Computing the partition function in Eq. 3.5, on the other hand, is computationally intractable in the general case [Dagum and Chavez, 1993] because it involves sum-

ming over every state. However, a number of rigorous approximation algorithms have been devised for performing inference in MRFs. Significantly, it has been shown that mathematically, these algorithms are equivalent to performing free-energy approximations [Yedidia et al., 2005]. This is not surprising, because inference and free energy calculations both require estimating a partition function. What is surprising, however, is that some existing inference algorithms are mathematically equivalent to specific free-energy approximations introduced by statistical physicists (e.g., Bethe [1935], Kikuchi [1951], Morita [1991], Morita et al. [1994]). For example, it is now known that Pearl's *Belief Propagation* (BP) algorithm [Pearl, 1986] is equivalent to the Bethe approximation [Bethe, 1935] of the free energy. Unless otherwise specified, we use Belief Propagation for inference in the following sections.

The term 'belief' in both BP refers to the marginal distributions over the random variables in the MRF. Briefly, each node in the graph keeps track of its own marginal probability distribution (i.e., belief). Belief Propagation algorithms start with random initial beliefs, and then use message passing between nodes to converge on a final set of beliefs. Informally, each node updates its own beliefs based on the beliefs of its neighbors in the graph, and the value of the potential function, Φ. When the algorithm converges, the final beliefs can be used to obtain the partition function (or an approximation thereof), and hence a free energy. If the MRF happens to form a tree (i.e., a graph with no cycles), Belief Propagation is exact and takes $O(|\mathcal{E}|)$ time, where $|\mathcal{E}|$ is the number of edges in the graph. The MRFs considered in this proposal, however, are not trees and have $O(|\mathcal{V}|)$ edges. In this case, we use a closely related algorithm known as Loopy Belief Propagation. Loopy BP is not guaranteed to converge, but has always done so in our experiments.

Using Loopy Belief Propagation on the MRF that encodes the conditional distribution,

we can obtain an estimate of the partition function of the conditional distribution $Z_{\mathbf{b}}$ for each backbone configuration $\mathbf{x_b}$ as we previously did for folding free energies [Kamisetty et al., 2008, 2007]; Z, the partition function over $\mathbf{x_c}$ can then be computed using Eq. 3.3.

3.2 Discretization

We now briefly discuss a subtle, yet important issue that we have glossed over so far in our presentation: the effects of discretizing the conformational space $\mathcal{C}$.

The assumption of a discrete rotamer library is fairly well-founded, cf. Canutescu et al. [2003], Ponder and Richards [1987], McGregor et al. [1987]. While a common use of such rotamer libraries is in performing side-chain placement, i.e. finding the single most energetically favorable side-chain conformation $\mathbf{x_r}$ [Yanover and Weiss, 2002, Xu, 2005, Kingsford et al., 2005, Canutescu et al., 2003], these rotamer libraries have also been used in computing free energies and conformational entropies of protein structures [Koehl and Delarue, 1994, Kamisetty et al., 2007, 2008, Lilien et al., 2005].

This approach of using a set of discrete rotameric states to compute the entropy faces a subtle problem. To understand this, let us consider an imaginary protein with exactly one residue whose side-chain atoms are unconstrained in configurational space, i.e. the energy $E_{\mathbf{r}}$ of the side-chain atoms is the same, no matter what configuration they are in.

This protein has a physically measurable amount of entropy $S_{physical}$. Now suppose we discretized the configurational space into n points, each representing an equal fraction of the space. It is clear that in this scenario, the probability of each rotamer will be equal to $1/n$(and indeed, this is what Belief Propagation would predict). The free energy in our discrete model, $H - TS$ equals $< E_{\mathbf{r}} > -T \sum_{i=1}^{n} -\frac{1}{n} \log(\frac{1}{n})$ = $E_{\mathbf{r}} - T \log(n)$. In other

words, as the granularity of the discretization increases, the discrete entropy increases and as $n \longrightarrow \infty$, $S \longrightarrow \infty$ and is completely unconnected to $S_{physical}$.

This problem arises in many scenarios, most notably for our purposes, in information-theoretic treatments of statistical physics [Jaynes, 1963, 1968]. Fortunately, a solution to this problem is available, which to the best of our knowledge is due to E.T. Jaynes [Jaynes, 1963]. By using a measure (i.e. a possibly unnormalized probability distribution) m over the configurational space and replacing the discrete entropy by the relative entropy $S = -\sum_{\mathbf{X_r}} P(\mathbf{X_r}) \log \frac{P(\mathbf{X_r})}{m(\mathbf{r})}$, we now obtain a quantity that behaves correctly in the limit. To use this for our purposes, we point out that the rotamer library we use [Canutescu et al., 2003] provides such a measure m_{dun} for each rotamer which we utilize.

Our earlier treatment of inference can be modified to use the relative entropy instead of the discrete entropy by observing that

$$S = -\sum_{\mathbf{X_r} \in \mathcal{C}} (P(\mathbf{X_r}) \log P(\mathbf{X_r}) - P(\mathbf{X_r}) \log m(\mathbf{X_r}))$$

and therefore, $G =$

$$\begin{aligned} &\sum_{\mathbf{X_r}} P(\mathbf{X_r}) E_{\mathbf{r}} + \sum_{\mathbf{X_r}} (P(\mathbf{X_r}) \log P(\mathbf{X_r}) - P(\mathbf{X_r}) \log P(m(\mathbf{X_r}))) \\ &\qquad = \sum_{\mathbf{X_r}} P(\mathbf{X_r})(E_{\mathbf{r}} - \log m(\mathbf{X_r})) - \sum_{\mathbf{X_r}} P(\mathbf{X_r}) \log P(\mathbf{X_r}) \end{aligned}$$

In other words, the move from the discrete entropy to the discrete relative entropy can be made by adding a $E_{rot} = -\log m(\mathbf{X_r})$ term to the energy function. A similar problem arises when summing over multiple backbone traces according to Eq. 3.3: by increasing the number of backbone traces, the value of Z monotonically increases. Again, this can be

easily fixed by assigning a fraction of volume of the conformational space to each trace. For our experiments we assume that the conformational space is uniformly sampled by the traces, i.e. each trace represents an equal volume of the conformational space.

Chapter 4

Applications: Model Quality Assessment

The protein structure prediction problem is one of the most challenging unsolved problems in Biology. Informally, it is the task of computing the three dimensional structure of a protein, given its chemical description as a sequence of amino acids. Knowing the three dimensional structure of a protein can provide deep insights into its working, and the mechanisms of its interaction with the environment. This can be used, for example, in the design of new drugs and bio-sensors. Unfortunately, despite significant progress, experimental methods (i.e., X-ray crystallography and Nuclear Magnetic Resonance) to *determine* protein structures still require months of effort and $O(\$100K)$ — per protein. Therefore there has been a lot of focus on *in-silico* approaches to Protein Structure *Prediction*.

Given a protein sequence s, a common feature of structure prediction algorithms is the ability to (stochastically) generate a large number of putative models, $\mathbf{b}_1, \mathbf{b}_2, \ldots \mathbf{b}_n$, and then assess the quality of each of these models using a ranking algorithm. Extant algorithms rank by first computing the optimal set $\mathbf{r}_{i_{opt}}$ of parameters $\mathbf{r}_i$ and then computing a point estimate $E(\mathbf{b}_i, \mathbf{r}_{i_{opt}}, \mathbf{s})$ of the model quality. A natural question to ask, is if a Bayesian estimate of

model quality can be computed efficiently. Such an estimate involves computing an integral over all possible $\mathbf{r}_i$, a computation that is infeasible to perform exactly for protein structures.

There are a variety of computational techniques for *estimating* this integral in the structural biology community. The more accurate amongst them require extensive sampling or molecular dynamics simulations (e.g., Alder and Wainwright [1959]), which can take hours to days on real-proteins, making them infeasible for the task of *in-silico* Protein Structure Prediction. Faster coarse-grained methods exist, e.g., Muegge [2006], but it has been argued [Thomas and Dill, 1994] that they are not accurate enough.

In contrast to these techniques, we estimate the integral for each $\mathbf{b}_i$ by first discretizing $\mathbf{r}_i$ and then performing approximate inference on a discrete Markov Random Field constructed over $\mathbf{r}_i, \mathbf{s}$ as described in Sec. 3.1. Hyper-parameters of our model are learnt by minimizing a loss-function for ranking over training data, using gradient descent.

Our results on a database of *in-silico* models for 32 proteins show that moving from a point estimate to a Bayesian estimate improves the accuracy of ranking by multiple criteria: the average rank of best model improves from nearly 27 to less than 7, the model ranked first is significantly better in quality, and the rank correlation improves by nearly 0.3 over point estimates.

4.1 Learning to Rank (log) Partition Functions

Given models $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots \mathbf{b}_n\}$, and a ranking (permutation over $1 \ldots n$) $\mathbf{y}$ for these models, the learning task involves finding a function G that computes a numerical score for each model that minimizes some loss-function $\mathcal{L}$ between G and $\mathbf{y}$ on $\mathbf{B}$. Many approaches have been developed for the task of learning to rank, especially in IR tasks like document-

retrieval [Herbrich et al., 2000] and web-search [Joachims, 2002]. These tasks differ in their choice of the loss function $\mathcal{L}$ and the algorithms used to minimize it. While initial approaches to ranking approached the ranking problem as a large number of pair-wise classifications [Herbrich et al., 2000, Joachims, 2002], recent approaches have shown the utility of using loss-functions based on the entire rank, or the so-called "list-wise" approaches [Cao et al., 2007, Xia et al., 2008]. Further, a "soft" approach to ranking [Burges et al., 2005] has allowed the use of gradient-based continuous optimization techniques instead of combinatorial optimization.

We use a "list-wise" soft-ranking approach to ranking since it has been shown to have good performance. We study the properties of two loss-functions: the negative log-likelihood and the cross-entropy. Each loss-function is minimized by a quasi-Newton method. Details of these metrics are available in our paper [Kamisetty and Langmead, 2009].

4.2 Results

We studied the efficacy of our approach on a database of 32 proteins selected from Wroblewska and Skolnick [2007]. For each protein, this database contains a set of 50 plausible models generated *in-silico* and the actual structure of the protein ("native"). Each of them contain $\mathbf{b}$ and $\mathbf{r}_{opt}$ and an associated "distance" – the root mean square displacement (RMSD) in Å (angstroms) between the coordinates of the atoms in the model to the native. This dataset covers all four classes of proteins according to the CATH classification, and the models for each protein cover a large fraction of the model space – very close (< 2 RMSD) to very distant (> 10 RMSD).

We split this database of 32 proteins into five, randomly generated, equal-sized train/test

Table 4.1: Rank of Native and Quality of Best Decoy across 5 test sets

Method	Rank of Native (out of 51 total)	Best - closest (RMSD)
Point Estimate	26.75	3.413
G	8.05	1.85
G-NegLL	21.19	3.71
G-Cross Entropy	6.60	1.743

partitions. Thus, each training set contains 16 proteins, containing the native and 50 plausible models for each of these proteins, along with the RMSD of these models to the native. Each test set contains 16 proteins containing 50+1 (in-silico+native) models which have to be ranked based on their quality. Ideally, the native should always be ranked first, along with the models in increasing order of RMSD to ground truth.

We compared the performance of four methods: (i) a point estimate obtained by computing $E_{Rosetta}$ on the optimized parameters, (ii) a Bayesian estimate G obtained using default hyper-parameters, and the Bayesian estimates of G using the learnt hyper-parameters with the two loss functions, which we shall refer to as (iii) G-neg LL and (iv) G-Cross Entropy.

Tab. 4.1 compares the average rank of the native structure across the 5 test sets and the average difference between the RMSD of the best *in-silico* model as predicted by the method, to the RMSD of the actual closest model. The average rank of the native structure is significantly improved by moving from a point estimate (26.75) to a Bayesian estimate (8.05). Further, by optimizing the hyper-parameters using the cross entropy loss function, this can be further improved to 6.6.

The average difference in RMSD between the predicted best model and the actual closest model is significantly reduced, from 3.4 Å to nearly half its value – 1.85 Å using G and 1.74 Å using G-Cross Entropy. Surprisingly, optimizing the hyper-parameters using the

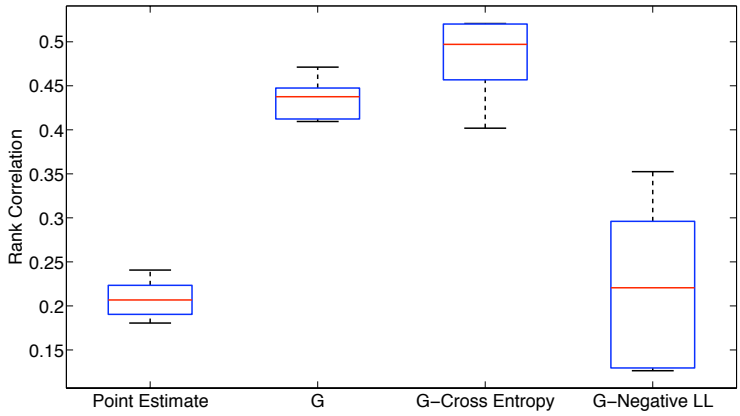


Figure 4.1: Pearson's rank correlation coefficient between predicted ranking and actual ranking for the four methods listed.

likelihood loss function is almost as bad as the point estimate, indicating that the likelihood loss function isn't suitable for this task. We believe that this is due to the fact the likelihood function neglects the RMSD information while computing the likelihood of a ranking. In a data-scarce setting such as ours, this could lead to a significant difference in the optimal solution.

While the rank of the top structure and the quality of top prediction are important metrics of performance, the quality of the overall ranking is also important. This is because, often, the models are iteratively generated, ranked, and selectively refined. Thus, it is important that the ranking is reasonably accurate at all positions.

To measure this, we compute the rank-correlation of the ranks with the ranks obtained by ranking according to RMSD from native. Fig. 4.1 shows the rank-correlation of the ranks computed in this manner. Again, it can be seen that the performance improves significantly by moving from a point-estimate to a Bayesian estimate, and learning using the cross-entropy loss function improves it further.

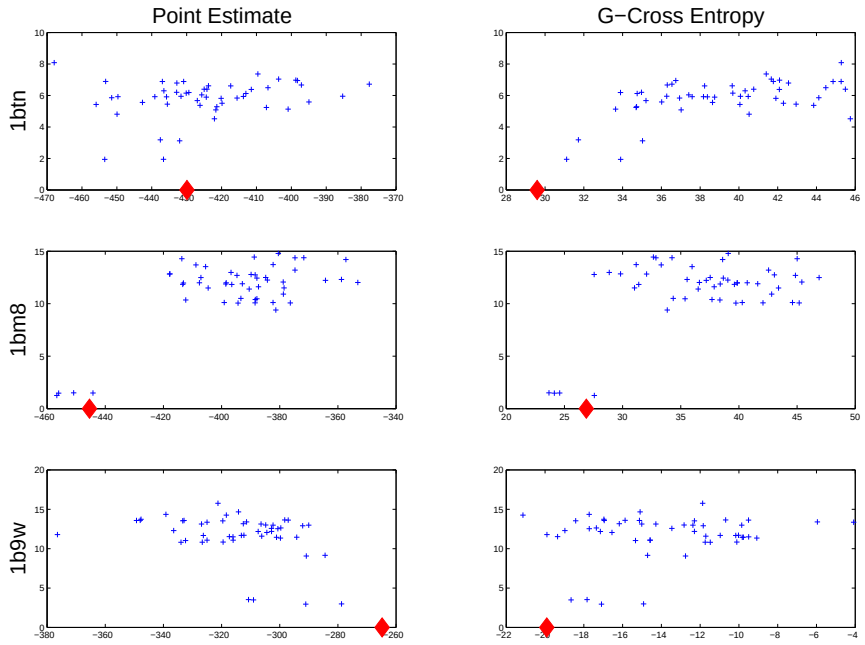


Figure 4.2: Scatter plots showing ranking for three different proteins in a test set, using the point estimate and the Bayesian estimate. The x-axis is the value of the corresponding estimate, while the y-axis shows the RMSD from the ground truth (shown with a red diamond)

Fig. 4.2 shows the values of the point estimates (first column) and the Bayesian estimates learnt using the cross-entropy (second column), for three protein structures (rows) in a particular test set. The native structure in each of these proteins is shown as a red asterisk while the 50 in-silico models are shown in blue.

These three proteins were selected to show the different types of behavior in learnt ranking. In 1btn, the first protein, using the Bayesian estimate the native structure is ranked correctly, the best in-silico model is ranked next and there is a good correlation between G and the RMSD. In the second protein, 1bm8, the native structure is not ranked the best, but

the best in-silico model is correctly identified. However, there is no strong correlation between RMSD and G for the distant models. Notice also, that in this case, the point estimate performs almost as well. In the third dataset, while the native structure is not ranked at the top, its rank is significantly better than using the point estimate. However, the model closest in RMSD is neither ranked well, nor is the top ranked structure of good quality. It must be noted that while there is variability in the ranking performance across the datasets, in all these cases, there is an improvement in results due to the Bayesian approach.

Chapter 5

Applications: Computing Binding Free Energies

5.1 Modeling Protein-protein Interactions with GOBLIN

Protein-protein interactions are essential to the molecular machinery of the cell; transient or persistent complexes mediate processes including regulation, signaling, transport, and catalysis. While coarse-grained, high-throughput techniques such as yeast two-hybrid [Fields and Song, 1989] are primarily focused on *which* proteins interact, finer-grained techniques based on structural analysis address questions of *how* and *why* these interactions occur. By modeling the physical interactions between constituent atoms, structure-based approaches provide deeper insights into, for example, the specificity of an interaction or its sensitivity to various mutations (e.g., [Weber and Harrison, 1999, Wang and Kollman, 2001]). In addition to answering questions of interest to basic science, such methods are also well-suited to designing variants with improved or novel properties [Kortemme and Baker, 2004, Lilien

et al., 2005, Joachimiak et al., 2006].

A fundamental law of thermodynamics states that interactions are governed by *binding free energies*. Free energy is a property of the *entire* configuration space and consists of both enthalpic and entropic contributions. There are a variety of techniques for estimating free energies computationally, and each method makes a different trade-off between computational efficiency and fidelity to the underlying physics.

This chapter introduces a method, called GOBLIN (*Graphical mOdel for BiomoLecular INteractions*), that lies between the extremes of detailed molecular dynamics and statistical potentials. GOBLIN models the energy landscape for a protein-protein complex as a probability distribution over an exponentially large number of configurations. This distribution is compactly encoded using an undirected probabilistic graphical model known as a Markov Random Field (MRF). Under this model, internal energies are calculated using standard atomic-resolution force-fields, and rigorous binding free energy calculations are performed using Belief Propagation [Pearl, 1986]. Our method runs in a few minutes and this therefore is significantly faster than MD. At the same time, GOBLIN is more rigorous than statistical methods.

GOBLIN significantly extends the state of the art of all-atom graphical models of proteins, including our own previous work [Kamisetty et al., 2007, 2008] in this area. Our earlier work focused on computing intra-molecular *folding free energies* for fixed backbone configurations, while GOBLIN is a technique for computing inter-molecular *binding free energies* under backbone flexibility. Additionally, in order to account for the particularities of the binding free energies of the system under study, we solve a non-linear optimization problem to fit the force-field parameters to predict the partition function correctly.

MRFs of apo and holo forms. The binding free energy of a protein complex is the difference between the free energies of the apo (unbound) and the holo (bound) forms. To compute this, it is therefore necessary to model both the apo and the holo forms. Thus, for a complex involving molecule A and B, we construct three separate MRFs: (i) one for the holo form, (ii) one for A in isolation, using the backbone of A from the holo form, and (iii) one for B in isolation, using the backbone of B from the holo form.

MRFs of mutants. Separate MRFs are constructed for each mutation considered. This is done by performing an *in silico* mutation to the PDB structure, and constructing a new MRF accordingly.

Data Preparation

The atomic coordinates for each complex were obtained from the PDB. Hydrogen atoms were then added using the REDUCE software program Word et al. [1999]. In order to compute $\Delta\Delta G$, we also need the structures of the individual partners. As is common in high-throughput approaches (e.g., Kortemme and Baker [2002], Guerois et al. [2002]), we assumed that the native backbone in the complex is also a good approximation for the apo and holo backbones of the engineered proteins. Thus, at the end of this process, we have generated plausible structures for the apo and holo forms of the engineered structures.

Backbone ensembles for the complexes were generated using the `-backrub` Smith and Kortemme [2008] option of Rosetta. The method performs independent Monte Carlo simulations with "generalized-backrub" moves and selects the lowest energy structure found in each simulation. When using the backrub option, we allowed all residues whose C_α atoms were within 6 Å of the mutated position (the distance suggested by Smith and Kortemme

[2008]) and ran 10^4 Monte Carlo steps within each simulation.

Learning force field parameters against free energies

A molecular mechanics force-field consists of: (i) a defined set of atom types; (ii) a function defining the internal energy of the system; and (iii) a set of parameters. It is common Guerois et al. [2002], Kortemme and Baker [2002], Benedix et al. [2009] to take the atom types and energy functions as fixed, but to adjust the parameters for a particular type of study.

A commonly used strategy for optimizing force field parameters is to minimize the sum of the squared errors between predicted and experimentally measured internal energies using fixed structures. In contrast, we consider the problem of minimizing the sum of of the squared errors in free energies, as computed using MRFs. The two problems are fundamentally different. In particular, whereas minimizing differences in internal energies gives rise to a simple linear regression problem, minimizing differences in free energies is a complicated non-linear regression problem involving the minimization of a functional (i.e., the partition function). We developed a novel algorithm to solve this problem in an efficient albeit approximate manner.

Given a training set of experimentally measured $\Delta\Delta G$ values for N mutants of that complex, along with the wild-type ΔG, consider the problem of learning force-field parameters to minimize the mean square error (MSE) between predicted and observed $\Delta\Delta G$. We do so by adjusting the vector of weights $\mathbf{w} = [w_{ljatr}, w_{ljrep}, w_{hbond}, w_{rot}, w_{sasa}, w_{coop}]$ with which we linearly combine the corresponding force-field terms.

In referring to the different observations and predictions, let us use superscripts e for experimental and p for predicted, and subscripts $i \in \{1, \ldots, N\}$ for the various datapoints

This allows us to express the MSE as

$$mse = \frac{1}{N}\sum_{i=1}^{N}(\Delta\Delta G_i^p - \Delta\Delta G_i^e)^2 \tag{5.1}$$

To minimize MSE subject to $\mathbf{w} \succeq 0$ by gradient descent, we must compute the gradient ∇mse:

$$\nabla mse = \left[\frac{\partial mse}{\partial w_{ljatr}}, \frac{\partial mse}{\partial w_{ljrep}}, \ldots, \frac{\partial mse}{\partial w_{coop}}\right] \tag{5.2}$$

$$\forall w, \frac{\partial mse}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} 2\left(\Delta\Delta G_i^p - \Delta\Delta G_i^e\right)\left(\frac{\partial \Delta\Delta G_i^p}{\partial w}\right) \tag{5.3}$$

Using $\frac{\partial G_{conf}}{\partial w} = \langle E \rangle_{\mathbf{c}}$ where $\langle E \rangle_{\mathbf{c}}$ is the expected value of the corresponding force-field terms over all $\mathbf{x_c}$ using the current value of w, and the fact that the derivative of differences is just the difference of derivatives, we have:

$$\frac{\partial \Delta\Delta G^p}{\partial w} = \begin{cases} \frac{\partial \Delta\Delta G^p_{conf}}{\partial w} = \Delta\Delta\langle E_i \rangle_{\mathbf{c}}, & \text{for } w \in \{w_{ljatr}, w_{ljrep}, w_{hbond}, w_{rot}\} \\ \Delta\Delta SASA, & \text{for } w_{sasa} \\ I_{wt}, & \text{for } w_{coop} \end{cases} \tag{5.4}$$

respectively.

In the case of a fixed backbone, the expectation $\langle E \rangle$ is over just $\mathbf{x_r}$. In both cases the expectations, and thus the gradient, can be computed along with the free energy during inference. Given this method of computing gradients, we performed gradient descent, updating the weights $\mathbf{w}$ at iteration i using the following equation $\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \eta \nabla mse$ where the step size η was set to 0.05. To test for the sensitivity of our results to η, we experimented

on one training set with various values between 0.01 and 0.15. We found that the step size affected the rate of convergence but had negligible effect on the final values indicating that the quality of results aren't sensitive to this parameter.

Given the enthalpies and free energies of each backbone trace, we can compute the free energy of the entire distribution and its derivatives.

5.2 Results

We studied the importance of entropy on a database of 704 single-point mutants from eight large and well studied complexes. For each of these mutants, the database contains the $\Delta\Delta G^e$, the experimental change in binding free energy upon mutation. The details of the datasets, along with the Protein Data Bank (PDB) Berman et al. [2000] ids of the wildtype complexes, are shown in Table 5.1. Of these, the three largest datasets (wildtype PDB ids: 1sgr, 1cho, 1ppf) are from the Kazal family of serine protease inhibitors Lu et al. [2001] while the rest of the interactions are part of an Alanine-scanning database previously used in Kortemme and Baker [2002] and Kortemme et al. [2003]. We note that the amount of thermodynamic data available for protein-protein interactions is limited, and the database we considered is among the largest of its kind.

The atomic coordinates for each complex were obtained from the PDB and converted into probabilistic graphical models (PGM), which are used to perform free energy calculations. Briefly, if $\mathbf{X} = \{X_1, ..., X_n\}$ is a vector encoding the conformation of the protein (or protein complex), each PGM encodes the distribution $P(\mathbf{X})$ using a factored representation. By construction, the probability of any particular conformation is inversely proportional to the exponential of its internal energy, as computed using a molecular mechanics force field.

Table 5.1: **Datasets for mutant protein-protein complexes.**

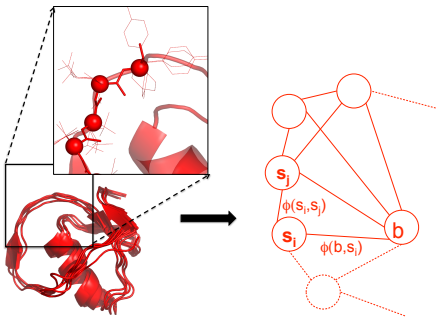
Wt PDB id	Partner A	Partner B	# residues in A and B	# mutants in A, B
1sgr	OMTKY	SGP B	236	150,0
1cho	OMTKY	Chymotrypsin	291	170,0
1ppf	OMTKY	Human LE	274	170, 0
1a22 (AS)	HGH	HGHBP	429	34,29
1gc1 (AS)	CD4	GP120	920	49,0
1dan (AS)	BCF VII-A	TF	587	20,23
1bxi (AS)	E9 Dnase	IM 9	212	30,0
3hfm (AS)	HYHEL	HEL	558	12,13

"AS": alanine-scanning experiments

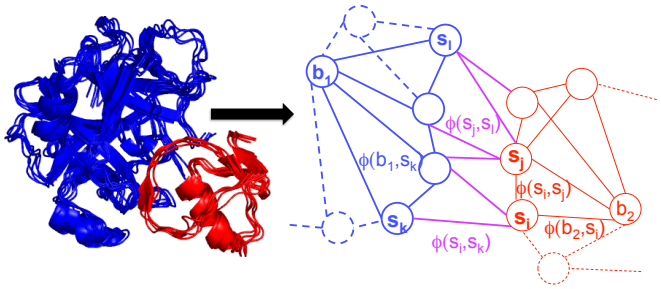
Interaction energies fall off quickly with distance leading to conditional independencies in the Boltzmann distribution.* GOBLIN takes advantage of these conditional independencies in its factorization, which leads to a compact encoding of the joint distribution (Figure 5.1). This factorization also leads to an efficient means for performing free energy calculations, and for optimizing force field parameters against experimentally observed $\Delta\Delta G$ (see Methods).

