Learning Generative Models for Protein Fold Families

Sivaraman Balakrishnan[1], **Hetunandan Kamisetty**[1],
Jaime G. Carbonell, Christopher James Langmead[*]

March 2010
CMU-CS-10-113

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1]These two authors contributed equally to this paper.
[*]Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA 15213.
E-mail: cjl@cs.cmu.edu

This research was supported by NSF IIS-0905193 and an award from Microsoft Research to CJL.

Keywords: Structure Learning, Generative Models, Probabilistic Graphical Models, Proteins

Abstract

Statistical models of the amino acid composition of the proteins within a fold family are widely used in science and engineering. Existing techniques for learning probabilistic graphical models from multiple sequence alignments either make strong assumptions about the conditional independencies within the model (e.g., HMMs), or else use sub-optimal algorithms to learn the structure and parameters of the model. We introduce an approach to learning the topological structure *and* parameters of an undirected probabilistic graphical model. The learning algorithm uses block-L_1 regularization and solves a *convex* optimization problem, thus guaranteeing a globally *optimal* solution at convergence. The resulting model encodes both the position-specific conservation statistics *and* the correlated mutation statistics between sequential and long-range pairs of residues. Our model is generative, allowing for the design of new proteins that have corresponding statistical properties to those seen in nature. We apply our approach to two widely studied protein families: the WW and the PDZ folds. We demonstrate that our model is able to capture interactions that are important in folding and allostery. Our results additionally indicate that while the network of interactions within a protein is sparse, it is richer than previously believed.

1 Introduction

The patterns in the amino acid composition of the proteins within a fold family provide insights into the constraints that govern structure, function, and dynamics. These constraints can reflect both position-specific conservation (e.g., 'position 3 is always a tryptophan'), or correlated mutations (e.g., 'position 4 is a tyrosine if position 10 is a glycine, but it is an arginine if position 10 is a valine'). Such covariation can exist between sequential and long-range pairs of residues, reflecting spatial proximity or functional coupling. In a series of elegant papers [22, 28, 24], Ranganathan and colleagues demonstrated that conservation and covariation constraints together contain virtually all the information required to specify a fold family. However, standard statistical models of amino acid sequences, such as HMMs and their variants [10], neglect covariation constraints between non-adjacent residues. The resulting model thus only partially reflects the evolutionary constraints imposed on a particular protein family.

More recently, Thomas and colleagues [33] introduced the use of undirected probabilistic graphical models, also known as a Markov Random Fields (MRFs), to compactly encode both the conservation and covariation constraints present in a given multiple sequence alignment (MSA). Their approach, including subsequent refinements [30, 31, 32], relies on a simple greedy algorithm for learning both the structure and parameters of the MRF. Unfortunately, their greedy algorithms have no guarantee as to the optimality of the resulting model. To address this deficiency, we introduce a principled approach to learning MRFs from MSAs. Our learning algorithm uses block-L_1 regularization and solves a convex optimization problem, and is thus guaranteed to produce a globally optimal solution at convergence.

Markov Random Fields are generative models, and can therefore be used in the context of protein design (i.e., generating novel sequences with prescribed structure and/or function). While structure-based approaches that explicitly consider physical constraints (e.g., [20]) have achieved a great deal of success, protein design methods based on structure alone cannot account for interactions that aren't evident in the native structure including those important for function, folding, or allosteric regulation. The consequences of ignoring such interactions can be significant. For example, even successfully-designed proteins often exhibit non-natural behavior in terms of their thermal stability and folding pathways [37]. Moreover, the amount of available sequence data for most protein families is often one-to-two orders of magnitude larger than the corresponding amount of structure data. Thus, there is a need for methods that incorporate information from MSAs. Indeed, as [28] show, the constraints present in the patterns of sequence variation is one such promising method.

We apply our approach to two widely studied protein families: the WW and the PDZ folds. We demonstrate that our model is able to capture interactions that are important in folding and allostery. Our results additionally indicate that while the network of interactions within a protein is sparse, it is richer than previously believed. While this paper is limited to new more powerful models constructed from MSAs, we have previously shown in [19], that it is possible to construct MRFs that integrate constraints learned from both sequence and structure.

(A) (B)

Figure 1: (A) A multiple sequence alignment (MSA) for a hypothetical domain family. (B) A portion of a Markov Random Field encoding the conservation in and the coupling in the MSA. The edge between random variables X_1 and X_4 reflects the coupling between positions 1 and 4 in the MSA.