For each complex $C = AB$ consisting of proteins A and B, we construct three separate PGMs (Figure 5.2). The first two PGMs model the Boltzmann distribution over the apo (unbound) conformations of A and B, and the third models the Boltzmann distribution over the holo (bound) conformation. Loopy Belief Propagation Pearl [1988] is then performed on each PGM to compute the free energies G_A, G_B, and G_C (i.e., before and after binding). The free energy of binding is computed as: $\Delta G = G_C - (G_A + G_B)$. Similarly, the change in binding free energy upon mutation, $\Delta\Delta G$, is computed by performing an *in silico* mutation, repeating the binding free energy calculation, and computing the difference:

*Two random variables X and Y are said to be conditionally independent, given Z, iff $P(X, Y|Z) = P(X|Z)P(Y|Z)$.



(A)



(B)

Figure 5.1: **Graphical models of protein complexes.** (A) Turkey ovomucoid third domain. (Left) Ensemble of backbones, with inset showing ensemble of side-chains for one backbone. (Right) Graphical model of the backbone and side-chain ensembles. For visual clarity, only the subscripts of the random variables are shown. The node labeled b corresponds to a random variable over the backbone ensemble, while the remaining nodes correspond to random variables over rotameric side-chain conformations. Edges capture intra-molecular interactions (vdW, hydrogen bonds, etc.) with ϕ functions according to a molecular mechanics force-field. The graphical model encodes a Boltzmann distribution over conformations in terms of the ϕ functions. Dashed nodes and edges represent a subset of the positions and interactions in the rest of the protein that GOBLIN models but have been omitted in this figure for simplicity. (B) Complex of chymotrypsin with turkey ovomucoid third domain. (Left) Ensemble of backbones. (Right) Graphical model. It combines the inhibitor model (red) with an analogous model for chymotrypsin (blue), and introduces inter-molecular edges (purple) with ϕ functions for inter-molecular interaction terms. This model encodes a Boltmann distribution over complex conformations.

$\Delta\Delta G = \Delta G_{mut} - \Delta G$. $\Delta\Delta E$ values were computed using max-product Belief Propagation that approximates the global minimum energy conformation (GMEC).

We considered two scenarios. In the first scenario, the PGM models the Boltzmann distribution over side-chain conformations, conditioned on a fixed backbone. Side-chains conformations are modeled using the backbone specific rotamer library described in Canutescu et al. [2003]. The second scenario models the Boltzmann distribution over both side-chain and backbone conformations. Alternative backbone conformations were obtained using the BACKRUB method Smith and Kortemme [2008].

Computation of interaction energies employs a molecular mechanics force-field with weights on different terms; GOBLIN learns those weights from a training set of $\Delta\Delta G$ data. We thus formed 20 random partitions of the 704 mutants into 352 training structures and 352 testing structures. The training structures were used to optimize the parameters of GOBLIN's force-field. The optimized parameters are shown in Table 5.2. We note that the parameters are optimized to maximize predictive performance, and not necessarily for biophysical interpretability. However, our optimized parameters are at least in part comparable to those in existing force fields. Our weight for the Lennard-Jones term, for example, is comparable to the weight used by Rosetta when modeling discrete rotameric states. Similarly, the weights on the $SASA$ and the I_{wt} terms are comparable in magnitude to Benedix et al. [2009]. The differences between our weights and other force fields is likely due to both differences in training data and the fact that our method models entropy explicitly, which leads to a different objective function. Our weight for hydrogen bonding is lower than expected. We note, however, that our method models solvent using terms proportional to the SASA. The SASA term, in turn, partially accounts for hydrogen bonding between solvent atoms and solvent-protein atoms. Thus, the weight on the hydrogen bond term may be lower than might be

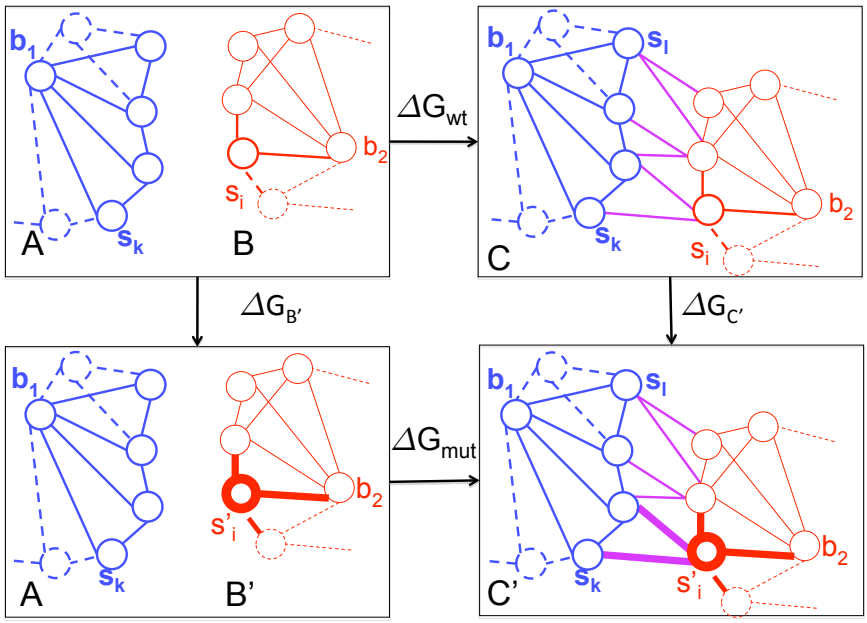


Figure 5.2: **Free energies in graphical models of protein complexes.** (Top-left) Two separate graphical models, A and B, encoding the wild-type apo forms of two proteins. (Top-right) A graphical model, C, encoding the wild-type complex. Binding free energies are obtained by computing the free energies of the three models, G_A, G_B, and G_C, and then calculating $\Delta G = G_C - (G_A + G_B)$. (Bottom-left, bottom-right) Graphical models of corresponding mutant forms. The mutated position and the interactions that are affected by it are shown in thick lines. While the energetic effect (via these interactions) is local, the entropic effect can be distal. GOBLIN accounts for both effects by performing variational inference to compute $\Delta G_{A'}$ and $\Delta G_{C'}$. $\Delta\Delta G$s are obtained by computing the binding free energy of the mutant, $\Delta G_{mut} = G_{C'} - (G_{A'} + G_B)$, and then calculating $\Delta\Delta G = \Delta G_{mut} - \Delta G$.

expected if the solvent was modeled explicitly.

Table 5.2: **Learned Force-field Parameters**

Name	Learned Value
w_{ljatr}	0.46
w_{ljrep}	0.70
w_{hb}	0.11
w_{rot}	0.23 kcal mol^{-1}
w_{sasa}	0.027 kcal mol^{-1} A$^{\circ -2}$
w_{iwt}	0.0006 kcal mol^{-1} A$^{\circ -2}$

Parameters corresponding to terms from ROSETTA's force-field are dimensionless since the corresponding force-field terms already have units of energy.

The trained model was used to predict the binding free energies for the test structures, and to identify the GMEC. In what follows, all errors are reported as averages over the 20 partitions. For comparison, we also used the programs FOLDX (version 3.0) and ROSETTA (version 2.3) to compute binding free energies.

Changes in Entropy Upon Binding

We first consider the nature of the changes in entropy caused by binding, and the effects of mutations on those changes. GOBLIN can compute detailed information on the changes in entropy (and enthalpy) for each residue. For example, Figure 5.3 illustrates the change in marginal over side chain configurations upon binding for residue Trp 304 in the HGH-HGHBP complex. Notice that the dominant rotamer has just under 50% of the probability mass before binding and that the rest of the mass is primarily distributed among four additional rotamers. After binding, there is a substantial shift in probability mass; the dominant rotamer now has almost 80% of the probability mass, and most of the rest of the mass is

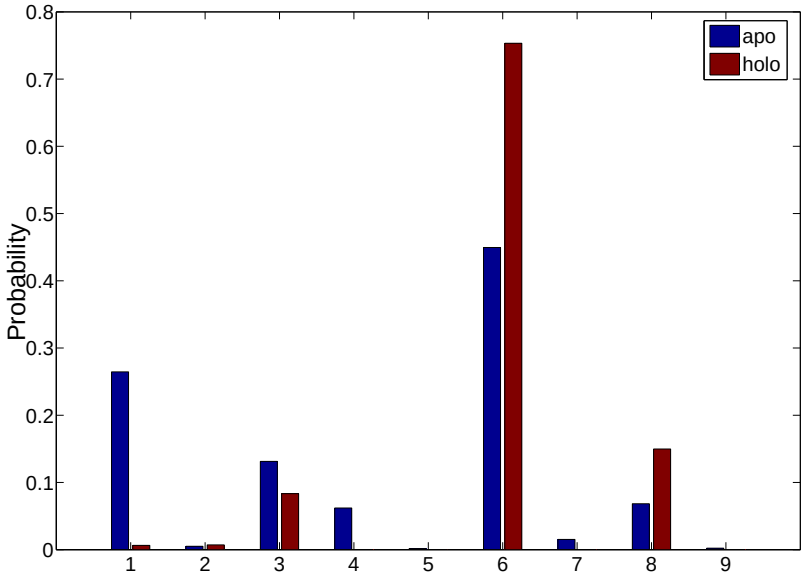
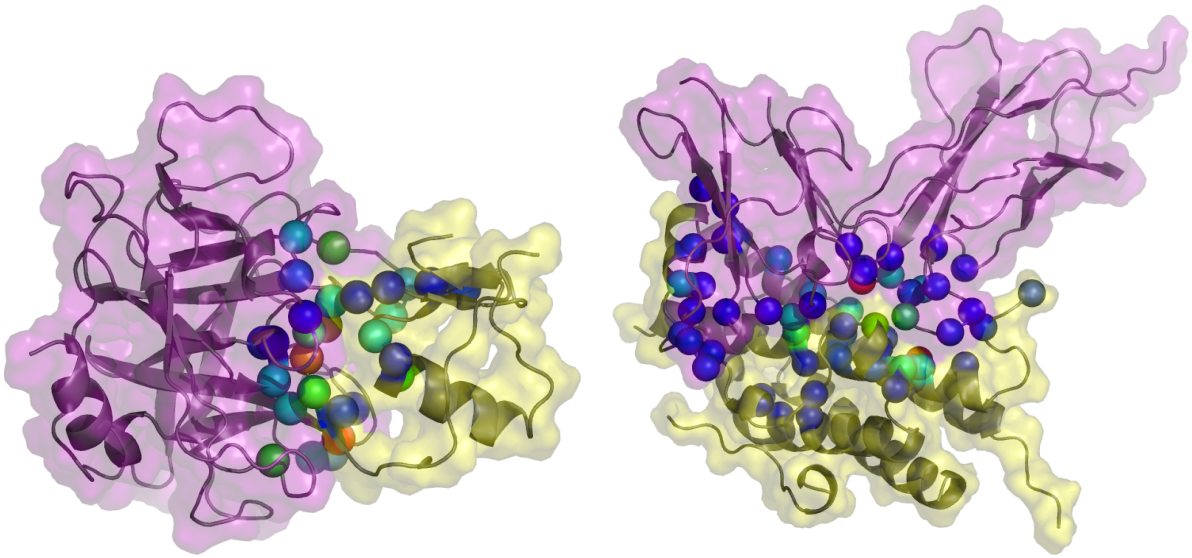


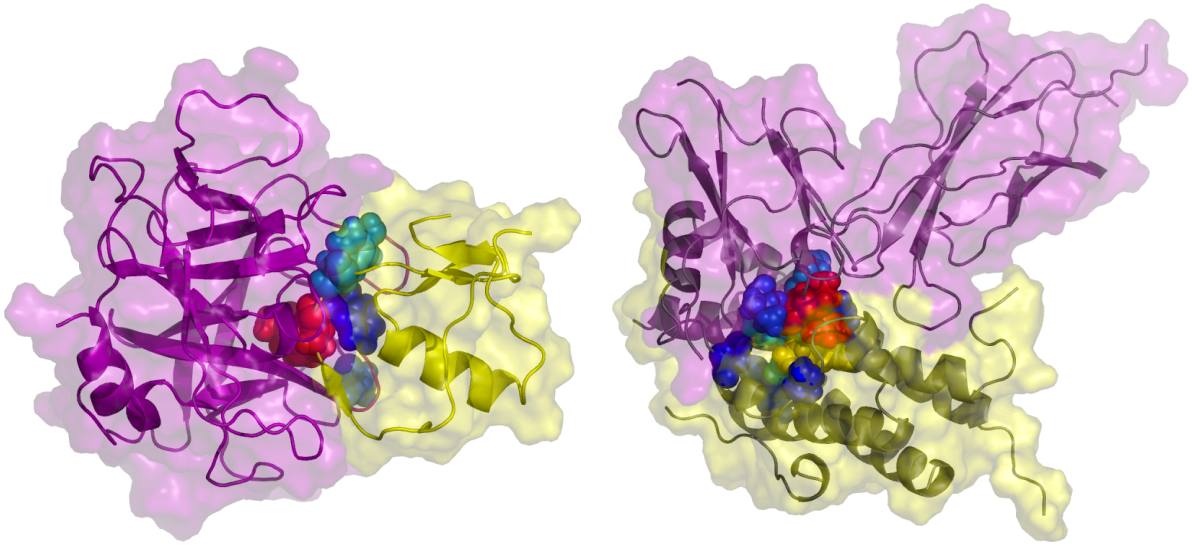
Figure 5.3: **Effect of rotameric-occupancy due to binding.** Change in rotameric probabilities for Trp 304 in the HGH-HGHBP complex. Trp has 9 rotamers in the rotamer library we use, 3 for each of $\chi_1 = \{60°, -180°, 60°\}$ respectively. The blue bars show the rotameric occupancies for the apo structure while the red bars show the occupancies in the holo structure. Upon binding, the probability mass redistributes.

distributed among two additional rotamers. Figure 5.4-A visualizes GOBLIN's predicted change in entropy upon binding for the wild-type Human Leukocyte Elastase : Turkey Ovomucoid and the Human Growth Hormone : Human Growth Hormone Binding Partner complexes. In these figures, the surfaces of the two partners are shown in purple and yellow respectively. Spheres mark the C_α atoms of residues that show a non-trivial change (absolute change in entropy ≥ 0.1 k_B units) in their entropy, with red spheres for the largest change and blue for the smallest. Not surprisingly, all the interface residues showed large decrease in entropy. More interestingly, the decrease in flexibility in the interface affects the entropy of neighboring residues. In the HLE-OMTKY complex, Ser 214 of HLE showed lower entropy in the holo form than the apo form, despite being > 10 Å away from the interface. In the HGH-HGHBP complex, these distal effects were stronger: Trp 86 and Glu 373 of HGH and Glu 373 of HGHBP (distances to interface: 17.2 Å, 12.9 Å and 12.9 Å resp.) all showing a non-trivial change in binding despite being far away from the interface. These changes in entropy, which are unfavorable, are compensated by a corresponding decrease in enthalpy, to make the binding favorable.

Figure 5.4-B shows the difference in binding entropy upon mutation (i.e., $\Delta\Delta S$) for one mutation from each of these two complexes: L18H of OMTKY and D171A of HGH. Again, the surface colors represent the two partners. All atoms of the residues showing a non-trivial binding entropy difference with respect to the wild-type ($\Delta\Delta S \geq 0.1$) are shown in spheres. Of the distal residues from panel A, Trp 86 of HGH and Ser 214 of HLE show a non-trivial change in entropy. In the former case, the change in entropy is actually positive (i.e., there is less entropic cost to binding in the alanine mutant than the wild-type) while in the latter case the change in entropy is negative. These results demonstrate that the distal entropic effects on binding of a mutation can be different from those in the wild-type, underlining



HLE-OMTKY (wild-type) HGH-HGHBP (wild-type)



HLE-OMTKY (L18G mutant) HGH-HGHBP (D171A mutant)

Figure 5.4: **Localized evaluation of change in entropy.** (Top) Change in entropy upon binding; (Bottom) Change in entropy upon mutation. (Left) HLE-OMTKY, wild-type and with L18G mutation; (Right) HGH-HGHBP, wild-type and with D171A mutation. The surface color distinguishes the partners. Spheres mark C_α atoms of residues whose marginal entropy changes by more than $0.1k_B$ (yielding $< 10\%$ of the residues).

the need to determine them accurately.

Quantitative Analysis

We next consider the quantitative accuracy of the free energy predictions, and the relative importance of side-chain and backbone conformational entropies.

Effects of Side-chain Entropy

The quantitative accuracies of GOBLIN under the fixed backbone scenario are presented in Table 5.3. The row labeled "GOBLIN" reports the root mean squared errors (RMSE) between prediction and observation for our method. The row labeled "GOBLIN-E" is the RMSE obtained when $\Delta\Delta G_{conf}$ is replaced with $\Delta\Delta E_{conf}$ — the change in internal energy for the GMECs in the apo and holo forms. The row labeled "GOBLIN-H" is the RMSE when $\Delta\Delta G_{conf}$ is replaced with $\Delta\Delta H_{conf}$, the expected energy averaged over the Boltzmann distribution, computed by neglecting the entropic component of the free energy computed by Belief Propagation. GOBLIN's RMSE is 1.6 kcal/mol, which is 9% lower than GOBLIN-E ($p < 0.05$), and about 12% lower than GOBLIN-H ($p < 0.01$). The drop in RMSEs persists when the 5th and 10th percentile of errors are removed (final two columns), suggesting that the difference in accuracies is robust to outliers. We conclude that entropic contributions play a significant role in protein-protein interactions, because ignoring them results in a significant increase in RMSE. We discuss the errors in more detail below.

Table 5.3 also compares GOBLIN's RMSE with that for the programs FOLDX and ROSETTA. GOBLIN outperforms FOLDX by nearly 10% ($p < 0.03$), and ROSETTA by 36% ($p < 6.8 \times 10^{-7}$). Significantly, GOBLIN continues to have lower RMSEs after removing each approach's least-accurate predictions, suggesting that the difference in accuracies

Table 5.3: $\Delta\Delta G$ **(kcal/mol) root mean squared errors.**

Method	Overall RMSE (std err)	95% RMSE (std err)	90% RMSE (std err)
GOBLIN	1.63 (0.06)	1.27 (0.05)	1.10 (0.04)
GOBLIN-E	1.80 (0.06)	1.40 (0.05)	1.22 (0.04)
GOBLIN-H	1.85 (0.09)	1.42 (0.08)	1.22 (0.07)
FOLDX	1.82	1.42	1.20
ROSETTA	2.54	1.92	1.68

Root mean squared error (RMSE) for GOBLIN, FOLDX, and ROSETTA. GOBLIN-E and GOBLIN-H refer to the RMSE for the GMEC and enthalpy, respectively. The values for GOBLIN and its variants are cross-validated test errors with the standard errors for these estimates reported in parentheses. The final two columns are the RMSE after the 5% and 10% worst outliers have been removed, respectively.

is robust to outliers. In particular, when the 5th (resp. 10th) percentile of errors are removed, GOBLIN outperforms FOLDX by 11% (resp. 9%) ($p < 0.01$; resp. $p < 0.08$), and outperforms ROSETTA by 34% (resp. 35%) ($p < 1.5 \times 10^{-5}$; resp. $p < 2.2 \times 10^{-4}$). Notice that the RMSE of FOLDX, which uses a knowledge-based approximation of the change in entropy, is approximately the same as GOBLIN-H, which ignores entropy altogether. These results suggest that GOBLIN's variational approach to free energy calculations is superior to the knowledge-based methods used by FOLDX and ROSETTA. The difference in accuracy is likely due to the fact that the variational approach considers not only the direct effects of each mutation on the free energies, but also the indirect effects on neighboring residues. Indeed, the very nature of Belief Propagation involves diffusing information throughout the graphical model.

Figure 5.5 shows a scatter plot comparing GOBLIN's predictions using the weights selected by cross-validation on the entire dataset. The correlation coefficient (R^2) across the entire dataset was 0.56. Outlier elimination improved this substantially, to 0.66 without the

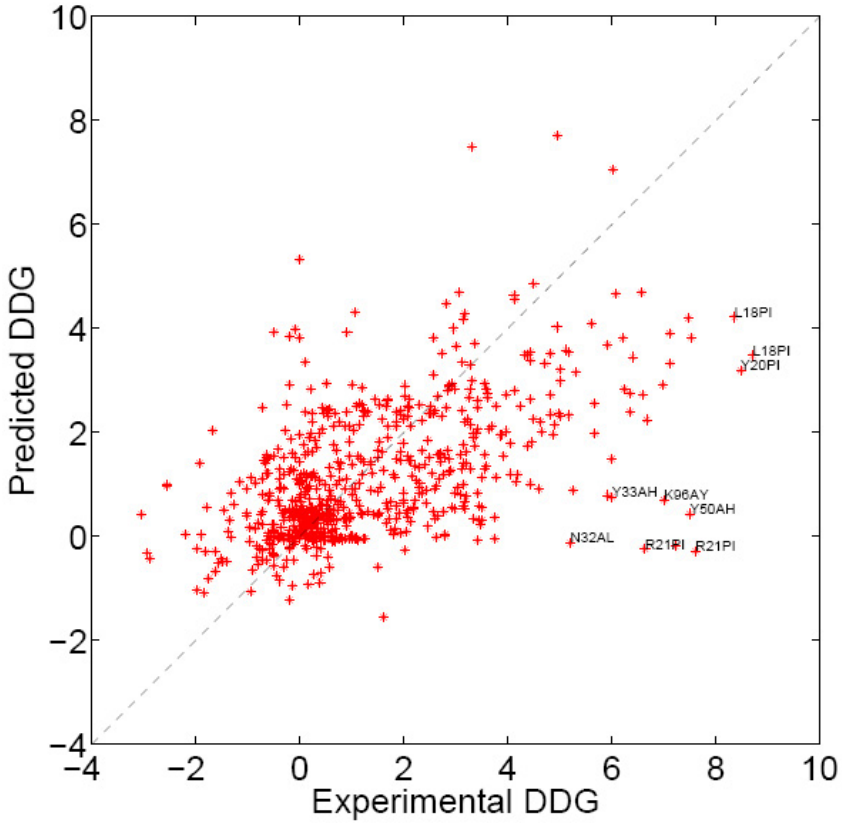


Figure 5.5: **Scatter plot comparing** GOBLIN**'s predictions with experimental values**. The correlation coefficient (R^2) was 0.56 across the entire dataset. The nine worst outliers are labeled with the mutation and chain id. When the worst 5% (resp. 10%) outliers are removed, $R^2 = 0.66$ (resp. $R^2 = 0.70$).

top 5% outliers and to 0.70 without the top 10% outliers. In comparison, the correlation coefficient for FoldX was 0.52 and increased to 0.62 (resp. 0.67) after removing the top 5% (resp. 10%) outliers. The corresponding values for Rosetta were 0.0, 0.06, and 0.15, respectively.

Errors for different residue types: Figure 5.6 (top and middle) show boxplots of the error in prediction according to mutant and wild-type amino acids respectively. In each box, the central red line is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually with red '+' marks. Note that not all 20 amino acids were mutated in our dataset: some (Cys, Gly) weren't mutated at all and a mutation from Met occurred only once, while mutations from Lys and Glu each accounted for about 12% of all mutations. The median signed error for most mutant amino acid types is close to zero, with a few notable exceptions, namely Pro, Lys, Asp, and Glu. Of these, mutations to and from Pro produced the highest errors. Indeed the three largest RMSEs in the entire data set were all proline mutations. This is to be expected since proline has an atypical backbone and a mutation to it can cause significant structural changes. The median signed error for mutations to lysine was nearly 2.0 kcal/mol and mutations to aspartic acid and glutamic acid had a larger spread in errors than other amino acids. These errors were likely caused due to their charged nature, and the fact that our force field does not presently account for electrostatics. Across the entire data set, GOBLIN's error on charged mutants was larger than on neutral residues, indicating the greater difficulty in modeling their interactions.

Errors due to change in charge or volume: Tables 5.4 and 5.5 stratify the errors in terms of the change in residue charge and volume after mutation, respectively. Relative to a global RMSE of 1.6 kcal/mol, certain kinds of mutations yield larger than average RMSEs, includ-

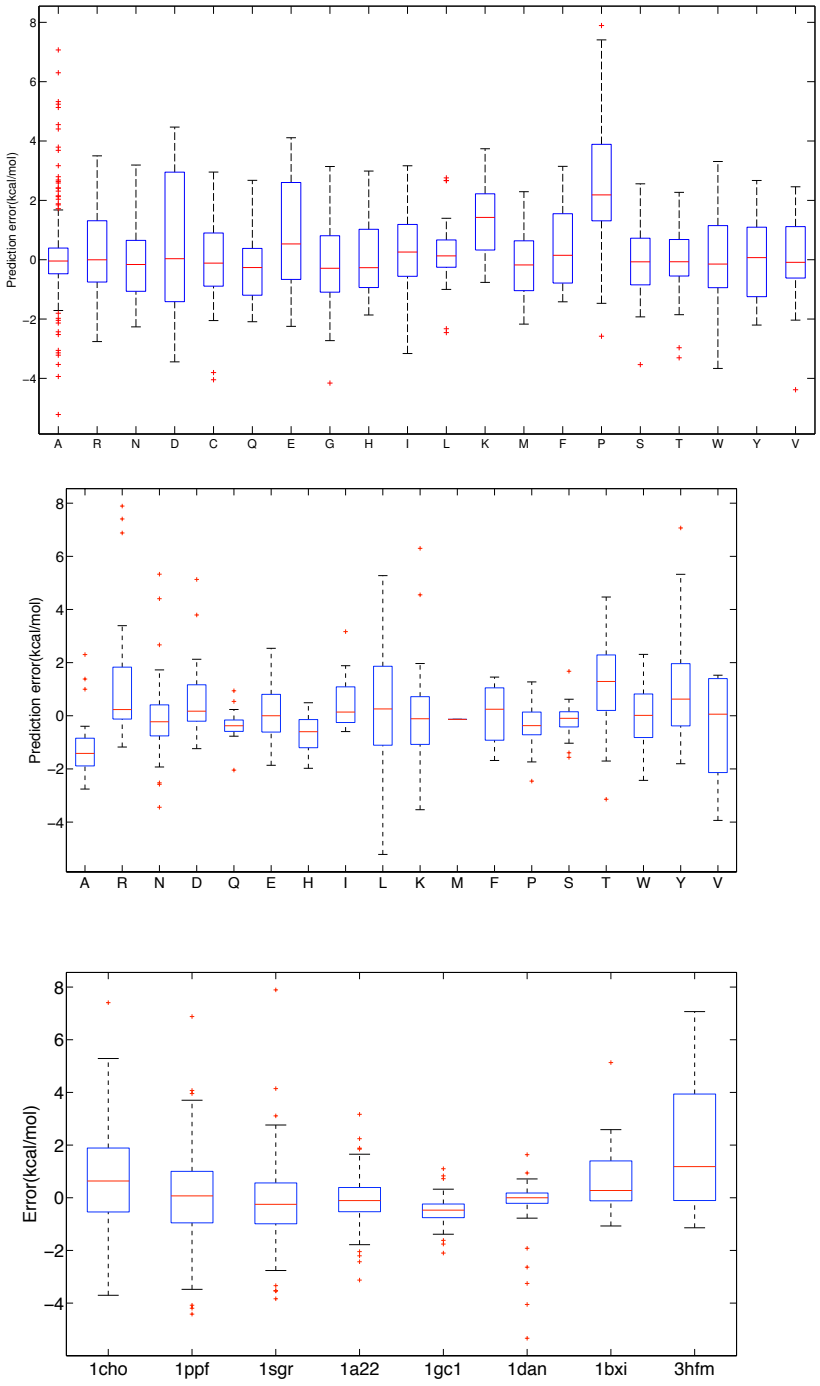


Figure 5.6: **Characterization of prediction error**. (Top) According to mutant residue type; (Middle) according to wild-type residue type; (Bottom) according to complex (in decreasing order by number of mutations). See text for an explanation of box plots. The error is actual minus predicted, so positive indicates an under-prediction, while negative means an over-prediction.

Table 5.4: **RMSEs (kcal/mol) according to charge of amino acid: negative (D, E), neutral (A, N, C, Q, G, H, I, L, M, F, P, S, T, W, Y, V), and positive (R, K). Values in brackets indicate net change in RMSE upon incorporating backbone flexibility.**

Wild-Type \Mutant	Negative	Neutral	Positive
Negative	1.71 [0.02]	1.20 [0.05]	1.38 [0.03]
Neutral	2.40 [0.04]	1.56 [0.11]	1.85 [-0.21]
Positive	1.57 [0.22]	1.80 [0.05]	0.96 [0.3]

Table 5.5: **RMSEs (kcal/mol) according to volume of amino acid: small (A, G, S), medium (N, D, C, Q, E, H, I, L, K, M, P, T, V), and large (R, F, W, Y). Values in brackets indicate net change in RMSE upon incorporating backbone flexibility.**

Wild-Type\Mutant	Small	Medium	Large
Small	0.75 [0.05]	1.57 [-0.29]	2.04 [-0.24]
Medium	1.47 [0.04]	1.64 [0.01]	1.45 [0.18]
Large	1.81 [0.35]	2.22 [0.05]	1.16 [0.44]

ing: mutations from one negatively charged residue to another; mutations from a neutral residue to a charged residue; and mutations from a positively charged residue to a neutral residue. These errors may reflect the fact that GOBLIN's force field, like ROSETTA's, does not explicitly account for electrostatic interactions other than hydrogen bonds. Mutations from small to large residues, and from large to either small or medium size residues are also associated with an increase in RMSE (Table 5.5). This is to be expected given that our backbones were held fixed for these experiments, and so no change is made to account for unfavorable packings. As explained subsequently, the error in such cases improves upon incorporating backbone flexibility.

Errors in different complexes: Figure 5.6-bottom shows the breakdown of GOBLIN's performance across the eight datasets listed in Table 5.1 arranged in decreasing order of

number of mutants. In four of the eight complexes, GOBLIN's error is around 1.5 kcal/mol or smaller. The largest error is in 3hfm where the RMSE is nearly 3.0 kcal/mol, which is marginally better than FOLDX's RMSE, and previously published results using ROSETTA Kortemme and Baker [2002]. One possible reason for such behavior might be due to conformational changes with distal effects, as suggested by Pons et al. [1999]. Additionally, the three programs (GOBLIN, FOLDX, and ROSETTA) assume that the apo backbones of the proteins are similar to their holo forms. When this assumption is violated, no program is expected to perform well, suggesting that it may be necessary to minimize the structure of the apo forms, or use apo forms deposited in the PDB.

Table 5.6: **Outliers**

Mutant	Error (kcal/mol)	Complex	Possible Reasons
R21PI	-7.68	1sgr	Mutation to proline; solvent interactions
R21PI	-7.37	1cho	Mutation to proline; solvent interactions
Y50AH	-7.08	3hfm	Large-scale rearrangement Pons et al. [1999]
R21PI	-6.88	1ppf	Mutation to proline; solvent interactions
K96AY	-6.29	3hfm	Large-scale rearrangement; loss of salt-bridge Pons et al. [1999]
N32AL	-5.47	3hfm	Large-scale rearrangement Pons et al. [1999]
Y33AH	-5.46	3hfm	Large-scale rearrangement Pons et al. [1999]
D51AA	-5.21	1bxi	Loss of strong electrostatic interaction Wallis et al. [1998]
L18PI	-5.10	1cho	Mutation to proline; possible destabilization of complex

Outliers: Table 5.6 lists GOBLIN's largest outliers (absolute error $\geq$ 5 kcal/mol). Four of the nine involve a mutation to proline from arginine in the serine protease inhibitor. Prolines have an atypical backbone, and often result in a substantial change in backbone configuration. Our means for sampling backbones does not, in general, handle such changes well. Of the remaining five outliers, four are mutants to the HyHEL-10 Fab-lysozyme complex (3hfm), a system that is known to undergo large scale re-arrangements upon binding. Here,

our assumption that the apo and holo backbones are approximately the same is inappropriate. Moreover, one of these mutations, K96A, has been postulated to result in a loss of a salt bridge Pons et al. [1999]. The force-field we currently use does not capture such interactions. The final outlier involves the loss of a strong electrostatic interaction, which, as previously mentioned, is not presently implemented in GOBLIN's force field.

Effects of Backbone and Side-Chain Entropy

We next consider PGMs modeling Boltzmann distributions over both side-chain and backbone conformations. Our expectation was that we would see a further reduction in RMSE, especially for disruptive mutations (e.g., those involving a large increase in the size of the side chain). We generated a set of nine alternative backbone conformations using a Backrub-like method developed by Kortemme and others Smith and Kortemme [2008] that is implemented in Rosetta. The method runs independent Monte-carlo simulations with "generalized backrub" moves and selects the lowest energy structure from each simulation. Along with the native backbone, this gave us an ensemble of ten backbones which is the size of the ensemble used by Smith and Kortemme [2008], Friedland et al. [2008] in their backrub studies. We then re-optimized the parameters and re-computed RMSEs in a cross-validated fashion.

Surprisingly, as shown in Figure 5.7, incorporating backbone flexibility did not change the prediction significantly in most cases. As expected, the error on incorporating disruptive mutations (small $\rightarrow$ large) decreases. However, on some other mutations, incorporating backbone flexibility tends to increase RMSE slightly rather than decrease it (Tables 5.4 and 5.5). The overall test error using backbone flexibility increased slightly from the rigid backbone case, albeit not significantly ($p = 0.11$), to 1.69 kcal/mol. This is still less than

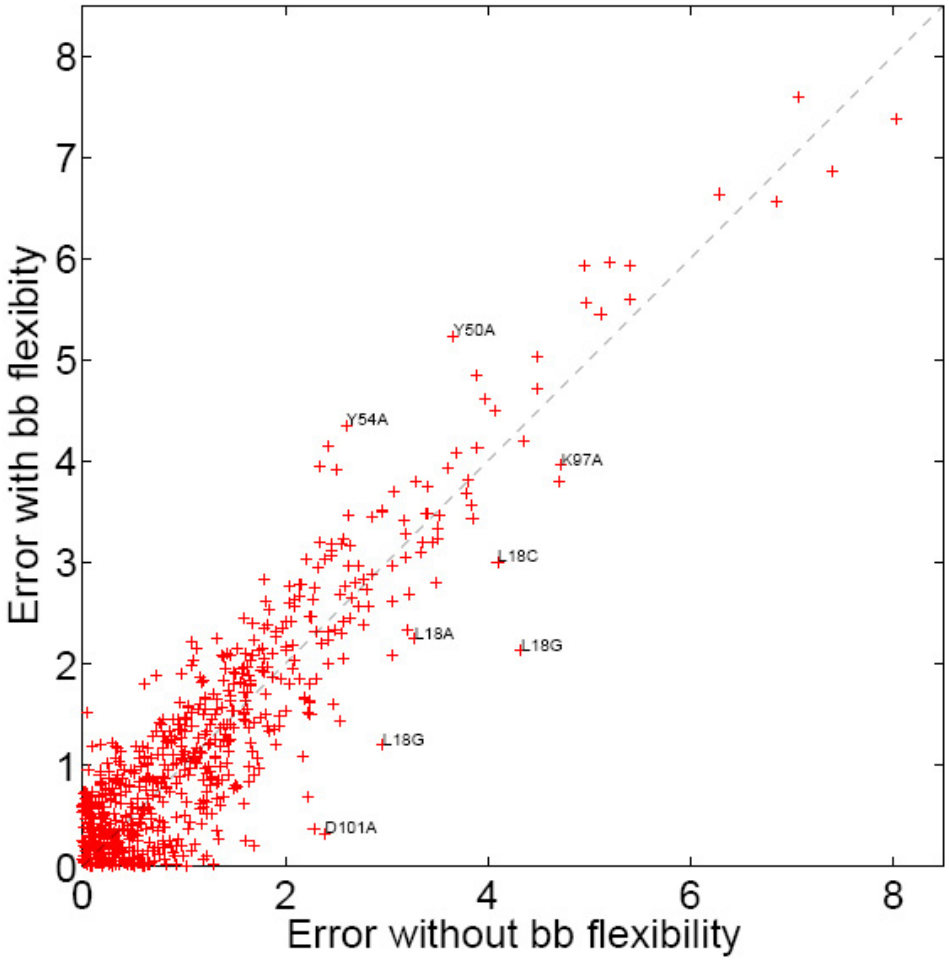


Figure 5.7: **Incorporation of backbone flexibility.** GOBLIN prediction error with (y-axis) and without (x-axis) backbone flexibility. Marked outliers indicate wild-type amino acid type, residue position and mutant amino acid type.

the RMSEs of FOLDX and ROSETTA. We note that our data is dominated by mutations to alanines (a small residue). GOBLIN tends to underestimate binding free energy in mutations from a large residue to a small one, and does so to a greater degree when accounting for backbone flexibility. This partially explains the increase in RMSE.

There are three cases where backbone flexibility does tend to decrease RMSE (Table 5.5): when the wild-type residue is small, and the mutant is either medium or large, or when the wild-type has neutral charge, and the mutant is positively charged. These reductions in RMSE are due to more favorable enthalpies made possible through alternative backbones, as opposed to an entropic contribution. There are some notable exceptions to this trend, as seen in Figure 5.7. For example, the incorporation of backbone flexibility in mutations L18G of the SGP B : OMTKY complex and D101A of the HYHEL : HEL complex lead to more than 2 kcal/mol reduction in error. In both cases, the improvement was caused by accounting for the increase in backbone entropy due to the mutation.

Further analysis from Molecular Dynamics simulations: To understand the cause of the outliers in our predictions, we investigated one of our largest outliers: R21P mutant (outlier) of the chymotrypsin : inhibitor complex (pdb id 1cho) by performing Molecular Dynamics simulations on the wild-type and the mutant in explicit solvent. The molecular dynamics simulations on the wild-type complex revealed a potential salt bridge between Arg 21 and Asp 35, and a hydrogen bond between the backbone of Arg 21 and Phe 41. Upon mutation to proline, the simulation revealed that both of these bonds were lost, resulting in a loop displacing further away from the serine protease inhibitor's hydrophobic pocket. This movement resulted in additional waters entering the binding site, further destabilizing the complex. This particular combination of changes (backbone conformational changes due to proline mutation, the loss of a salt-bridge, and solvent effects) explains why GOBLIN

underestimates the change in free energy. In particular, the BACKRUB-generated backbone ensemble did not provide adequate sampling, GOBLIN's force-field doesn't account for electrostatics and doesn't explicitly model solvent.

To demonstrate the importance of backbone sampling, we replaced the BACKRUB-generated backbones in the graphical model with the 3,000 backbones generated via MD. The modified graphical model thus encoded a Boltzmann distribution over the side chains and this larger backbone ensemble. We then re-computed the free energies. We note that the complexity of performing this calculation scales linearly with the number of backbones, and is trivially distributed across a computer cluster. More importantly, there was a dramatic drop in error — from ≈ 8 kcal/mol to ≈ 1 kcal/mol. We note, however, that only 10 of the MD-generated backbones had substantial probability mass. This suggest that it may be necessary to generate a significantly larger backbone ensemble for such outliers. As a control experiment, we performed additional Molecular Dynamics simulations for L18GI, a mutation where GOBLIN was fairly accurate in its prediction when using the BACKRUB ensemble ($\Delta\Delta G^e$: 6 kcal/mol, $\Delta\Delta G^p$: 5.2 kcal/mol). On this mutant, the prediction did not change appreciably ($\Delta\Delta G^p$: 5.4 kcal/mol). This highlights the importance of obtaining a well-sampled ensemble of backbones when performing free energy calculations of protein-protein complexes and suggests that when given such a set of backbones, GOBLIN predictions can be substantially more accurate.

5.3 Extensions to Model Protein-Ligand Interactions

While the previous sections describe the results of using GOBLIN to model protein-protein interactions, the framework itself is much more general and can model biomolecular interac-

tions in general provided an appropriate discretization of conformational states is available.

To demonstrate this framework, we extended GOBLIN to model protein-ligand interactions. We implemented AutoDock 4.2sHuey et al. [2007] force-field to model Protein-ligand interactions. Intra-protein interactions were modeled using the Rosetta force-field as previously. The protein-ligand force-field includes an explicit term for electrostatics which are known to be important in such interactions. Partial charges were assigned using Gasteiger charges and rotatable bonds in the ligands were identified using AutoDock tools.

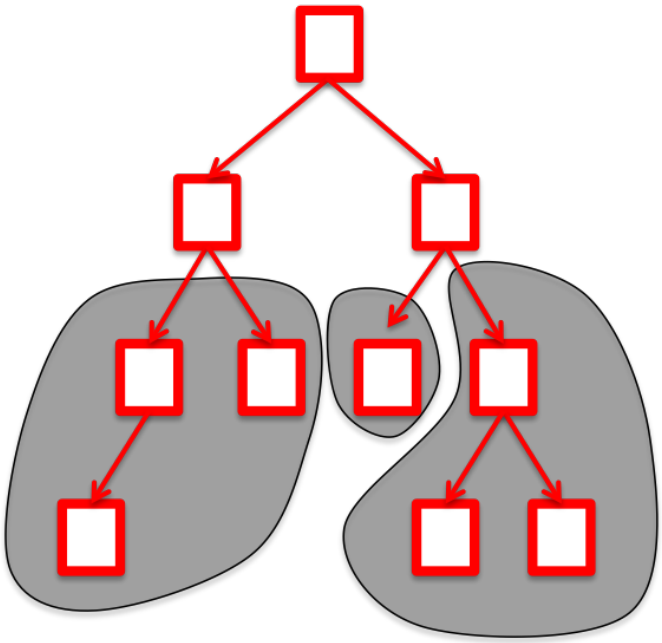
Fig. 5.8-(A) shows a torsion tree of a ligand as generated by Autodock Tools. The root of the tree corresponds to the core of the ligand; each rotatable bond connects two rigid parts of the ligand. Starting from this torsion tree representation, GOBLIN-Ligand constructs conformations of the ligand by discretizing each rotatable bond. The resulting conformations behave like side-chains of an amino-acid and can be used within the MRF framework. Fig. 5.8-(B) shows a portion of the MRF generated for one such protein-ligand interaction.

A dataset of 122 protein-ligand crystal complexes for the Ligand Protein Database Roche et al. [2001] was used to evaluate GOBLIN . As previously, for each complex in the database, the crystal structure of the bound complex as well as binding energy terms were available. Unlike previously where only changes in ΔG_{bind}upon mutation were available for the protein-protein interaction dataset, in this case the experimentally calculated values for binding free energies ΔG_{bind} were available.

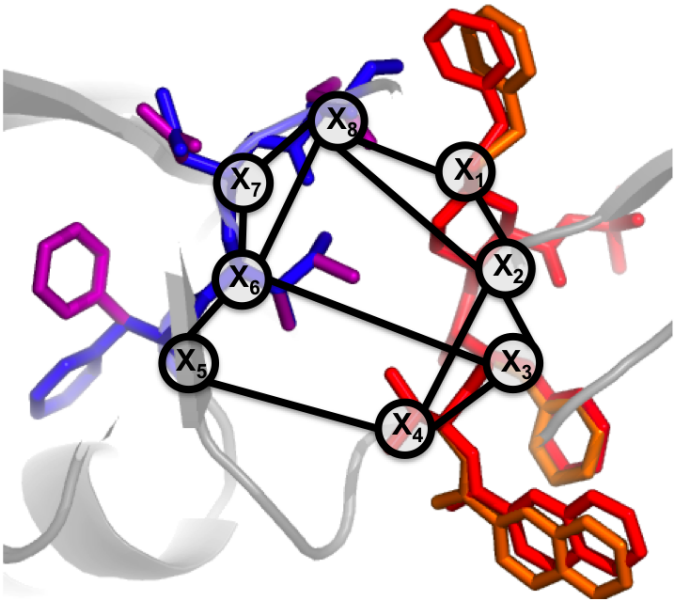
The dataset was split into equal sized randomly selected train/test partitions. Force-field parameters were trained to experimental values in the training set and the errors were computed on the test set. This process was repeated for 20 random train/test partitions.

Fig. 5.9-(A) shows the root mean square test error across these twenty partitions for

GOBLIN-Ligand and Autodock with default and re-trained parameters. In addition, the accuracy of GOBLIN-Ligand's predictions without entropic contributions is also shown. Fig. 5.9-(B) shows a boxplot the breaks down GOBLIN-Ligand's performance as a function of the number of rotatable bonds.

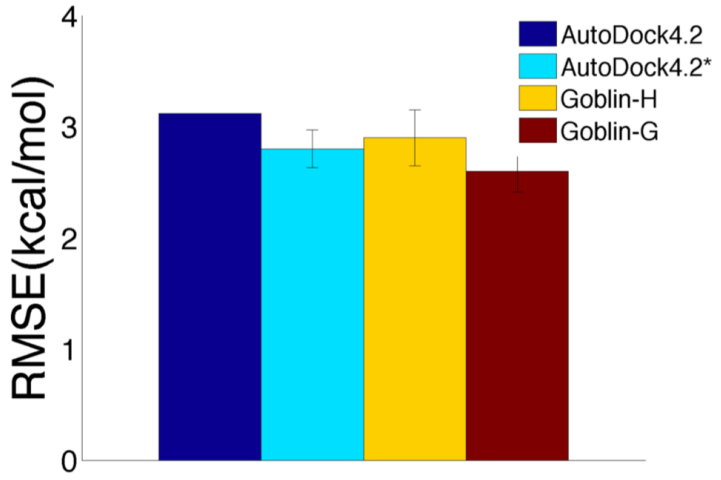


(A)

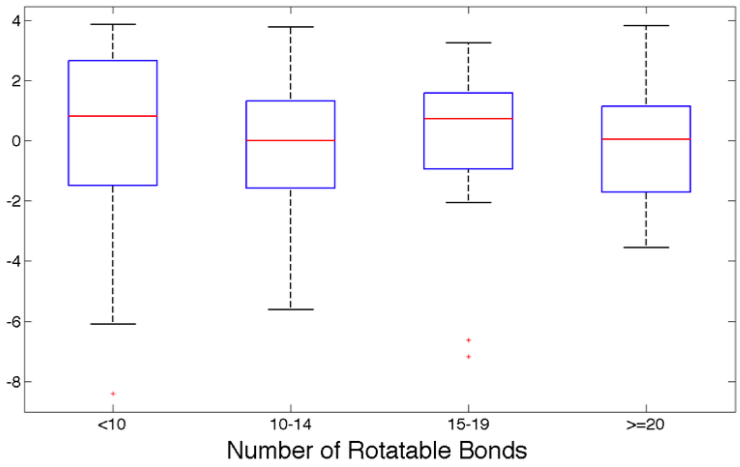


(B)

Figure 5.8: **Graphical models of protein-ligand complexes.** (A) Figure showing an example torsion tree of a ligand. Each directed edge corresponds to rotatable bond and is a degree of freedom. The core of the ligand corresponding to the nodes of the torsion tree around the root are kept rigid while the rest of the ligand is allowed to rotate. (B) Graphical model of the backbone and side-chain ensembles. The nodes on the protein correspond to random variables over rotameric side-chain conformations while the nodes on the ligand correspond to torsional degrees of freedom. (B)



(A)



(B)

Figure 5.9: **Results of GOBLIN-Ligand on 122 protein-ligand complexes** (A)Comparison of root mean square errors with experimentally measured ΔGvalues Figure shows error in prediction for multiple approaches: AutoDock 4.2 run with its default settings; Autdock 4.2 with force-field parameters re-trained on this dataset; using the predicted free energy according to GOBLIN (GOBLIN-G) and using only the enthalpic component of GOBLIN (GOBLIN-H). GOBLIN-Ligand outperforms all other approaches on this dataset. (B) Boxplot showing variation of GOBLIN-Ligand's error as the number of rotatable bonds in the ligand increase.

Chapter 6

Structured Priors over Sequence Space

We introduce a new approach to learning statistical models from multiple sequence alignments (MSA) of proteins. Our method, called GREMLIN (Generative REgularized ModeLs of proteINs), learns an undirected probabilistic graphical model of the amino acid composition within the MSA. The resulting model encodes both the position-specific conservation statistics *and* the correlated mutation statistics between sequential and long-range pairs of residues. Existing techniques for learning graphical models from multiple sequence alignments either make strong, and often inappropriate assumptions about the conditional independencies within the MSA (e.g., Hidden Markov Models), or else use sub-optimal algorithms to learn the parameters of the model. In contrast, GREMLIN makes no *a priori* assumptions about the conditional independencies within the MSA. We formulate and solve a *convex* optimization problem, thus guaranteeing that we find a *globally optimal* model at convergence. The resulting model is also generative, allowing for the design of new protein sequences that have the same statistical properties as those in the MSA. We perform a detailed analysis of covariation statistics on the extensively studied WW and PDZ domains

and show that our method out-performs an existing algorithm for learning undirected probabilistic graphical models from MSA. We then apply our approach to 71 additional families from the PFAM database and demonstrate that the resulting models significantly out-perform Hidden Markov Models in terms of predictive accuracy.

6.1 Introduction

A protein family* is a set of evolutionarily related proteins descended from a common ancestor, generally having similar sequences, three dimensional structures, and functions. By examining the statistical patterns of sequence conservation and diversity within a protein family, we can gain insights into the constraints that determine structure and function. These statistical patterns are often learned from multiple sequence alignments (MSA) and then encoded using probabilistic graphical models (e.g., Krogh et al. [1994], Karplus et al. [1997, 1998], Bateman et al. [2002], Liu et al. [2006]). The well-known database PFAM Bateman et al. [2002], for example, contains more than 11,000 profile Hidden Markov Models (HMM) Eddy [1998] learned from MSAs. The popularity of generative graphical models is due in part to the fact that they can be used to perform important tasks such as structure and function classification (e.g., Karplus et al. [1997], Liu et al. [2006]) and to design new protein sequences (e.g., Thomas et al. [2009a]). Unfortunately, existing methods for learning graphical models from MSAs either make unnecessarily strong assumptions about the nature of the underlying distribution over protein sequences, or else use greedy algorithms that are often sub-optimal. The goal of this chapter is to introduce a new algorithm that addresses these two issues simultaneously and to demonstrate the superior performance of

*In this chapter, the expression *protein family* is synonymous with *domain family*.

the resulting models.

A graphical model encodes a probability distribution over protein sequences in terms of a graph and a set of functions. The nodes of the graph correspond to the columns of the MSA and the edges specify the *conditional independencies* between the columns. Each node is associated with a local function that encodes the column-specific conservation statistics. Similarly, each edge is associated with a function that encodes the correlated mutation statistics between pairs of residues.

The task of learning a graphical model from an MSA can be divided into two subproblems: (i) learning the topology of the graph (i.e., the set of edges), and (ii) estimating the parameters of the functions. The first problem is especially challenging because the number of unique topologies on a graph consisting of p nodes is $O(2^{p^2})$. For that reason, it is common to simply *impose* a topology on the graph, and then focus on parameter estimation. An HMM, for example, has a simple topology where each column is connected to its immediate neighbors. That is, the model assumes each column is conditionally independent of the rest of the MSA, given its sequential neighbors. This assumption dramatically reduces the complexity of learning the model but is not well justified biologically. In particular, it has been shown by Ranganathan and colleagues that it is necessary to model correlated mutations between non-adjacent residues Lockless and Ranganathan [1999], Socolich et al. [2005], Russ et al. [2005].

Thomas and colleagues Thomas et al. [2005] demonstrated that correlated mutations between non-adjacent residues can be efficiently modeled using a different kind of graphical model known as a Markov Random Field (MRF). However, when using MRFs one must first identify the conditional independencies within the MSA. That is, one must learn the topology of the model. Thomas and colleagues address that problem using a greedy algo-

rithm, called GMRC, that adds edges between nodes with high mutual information Thomas et al. [2005, 2008b, 2009c,b]. Unfortunately, their algorithm provides no guarantees as to the optimality of the resulting model.

The algorithm presented in this chapter, called GREMLIN (Generative REgularized ModeLs of proteINs), solves the same problem as Thomas et al. [2005] but does so using a method with strong theoretical guarantees. In particular, our algorithm is *consistent*, i.e. it is guaranteed to yield the true model as the data increases, and it has low *sample-complexity*, i.e. it requires less data to identify the true model than any other known approach. GREMLIN also employs *regularization* to penalize complex models and thus reduce the tendency to over-fit the data. Finally, our algorithm is also computationally efficient and easily parallelizable. We demonstrate GREMLIN by performing a detailed analysis on the well-studied WW and PDZ domains and demonstrate that it produces models with higher predictive accuracy than those produced using the GMRC algorithm. We then apply GREMLIN to 71 other families from the PFAM database and show that our algorithm produces models with consistently higher predictive accuracy than profile HMMs.

6.2 Modeling Domain Families with Markov Random Fields

Let S_i be a finite discrete random variable representing the amino-acid composition at position i of the MSA of the domain family taking values in $\{1...k\}$ where the number of states, k, is 21 (20 amino acids with one additional state corresponding to a gap). Let $\mathbf{S} = \{S_1, S_2, ..S_p\}$ be the multi-variate random variable describing the amino acid composition of an MSA of length p. Our goal is to model $P(\mathbf{S})$, the amino-acid composition of the domain family.

Unfortunately, P(**S**) is a distribution over a space of size k^p, rendering the explicit modeling of the joint distribution computationally intractable for naturally occurring domains. However, by exploiting the properties of the distribution, one can significantly decrease the number of parameters required to represent this distribution. To see the kinds of properties that we can exploit, let us consider a toy domain family represented by an MSA as shown in Fig. 6.1-(A). A close examination of the MSA reveals the following statistical properties of its composition: (i) the Tyrosine ('Y') at position 2 is conserved across the family; (ii) positions 1 and 4 are co-evolving – sequences with a (S) at position 1 have a Histidine (H) at position 4, while sequences with a Phenylalanine (F) at position 1 have a Tryptophan (W) at position 4; (iii) the remaining positions appear to evolve independently of each other. In probabilistic terms we say that S_1, S_3 are co-varying, and that the remaining S_i's are statistically independent. We can therefore encode the joint distribution over all positions in the MSA by storing one joint distribution $P(S_1, S_4)$, and the univariate distributions $P(S_i)$, for the remaining positions (since they are all statistically independent of every other variable).