2 Modeling Domain Families with Markov Random Fields

A protein is a polypeptide chain consisting of one or more components called *domains*. A set of evolutionarily related domains from different proteins is called a family[1]. The domains within a family tend to have similar biological functions and three dimensional structures. Thus, by examining the statistical patterns of sequence conservation and diversity within a domain family, we can gain insights into the constraints that determine structure and function. In what follows, we describe an approach to learning these statistical patterns from a given multiple sequence alignment. The resulting model is a probability distribution over amino acid sequences for a particular domain family.

Let X_i be the multinomial 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 $\mathbf{X} = \{X_1, X_2, ..X_p\}$ be the multi-variate random variable describing the amino acid composition of an MSA of length p. Our goal is to model $P(\mathbf{X})$, the amino-acid composition of the domain family.

Unfortunately, P($\mathbf{X}$) 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. 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 independent of each other. In probabilistic terms we say that X_1, X_3 are co-varying, and that the remaining X_i's are statistically independent.

[1] In this paper, the expression *domain family* is synonymous with *protein family*.

We can therefore encode the joint distribution over all positions in the MSA by storing one joint distribution $P(X_1, X_4)$, and the uni-variate distributions $P(X_i)$, for the remaining positions (since they are all statistically independent of every other variables). The ability to factor the full joint distribution, $P(\mathbf{X})$, in this fashion has an important consequence in terms of space complexity. Namely, we can reduce the space requirements from 21^7 to $21^2 + 5 * 21$ 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 over a graph; while any distribution can be modeled by a PGM with a complete graph, exploiting the conditional independencies in the distribution leads to a PGM with a (structurally) sparse graph. Following [30], 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 $\mathcal{M} = (\mathbf{X}, \mathcal{E}, \Phi, \Psi)$ where $(\mathbf{X}, \mathcal{E})$ is an undirected graph over the random variables, and Φ, Ψ are a set of node and edge potentials, respectively, usually chosen to be log-linear functions of the form:

$$\phi_s = (e^{v_1^s} e^{v_2^s} ... e^{v_k^s}); \qquad \psi_{st} = \left\{ \begin{array}{c} e^{w_{11}^{st}} e^{w_{12}^{st}} ... e^{w_{1k}^{st}} \\ e^{w_{21}^{st}} e^{w_{22}^{st}} ... e^{w_{2k}^{st}} \\ ... \\ e^{w_{k1}^{st}} e^{w_{k2}^{st}} ... e^{w_{kk}^{st}} \end{array} \right\} \tag{1}$$

where $\mathbf{v} = \{v_s | s = 1 \dots p\}$ and $\mathbf{w} = \{w^{st} | (s,t) \in \mathcal{E}\}$ are node and edge "weights", and k is the number of states that each random variable can take.

The probability of a particular sequence $x = \{x_1, x_2, \dots, x_p\}$ according to $\mathcal{M}$ is defined as:

$$P_{\mathcal{M}}(X) = \frac{1}{Z} \prod_{s \in V} \phi_s(X_s) \prod_{(s,t) \in E} \psi_{st}(X_s, X_t) \tag{2}$$

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

$$Z = \sum_{X \in \mathbf{X}} \prod_{s \in V} \phi_s(X_s) \prod_{(s,t) \in E} \psi_{st}(X_s, X_t) \tag{3}$$

The structure of the MRF for the MSA shown in Fig. 1(A) is shown in Fig. 1(B). The edge between variables X_1 and X_4 reflects the statistical coupling between those positions in the MSA.

3 Structure learning with L_1 Regularization

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

problem can be divided into two parts: (i) *structure learning* — learning the edges of the graph, and (ii) and *parameter estimation* — learning $\mathbf{v}, \mathbf{w}$ (since they completely define the potentials Φ, Ψ), given the structure of the graph.

Due to its importance and applicability in a broad spectrum of areas, the problem of structure learning for graphical models has received considerable attention from several communities. Broadly, the previously considered approaches to this problem are either constraint-based [30, 29] or score-based [8]. Constraint-based methods estimate conditional independencies from data using hypothesis testing and then determine a graph that represents these independencies. Score-based approaches combine a metric to measure goodness of fit with a metric to measure complexity of the graph to *score* each graph. This is combined with a (typically greedy) search procedure that generates candidate graphs. However, since the number of possible graphs is super exponential in the number of vertices the search problem is computationally intractable, in general.

More recently several authors [36, 21, 18, 25] have considered convex approximations to the complexity metric and tractable (convex) approximations to the goodness of fit metric. Of these, those based on L_1 regularization are the most interesting because of their strong theoretical guarantees (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 [35] for a recent review of L1-regularization. We use a similar convex optimization based approach for both structure learning and parameter estimation. To that end, we first describe a suitable objective function for the problem.