The ability to factor the full joint distribution, $P(\mathbf{S})$, in this fashion has an important consequence in terms of space complexity. Namely, we can reduce the space requirements from 21^7 to $21^2 + 7 * 21$ parameters. This drastic reduction in space complexity translates to a corresponding reduction in time complexity for computations over the distribution. While this simple example utilizes independencies in the distribution; this kind of reduction is possible in the more general case of *conditional independencies*. A Probabilistic Graphical Model (PGM) exploits these (conditional) independence properties to store the joint probability distribution using a small number of parameters.

Intuitively, a PGM stores the joint distribution of a multivariate random variable in a graph; while any distribution can be modeled by a PGM with a complete graph, exploit-

ing the conditional independencies in the distribution leads to a PGM with a (structurally) sparse graph. Following Thomas et al. [2008b], we use a specific type of probabilistic graphical model called a Markov Random Field (MRF). In its commonly defined form with pair-wise log-linear potentials, a Markov Random Field (MRF) can be formally defined as a tuple $\mathcal{M} = (\mathbf{S}, \mathcal{E}, \Phi, \Psi)$ where $(\mathbf{S}, \mathcal{E})$ is an undirected graph over the random variables. $\mathbf{S}$ represents the set of vertices and $\mathcal{E}$ is the set of edges of the graph. The graph succinctly represents conditional independencies through its Markov properties, which state for instance that each node is independent of all other nodes given its neighbors. Thus, graph separation in $(\mathbf{S}, \mathcal{E})$ implies conditional independence. Φ, Ψ are a set of node and edge potentials, respectively, usually chosen to be log-linear functions of the form:

$$\phi_i = [e^{v_1^i} \;\; e^{v_2^i} \;\; ... \;\; e^{v_k^i}]; \qquad \psi_{ij} = \begin{bmatrix} e^{w_{11}^{ij}} & e^{w_{12}^{ij}} & ... & e^{w_{1k}^{ij}} \\ e^{w_{21}^{ij}} & e^{w_{22}^{ij}} & ... & e^{w_{2k}^{ij}} \\ & & ... & \\ e^{w_{k1}^{ij}} & e^{w_{k2}^{ij}} & ... & e^{w_{kk}^{ij}} \end{bmatrix} \tag{6.1}$$

where i is a position in the MSA, and (i, j) is an edge between the positions i and j in the MSA. ϕ_i is a $(k \times 1)$ vector and ψ_{ij} is a $(k \times k)$ matrix. For future notational simplicity we further define

$$\mathbf{v}^i = [v_1^i \;\; v_2^i \;\; ... \;\; v_k^i] \qquad \mathbf{w}^{ij} = \begin{bmatrix} w_{11}^{ij} & w_{12}^{ij} & ... & w_{1k}^{ij} \\ w_{21}^{ij} & w_{22}^{ij} & ... & w_{2k}^{ij} \\ & & ... & \\ w_{k1}^{ij} & w_{k2}^{ij} & ... & w_{kk}^{ij} \end{bmatrix} \tag{6.2}$$

where $\mathbf{v}^i$ is a $(k \times 1)$ vector and $\mathbf{w}^{ij}$ is a $(k \times k)$ matrix. $\mathbf{v} = \{\mathbf{v}^i | i = 1 \dots p\}$ and

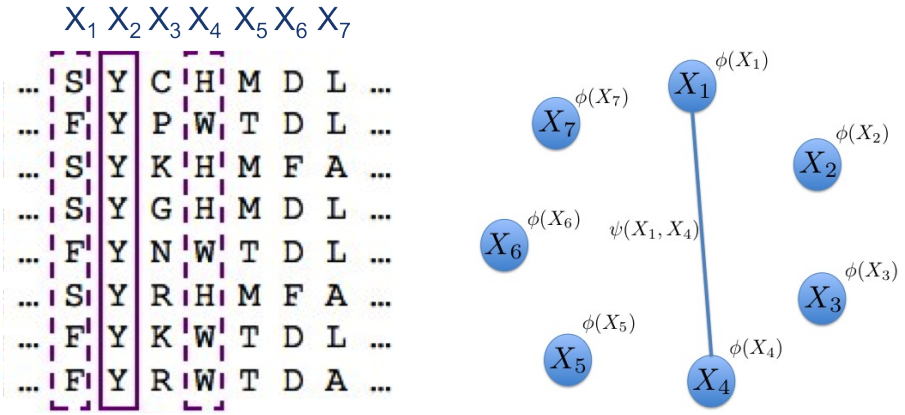


Figure 6.1: (A) A multiple sequence alignment (MSA) for a hypothetical domain family. (B) The Markov Random Field encoding the conservation in and the coupling in the MSA. The edge between random variables S_1 and S_4 reflects the coupling between positions 1 and 4 in the MSA.

$\mathbf{w} = \{\mathbf{w}^{ij} | (i,t) \in \mathcal{E}\}$ are node and edge "weights". $\mathbf{v}$ is a collection of p, $(k \times 1)$ vectors and $\mathbf{w}$ is a collection of p, $(k \times k)$ matrices.

The probability of a particular sequence $\mathbf{s} = \{s_1, s_2, \ldots, s_p\}$ according to $\mathcal{M}$ is defined as:

$$P_{\mathcal{M}}(\mathbf{S} = \mathbf{s}) = \frac{1}{Z} \prod_{i \in V} \phi_i(s_i) \prod_{(i,j) \in E} \psi_{ij}(s_i, s_j) \tag{6.3}$$

where Z, the so-called partition function, is a normalizing constant defined as a sum over all possible assignments to $\mathbf{S}$:

$$Z = \sum_{\mathbf{s} \in \mathbf{S}} \prod_{i \in V} \phi_i(s_i) \prod_{(i,j) \in E} \psi_{ij}(s_i, s_t) \tag{6.4}$$

The structure of the MRF for the MSA shown in Fig. 6.1(A) is shown in Fig. 6.1(B). The edge between variables S_1 and S_4 reflects the statistical coupling between those positions in

the MSA.

6.3 Structure learning with L_1 Regularization

In the previous section we outlined how an MRF can parsimoniously model the probability distribution $P(\mathbf{S})$. In this section we consider the problem of *learning* the MRF from an MSA.

Eq. 6.3 describes the probability of a sequence $\mathbf{s}$ for a specific model $\mathcal{M}$. Given a set of independent sequences $\mathcal{S} = \{\mathbf{s^1}, \mathbf{s^2}, \mathbf{s^3}, ..., \mathbf{s^n}\}$, the log-likelihood of the model parameters $\Theta = (\mathcal{E}, \mathbf{v}, \mathbf{w})$ is then:

$$\text{ll}(\Theta) = \sum_{\mathbf{s}^d \in \mathcal{S}} \left[\sum_{i \in V} \log \phi_i(\mathbf{s}_i^d) + \sum_{(i,j) \in E} \log \psi_{ij}(\mathbf{s}_i^d, \mathbf{s}_t^d) \right] - \log Z \tag{6.5}$$

where the term in the braces is the unnormalized likelihood of each sequence, and Z is the global partition function. The problem of learning the structure *and* parameters of the MRF is now simply that of maximizing ll(Θ).

$$MLE(\theta) = \max_{\Theta} \text{ll}(\Theta) \tag{6.6}$$

This Maximum Likelihood Estimate (MLE) is guaranteed to recover the true parameters as the amount of data increases. However, this formulation suffers from two significant shortcomings: (i) the likelihood involves the computation of the global partition function which is computationally intractable and requires $O(k^p)$ time to compute, and (ii) in the absence of infinite data, the MLE can significantly over-fit the training data due to the potentially large number of parameters in the model.

An overview of our approach to surmount these shortcomings is as follows: first, we approximate the likelihood of the data with an objective function that is easier to compute, yet retains the optimality property of MLE mentioned above. To avoid over-fitting and learning densely connected structures, we then add a regularization term that penalizes complex models to the likelihood objective. The specific regularization we use is particularly attractive because it has high statistical efficiency.

The general regularized learning problem is then formulated as:

$$\max_{\Theta} \text{pll}(\Theta) - \text{Reg}(\Theta) \tag{6.7}$$

where the pseudo log-likelihood $\text{pll}(\Theta)$ is an approximation to the exact log-likelihood and $\text{Reg}(\Theta)$ is a regularization term that penalizes complex models.

While this method can be used to jointly estimate both the structure $\mathcal{E}$ and the parameters $\mathbf{v}, \mathbf{w}$, it will be convenient to divide the learning problem into two parts: (i) *structure learning* — which learns the edges of the graph, and (ii) and *parameter estimation* — learning $\mathbf{v}, \mathbf{w}$ given the structure of the graph. We will use a regularization penalty in the structure learning phase that focuses on identifying the correct set of edges. In the parameter estimation phase, we use these edges and learn $\mathbf{v}$ and $\mathbf{w}$ using a different regularization penalty that focuses on estimating $\mathbf{v}$ and $\mathbf{w}$ accurately. We note that once the set of edges has been fixed, the parameter estimation problem can be solved efficiently. Thus, we will focus on the problem of learning the edges or, equivalently, the set of conditional independencies within the model.

6.3.1 Pseudo Likelihood

The log-likelihood as defined in Eq. 6.5 is smooth, differentiable, and concave. However, maximizing the log-likelihood requires computing the global partition function Z and its derivatives, which in general can take up to $\mathcal{O}(k^p)$ time. While approximations to the partition function based on Loopy Belief Propagation [Lee et al., 2007b] have been proposed as an alternative, such approximations can lead to inconsistent estimates.

Instead of approximating the true-likelihood using approximate inference techniques, we use a different approximation based on a pseudo-likelihood proposed by Besag [1977], and used in Wainwright et al. [2007], Schmidt et al. [2008]. The pseudo-likelihood is defined as:

$$\begin{aligned} \text{pll}(\Theta) &= \frac{1}{n}\sum_{\mathbf{s}^d \in \mathcal{S}}\sum_{i=1}^{p} \log(P(\mathbf{s}_i^d | \mathbf{s}_{-i}^d)) \\ &= \frac{1}{n}\sum_{\mathbf{s}^d \in \mathcal{S}}\sum_{j=1}^{p} \left[\log \phi_i(\mathbf{s}_i^d) + \sum_{j \in V_i'} \log \psi_{ij}(\mathbf{s}_i^d, \mathbf{s}_j^d) - \log Z_i \right] \end{aligned}$$

where $\mathbf{s}_i^d$ is the residue at the i^{th} position in the d^{th} sequence of our MSA, $\mathbf{s}_{-i}^d$ denotes the "Markov blanket" of $\mathbf{s}_i^d$, and Z_i is a local normalization constant for each node in the MRF. The set V_i' is the set of all vertices which connect to vertex i in the PGM. The only difference between the likelihood and pseudo-likelihood is the replacement of a global partition function with local partition functions (which are sums over possible assignments to single nodes rather than a sum over all assignments to *all* nodes of the sequence). This difference makes the pseudo-likelihood significantly easier to compute in general graphical models.

The pseudo-likelihood retains the concavity of the original problem, and this approx-

imation makes the problem tractable. Moreover, this approximation is known to yield a consistent estimate of the parameters under fairly general conditions if the generating distribution is in fact a pairwise MRF defined by a graph over **S** [Gidas, 1988]. That is, under these conditions, as the number of samples increases, parameter estimates using pseudo-likelihood converge to the true parameters.

6.3.2 L1 Regularization

The study of convex approximations to the complexity and goodness of fit metrics has received considerable attention recently [Wainwright et al., 2007, Lee et al., 2007b, Hofling and Tibshirani, 2009, Schmidt et al., 2008]. Of these, those based on L_1 regularization are the most interesting because of their strong theoretical guarantees. In particular methods based on L_1 regularization exhibit consistency in both parameters and structure (i.e., as the number of samples increases we are guaranteed to find the true model), and high statistical efficiency (i.e., the number of samples needed to achieve this guarantee is small). See Tropp [2006] for a recent review of L_1-regularization. Our algorithm uses L_1-regularization for both structure learning and parameter estimation.

For the specific case of block-L_1 regularization, $Reg(\Theta)$ usually takes the form:

$$Reg(\Theta) = \lambda_{node} \sum_{i=1}^{p} \|\mathbf{v}^i\|_2^2 + \lambda_{edge} \sum_{i=1}^{p} \sum_{j=i+1}^{p} \|\mathbf{w}^{ij}\|_2 \tag{6.8}$$

where λ_{node} and λ_{edge} are regularization parameters that determine how strongly we penalize higher (absolute) weights. The value of λ_{node} and λ_{edge} control the trade-off between the log-likelihood term and the regularization term in our objective function.

The regularization described above groups all the parameters that describe an edge to-

gether in a *block*. The second term in Eq. 6.8 is the sum of the L_2 norms of each block. Since the L_2 norm is always positive, our regularization is exactly equivalent to penalizing the L_1 norm of the vector of norms of each block with the penalty increasing with higher values of λ_{edge}. It is important to distinguish the block-L_1 regularization on the edge weights from the more traditional L_2 regularization on the node weights where we sum the *squares* of the L_2 norms.

The L_1 norm is known to encourage sparsity (by setting parameters to be exactly zero), and the *block* L_1 norm we have described above encourages group sparsity (where *groups* of parameters are set to zero). Since, each group corresponds to all the parameters of a single edge, using the block L_1 norm leads to what we refer to as structural sparsity (i.e. sparsity in the edges). In contrast, the L_2 regularization also penalizes high absolute weights, but does not usually set any weights to zero, and thus does not encourage sparsity.

6.3.3 Optimizing Regularized Pseudo-Likelihood

In the previous two sections we described an objective function, and then a tractable and consistent approximation to it, given a set of weights (equivalently, potentials). However, to solve this problem we still need to be able to find the set of weights that maximizes the likelihood under the block-regularization form of Eq. 6.7. We note that the objective function associated with block-L_1 regularization is no longer smooth. In particular, its derivative with respect to any parameter is discontinuous at the point where the group containing the parameter is 0. We therefore consider an equivalent formulation where the non-differentiable part of the objective is converted into a constraint making the new objective function differ-

entiable.

$$\max_{\Theta,\alpha} \text{pll}(\Theta) - \lambda_{node} \sum_{i=1}^{p} \|\mathbf{v}^i\|_2^2 - \lambda_{edge} \sum_{i=1}^{p} \sum_{j=i+1}^{p} \alpha_{ij}$$

$$\text{subject to:} \quad \forall (1 \leq i < j \leq p) : \alpha_{ij} \geq \|\mathbf{w}^{ij}\|_2$$

where the constraints hold with equality at the optimal (Θ,α). Intuitively, α_{ij} behaves as a differentiable proxy for the non-differentiable $\|\mathbf{w}^{ij}\|_2$, making it possible to solve the problem using techniques from smooth convex optimization. Since the constraints hold with equality at the optimal solution (ie $\alpha_{ij} = \|\mathbf{w}^{ij}\|_2$), the solutions and therefore, the formulations are identical.

We solve this reformulation through the use of projected gradients. We first ignore the constraints, compute the gradient of the objective, and take a step in this direction. If the step results in any of the constraints being violated we solve an alternative (and simpler) Euclidean projection problem:

$$\min_{\Theta'\alpha'} \| \begin{bmatrix} \Theta' \\ \alpha' \end{bmatrix} - \begin{bmatrix} \Theta \\ \alpha \end{bmatrix} \|_2^2$$

$$\text{subject to:} \quad \forall (1 \leq i < j \leq p) : \alpha_{ij} \geq \|\mathbf{w}^{ij}\|_2$$

which finds the closest parameter vector to the vector obtained by taking the gradient step (in Euclidean distance), which satisfies the original constraints. In this case the projection problem can be solved extremely efficiently (in linear time) using an algorithm described in Schmidt et al. [2008]. Methods based on projected gradients are guaranteed to converge to a

stationary point [Boyd and Vandenberghe, 2004], and convexity ensures that this stationary point is globally optimal.

In order to scale the method to significantly larger domains, we can sub-divide the structure learning problem into two steps. In the first step, each node is considered separately to identify its neighbors. This may lead to an asymmetric adjacency matrix, and so in the second step the adjacency matrix is made symmetric. This two-step approach to structure learning has been extensively compared to the single step approach by Hofling and Tibshirani [2009] and has been found to have almost identical performance. The two-step approach however has several computational advantages. The problem of learning the neighbors of a node is exactly equivalent to solving a logistic regression problem with block-L_1 regularization, and this problem can be solved quickly and with low memory requirements. Additionally, the problem of estimating the graph can now be trivially parallelized across nodes of the graph since these logistic regression problems are completely decoupled. Parameter learning of the graph with just L_2 regularization can then be solved *extremely* efficiently using quasi-Newton methods [Nocedal, 1980].

Chapter 7

Application: Learning Generative Models over Protein Fold Famlies

7.1 Results

The probabilistic framework defined in Sec. 6.2 and the optimization objectives and algorithms defined in Sec. 6.3 constitute a method for learning a graphical model from a given MSA. The optimization framework has two major penalty parameters that can be varied (λ_v, λ_e). To understand the effects of these parameters, we first evaluated GREMLIN on artificial protein families whose sequence records were generated from known, randomly generated models. This lets us evaluate the success of the various components of GREMLIN in a controlled setting where the ground truth was known.

Our experiments involve comparing the performance of ranking edges and learning a graph structure using a variety of techniques, including: (i) our algorithm, GREMLIN; (ii) the greedy algorithm of Thomas et al. [2005, 2008b], denoted GMRC method'; and (iii)

Profile Hidden Markov Models [Eddy, 1998] used by Bateman et al. [2002].

We note that the GMRC method only considers edges that meet certain coupling criteria (see Thomas et al. [2005, 2008b] for details). In particular, we found that it returns sparse graphs (fewer than 100 edges), regardless of choice of run-time parameters. GREMLIN, in contrast, returns a full spectrum from disconnected to completely connected graphs depending on the choice of the regularization parameter. In our experiments, we use our parameter estimation code on their graphs, and compare ourselves to the best graph they return.

In the remainder of this section, we demonstrate that GREMLIN significantly out-performs other algorithms. In particular, we show that GREMLIN achieves higher goodness of fit to the test set, and has lower prediction error than the GMRC method - *even when we learn models of similar sparsity*. Finally, we show that GREMLIN also significantly out performs profile HMM-based models for 71 real protein families, in terms of goodness of fit. These results demonstrate that the use of block-regularized structure learning algorithms can result in higher-quality MRFs than those learnt by the GMRC method, and that MRFs produce higher quality models than HMMs.

7.1.1 Simulations

We generated 32-node graphs. Each node had a cardinality of 21 states, and each edge was included with probability ρ. Ten different values of ρ varying from 0.01 and 0.45 were used; for each value of ρ, twenty different graphs were generated resulting in a total of 200 graphs. For each edge that was included in a graph, edge and node weights were drawn from a Normal distribution (weights $\sim \mathcal{N}(0,1)$). Since each edge involves sampling 441 weights from this distribution, the edges tend to have many small weights and a few large ones. This reflects the observation that in positions with known correlated mutations, a few

favorable pairs of amino acids are usually much more frequent than most other pairs. When we sample from our simulated graphs using these parameters, we therefore tend to generate such sequences.

For each of these 200 graphical models, we then sampled 1000 sequences using a Gibbs sampler with a burn-in of 10,000 samples and discarding 1,000 samples between each accepted sequence. These 1000 sequences were then partitioned into two sets: a training set containing 500 sequences and a held-out set of 500 sequences used to test the model. The training set was then used to train a model using the block regularization norm.

We first test our accuracy on structure learning. We measure accuracy by the F-score which is defined as

$$\text{F-score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Precision and recall are in turn defined in terms of the number of true positives (tp), false positives (fp) and false negatives (fn) as precision $= \frac{tp}{tp+fp}$ and recall $= \frac{tp}{tp+fn}$.

Since the structure of the model directly depends only on the regularization weight on the edges, the structures were learnt for each norm and each training set with different values of λ_e (between 1 and 500), keeping λ_v fixed at 1.

Fig. 7.1 shows our performance in predicting the true structure by using L_1-L_2 (Fig. 7.1) We observe that for all settings of ρ GREMLIN learns fairly accurate graphs at some value of λ_e.

Figure 7.2-A compares our structure learning method with the algorithm in Thomas et al. [2008b]. We evaluate their method over a wide range of parameter settings and select the best model. Figure 7.2-A shows that our method significantly out-performs their method for *all* values of ρ. We see that over all settings our best model has an average F-score of *at least* 0.63. We conclude that we are able to infer accurate structures given the proper choice

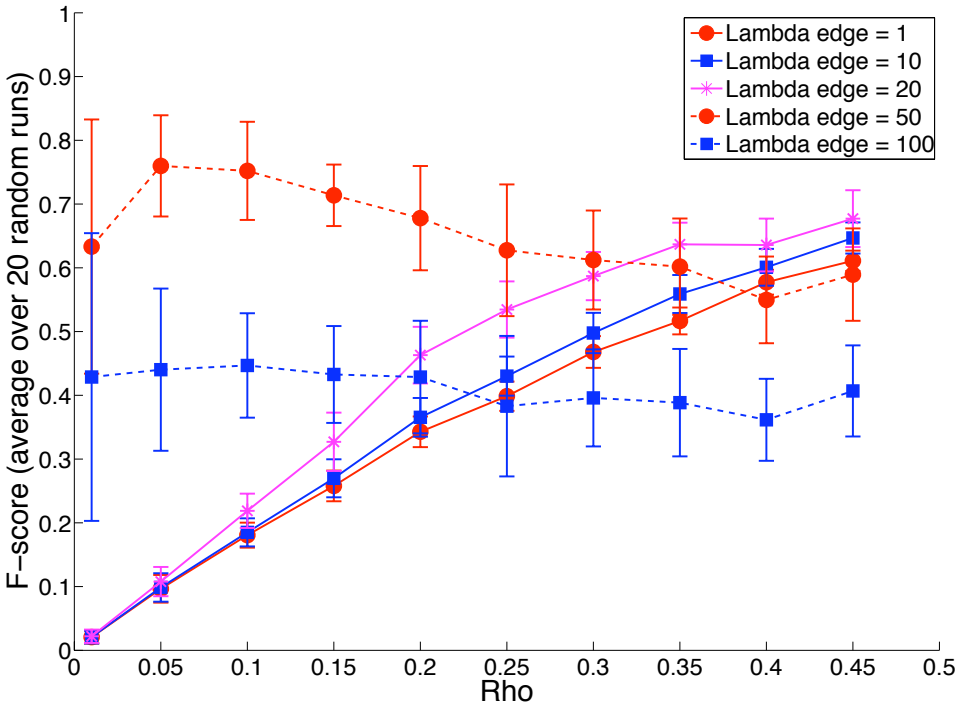
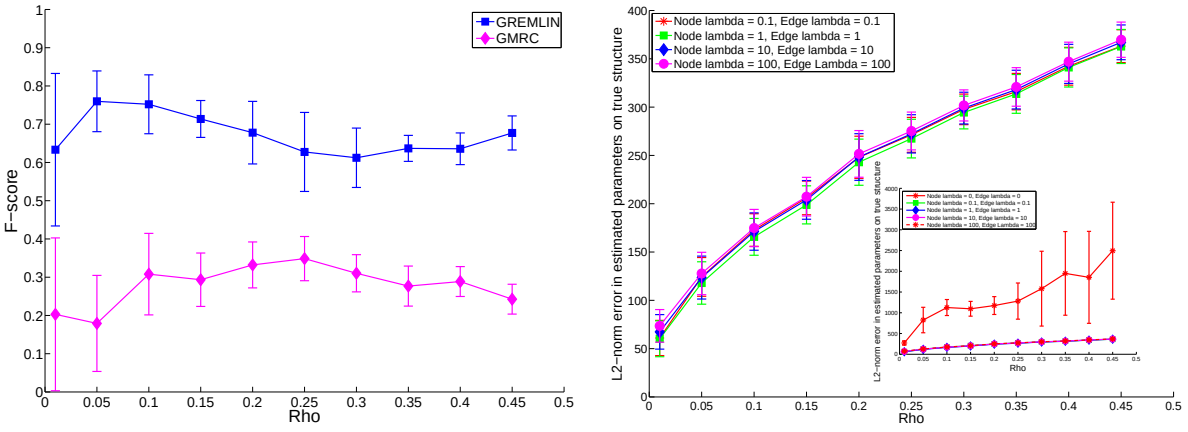


Figure 7.1: F-scores of structures learnt by using L_1-L_2 norm The figure shows the average and standard deviation of the F-score across 20 different graphs as a function of ρ, the probability of edge-occurrence.



(A) (B)

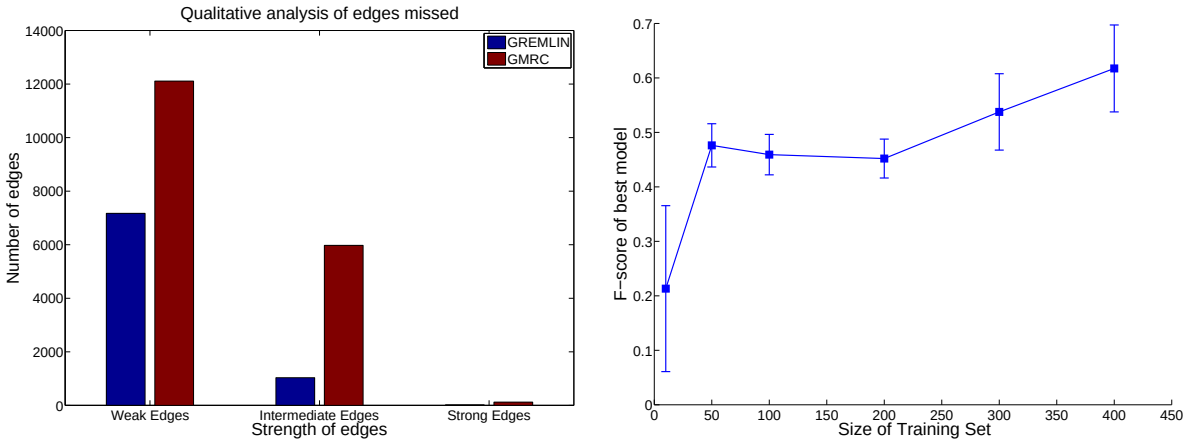
Figure 7.2: (A) Edge occurrence probability ρ versus F-score for the structure learning methods we propose, and the method proposed in Thomas et al. [2008b]. (B) L_2 norm of the error in the estimated parameters as a function of the weight of the regularization in stage two. The inset shows the case when no regularization is used in stage two. The much higher parameter estimation error in this case highlights the need for regularization in *both* stages.

of settings.

Figure 7.2-B, shows the error in our parameter estimates given the true graph as a function of ρ. We also find that parameter estimation is reasonably robust to the choice of the regularization weights, as long as the regularization weights are non-zero.

Fig. 7.3-A shows a qualitative analysis of edges missed by each method (we consider all simulated graphs and the best learnt graph of each method). We divide the missed edges into three groups (weak, intermediate and strong) based on their true L_2 norm. We see again that the three norms perform comparably, significantly out-performing the GMRC method in all three groups.

Finally, Fig. 7.3-B shows the sensitivity of our structure learning algorithms to the size of training set. In particular, we see that for the simulated graphs around 400 sequences



(A) (B)

Figure 7.3: (A) Qualitative grouping of edges missed by GREMLIN and the GMRC method (B) Sensitivity of structure learning to size of training set.

results in us learning very accurate structures. However, as few as 50 sequences are enough to infer reasonable structures.

7.1.2 Evaluating Structure and Parameters Jointly

In a simulated setting, structure and parameter estimates can be compared against known ground truth. However, for real domain families we need other evaluation methods. We evaluate the structure and parameters for real domain families by measuring the imputation error of the learnt models. Informally, the imputation error measures the probability of *not* being able to "generate" a complete sequence, given an incomplete one. The imputation error of a column is measured by erasing it in the test MSA, and then computing the probability that the true (known) residues would be predicted by the learnt model. This probability is calculated by performing inference on the erased columns, conditioned on the rest of the MSA. The imputation error of a model is the average of its imputation error over

columns.

Using imputation error directly for model selection generally gives us models that are too dense. Intuitively, once we have identified the true model, adding extra edges decreases the imputation error by a very small amount, probably a reflection of the finite-sample bias.

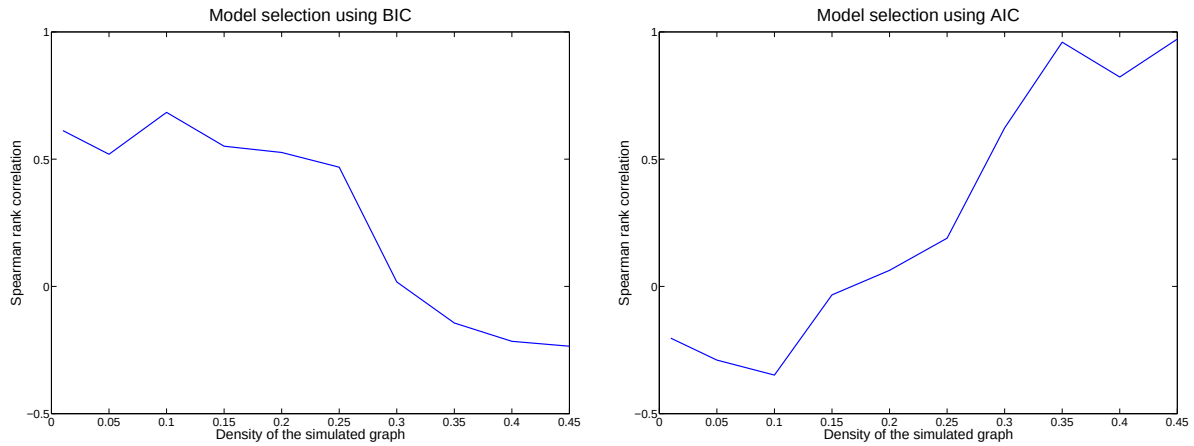
7.1.3 Model selection using information criteria

We consider modifications to two widely used model selection strategies. The Bayesian Information Criterion (BIC) [Schwarz, 1978], is used to select parsimonious models and is known to be asymptotically consistent in selecting the true model. The Akaike Information Criterion (AIC) [Akaike, 2003], typically selects denser models than the BIC, but is known to be asymptotically consistent in selecting the model with lowest predictive error (risk). In general, they do not however select the same model [Yang, 2003].

We use the following definitions:

$$\begin{aligned} \text{pseudo-BIC}(\lambda) &= -2\text{pll}(\lambda) + \log(\text{n})\text{df}(\lambda) \\ \text{pseudo-AIC}(\lambda) &= -2\text{pll}(\lambda) + 2\text{df}(\lambda) \end{aligned}$$

Where we use the pseudo log-likelihood approximation to the log-likelihood. While it may be expected that using the pseudo log-likelihood instead of the true log-likelihood may in fact lead to inconsistent selection a somewhat surprising result [Csiszar and Talata, 2006] shows that in the case of BIC using pseudo log-likelihood is in fact also consistent for model selection. Although we aren't aware of the result, we expect a similar result to hold for the risk consistency of the pseudo-AIC.



(A) (B)

Figure 7.4: Graph density versus the rank correlation for ranking and selection using (A) BIC (B) AIC.

We evaluate the likelihood on the *training* sample to score the different models. n is the number of training sequences.

Estimating the degrees of freedom of a general estimator is quite hard in practice. This has lead to use of various heuristics in practice. For the LASSO estimator which uses a pure-L1 penalty, it is known that the number of non-zeros in the regression vector is a good estimate of the degrees of freedom. A natural extension when using a *block*-L1 penalty is the number of non-zero blocks (i.e. edges). Since this does not differentiate between weak and strong edges, we used the block-L1 norm as an estimate of the degress of freedom. In our simulations, we find that choice often results in good model selection.

Figure 7.4 shows the performance of the two model selection strategies at different sparsity levels. We evaluate the performance by learning several graphs (at different levels of regularization) and comparing the Spearman rank-correlation between the F-score of the graphs and their rank. We can clearly see that when the true graph is sparse the modified

BIC has a high rank-correlation, whereas when the true graph is dense the modified AIC does well, with neither method providing reliable model selection for all graphs.

For this reason, we considered an approach to model selection based on finite sample error control. We chose to control the false discovery rate (FDR) in the following way. Consider permuting the each column of the MSA independently (and randomly). Intuitively, the true graph is now a graph with no edges. Thus, one approach to selecting the regularization parameter is to find the value that yields no edges on the permuted MSA. A more robust method, which we use, is to use the average regularization parameter obtained from multiple random permutations as in Listgarten and Heckerman [2007]. In the results that follow we use 20 random permutations.

Given the success of GREMLIN on simulated data, and equipped with a method for model selection described above, we proceed to apply GREMLIN to real protein MSAs. We consider the WW and PDZ families in some detail since the extensive literature on these families allows us to draw meaningful conclusions about the learnt models.

7.1.4 A generative model for the WW domain

The WW domain family (Pfam id: PF00397 [Bateman et al., 2002]) is a small protein interaction module with two highly conserved tryptophans that adopts a curved three-stranded β-sheet structure with a binding site for proline-containing peptides. In Socolich et al. [2005] and Russ et al. [2005], the authors determine, using Statistical Coupling Analysis (SCA), that the residues can be divided into two clusters: the first cluster contains a set of 8 strongly coupled residues and the second cluster contains everything else. Based on this finding, the authors then designed 44 sequences that satisfy co-evolution constraints of the first cluster, of which 12 actually fold *in vitro*. An alternative set of control sequences,

which did not satisfy the constraints, failed to fold.

We first constructed an MSA by starting with the PFAM alignment and removing sequences to construct a non-redundant alignment (no pair of sequences was greater than 80% similar). This resulted in an MSA with 700 sequences of which two thirds were used as a training set and the rest were used as a test set. Each sequence in the alignment had 30 positions. The training set was used to learn the model, for multiple values of λ_e. Given the structure of the graph, parameters were learned using $\lambda_v = 1, \lambda_e = 1$. The learnt model is presented in Figure 7.5.

Figure 7.6 compares the imputation errors of our approach (in red and yellow) with the GMRC method of Thomas et al. [2008b] and Profile HMMs [Eddy, 1998]. The model in red was learnt using λ_e selected by performing a permutation study. Since this model had more edges than the model learnt by GMRC, we used a higher λ_e to learn a model that had fewer edges than the GMRC model. The x-intercept was based on a loose lower bound on the error and was estimated by computing the imputation error on the test-data of a completely connected model *learnt on the test data*. Due to over-fitting, this is likely to be a very loose estimate of the lower bound. We find that our imputation errors are lower than the methods we compare to (even at comparable levels of sparsity).

To see which residues are affected by these edges, we construct a "coupling profile" (Fig. 7.5-C). We construct a shuffled MSA by taking the natural MSA and randomly permuting the amino acids within the same position (column of MSA) for each position. The new MSA now contains no co-evolving residues but has the same conservation profile as the original MSA. To build a coupling profile, we calculate the difference in the imputation error of sequences in a held-out test set and the shuffled MSA. Intuitively, having a high imputation error difference means that the position was indeed co-evolving with some other positions

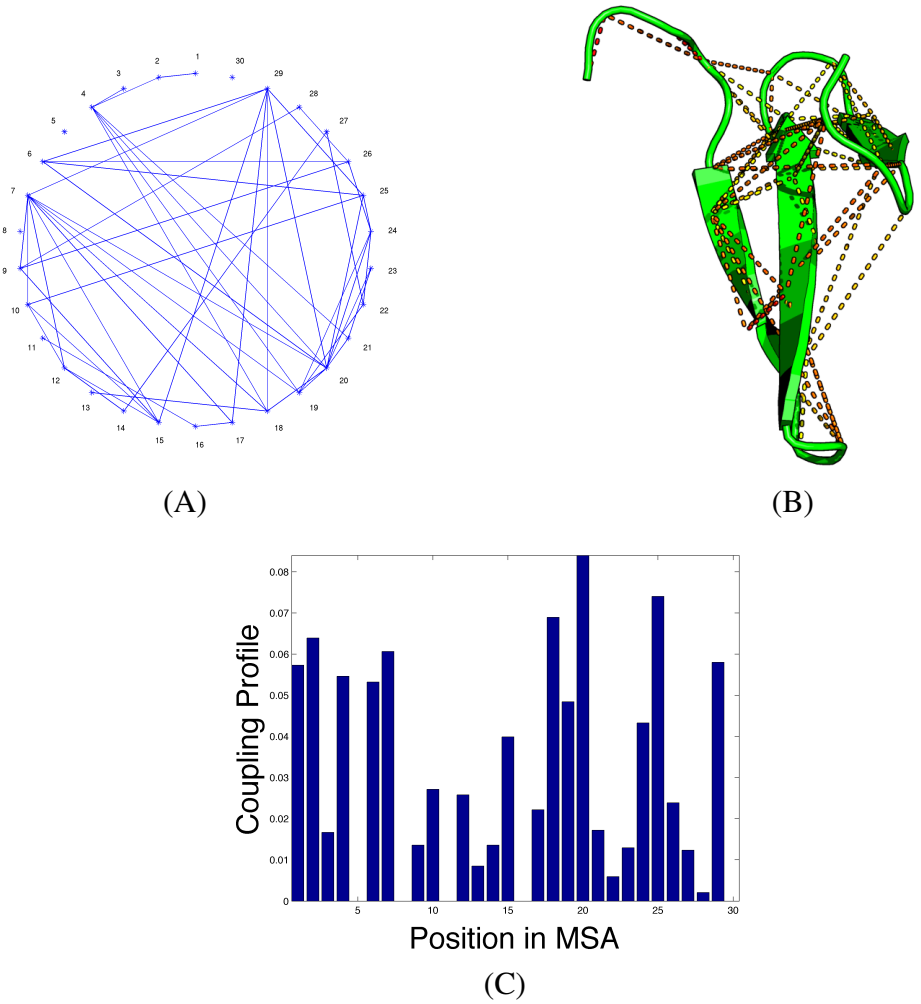


Figure 7.5: **WW domain model**. Edges returned by GREMLIN overlayed on a circle (**a**) and on the structure (**b**) of the WW domain of Transcription Elongation Factor 1 (PDB id: 2DK7) [Berman et al., 2000]. (**c**) Coupling profile (see text).

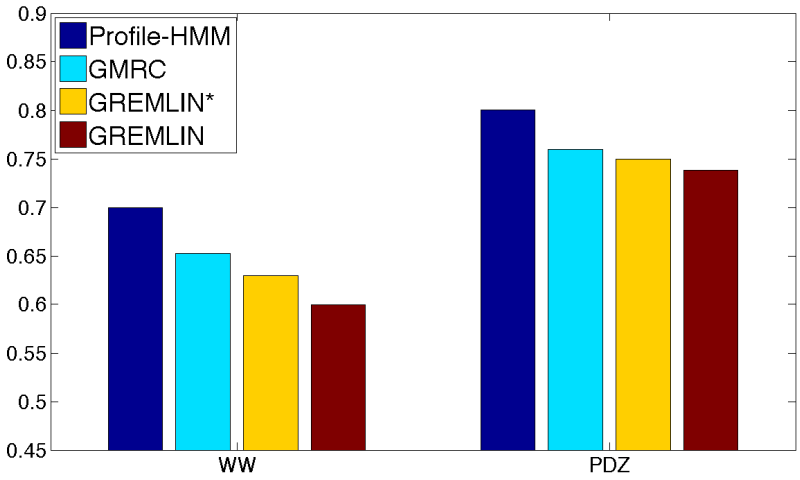


Figure 7.6: Comparison of Imputation errors on WW and PDZ families. We consider two variants of GREMLIN - with the regularization parameter selected either to produce a model with a smaller number of edges than GMRC (third bar in each group, shown in yellow) or to have zero edges on 20 permuted MSAs (last bar, shown in red). The x-intercept was chosed by estimating a lower bound on the imputation error as described in the text.

in the MSA. The other positions would also have a high imputation error difference in the coupling profile.

We also performed a retrospective analysis of the artificial sequences designed by Russ et al. [2005]. We attempt to distinguish sequences that folded from those that didn't.

Although this is a discriminative test (folded or not) of a generative model, we nevertheless achieve a high AUC of 0.87Fig. 7.7. We therefore postulate that the additional constraints we identify are indeed critical to the stability of the WW fold. In comparing our AUC to the published results of Thomas et al. [2008b] (AUC of 0.82) and the Profile HMM (AUC of 0.83) we see that we are able to better distinguish artificial sequences that fold from those that don't.

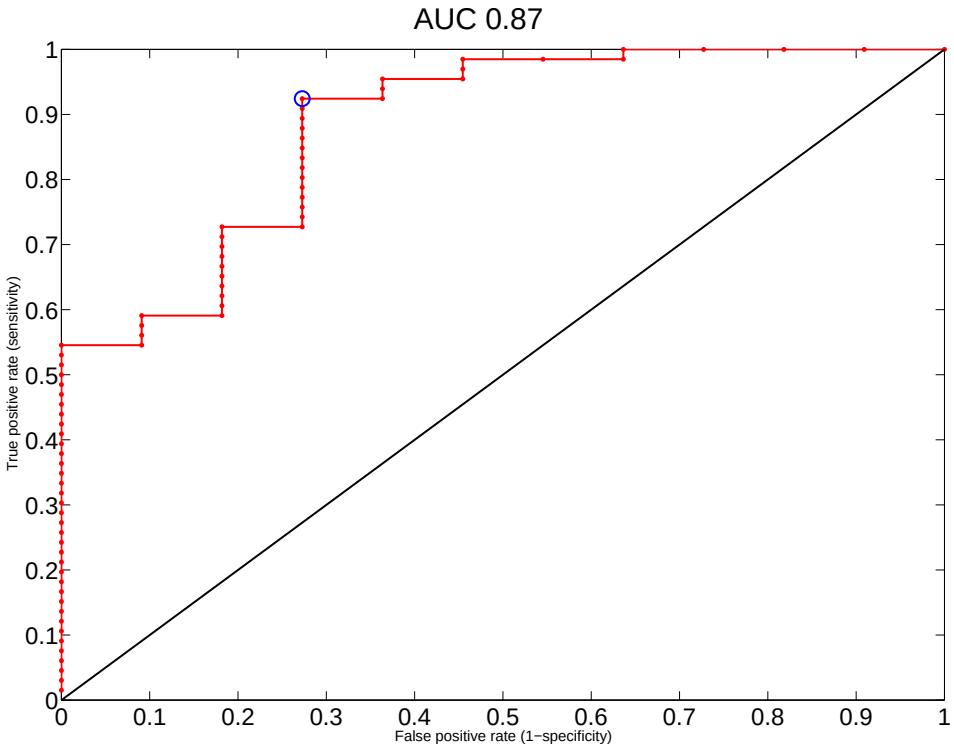


Figure 7.7: Receiver operating characteristic (ROC) curve of GREMLIN for the task of distinguishing artificial WW sequences that fold from those that don't.

7.1.5 Allosteric regulation in the PDZ domain

The PDZ domain is a family of small, evolutionarily well represented protein binding motifs. The domain is most commonly found in signaling proteins and helps to anchor transmembrane proteins to the cytoskeleton and hold together signaling complexes. The PDZ domain is also interesting because it is considered an *allosteric* protein. The domain, and its members have been studied extensively, in multiple studies, using a wide range of techniques ranging from computational approaches based on statistical coupling ([Lockless and Ranganathan, 1999]) and Molecular Dynamics simulations [Dhulesia et al., 2008], to NMR based experimental studies ([Fuentes et al., 2004]).

We use the MSA from Lockless and Ranganathan [1999]. The MSA is an alignment of 240 non-redundant sequences, with 92 positions. We chose a random sub-sample with two-thirds of the sequences as the training set and use the rest as a test set. Using this training set, we learnt generative models for each of the block regularizers, and choosing the smallest value of λ_e that gave zero edges for 20 permuted MSAs as explained previously. The resulting model had 112 edges (Fig. 7.8). Figure 7.6 summarizes the imputation errors on the PDZ domain. We again observe that the model we learn is denser than that learnt by GMRC and has lower imputation error. However, even at comparable sparsity GREMLIN out-performs the Profile HMM and GMRC.

The SCA based approach of Lockless and Ranganathan [1999] identified a set of residues that were coupled to a residue near the active site (HIS-70) including a residue at a distal site on the other end of the protein (GLY-49 in this case). Since the SCA approach can only determine the presence of a dependence but cannot distinguish between direct and indirect couplings, only a cluster of residues was identified. Our model also identifies this interaction, but more importantly, it determines that this interaction is mediated by ALA-74 with

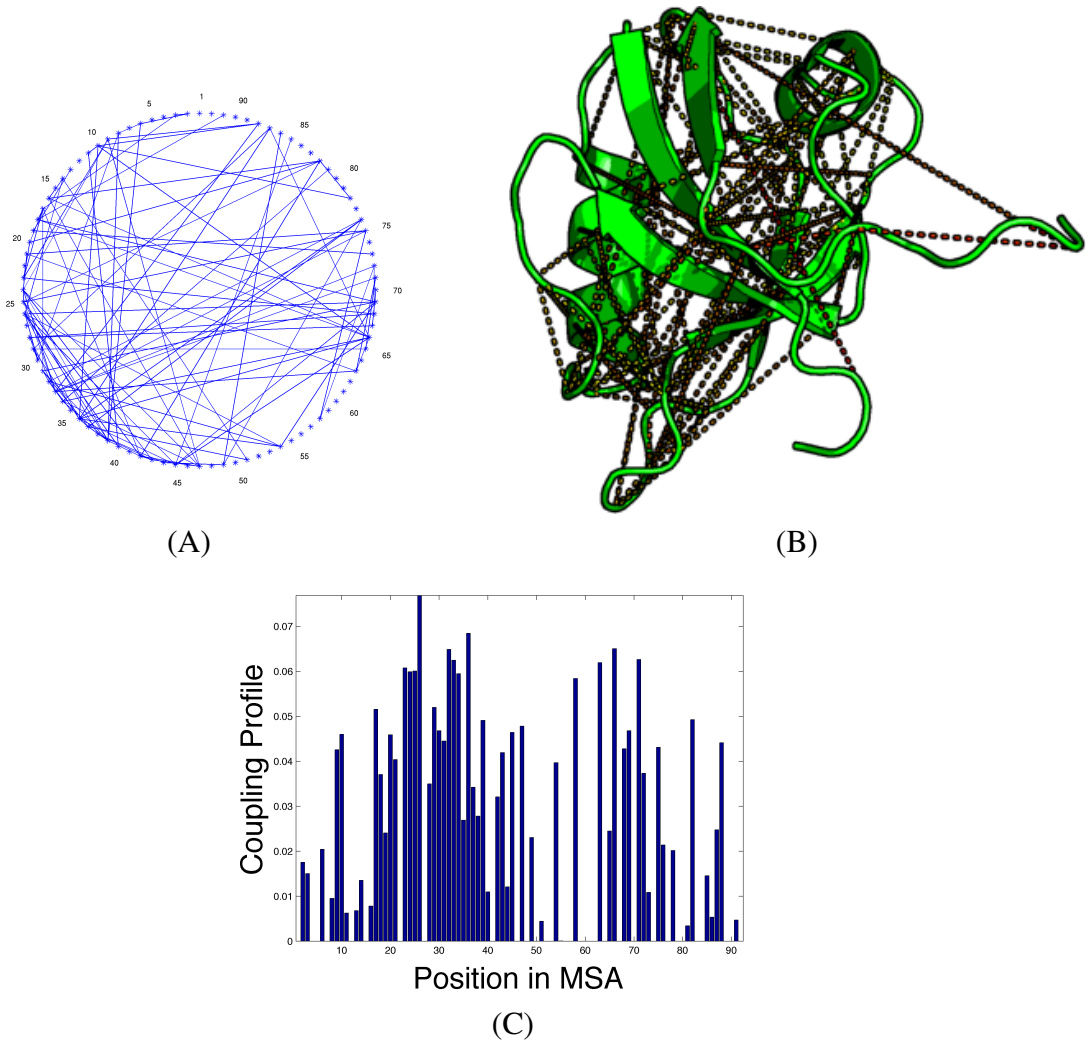


Figure 7.8: **PDZ domain model**. Edges returned by GREMLIN overlayed on a circle (**a**) and on the structure (**b**) of PDZ domain of PSD-95 (PDB id:1BE9). (**c**) Coupling profile (see text).

position 74 *directly* interacting with both these positions. By providing such a list of sparse interactions our model can provide a small list of hypotheses to an experimentalist looking for possible mechanisms of such allosteric behavior.

In addition to the pathway between HIS-70 and GLY-49, we also identify residues not on the pathway that are connected to other parts of the protein including, for example ASN-61 of the protein. This position is connected to ALA-88 and VAL-60 in our model, and does not appear in the network suggested by Lockless and Ranganathan [1999], but has been implicated by NMR experiments [Fuentes et al., 2004] as being dynamically linked to the active site.

From our studies on the PDZ and WW families we find that GREMLIN produces higher quality models than GMRC and profile HMMs, and identifies richer sets of interactions. In the following section we consider the application of GREMLIN to a larger subset of the PFAM database. Since the greedy algorithm of GMRC does not scale to large families, our experiments are restricted to comparing the performance of GREMLIN with that of profile HMMs.

7.1.6 Large-scale analysis of families from Pfam

We selected all protein families from PFAM [Bateman et al., 2002] that had at least 300 sequences in their seed alignment. We restricted ourselves to such families because the seed alignments are manually curated before depositing and are therefore expected to have higher quality than the whole alignments. We pre-processed these alignments to remove redundant sequences (sequence similarity $> 80\%$) in order to generate non-redundant alignments. From each alignment, we then removed columns that had gaps in more than half the sequences, and then removed sequences in the alignment that had more than insertions

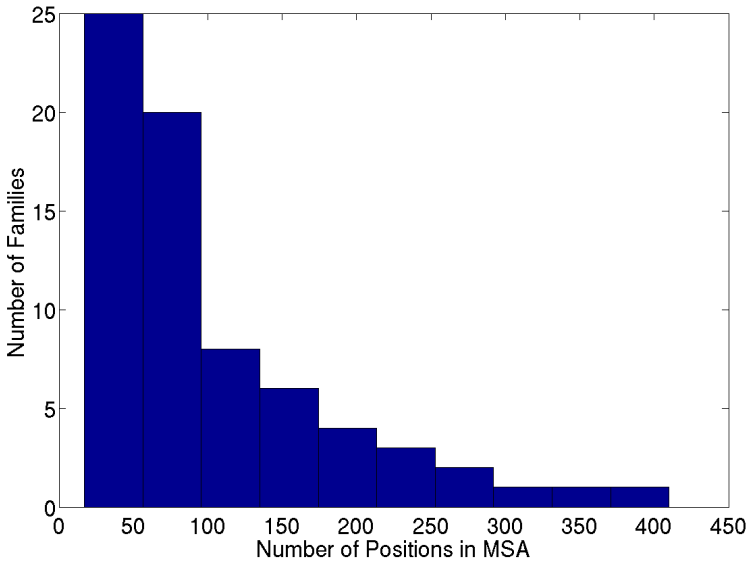
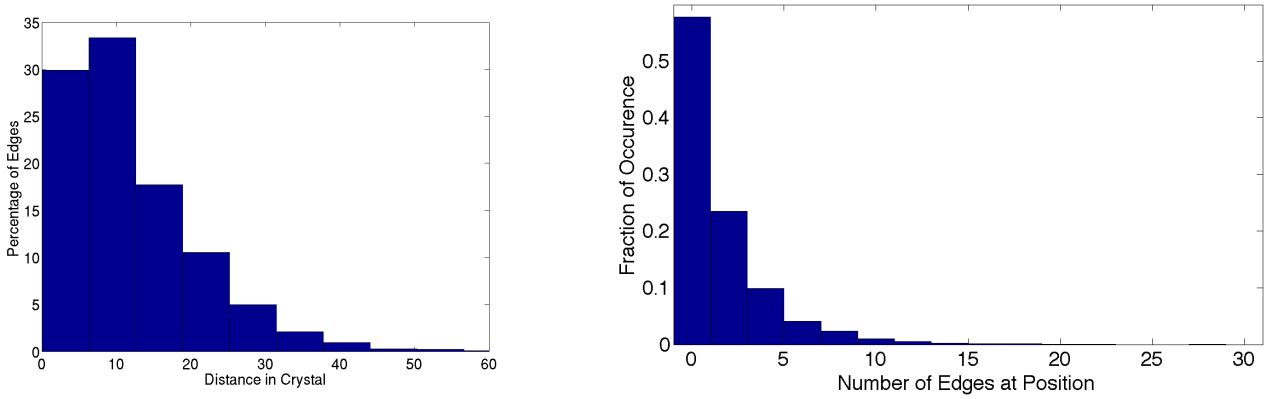


Figure 7.9: Histogram of MSA lengths of the 73 PFAM families in our study.

at more than 10% of these columns. Finally, we removed sequences that had more than 20% gaps in their alignment. If this post-processing resulted in an alignment with less than 300 sequences, it was dropped from our analysis. 71 families remained at the end of this process. These families varied greatly in their length with the shortest family having 15 positions and the longest having more than 450 positions and the median length being 78 positions. Figure 7.9 shows the distribution of lengths.

For each of these families, we created a random partition of the alignment into training (with 2/3 of the sequences) and test (with 1/3 of the sequences) alignments and trained an MRF using our algorithm. As mentioned earlier, we chose λ_e by performing 20 random permutations of each column and choosing the smallest λ_e that gave zero edges on all 20 permutations. As a baseline comparison, we also trained a profile-HMM using the Bioinformatics toolkit in Matlab on the training alignments. We then used the learnt models to



(A) (B)

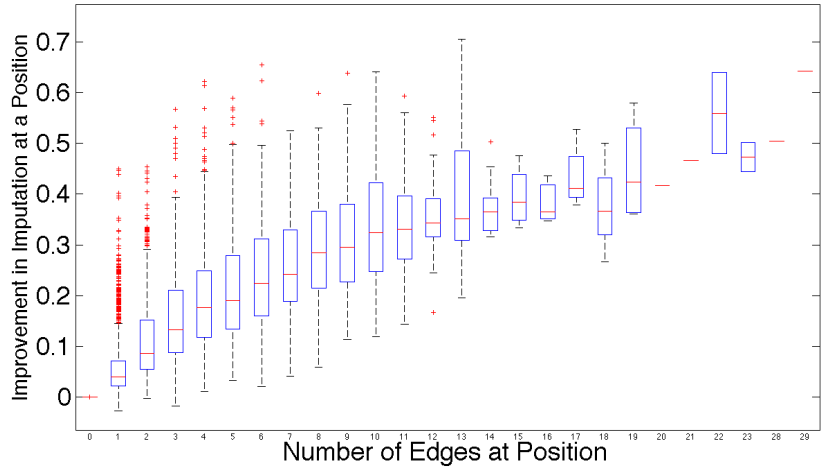
Figure 7.10: (A) Histogram of the distance in crystal structure. (B) Degree distribution across all proteins.

impute the composition of each position of the test MSA and computed the overall and per-position imputation errors for both models. We provide the models and detailed analyses for each family on a supporting website * and focus on overall trends in the rest of this section.

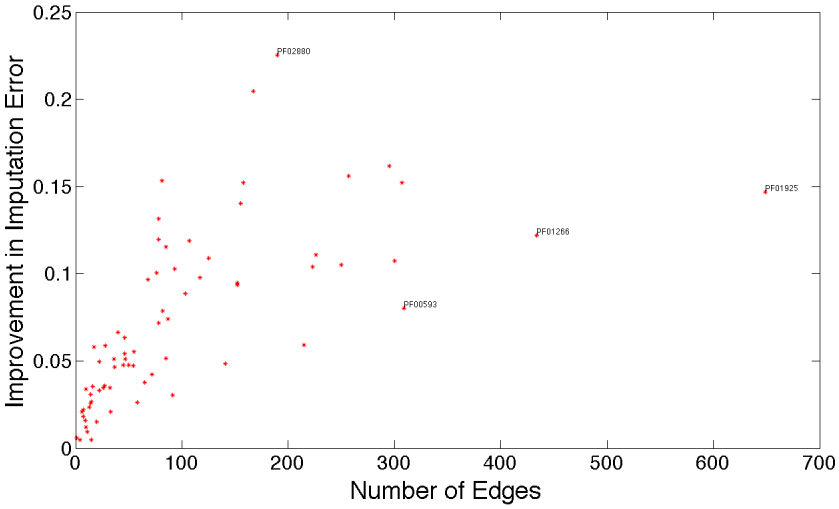
Figure 7.10 shows the histograms of the distance between residues connected by an edge and the degree of the nodes. Approximately 30% of the edges are between residues that are more than 10 Å of each other. That is, GREMLIN learns edges that are different than those that would be obtained from a contact map. Despite the presence of long-range edges, GREMLIN does learn a sparse graph; most nodes have degree less than 5, and the majority have 1 or fewer edges.

Fig. 7.11-(A) shows a boxplot demonstrating the effect of incorporating co-evolution information according to our model. The y-axis shows the decrease in the per-position imputation error when moving from a profile-HMM model to the corresponding MRF, while the x-axis bins this improvement according to the number of edges in the MRF at that

*http://www.cs.cmu.edu/~cjl/gremlin/



(A)



(B)

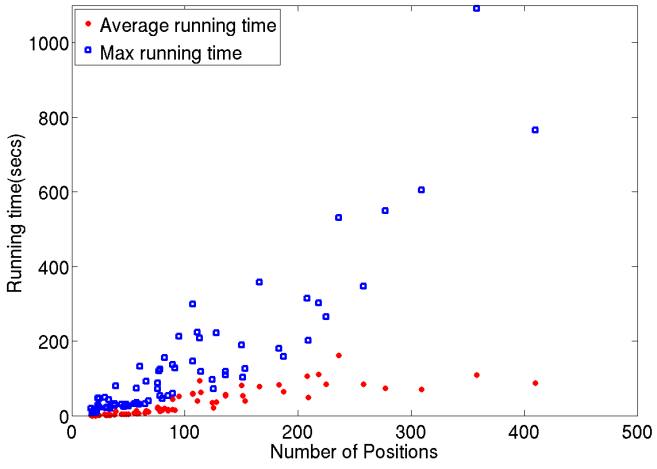
Figure 7.11: (A) Boxplot displaying the effect of coupling on improvement in imputation error at a position when compared to a profile-HMM. The median imputation error shows a near-linear decrease as the number of neighbors learnt by the model increases. (B) Improvement in overall imputation error across all positions for each family.

position. In each box, the central red line is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually with red '+' marks. As the figure shows, moving from a profile-HMM model to an MRF never hurts: for positions with 0 edges, there is no difference in imputation; for positions with at least one edge, the MRF model *always* results in lower error. While this is not completely surprising given that the MRF has more parameters and is therefore more expressive, it is not obvious that these parameters can be learnt from such little data. Our results demonstrate that this is indeed possible. While there are individual variations within each box, the median improvement in imputation error shows a clear linear relationship to the number of neighbors of the position in the model. This linear effect falls off towards the right in the high-degree vertices where the relationship is sub-linear. Fig. 7.11-(B) shows the effect of this behavior on the improvement in overall imputation error across all positions for a family.

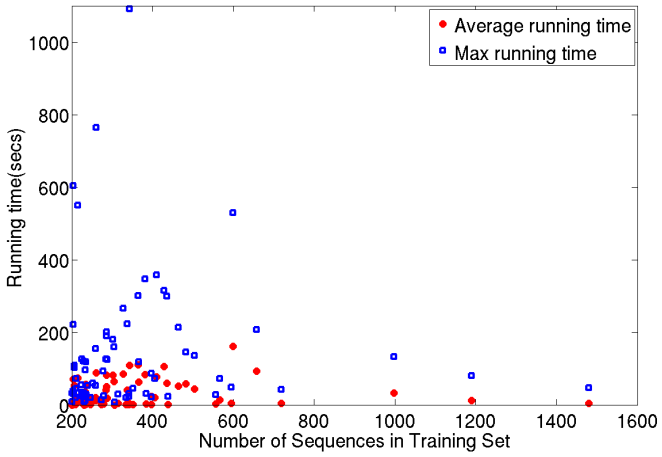
7.1.7 Computational efficiency

In this subsection we briefly discuss the computational efficiency of GREMLIN . The efficiency of GREMLIN was measured based on the running time (i.e. CPU seconds until a solution to the convex optimization problem is found). GREMLIN was run on a 64 node cluster. Each node had 16GB DRAM and 2xquad-cores (each with 2.8-3 GHZ), allowing us to run 512 jobs in parallel with an average of 2GB RAM per job.

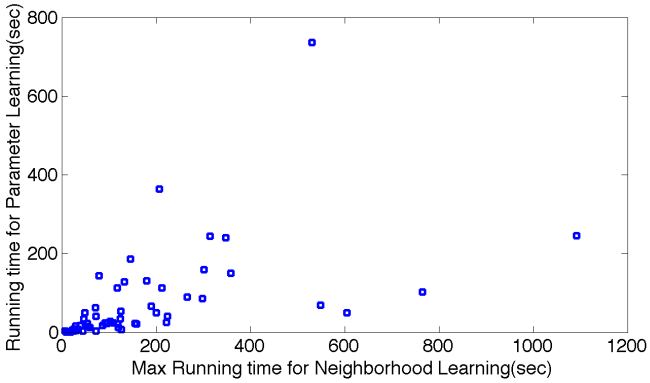
Fig. 7.12 shows a plot of the running time for a given λ_e on all the PFAM MSAs. Fig. 7.12-(A) plots the running time for learning the neighbors of a position, against the number of columns (positions) in the MSA (A) while 7.12-(B) plots it against number of rows (sequences) in the training MSA. In both, the average running time *per column* is



(A)



(B)



(C)

Figure 7.12: (A) Number of Positions in the MSA versus runtime of Neighborhood learning (in seconds) (B) Number of sequences in the MSA versus runtime of Neighborhood learning (C) Runtime of Neighborhood learning versus runtime of Parameter learning

shown in red circles. While learning the neighbors at a position, since GREMLIN is run in parallel for each column of the MSA, the actual time to completion for each protein depends on the maximum running time across these columns. This number is shown in blue squares. Fig. 7.12-(C) plots the running time for parameter learning against the maximum running time to learn the neighbors at a position. Recall that this task is performed serially. As the figure demonstrates, GREMLIN takes roughly similar amounts of time in its parallel stage (neighborhood learning) as it does in its serial stage (parameter learning).

The plots show that the running time has an increasing trend as the size of the MSA increases (number of positions and number of sequences). Also, the dependence of the running time on the number of columns is stronger than its dependence on the number of rows. This is consistent with the analysis in Wainwright et al. [2007] which shows that a similar algorithm for structure learning with a pure L_1 penalty has a computational complexity that scales as $\mathcal{O}(\max(n,p)p^3)$, where n corresponds to the number of rows and p to the number of columns in the MSA.

7.2 Discussion

7.2.1 Related Work

The study of co-evolving residues in proteins has been a problem of much interest due to its wide utility. Much of the early work focused on detecting such pairs in order to predict contacts in a protein in the absence of a solved structure [Altschuh et al., 1988, Göbel et al., 1994] and to perform fold recognition. The pioneering work of Lockless and Ranganathan [1999] used an approach to determine probabilistic dependencies they call SCA and observed that analyzing such patterns could provide insights into the allosteric behav-

ior of the proteins and be used to design new sequences [Socolich et al., 2005]. Others have since developed similar methods [Fatakia et al., 2009, Fodor and Aldrich, 2004, Fuchs et al., 2007]. By focusing on co-variation or probabilistic *dependencies* between residues, such methods conflate direct and indirect influences and can lead to incorrect estimates. In contrast, Thomas et al. [2008b] developed an algorithm for learning a Markov Random Field over sequences. Their constraint-based algorithm proceeds by identifying conditional independencies and adding edges in a greedy fashion. However, the algorithm can provide no guarantees on the correctness of the networks it learns. They then extended this approach to incorporate interaction data to learn models over pairs of interacting proteins [Thomas et al., 2009c] and also develop a sampling algorithm for protein design using such models [Thomas et al., 2009b]. More recently, Weigt et al. [2009] use a similar approach to determine residue contacts at a protein-protein interface. Their method uses a gradient descent approach using Loopy Belief Propagation to approximate likelihoods. Additionally, their algorithm does not regularize the model and may therefore be prone to over-fitting. In contrast, we use a Pseudo-Likelihood as our objective function thereby avoiding problems of convergence that Loopy BP based methods can face and regularize the model using block regularization to prevent over-fitting.

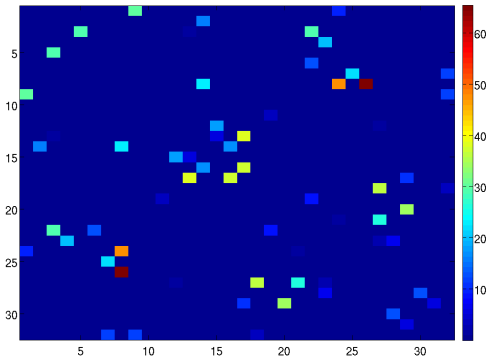
Block regularization is most similar in spirit to the group Lasso [Yuan and Lin, 2006] and the multi-task Lasso [Argyriou et al., 2007]. Lasso [Tibshirani, 1994] is the problem of finding a linear predictor, by minimizing the squared loss of the predictor with an L_1 penalty. It is well known that the shrinkage properties of the L_1 penalty lead to sparse predictors. The group Lasso extends this idea by grouping the weights of some features of the predictor using an L_2 norm, [Yuan and Lin, 2006] show that this leads to sparse selection of groups. The multi-task Lasso solves the problem of multiple separate (but

similar) regression problems by grouping the weight of a single feature across the multiple tasks. Intuitively, we solve a problem similar to a group Lasso, replacing the squared loss with an approximation to the negative log-likelihood, where we group all the feature weights of an edge in an undirected graphical model. Thus, sparse selection of groups gives our graphs the property of structural sparsity.

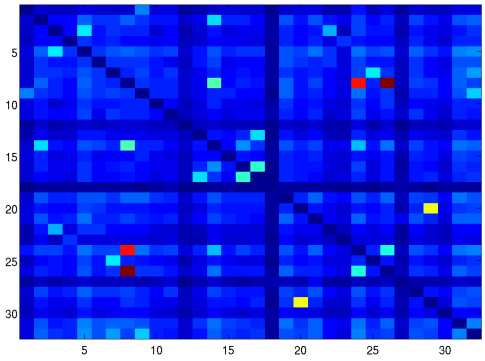
Lee and co-workers [Lee et al., 2007a] introduced structure learning in MRFs with a pure L_1 penalty, but do not go further to explore block regularization. They also use a different approximation to the likelihood term, using Loopy Belief Propagation. Schmidt and co-workers [Schmidt et al., 2008] apply block-regularized structure learning to the problem of detecting abnormalities in heart motion. They also developed an efficient algorithm for tractably solving the convex structure learning problem based on projected gradients.

7.2.2 Mutual Information performs poorly in the structure learning task

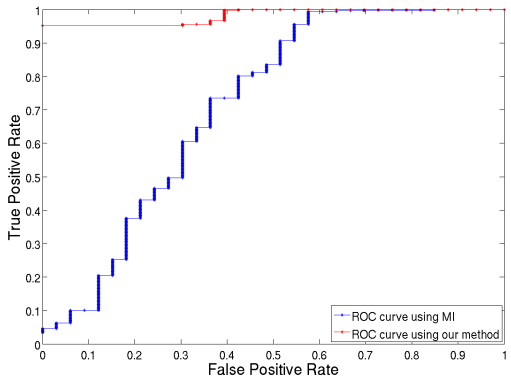
One of the key advantages of a graphical model based approach to modeling protein families is that the graph reveals which interactions are direct and which are indirect. One might assume that alternative quantities, like Mutual Information, might yield similar results. We now demonstrate with an example that a simple Mutual Information based metric cannot distinguish well between direct and indirect interactions. Fig. 7.13-(A) shows the adjacency matrix of a Probabilistic Graphical Model. The elements of the matrix are color-coded by the strength of their interaction: blue represents the weakest interaction (of strength 0, i.e. a non-interaction) and red the strongest interaction in this distribution. Fig. 7.13-(B) shows the mutual information induced between the variables by this distribution as measured from 500 sequences sampled from the graphical model (the diagonal elements



(A)



(B)



(C)

Figure 7.13: (A) Adjacency matrix of a Boltzmann distribution colored by edge strength. (B) Mutual Information between positions induced by this Boltzman distribution. While the mutual information of the strongest edges is highest; a large fraction of the edges have MI comparable to many non-interactions. (C) Shows the weak ability of MI to distinguish between edges and indirect interactions in contrast to GREMLIN . AUC using MI: 0.71; AUC using GREMLIN : 0.98.

of the mutual information matrix have been omitted to highlight the information between different positions). While it may appear visually that (B) shares a lot of structure with (A), it isn't actually the case. In particular, the edges with the highest mutual information indeed tend to be direct interactions; however a large fraction of the direct interactions might not have high MI. This is demonstrated in Fig. 7.13-(C) where MI is used as a metric to classify edges into direct and indirect interactions. The blue line shows the ROC curve using MI as a metric and has only moderate discriminatory power for this task (AUC: 0.71). In contrast, our approach, shown in red, is much more successful at discriminating between direct and indirect interactions: the AUC of our approach is a near-perfect 0.98.

7.2.3 Influence of Phylogeny

One limitation associated with a sequence-only approach to learning a statistical model for a domain family is that the correlations observed in the MSA can be inflated due to phylogeny [Pollock and Taylor, 1997, Felsenstein, 2003]. A pair of co-incident mutations at the root of the tree can appear as a significant dependency even though they correspond to just once co-incident mutation event. To test if this was the case with the WW domain, we constructed a phylogenetic tree from the MSA using Junes-Cantor measure of sequence dissimilarity. In the case of WW, this resulted in a tree with two clear sub-trees, corresponding to two distinct (nearly equal-sized) clusters in sequence space. Since each sub-tree had a number of sequences, we re-learnt MRFs for each sub-tree separately. The resulting models for each sub-tree did not vary significantly from our original models – a case that would have occurred if there were co-incident mutations at the root that lead to spurious dependencies. Indeed the only difference between the models was in the C-terminal end was an edge between positions 1 and 2 that was present in sequences from the first sub-tree but

was absent in the second sub-tree. This occurred because in the second sub-tree, these positions were completely conserved due to which our model was not able to determine the dependency between them. While this does not eliminate the possibility of confounding due to phylogeny, we have reason to believe that our dependencies are robust to significant phylogenetic confounding in this family. A similar analysis for the PDZ domain, found 3 sub-trees, and again we found that the strongest dependencies were consistent across models learnt on each sub-tree separately. Nevertheless, we believe that incorporating phylogenetic information into our method is an important direction for future research.

Chapter 8

Graphical Games over Fitness Landscapes

The previous chapters have dealt with probabilistic scenarios and modeled the equilibria that arise in such scenarios. In the case of GOBLIN, the motivation for such equilibrium distributions arose directly from the thermodynamics of the physical system being modeled; GREMLIN was motivated by analogy as a coarse-grained model of the physical systems considered by GOBLIN. In contrast, this chapter deals with scenarios that are strategic in nature. The laws of thermodynamics, and the resulting Boltzmann equilibria don't apply in such scenarios. We will look at alternate notions of equilibria that are applicable in such scenarios with an emphasis on their computability.

Such strategic scenarios are ubuiquitous in Biology. Indeed, it can be argued that the process of evolution is best modeled in such a strategic manner: organisms adapt to the behavior of other organisms and the environment to maximize their chances of survival. While these adaptations by themselves are not the output of strategic behavior, the combination of

random mutations with non-random selective pressures can, and is often modeled as such.

Game theory deals with the behavior of individual entities in such strategic scenarios. It has been widely used to model evolutionary behavior since the pioneering work of Hamilton [1964a,b] and Smith and Price [1973]. However, these analyses have been largely focused on modeling the outcomes of simple strategies under ideal settings. For example, the classical "Hawk-Dove" game described in detail later in this chapter analyzes outcomes of a game between two populations: one population is aggressive and seeks conflict ("Hawk") while the other is submissive and seeks to avoid conflict ("Dove").

In contrast, we will analyze games where the different strategies correspond to the amino acid composition of a position in the protein (or similar building blocks of molecular diversity). By modeling the smallest unit of evolutionary change (the mutation of an amino acid) this chapter lays the groundwork for an approach that models strategic behavior in much higher-resolution than previous game-theoretic approaches in biology.

This chapter starts with an introduction to the basics of Game theory focussing on classical representations of strategic games, notions of equilibria and their computability. The subsequent section will then introduce more recent developments in representations of games, followed by a Linear Programming formulation of equilibria on acyclic and cyclic graphs and introduce new relaxations that allow efficient computation of bounds on the properties of these equilibria. The section introducing Game theory follows standard treatments. A reader interested in more details could refer [Leyton-Brown and Shoham, 2008] and references therein.

8.1 Introduction to Game Theory

Definition 1 (Normal form Game)**.** *An n-person normal-form game G is a tuple (X, A, u) where*

- *X is a finite set of n players;*
- *$A = \prod A_i$ where A_i is the set of actions available to player i;*
- *$u = (u_1, \ldots u_i, \ldots u_n)$ where $u_i : A \to \mathcal{R}$ is a* utility *function that player i seeks to maximize.*

Given the set of possible actions available to a player, a strategy is a choice of action(s). A strategy that chooses a single action to play is called a *pure strategy*. In contrast, a *mixed strategy* each player *independently* chooses a probability distribution over the set of actions and chooses an action by sampling from this distribution.

The n player normal form game is commonly represented as n, n dimensional matrices, one for each u_i. We will first look at specific examples of two player normal form games while introducing additional concepts from the theory of games. In subsequent sections, we will introduce structured representations that exploit sparsity to efficiently store the game more compactly and develop algorithms that compute equilibria.

8.1.1 Two-player zero sum games and Minimax

Definition 2 (Two player zero-sum)**.** *A game with $n = 2$ and $u_1 = -u_2$.*

Theorem 2 (The Minimax theorem. von Neumann, 1951)**.** *[Von Neumann, 1928, Von Neumann et al., 2007] For every two-person zero-sum game with finite strategies, there exists a*

	R	P	S
R	(0,0)	(-1, 1)	(1,-1)
P	(1,-1)	(0,0)	(-1, 1)
S	(-1,1)	(1, -1)	(0,0)

Table 8.1: "Rock Paper Scissors". Rock beats Scissors; Paper beats Rock and Scissors beats Paper. The minimax mixed strategy places equal probability on each action for both players and results in a value of 0

	H	T
H	(1,-1)	(-1, 1)
T	(-1,1)	(1, -1)

Table 8.2: "Matching Pennies". Row player wins if both choose same strategy; column player wins otherwise

value V and a mixed strategy for each player such that (a) given the column player's strategy, the best payoff possible for the row player is V, and (b) given the row player's strategy, the best payoff possible for the column player is -V.

The minimax value, and the corresponding strategy reflect the best worst-case scenario for both players. Rational players in a two player zero-sum game would therefore play the minimax strategy.

Tab. 8.2 and Tab. 8.1 show the utilities of two classical zero-sum games: "Matching Pennies" and "Rock Paper Scissors". By convention, the first value in each entry of the pay-off matrix is the utility of the row-player while the second value is the utility of the column player. Naturally, in zero sum games these values sum to zero in zero-sum games.

8.1.2 Nonzero sum games and Equilibria

Games in which $u_1 \neq -u_2$ are called non-zero sum games. In general, in such games, it is not possible to define a unique minimax value. A generalization of the minimax idea to non-zero sum games is the notion of an equilibrium.

The idea is as follows: suppose $s = (s_1, s_2)$ was a particular strategy profile. Given that the column player is playing s_2, the row player has no incentive to deviate from s_1 if it is the best response to s_2. Additionally, if s_2 is the best response for the column player given that the row player is playing s_1, then neither has an incentive to deviate from this strategy. Such a state is called an equilibrium state of the game.

Definition 3 (Best-response)**.** *Player i's best response to the strategy profile s_{-i} is a (possibly mixed) strategy $s_i^* \in S_i$ such that $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}) \forall s_i \in S_i$*

Definition 4 (Nash Equilibrium)**.** *A strategy profile $s = (s_1, \ldots s_i, \ldots s_n)$ is a nash equilibrium if s_i is the best responses to s_{-i} for all players i.*

In other words, a strategy profile is a Nash Equilibrium (NE) if no player has an incentive to unilaterally deviate from this profile.

Theorem 3 (Existence of Nash Eq. Nash, 1951)**.** *Every game with a finite number of players and action profiles has atleast one Nash Equilibrium.*

Tab. 8.3 and Tab. 8.4 show the utilities of two classic two player non-zero sum games.

8.1.3 Correlated Equilibria

If we associate the strategy profile $s = (s_1, \ldots, s_n)$ with a probability vector $\pi = (\pi_1, \ldots, \pi_n)$ where π_i is the (possibly) mixed strategy of player i, the requirements for π to be a Nash

	C	S
C	(-4,-4)	(-1, -5)
S	(-5, -1)	(-2, -2)

Table 8.3: Prisoner's dilemma. Two accomplices in a crime are caught and interrogated separately. Each can choose to confess (C) or stay silent (S). The Nash Equilibrium of this game is (C,C)

	B	S
B	(5,6)	(1, 1)
S	(2,2)	(6, 5)

Table 8.4: The "Battle of the Sexes". A couple can either go visit a Baseball game (B) or a softball game (S). They have differing incentives for each but prefer going together to going alone. This game has two Nash Equilibria: (B,B) and (S,S)

Equilibrium can be written as:

$$\forall i, \forall a^i, a'_i \in A^i, \sum_{a^{-i}} \pi(a^i, a^{-i}) u_i(a^i, a^{-i}) \geq \sum_{a^{-i}} \pi(a^i, a^{-i}) u_i(a'_i, a^{-i})$$

where $\pi(a^i, a^{-i}) = \pi(a^1) \times \ldots \pi(a^i) \ldots \pi(a^n)$ since the definition of a mixed strategy profile specifies that each player samples from π_i independent of other players.

Relaxing this requirement of independence results in equilibria called correlated equilibria (CE)[Aumann, 1974]. Thus, a CE is any joint distribution π over the player's actions such that

$$\forall i, \forall a^i, a'_i \in A^i, \sum_{a^{-i}} \pi(a^i, a^{-i}) u_i(a^i, a^{-i}) \geq \sum_{a^{-i}} \pi(a^i, a^{-i}) u_i(a'_i, a^{-i})$$

It is easy to see that this relaxation results in constraints that are linear in π. Thus the

problem of finding a π that is a CE can be solved using a Linear Program as follows:

$$
\begin{aligned}
\min \quad & 1 \\
\text{s.t.} \quad & \forall i, \forall a^i, a'^i \in A^i, \textstyle\sum_{a^{-i}} \pi(a^i, a^{-i}) u_i(a^i, a^{-i}) \geq \sum_{a^{-i}} \pi(a^i, a^{-i}) u_i(a'^i, a^{-i}) \\
& \forall a \in A, \pi(a) \geq 0 \\
& \textstyle\sum_{a \in A} \pi(a) = 1
\end{aligned}
$$

CE have several properties that make it more attractive than NE: they can lead to more efficient outcomes; they can be viewed as a Bayesian alternative to NE [Aumann, 1987], they are easier to compute than NE (which are PPAD-complete [Daskalakis et al., 2009]). Finally, there exist natural algorithms that allow players of a game to converge to a CE[Foster and Vohra, 1997]; no such algorithms are known to exist for NE in general.

Fig. 8.1 shows the CE and NE constraints for the "Battle of the Sexes" game. Each point in the green polytope corresponds to a valid CE. The hyperbolic surface represents the points that satisfy the constraint $\pi = \pi_1 \pi_2$. Any point that is both a CE and on the hyperbolic surface is an NE. This game has 3 NE.

In an n-player game where each player has m actions, this LP formulation has $O(n \times m^2 + m^n)$ constraints over m^n variables. While this might appear to be excessive, it must be noted that the matrix representation of the game itself takes $O(n \times m^n)$ space. In the following section, we'll explore formulations that could potentially result an exponential saving in the space required to represent the game and corresponding LP formulations over the sparse representations for computing CE.

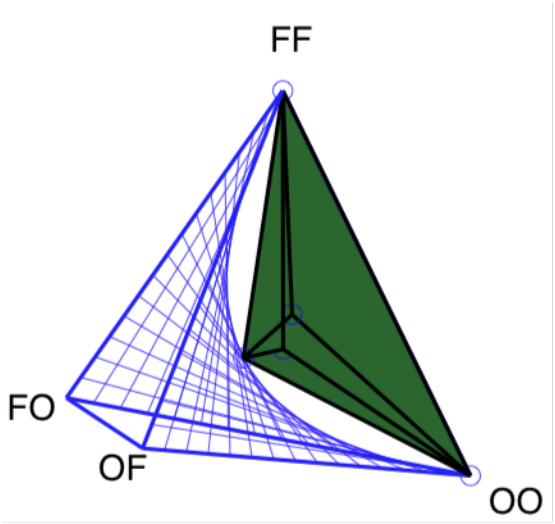


Figure 8.1: The constraints on Nash and Correlated Equilibria illustrated for a simple two-player game. The set of correlated equilibria form a linear polytope while the constraints for a Nash Equilibrium form a non-convex surface. Figure thanks to Prof. Geoff Gordon

8.2 Graphical Games

A Graphical Game (G, M) is a tuple consisting of a graph G and a set of local utility functions $M = (M_1, ..., M_n)$ that compactly represents a multi-player game. Each player is represented by a vertex in the graph G and edges are drawn between vertices i and j if the utility function of either depends on the actions of the other. If M_i depends on the actions of all remaining players, this representation offers no advantages; however, when M_i only depends on a subset of players, the graphical game representation can be more compact than the matrix form representation. In particular, the graphical game requires space exponential in the largest degree d of the graph.

This representation is evocative of a Markov Random Field. While MRFs represent probabilistic interactions, the graphical game represents strategic interactions. An important result due to Kakade et al. [2003] shows that there is a natural relationship between the

structure of the graphical game and the probabilistic relations of a subset of its CE. To describe this result, we first need the following definition:

Definition 5 (Expected Payoff-Equivalence)**.** *For a graph G, two distributions P, Q are equivalent up to expected payoff if for all players i, and actions $\vec{a_i}$ over Δ_i, $\mathbb{E}_{a\sim P}u_i(\vec{a_i}) = \mathbb{E}_{a\sim Q}u_i(\vec{a_i})$.*

Here, Δ_i refers to the set containing i and its neighbors $N(i)$ in G, and $\vec{a_i}$ refers to an assignment of actions to Δ_i.

Kakade et al. [2003] show that any CE of the game that does not factorize according to G is equivalent (up to expected payoff) to some CE of the game that does. The rest of this work therefore limits itself to the CE that factorize according to G.

8.3 CE in Graphical Games

Before we discuss algorithms, it will be useful to define some notation and sets of interest. We use indexed superscripts to denote components of these assignments: $\vec{a_i}^j$ corresponds to the assignment of an action to player j according to $\vec{a_i}$. We use the notation $\vec{a_i}[i : a']$ to denote the vector that is obtained by replacing the i^{th} component of $\vec{a_i}$ with action a'. $\{\mu_i\}$ will refer to the set of marginals that each store the marginal probability over variable i and its neighbors.

8.3.1 Exact CE

If G has no cycles, computing a CE of the graphical game can be solved satisfying a set of linear constraints. An important departure from the standard MRF treatment is that the

potentials of a graphical game are usually *not* pairwise. This is due to the fact that the utility is a function from an assignment $\vec{a_i}$ of actions to all neighbors of i. Thus, in general, the smallest region over i that would allow incorporation of the CE constraints would need to include all neighbors of i.

The marginal polytope $\mathcal{M}_\Delta$ is defined to be the set of marginals $\mu_i(\vec{a_i})$ (over sets of variables of size Δ_i) that are realizable by a joint distribution. This is analogous to the common definition of a marginal polytope over pair-wise marginals commonly used in approximate inference [Wainwright and Jordan, 2008], a set we shall refer to as $\mathcal{M}_2$.

$$\begin{aligned}\mathcal{M}_\Delta(G) &= \{\mu \in \Re^d | \exists p \text{ with marginals } \mu_i(\vec{a_i})\} \\ \mathcal{M}_2(G) &= \{\mu \in \Re^d | \exists p \text{ with marginals } \mu_i(a_i, a_j) \forall (i,j) \in E\}\end{aligned}$$

For an acyclic graph, the marginal polytope $\mathcal{M}_\Delta$ can be expressed using the following linear constraints over each Δ_i and their pair-wise intersections:

$$\begin{aligned}\text{Positivity:} \quad & \forall i, \forall \vec{a_i}, \mu_i(\vec{a_i}) \geq 0 \\ \text{Local Normalization:} \quad & \forall i, \sum \mu_i(\vec{a_i}) = 1 \\ \text{Local consistency:} \quad & \forall i, j, \forall N_{ij} \in \Delta_i \cap \Delta_j, \\ & \text{and assignments } y^{N_{ij}}\end{aligned}$$

$$\sum_{\vec{a_i}:\vec{a_i}^{N_{ij}} = y^{N_{ij}}} \mu_i(\vec{a_i}) = \sum_{\vec{a_j}:\vec{a_j}^{N_{ij}} = y^{N_{ij}}} \mu_j(\vec{a_j})$$

Since these constraints involve variables that are adjacent to each other in G, the set of marginals that satisfy them are referred to as $\mathrm{LOCAL}_\Delta(G)$. In a graph without cycles, $\mathrm{LOCAL}_\Delta(G) = \mathcal{M}_\Delta(G)$, since the joint can always be uniquely reconstructed from the

μ_i's in such graphs. Since the definition of a CE only imposes constraints on the solution, it is common to determine a CE that optimizes some objective function, for example, the expected social utility $\sum_i \sum_{\vec{a_i}} \mu(\vec{a_i}) u_i(\vec{a_i})$. We will use $f(\mu)$ to refer to any such objective function. While we assume that f is linear, our approach can be naturally extended to any class of functions as long as we can efficiently perform optimization, for example, minimizing convex functions.

The LP for computing a CE that maximizes some function f can be expressed as:

$$\max f(\mu)$$

s.t.

$$\forall i, \forall a, a' \in A^i, \sum_{\vec{a_i}:\vec{a_i}^i = a} \mu_i(\vec{a_i})(u_i(\vec{a_i}) - u_i(\vec{a_i}[i:a'])) \geq 0$$

(CE constraints)

$$\mu = [\mu_1 \dots \mu_n] \in \mathcal{M}_\Delta$$

8.3.2 Relaxations to outer marginal polytopes

If G has cycles, $\mathrm{LOCAL}_\Delta(G) \neq \mathcal{M}_\Delta(G)$. In such cases, one option is to run the algorithm on the junction tree $JT(G)$ instead of the original graph. Since by the junction tree property, $\mathrm{LOCAL}_\Delta(JT(G)) = \mathcal{M}_\Delta(JT(G))$ [Wainwright and Jordan, 2008] this process is guaranteed to give the exact CE. However, the algorithm has to maintain marginals over the size of tree-width of the graph which can be prohibitively expensive in games with many players or actions even if each vertex of the graph has a small degree. For example, a grid-structured graphical game of size $n \times n$ with k actions for each player has a maximum degree of 4

resulting in representation cost of $O(nk^4)$ but its tree-width is n.

If we are primarily interested in the value of the objective function $f(\mu)$ or in the marginal distribution induced by a CE, we can trade-off accuracy for time by approximating $\mathcal{M}_\Delta(G)$ with $\mathrm{LOCAL}_\Delta(G)$. This is analogous to a Generalized Belief Propagation approach to inference on an MRF. Since $\mathrm{LOCAL}_\Delta(G) \supset \mathcal{M}_\Delta(G)$, this is an outer relaxation of the problem. Solving it will therefore give a lower bound on the objective function.

The CE constraints require all marginals of size Δ_i implying that a relaxation looser than $LOCAL_\Delta$ is not possible for general games. However, in cases where the utility function additional structure, it is possible to construct further relaxations.

8.3.3 Pair-wise additive utility functions

Consider a setting where the utility of a particular player $u_i(\vec{a_i})$ can be expressed as the sum of pair-wise functions over the actions of the player's neighbors

$$u_i(\vec{a_i}) = \sum_{j \in N(i)} g_{i,j}(\vec{a_i}^i, \vec{a_i}^j)$$

The expression in the CE constraint can then be expressed as

$$\begin{aligned}
\sum_{\vec{a_i}:\vec{a_i}^i = a} \mu_i(\vec{a_i}) u_i(\vec{a_i}) = & \sum_{\vec{a_i}:\vec{a_i}^i = a} \mu_i(\vec{a_i}) \sum_{j \in N(i)} g_{i,j}(\vec{a_i}^i, \vec{a_i}^j) \\
= & \sum_{j \in N(i)} \sum_{\vec{a_i}:\vec{a_i}^i = a} \mu_i(\vec{a_i}) g_{i,j}(\vec{a_i}^i, \vec{a_i}^j) \\
= & \sum_{j \in N(i)} \sum_{a_j} \mu_i(a, a_j) g_{i,j}(a, a_j)
\end{aligned}$$

Thus, if the utility can be expressed as a sum of pair-wise functions, the constraint $\mu \in \mathcal{M}_\Delta$ in the LP for the exact CE can be replaced with $\mu \in \mathcal{M}_2$. It is possible to then construct an outer relaxation to $\mathcal{M}_2$ using local constraints over edges and vertices of the graph. We will therefore refer to this relaxation as $LOCAL_2$.

8.3.4 Cycle Inequalities

Cycle inequalities are among the constraints that are satisfied by $\mathcal{M}_2$ and not explicitly enforced by $LOCAL_2$. They arise from the following simple observation on a graph with binary variables: if we start from a node in a cycle and traverse the cycle to come back to the node this traversal must have seen an even number of edges where adjacent variables had different values (if not, the value at the end of the traversal must be different from at the beginning, a contradiction). The advantage of these constraints is that their violation can be detected and a violated constraint identified in graphs in polytime [Barahona and Mahjoub, 1986] by computing shortest paths in a related graph. Violated cycle inequalities are incorporated incrementally into the constraint set and the LP is re-solved until no more violations occur. Sontag and Jaakkola [2007] devise an extension of this idea for graphs over variables with k values that first constructs different binary instantiations of these variables and identifies violated constraints on these instantiations. Since there are 2^k possible instantiations for each variable in the graph, we restrict ourselves to the *k-projection graph*[Sontag and Jaakkola, 2007] that considers k different binary instantiations for a variable with k values.

We will refer to the set of solutions that satisfy these constraints as $Cycle - Ineq$. $LOCAL_2 + Cycle - Ineq$ will therefore refer to solutions that are in both $LOCAL_2$ and $CYCLE - Ineq$.

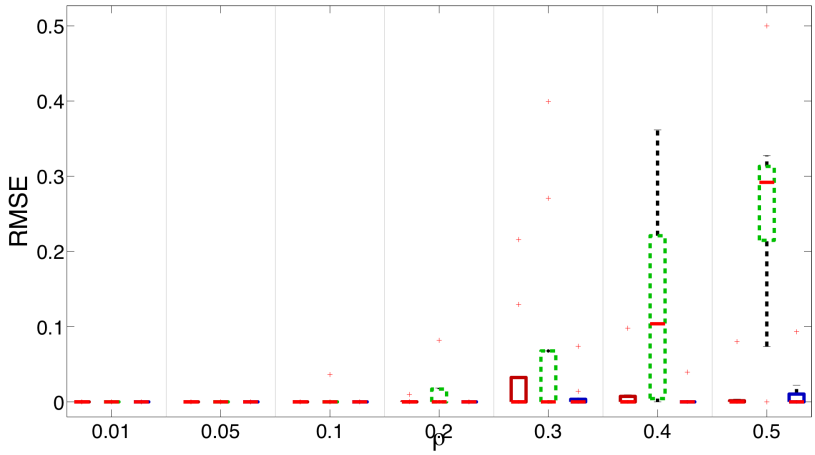
8.4 Simulation Results

We generated 16-node graphs corresponding to games with 16 players. Each player had two actions, and an edge was included between any pair of vertices with probability ρ. We generated graphs with the value of ρ varying from 0.01 and 0.5 (corresponding to a graph with half the density of a completely connected graph). We generated 10 graphs for each setting for ρ resulting in a total of 70 graphs. For each player, each element in the utility matrix was randomly generated from $\mathcal{U}(-1, 1)$. We note that our choice of values was limited by a need to be able to compute CE exactly so as to compare the accuracy of our relaxations. The relaxations themselves can be used on games with many more players and actions.

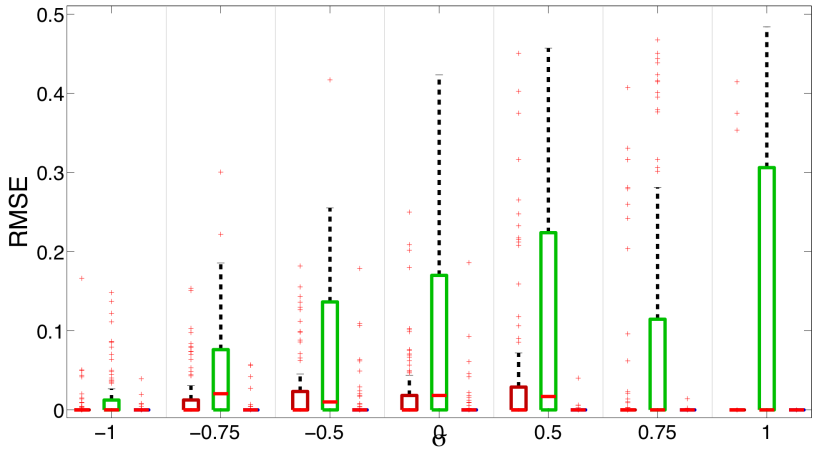
For each of these graphs, we computed exact Correlated Equilibria while optimizing for social utility and computed the marginal distributions of the CE strategy for each player. We then computed approximate marginal distributions that optimized the same objective function but made progressively looser outer relaxations: the first relaxation stored marginal distributions over a player and all its neighbors, for each player while the second relaxation only stored pair-wise marginal distributions, one for each edge in graph.

Fig. 8.2-A shows a boxplot of the error induced in the individual marginal distributions of each player due to approximate inference as ρ, the density of the graph, increases. In each group, the first box shows the error when using the $LOCAL_\Delta$ relaxation, the second box shows the errors when using $LOCAL_2$ and the third, when using $LOCAL_2 + Cycle - Ineq$. Increasing ρ has marginal effect on the error of $LOCAL_\Delta$ but increases the error of $LOCAL_2$. Remarkably, adding the cycle inequality constraints drastically improves the accuracy of $LOCAL_2$: the error is nearly zero across all settings of ρ.

To test the sensitivity of these results to the choice of utility and objective function, we



(A)



(B)

Figure 8.2: Boxplot showing L_2 error of marginal distributions on randomly generated graphs with utilities sampled from $\mathcal{U}(-1, 1)$ as the density, ρ, increases (A) and from a coupled distribution as the correlation between utilities, σ, increases(B). In each group, the first boxplot corresponds to $LOCAL_\Delta$, the second to $LOCAL_2$ and the third to $LOCAL_2 + Cycle - Ineq$.

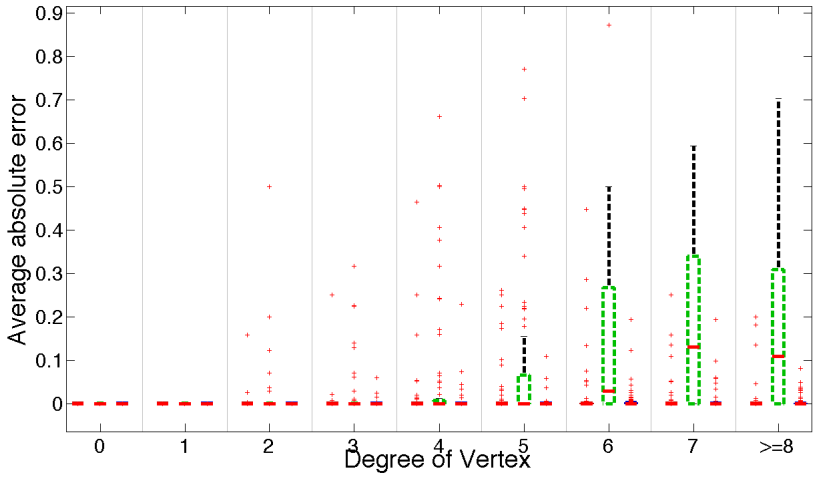


Figure 8.3: Boxplot showing absolute error of the marginal distribution at a variable across all simulated graphs as a function of the degree of the position. As the degree increases, the error tends to increase although the median error remains very low. Order within each group: $LOCAL_{\Delta}, LOCAL_2, LOCAL_2 + Cycle - Ineq$.

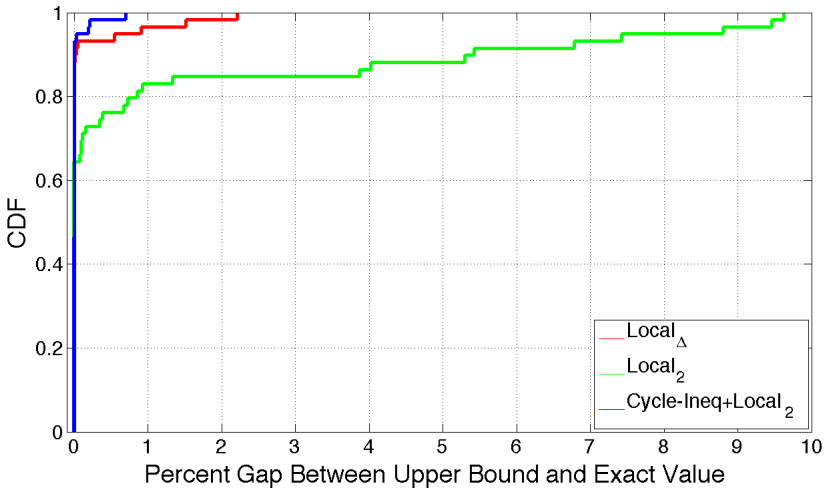


Figure 8.4: Histogram of the percentage gap between the upper bound of the objective function using an outer approximation and the exact value. Both $LOCAL_2$ and $LOCAL_{\Delta}$ relaxations produce bounds that are remarkably close to the actual value of the objective function.

generated 100, 16-node graphs with $\rho = 0.3$. For each of these graphs, we then generated graphical games where the utilities for each action-pair on each edge were sampled from a bi-variate normal distribution $\mathcal{N}([0,0],[1,\sigma;\sigma,1])$ where $-1 \leq \sigma \leq 1$ resulting in a total of 700 graphical games with wide variety of utilities. At $\sigma = -1$, for example, the utilities for any action pair on each edge sum to zero, while for $\sigma = 1$, the utilities on a particular edge are the same for both players. Instead of the social utility, we optimized a random linear function of the marginals. Fig. 8.2-B shows the error in marginals using the outer relaxations as σ increased. $LOCAL_\Delta$ still has consistently low error while the spread of the distribution of errors for $LOCAL_2$ appears to increase as utilities become more coupled. Again incorporating the cycle inequalities to $LOCAL_2$ drastically improves the accuracy to near zero error.

Fig. 8.3 shows a boxplot of the error induced in the individual marginal distributions of each player due to approximate inference as ρ, the density of the graph, increases. As previously, $LOCAL_\Delta$ and $LOCAL_2 + Cycle - Ineq$ are accurate in all settings while $LOCAL_2$'s error increases as the degree of the vertex increases.

Fig. 8.4 shows the CDF of the percentage gap between the upper bound as computed by an outer relaxation and the exact value using the marginal polytope. The bounds provided by all relaxations are $< 10\%$ away from the optimal value for all graphs with the maximum error of $LOCAL_2 + Cycle - Ineq$ being *less than* 1%.

Chapter 9

Application: Games of Molecular Conflict

9.1 Introduction

An important type of protein interactions is their interaction with drugs. Drug-design is the process of engineering a drug to selectively bind to specific proteins and modify their behavior. In this task, it is traditional to assume that the protein itself does not mutate. While this approach has led to a lot of successful drugs, it has also led to the evolution of proteins that evade this interaction through mutations. This phenomena is commonly referred to as *drug resistance*.

This resistance could be caused due to mutations in the drug target or elsewhere that impair the functioning of the drug, or, in cases of single-celled organisms, through the direct transfer of entire genes (horizontal gene transfer) between organisms. Drug resistance is a significant topical health problem with the WHO estimating that, "...resistance to first-line

drugs in most of the pathogens causing these diseases ranges from zero to almost 100%. In some instances resistance to second- and third line agents is seriously compromising treatment outcome. . . ." It is particularly common in scenarios where rapid evolution is possible – cancer [Sakai et al., 2008, Eliopoulos et al., 1995] and HIV [Clavel and Hance, 2004] for example – although bacteria and other microorganisms can also exhibit drug resistance from prolonged exposure [Soulsby, 2005].

To address this problem, this chapter develops a technique called GAMUT(GAmes of MolecUlar conflicT) that applies a game-theoretic approach to drug design. We focus on modeling spontaneous mutations in the drug target that confer an evolutionary benefit to the pathogen and therefore establish themselves in the population. The game models the interactions between the drug designer and the protein: the protein makes "moves" by mutating its positions and the drug designer makes moves by choosing one of several candidate drugs. If these moves result in a protein-drug pair that has high affinity, the drug designer wins; else, the protein wins. The aim of the drug designer is to therefore design a drug that plays well against all moves of the protein.

9.2 HIV Protease

The HIV-1 Protease (HIVPR) is a HIV protein that cleaves HIV's other gene products in order to make them into functional proteins. The normal functioning of HIV1PR i s essential for the propagation of the virus. Due to its importance in HIV replication, it has been extensively studied as a drug target. More than 10 protease inhibitors that target HIVPR are currently approved by the FDA.

HIV is a retrovirus. This means that the genetic code of the virus is transmitted in the

RNA form. Inside the host cell, the genetic code is then reverse-transcribed into DNA and is integrated into the host's DNA. When the host cell's ribosome transcribes and translates this DNA into protein form, the HIV proteins are activated eventually shutting down the host's immune system and causing AIDS. The cellular transcription machinery in human's is remarkably accurate and rarely induces random mutations. In contrast, the reverse transcriptase is more error-prone. This erroneous nature of the reverse transcriptase frequently induces random mutations into HIV's genetic code. While this can induce deleterious mutations to the virus, it also allows it to rapidly evolve in the face of evolutionary pressure. Any mutations that positively affect HIV's fitness are therefore quickly introduced (by this erroneous copying mechanism) and establish themselves into the HIV population.

When Saquinavir, the first protease inhibitor approved by the FDA, was administered to patients, the virus quickly evolved to develop drug-resistance. Since then, designing inhibitors that are robust to mutant forms of HIV has been a subject of active research. A common current strategy is to use a combination of drugs to treat the disease. Our goal here is to predict the possibility of resistance, estimate optimal combinations of drugs in the light of such resistance and determine the success (or lack thereof) of such combinations.

Fig. 9.1 shows the strategy profile of the worst-case equilibrium strategy (equilibrium with weakest binding). According to our model, the expected interaction energy of this game is $\sim$ -8 kcal/mol indicating that all likely mutants of HIV are expected to be inhibited by the cocktail. Interestingly, when the game is analyzed with the drug player allowed the use of Saquinavir only, the result is very different: under the *best* case scenario, the expected binding energy is 0 kcal/mol with the worst case scenario having expected binding energy of 2 kcal/mol. Our model is therefore able to accurately predict that Saquinavir is prone to drug resistance. In addition it predicts that no combination of any known mutations around

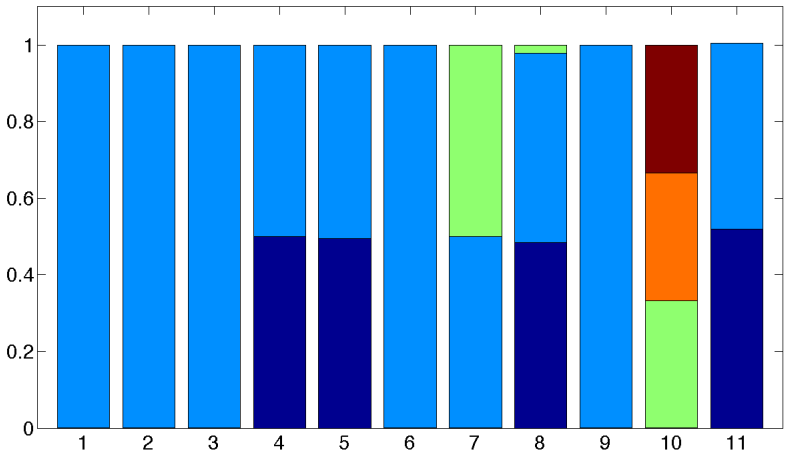


Figure 9.1: Equilibrium Strategy profile under worst-case scenarios with Drug player allowed to choose a cocktail of all three drugs. The first 10 columns correspond to HIV positions around the active site while the last player corresponds to the Drug. In each column of the HIV player, the dark blue color corresponds to the wildtype – the rest are mutants.

the active site of HIV can successfully destabilize the interaction if a three drug cocktail with Saquinavir, Amprenavir and Darunavir is used.

9.3 PDZ

The PDZ domain is a family of small, evolutionarily well represented protein binding motifs. The domain is most commonly found in signaling proteins and helps to anchor transmembrane proteins to the cytoskeleton and hold together signaling complexes. Due to its frequent occurrence in protein-protein interactions, it is a very well studied domain. Since some of these interactions are affected in cancerous cells, members of this family are being studied as targets for anti-cancer drugs.

In our game, these peptides are the drugs and the binding affinities are the utilities.

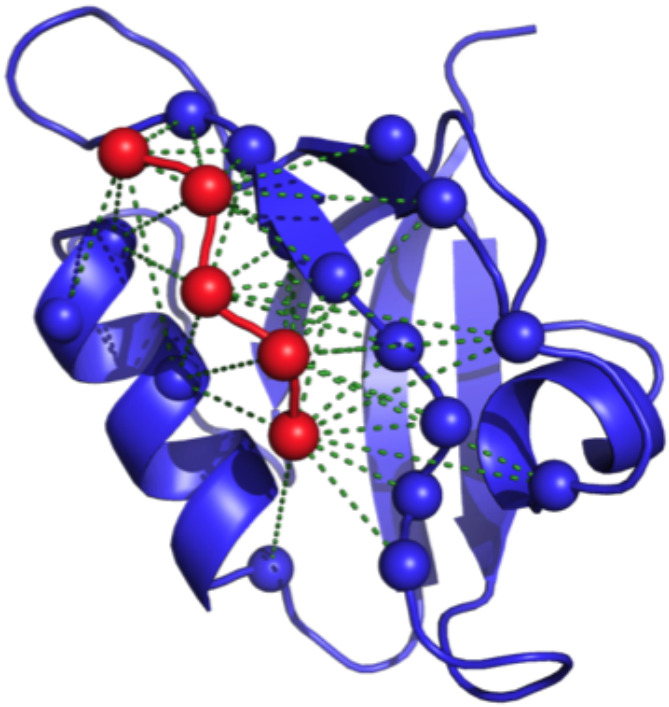
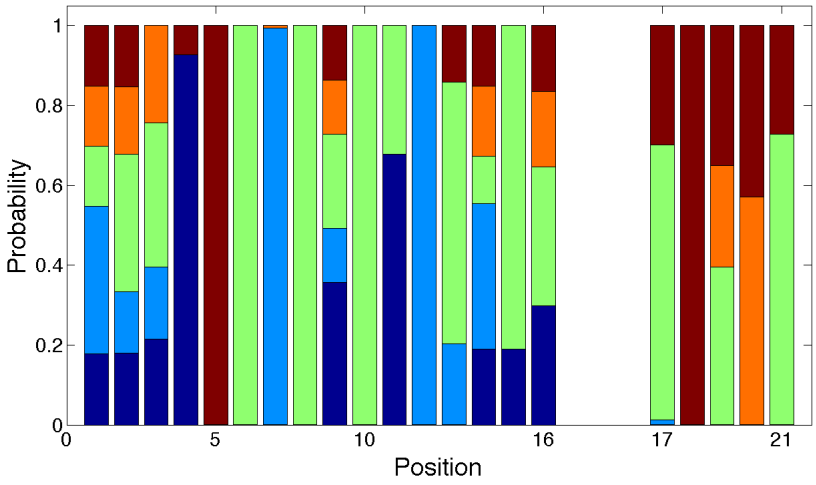
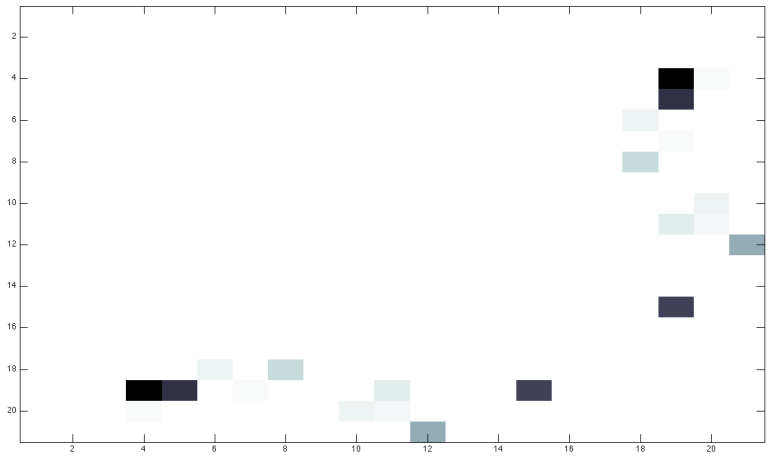


Figure 9.2: The graphical game describing the pdz-drug game is shown overlaid on a protein structure. The players are labeled with spheres (pdz in blue, drug in red) and the edges of the game are shown with dashed lines. The game has 21 players and 38 edges.

Along with Prof. Bailey Kellog's lab at Dartmouth, we have developed a sparse linear model that accurately predicts the binding affinity of PDZ domains to short peptides. The utilitiies (Fig. 9.2) were learnt using block sparse linear regression from experimental data and consists of thirty eight blocks (corresponding to edges here) and twenty one variable positions (corresponding to sixteen 'protein' players and five 'drug' players). Each protein player has five actions corresponding to the wild-type and the four other most likely amino acid positions at this position, as observed in nature. This restricts the game to mutations that are energetically viable.



(A)



(B)

Figure 9.3: (A) The marginal profile of each player for the worst-case equilibrium scenario. The 16 positions corresponding to the PDZ and the 5 positions corresponding to drug are shown in separate groups. (B) The expected binding energy for each edge in. By choosing mutations that maximize disruption to binding, PDZ is able to shut off interactions on all but eleven edges. However, these eleven edges together ensure that the overall binding energy is negative indicating that the cocktail is successful.

The maximum degree of this graph was 10; neither exact computation of the CE nor the $LOCAL_{\Delta}$ approximation had sufficient memory to complete this computation on a standard desktop machine with 4GB RAM. We therefore only report the results of $LOCAL_2+Cycle$.

In the drug-design game, a CE encodes a distribution over drugs (known as a drug cocktail) *and* a distribution over PDZ sequences. The distribution over PDZ mutations can be interpreted as the set of mutations that are likely to arise in response to the cocktail. The key question is whether the expected binding energy of the cocktail and the PDZs is negative (i.e., favorable) or positive (i.e., unfavorable). If positive, then the PDZ is resistant to the cocktail. Thus, it is sufficient to determine whether the expected binding energy is guaranteed to be negative. Therefore, we computed the (approximate) CE that would maximize the binding energy (i.e a "pessimistic" CE) and obtained an upper bound on the maximum binding energy. Naturally, if this upper bound is negative then the true worst-case expected binding energy is also negative which, in turn, implies that the cocktail is successful at *any* CE of the game.

The expected binding energy was negative (-5.83 kcal/mol) indicating that this cocktail-drug is predicted to successfully bind to all viable mutants of the PDZ. Fig. 9.3-A shows the marginal profile for each position across its five actions. It is interesting to note that compared to the PDZ, the profile of the drug is limited to a few actions per position, possibly due to the pessimistic nature of our prediction.

Fig. 9.3-B shows the breakdown of this total binding energy across the 38 edges. Only eleven of the 38 edges had a non-zero contribution to the binding energy. In contrast, the "optimistic" CE of this game (obtained by minimizing binding energy) had 20 edges that had non-zero contribution to the binding energy.

Chapter 10

Conclusions and Future Work

In Ch. 4, we presented a Bayesian alternative to traditional methods for evaluating the quality of predicted protein structures. Our experimental results show that this Bayesian approach significantly out performs MAP estimates of quality assessment. Additionally, we presented a practical algorithm for learning to rank using partition functions by optimizing a list-wise loss function over training data. We compared two loss functions, the negative log-likelihood and the cross entropy, and found that optimizing the cross-entropy objective function improves on the unoptimized hyper-parameters.

Protein structure prediction is an important, and unsolved problem in Biology, and we believe that our method might be able to improve the accuracy of existing techniques. There are a number of areas for future improvement, the most important being incorporating Bayesian Model Averaging by modeling a limited amount of backbone flexibility.

Ch. 5 made three primary contributions (i) the first graphical model for protein-protein complexes; (ii) the first graphical model for simultaneously modeling backbone and side-chain flexibility; and (iii) a novel algorithm for optimizing force fields by minimizing differ-

ences in free energies. Our method is efficient and accurate on the task of computing the free energy of protein-protein complexes. In particular, our method outperformed ROSETTA and FOLDX on a benchmark set of more than 700 mutants.

Our results indicate that an explicit incorporation of backbone and side-chain flexibility is feasible. Interestingly, backbone flexibility did not substantially improve our results relative to the fixed-backbone case on our benchmark set. There are two likely reasons for this. First, our chosen benchmark set contained relatively rigid complexes. That is, the backbones are largely stable *in vivo*, as previously suggested in Lu et al. [2001]. Second, our backrub-generated backbones generally have high internal energies and therefore do not contribute substantially to the total free energy. Addressing these problem requires improved methods for generating realistic backbone conformations, an important problem for future research.

We extended GOBLIN to model protein-ligand interactions. GOBLIN-Ligand is a fast method of incorporating entropic contributions in computing protein-ligand interaction free energies. Despite using similar force-fields, GOBLIN-Ligand can improve upon AutoDock's accuracy. It thus strikes a balance between rigor of Molecular Dynamics and the speed of enthalpic approaches. While accounting for entropic contributions, GOBLIN -Ligand currently keeps the core of the ligand fixed. The accuracy of the entropic estimates might be improved in the future by accounting for the flexibility of the core, by including other binding poses of the complex.

Ch. 7 proposed a new algorithm for discovering and modeling the statistical patterns contained in a given MSA. By employing sound probabilistic modeling and convex structure (and parameter) learning, we are able to find a good balance between structural sparsity (simplicity) and goodness of fit. One of the key advantages of a graphical model approach is that the graph reveals the direct and indirect constraints that can further our understanding

of protein function and regulation. MRFs are generative models, and can therefore be used design new protein sequences via sampling and inference. However, we expect that the utility of our model in the context of protein design could be greatly enhanced by incorporating structure based information which explicitly models the physical constraints of the protein. As shown in Kamisetty et al. [2009], it is possible to construct MRFs that integrate both sequence and structure information. We believe an interesting direction for future work is to apply structure learning to MSAs enhanced with physical constraints (e.g., interactions energies) in the form of informative priors or as edge features. The learning algorithm would then select the type of constraint (i.e., sequence vs structure) that best explains the covariation in the MSA.

We note that there are a number of other ways to incorporate phylogenetic information directly into our model. For example, given a phylogenetic clustering of sequences, we can incorporate a single additional node in the graphical model reflecting the cluster to which the sequence belongs. This would allow us to distinguish functional coupling from coupling caused due to phylogenetic variations.

Together, GREMLIN and GOBLIN provide a powerful framework for the modeling of proteins and their interactions with other molecules. In Ch. 8, we demonstrated an approach to approximate the properties of a CE in graphical games. Our approach, based on outer relaxations to the marginal polytope computes these approximations efficiently. On a large set of games with different types of utilities, we demonstrated that these relaxations are also remarkably accurate, often giving the exact solution. In addition, our approach bounds the objective function. When used with the social utility for example, our approach can be used to bound the price of anarchy of the game Koutsoupias and Papadimitriou [2009].

In our application of game-theory to a biological problem in Ch. 9, we follow a rich

body of prior work that model evolutionary behavior in this manner, starting with the pioneering work of Hamilton [1964a,b] and Smith and Price [1973]. Indeed, it can be argued that the process of evolution is best modeled in such a strategic manner: organisms adapt to the behavior of other organisms and the environment to maximize their chances of survival. While these adaptations by themselves are not the output of strategic behavior, the combination of random mutations with non-random selective pressures can, and is often modeled as such.

By modeling the smallest unit of evolutionary change (the mutation of an amino acid) this chapter considers strategic behaviors at a much higher-resolution than previous applications of game-theory to the study of evolution. In the future, we hope to extend this promising approach to learning the structure and utilities of graphical games from data and in applying such models to the design of new drugs.

Bibliography

H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, January 2003.

B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.

D. Altschuh, T. Vernet, P. Berti, D. Moras, and K. Nagai. Coordinated amino acid changes in homologous protein families. *Protein Eng.*, 2(3):193–199, September 1988.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*. MIT Press, 2007.

R.J. Aumann. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica: Journal of the Econometric Society*, 55(1):1–18, 1987. ISSN 0012-9682.

R.J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974.

F. Barahona and A.R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36(2): 157–173, 1986.

Donald Bashford and Martin Karplus. Multiple-site titration curves of proteins: an analysis of exact and approximate methods for their calculation. *The Journal of Physical Chemistry*, 95(23):9556–9561, 1991.

Donald Bashford and Martin Karplus. pKa's of ionizable groups in proteins: atomic detail from a continuum electrostatic model. *Biochemistry*, 29(44):10219–10225, November 1990.

A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S.R. Eddy, S. Griffiths-Jones, K.L. Howe, M. Marshall, and E.L.L. Sonnhammer. The Pfam protein families database. *Nucleic acids research*, 30(1):276, 2002.

Alexander Benedix, Caroline M. Becker, Bert L. de Groot, Amedeo Caflisch, and Rainer A. Bockmann. Predicting free energy changes using structural ensembles. *Nature Methods*, 6(1):3–4, January 2009.

H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucl. Acids Res.*, 28:235–242, 2000.

J. Besag. Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika*, 64(3):616–618, 1977.

H. A. Bethe. Statistical theory of superlattices. *Proc. Roy. Soc. London A*, 150:552–575, 1935.

Parbati Biswas, Jinming Zou, and Jeffery G. Saven. Statistical theory for protein ensembles with designed energy landscapes. *The Journal of Chemical Physics*, 123(15), 2005.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, volume 22, pages 89–96, 2005.

A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack Jr. A graph theory algorithm for protein side-chain prediction. *Protein Sci.*, 12:2001–2014, 2003.

Z. Cao, T. Qin, T.Y. Liu, M.F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM Press New York, NY, USA, 2007.

F. Clavel and A.J. Hance. HIV drug resistance. *The New England journal of medicine*, 350 (10):1023, 2004.

P. Clifford. Markov random fields in statistics. In G. R. Grimmett and D. J. A. Welsh, editors, *Disorder in Physical Systems. A Volume in Honour of John M. Hammersley*, pages 19–32, Oxford, 1990. Clarendon Press.

Imre Csiszar and Zsolt Talata. Consistent estimation of the basic neighborhood of markov random fields. *The Annals of Statistics*, 34(1):123–145, 2006.

P. Dagum and R. M. Chavez. Approximating probabilistic inference in bayesian belief networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(3):246–255, 1993.

B.I. Dahiyat and S.L. Mayo. De novo protein design: fully automated sequence selection. *Science*, 278(5335):82–87, Oct 1997.

C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

Anne Dhulesia, Joerg Gsponer, and Michele Vendruscolo. Mapping of two networks of residues that exhibit structural and dynamical changes upon binding in a pdz domain protein. *Journal of the American Chemical Society*, 130(28):8931–8939, July 2008.

S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14:755–763, 1998.

A.G. Eliopoulos, D.J. Kerr, J. Herod, L. Hodgkins, S. Krajewski, J.C. Reed, and L.S. Young. The control of apoptosis and drug resistance in ovarian cancer: influence of p53 and Bcl-2. *Oncogene*, 11(7):1217, 1995.

S. N. Fatakia, S. Costanzi, and C. C. Chow. Computing highly correlated positions using mutual information and graph theory for g protein-coupled receptors. *PLoS ONE*, 4(3): e4681, 03 2009.

Joseph Felsenstein. *Inferring Phylogenies*. Sinauer Associates, September 2003.

S. Fields and O. Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245–246, July 1989. ISSN 0028-0836.

Anthony A. Fodor and Richard W. Aldrich. On evolutionary conservation of thermodynamic coupling in proteins. *Journal of Biological Chemistry*, 279(18):19046–19050, April 2004.

D.P. Foster and R.V. Vohra. Calibrated Learning and Correlated Equilibrium* 1. *Games and Economic Behavior*, 21(1-2):40–55, 1997.

G.D. Friedland, A.J. Linares, C.A. Smith, and T. Kortemme. A simple model of backbone flexibility improves modeling of side-chain conformational variability. *J. Mol. Biol.*, 380 (4):757–774, July 2008.

M. Fromer and C. Yanover. Accurate prediction for atomic-level protein design and its application in diversifying the near-optimal sequence space. *Proteins: Structure, Function, and Bioinformatics*, page In Press, 2008.

Angelika Fuchs, Antonio J. Martin-Galiano, Matan Kalman, Sarel Fleishman, Nir Ben-Tal, and Dmitrij Frishman. Co-evolving residues in membrane proteins. *Bioinformatics*, 23 (24):3312–3319, December 2007.

E.J. Fuentes, C.J. Der, and A.L. Lee. Ligand-dependent dynamics and intramolecular signaling in a PDZ domain. *Journal of molecular biology*, 335(4):1105–1115, 2004.

B. Gidas. Consistency of maximum likelihood and pseudo-likelihood estimators for Gibbs Distributions. *Institute for Mathematics and Its Applications*, 10:129–+, 1988.

M. K. Gilson. Multiple-site titration and molecular modeling: two rapid methods for computing energies and forces for ionizable groups in proteins. *Proteins*, 15(3):266–282, 1993.

Ulrike Göbel, Chris Sander, Reinhard Schneider, and Alfonso Valencia. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Genetics*, 18 (4):309–317, April 1994.

R. Guerois, J. E. Nielsen, and L. Serrano. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *Journal of Molecular Biology*, 320:369–387, 2002.

W. Hamilton. The genetical evolution of social behaviour. I. *Journal of Theoretical Biology*, 7(1):1–16, July 1964a.

W. D. Hamilton. The genetical evolution of social behaviour. II. *Journal of theoretical biology*, 7(1):17–52, July 1964b.

J. J. Havranek and P. B. Harbury. Automated design of specificity in molecular recognition. *Nat Struct Biol*, 10(1):45–52, January 2003. ISSN 1072-8368.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. *Large margin rank boundaries for ordinal regression*, pages 115–132. MIT Press, Cambridge, MA, 2000.

Ruth Huey, Garrett M. Morris, Arthur J. Olson, and David S. Goodsell. A semiempirical free energy force field with charge-based desolvation. *J. Comput. Chem.*, 28(6):1145–1152, 2007.

Holger Hofling and Robert Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10:883–906, April 2009.

E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620+, May 1957.

E. T. Jaynes. Information theory and statistical mechanics. *Statistical Physics*, pages 181–218, 1963.

E. T. Jaynes. Prior probabilities. *Systems Science and Cybernetics, IEEE Transactions on*, 4(3):227–241, 1968.

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

T. Joachims. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM New York, NY, USA, 2002.

S. Kakade, M. Kearns, J. Langford, and L. Ortiz. Correlated equilibria in graphical games. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 42–47. ACM, 2003.

H. Kamisetty, E.P. Xing, and C.J. Langmead. Free Energy Estimates of All-atom Protein Structures Using Generalized Belief Propagation. In *Proc. 7th Ann. Intl. Conf. on Research in Comput. Biol. (RECOMB)*, pages 366–380, 2007.

H. Kamisetty, E. P. Xing, and C. J. Langmead. Free energy estimates of all-atom protein structures using generalized belief propagation. *J. Comp. Biol.*, 15(7):755–766, September 2008. ISSN 1557-8666.

H. Kamisetty, B. Ghosh, C. Bailey-Kellogg, and C.J. Langmead. Modeling and Inference of Sequence-Structure Specificity. In *Proc. of the 8th International Conference on Computational Systems Bioinformatics (CSB)*, pages 91–101, 2009.

Hetunandan Kamisetty and Christopher J. Langmead. A bayesian approach to protein model quality assessment. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 481–488, New York, NY, USA, 2009. ACM.

S. Kamtekar, J.M. Schiffer, H. Xiong, J.M. Babik, and M.H. Hecht. Protein design by binary patterning of polar and nonpolar amino acids. *Science*, 262(5140):1680–1685, Dec 1993.

Kevin Karplus, Kimmen Sjlander, Christian Barrett, Melissa Cline, David Haussler, Richard Hughey, Liisa Holm, Chris Sander, Ebi England, and Ebi England. Predicting protein structure using hidden markov models. In *Proteins: Structure, Function, and Genetics*, pages 134–139, 1997.

Kevin Karplus, Christian Barrett, and Richard Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14:846–856, 1998.

R. Kikuchi. A theory of cooperative phenomena. *Phys. Rev*, 81:988–1003, 1951.

C. L. Kingsford, B. Chazelle, and M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics*, 21:1028–1036, 2005.

P. Koehl and M. Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J. Mol. Biol.*, 239:249–275, 1994.

P. Kollman. Free energy calculations: Applications to chemical and biochemical phenomena. *Chemical Reviews*, pages 2395–2417, 1993.

T. Kortemme and D. Baker. Computational design of protein-protein interactions. *Curr. Opin. Chem. Biol.*, 8(1):91–97, February 2004.

T. Kortemme and D. Baker. A simple physical model for binding energy hot spots in protein-protein complexes. *PNAS*, 99(22):14116–14121, October 2002. ISSN 0027-8424.

T. Kortemme, A. V. Morozov, and D. Baker. An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein-protein complexes. *J. Mol. Biol.*, 326(4):1239–1259, February 2003.

E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *Computer science review*, 3 (2):65–69, 2009.

Anders Krogh, Michael Brown, I. Saira Mian, Kimmen Sjlander, and David Haussler. Hidden markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 1994.

B. Kuhlman and D. Baker. Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences*, 97(19):10383–10388, 2000.

B. Kuhlman, G. Dantas, G.C. Ireton, G. Varani, B.L. Stoddard, and D. Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649):1364–1368, Nov 2003.

Shankar Kumar, John M. Rosenberg, Djamal Bouzida, Robert H. Swendsen, and Peter A. Kollman. THE weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *J. Comput. Chem.*, 13(8):1011–1021, 1992.

J.D. Lafferty, A. McCallum, and F.C.N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2001.

T. Lazaridis and M. Karplus. Effective energy function for proteins in solution. *Proteins*, 35(2):133–152, May 1999. ISSN 0887-3585.

Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of markov networks using l_1-regularization. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 817–824. MIT Press, Cambridge, MA, 2007a.

Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of markov networks using l_1-regularization. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 817–824. MIT Press, Cambridge, MA, 2007b.

K. Leyton-Brown and Y. Shoham. *Essentials of game theory*. Morgan & Claypool Publishers, 2008.

R. Lilien, B. Stevens, A. Anderson, and B. R. Donald. A novel ensemble-based scoring and search algorithm for protein redesign, and its application to modify the substrate specificity of the gramicidin synthetase A phenylalanine adenylation enzyme. *J. Comp. Biol.*, 12(6-7):740–761, 2005.

Jennifer Listgarten and David Heckerman. Determining the number of non-spurious arcs in a learned dag model: Investigation of a bayesian and a frequentist approach. *23rd annual conference on Uncertainity in Artificial Intelligence*, 2007.

Yan Liu, Jaime G. Carbonell, Peter Weigele, and Vanathi Gopalakrishnan. Protein fold recognition using segmentation conditional random fields. *Journal of Computational Biology*, 13(2):394–406, 2006.

S. W. Lockless and R. Ranganathan. Evolutionarily conserved pathways of energetic connectivity in protein families. *Science*, 286:295–299, Oct 1999.

C. Loose, K. Jensen, I. Rigoutsos, and G. Stephanopoulos. A linguistic model for the rational design of antimicrobial peptides. *Nature*, 443(7113):867–869, Oct 2006.

S.C. Lovell, J.M. Word, J.S. Richardson, and D.C. Richardson. The penultimate rotamer library. *Proteins*, 40:389–408, 2000.

S.M. Lu, W. Lu, M.A. Qasim, others, and M. Laskowski, Jr. Predicting the reactivity of proteins from their sequence alone: Kazal family of protein inhibitors of serine proteinases. *PNAS*, 98(4):1410–1415, February 2001.

M. J. McGregor, S. A. Islam, and M. J. Sternberg. Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. *J. Mol. Biol.*, 198 (2):295–310, November 1987. ISSN 0022-2836.

T. Morita. Cluster variation method for non-uniform ising and heisenberg models and spin-pair correlation function. *Prog. Theor. Phys.*, 85:243 – 255, 1991.

T. Morita, T. M. Suzuki, K. Wada, and M. Kaburagi. Foundations and applications of cluster variation method and path probability method. *Prog. Theor. Phys. Supplement*, 115, 1994.

I. Muegge. PMF scoring revisited. *J. Med. Chem.*, 49(20):5895–5902, 2006.

R Neal. Probabilistic inference using markov chain monte carlo methods. *Technical Report Dept. of Computer Science, University of Toronto*, 1993.

R. M Neal. Estimating ratios of normalizing constants using linked importance sampling. *Technical Report No. 0511, Dept. of Statistics, University of Toronto*, 2005.

J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

C.R. Otey, J.J. Silberg, C.A. Voigt, J.B. Endelman, G. Bandara, and F.H. Arnold. Functional evolution and structural conservation in chimeric cytochromes P450: Calibrating a structure-guided approach. *Chem. Biol.*, 11(3):309–318, Mar 2004.

J. Pearl. Fusion, propagation, and structuring in belief networks. *Artif. Intell.*, 29(3):241–288, 1986.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

D. D. Pollock and W. R. Taylor. Effectiveness of correlation analysis in identifying protein residues undergoing correlated evolution. *Protein Eng.*, 10(6):647–657, June 1997.

J. W. Ponder and F. M. Richards. Tertiary templates for proteins. use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.*, 193 (4):775–791, February 1987. ISSN 0022-2836.

J. Pons, A. Rajpal, and J. F. Kirsch. Energetic analysis of an antigen/antibody interface: alanine scanning mutagenesis and double mutant cycles on the HyHEL-10/lysozyme interaction. *Protein Science*, 8(5):958–968, May 1999.

O. Roche, R. Kiyama, and C. L. Brooks. Ligand-protein database: linking protein-ligand complex structures to binding data. *Journal of medicinal chemistry*, 44(22):3592–3598, 2001.

W. P. Russ, D. M. Lowery, P. Mishra, M. B. Yaffe, and R. Ranganathan. Natural-like function in artificial WW domains. *Nature*, 437:579–583, Sep 2005.

W. Sakai, E.M. Swisher, B.Y. Karlan, M.K. Agarwal, J. Higgins, C. Friedman, E. Villegas, C. Jacquemont, D.J. Farrugia, F.J. Couch, et al. Secondary mutations as a mechanism of cisplatin resistance in BRCA2-mutated cancers. *Nature*, 451(7182):1116–1120, 2008.

Mark Schmidt, Kevin Murphy, Glenn Fung, and Rmer Rosales. Structure learning in random fields for heart motion abnormality detection. In *CVPR*. IEEE Computer Society, 2008.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978.

C. A. Smith and T. Kortemme. Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. *J. Mol. Biol.*, 380(4):742–756, July 2008. ISSN 1089-8638.

J. Maynard Smith and G. R. Price. The Logic of Animal Conflict. *Nature*, 246(5427):15–18, November 1973.

M. Socolich, S. W. Lockless, W. P. Russ, H. Lee, K. H. Gardner, and R. Ranganathan. Evolutionary information for specifying a protein fold. *Nature*, 437:512–518, Sep 2005.

D. Sontag and T. Jaakkola. New outer bounds on the marginal polytope. *Advances in Neural Information Processing Systems*, 20, 2007.

E.J. Soulsby. Resistance to antimicrobials in humans and animals. *British Medical Journal*, 331(7527):1219, 2005.

J. Thomas, N. Ramakrishnan, and C. Bailey-Kellogg. Graphical models of residue cou-

pling in protein families. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 5(2):183–197, 2008a.

J. Thomas, N. Ramakrishnan, and C. Bailey-Kellogg. Graphical models of residue coupling in protein families. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 5(2):183–197, 2008b.

J. Thomas, N. Ramakrishnan, and C. Bailey-Kellogg. Protein design by sampling an undirected graphical model of residue constraints. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 2009a. In press.

J. Thomas, N. Ramakrishnan, and C. Bailey-Kellogg. Protein Design by Sampling an Undirected Graphical Model of Residue Constraints. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 6(3):506–516, 2009b.

J. Thomas, N. Ramakrishnan, and C. Bailey-Kellogg. Graphical models of protein-protein interaction specificity from correlated mutations and interaction data. *Proteins: Structure, Function, and Bioinformatics*, 76(4):911–29, 2009c.

John Thomas, Naren Ramakrishnan, and Chris Bailey-Kellogg. Graphical models of residue coupling in protein families. In *BIOKDD '05: Proceedings of the 5th international workshop on Bioinformatics*, pages 12–20, New York, NY, USA, 2005. ACM. ISBN 1-59593-213-5.

P. D. Thomas and K. A. Dill. Statistical potentials extracted from protein structures: How accurate are they? *J. Mol. Biol.*, 257:457–469, 1994.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

JA Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

J. Von Neumann. Zur Theorie der Gesellschaftsspiele Math. *Annalen*, 100:295–320, 1928.

J. Von Neumann, O. Morgenstern, A. Rubinstein, and H.W. Kuhn. *Theory of games and economic behavior*. Princeton Univ Pr, 2007.

Martin J. Wainwright, Pradeep Ravikumar, and John D. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1465–1472. MIT Press, Cambridge, MA, 2007.

M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

R. Wallis, K.Y. Leung, M.J. Osborne, R. James, G.R. Moore, and C. Kleanthous. Specificity in protein-protein recognition: conserved Im9 residues are the major determinants of stability in the colicin E9 DNase-Im9 complex. *Biochemistry*, 37(2):476–485, 1998.

W. Wang and P.A. Kollman. Computational Study of Protein Specificity: The molecular basis of HIV-1 protease drug resistance. *PNAS*, 98(26):14937–14942, 2001.

A. L. Watters, P. Deka, C. Corrent, D. Callender, G. Varani, T. Sosnick, and D. Baker. The highly cooperative folding of small naturally occurring proteins is likely the result of natural selection. *Cell*, 128(3):613–624, February 2007.

I. T. Weber and R.W. Harrison. Molecular mechanics analysis of drug-resistant mutants of HIV protease. *Protein Engineering*, 12(6):469–474, 1999.

M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, and T. Hwa. Identification of direct residue contacts in protein-protein interaction by message passing. *Proc. Natl. Acad. Sci. U.S.A.*, 106:67–72, Jan 2009.

J.M. Word, S.C. Lovell, J.S. Richardson, and D.C. Richardson. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *J. Mol. Biol.*, 285(4):1735–1747, 1999.

L. Wroblewska and J. Skolnick. Can a physics-based, all-atom potential find a protein's native structure among misfolded structures? i. large scale amber benchmarking. *Journal of Computational Chemistry*, 28(12):2059–2066, 2007.

Fen Xia, Tie Y. Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, New York, NY, USA, 2008. ACM.

J. Xu. Rapid protein side-chain packing via tree decomposition. In *Proc. 9th Ann. Intl. Conf. on Comput. Biol. (RECOMB)*, pages 423–439, 2005.

Yuhong Yang. Can the strengths of aic and bic be shared? *Biometrica*, 92:2003, 2003.

C. Yanover and Y. Weiss. Approximate inference and protein folding. *Proc. NIPS*, pages 84–86, 2002.

J.S. Yedidia, W.T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, 2005.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.