The log-likelihood of the parameters $\Theta = (\mathcal{E}, \mathbf{v}, \mathbf{w})$, given a set of sequences, $\mathcal{X} = \{\mathbf{X^1}, \mathbf{X^2}, \mathbf{X^3}, ..., \mathbf{X^n}\}$, is

$$\text{ll}(\Theta) = \frac{1}{n} \sum_{X^i \in \mathcal{X}} \left[\sum_{s \in V} \log \phi_s(X_s^i) + \sum_{(s,t) \in E} \log \psi_{st}(X_s^i, X_t^i) \right] - \log Z \tag{4}$$

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). To avoid over-fitting and learning densely connected structures, we need to regularize the log-likelihood. In what follows we describe a method to learn sparse structures by optimizing the pseudo-likelihood using block-L_1 regularizers.

The general regularized structure learning problem can be formulated as:

$$\max_{\Theta} \text{ll}(\Theta) - \mathbf{R}(\Theta) \tag{5}$$

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

$$R(\Theta) = \lambda_{node} \|\mathbf{v}\|_2 + \lambda_{edge} \sum_{1 \le s < t \le p} \|\mathbf{w}^{st}\|_q \tag{6}$$

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 together in a *block*. The second term in Eq. 6 is the sum of the norms of each block. The choice of norm is usually selected from $q \in \{1, 2, \infty\}$. Since norms are always positive, this 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}.

The choice of norm affects the nature of the sparsity. Using $q = 1$ is equivalent to penalizing the likelihood by the sum of the L_1 norms of the individual parameters. That is, using $q = 1$ encourages sparsity in the parameters, and this leads to sparsity in the edges indirectly (when all parameters of an edge are zeroed out the edge is removed). In contrast, using $q = \{2, \infty\}$ directly encourages structural sparsity (sparsity in the edges) through the implicit L_1 norm described above, but does not directly encourage individual parameters to be zeroed out.

In the following sections we present a method to tractably compute the objective, followed by a method to optimize it.

3.1 Pseudo Likelihood

The log-likelihood as defined in Eq. 4 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 upto $\mathcal{O}(k^p)$ time. While approximations to the partition function based on Loopy Belief Propagation [21] 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 [6], and used in [36, 25]. The pseudo-likelihood is defined as:

$$\begin{aligned} \text{pll}(\Theta) &= \frac{1}{n} \sum_{X^i \in \mathcal{X}} \sum_{j=1}^{p} \log(P(X_j^i | X_{-j}^i)) \\ &= \frac{1}{n} \sum_{X^i \in \mathcal{X}} \sum_{j=1}^{p} \left[\log\phi_j(X_j^i) + \sum_{k \in V_j'} \log\psi_{jk}(X_j^i, X_k^i) - \log Z_j \right] \end{aligned}$$

where X_j^i is the residue at the j^{th} position in the i^{th} sequence of our MSA, X_{-j}^i denotes the "Markov blanket" of X_j^i, and Z_j is a local normalization constant for each node in the MRF. The set V_j' is the set of all vertices which connect to vertex j 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 so this approximation makes the problem tractable. Moreover, this approximation is known to yield a consistent estimate of the parameters [16]. That is, as the number of samples increases, parameter estimates using pseudo-likelihood converge to the parameter values using true likelihood.

3.2 Optimizing L1-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. 5. 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 differentiable.

$$\begin{aligned} & \max_{\Theta,\alpha} \mathrm{ll}(\Theta) - \lambda_{node}\|\mathbf{v}\|_2 - \textstyle\sum_{1\le s<t\le p} \alpha_{st} \\ \text{subject to:} \quad & \forall(1 \le s < t \le p) : \alpha_{st} \ge \|w^{st}\|_q \end{aligned}$$

where the constraints hold with equality at the optimal (Θ,α).

One way to solve this reformulation is through a two stage procedure involving the use of projected gradients. In the first stage, we ignore the constraints, compute the gradient of the objective, and then 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:

$$\begin{aligned} & \min_{\Theta'\alpha'} \left\| \begin{bmatrix} \Theta' \\ \alpha' \end{bmatrix} - \begin{bmatrix} \Theta \\ \alpha \end{bmatrix} \right\|_2 \\ \text{subject to:} \quad & \forall(1 \le s < t \le p) : \alpha_{st} \ge \|w^{st}\|_q \end{aligned}$$

which finds the closest parameter vector to the vector obtained by taking the gradient step (by minimizing the Euclidean distance), while satisfying the original constraints. This problem can be solved efficiently for block-L_1 norms using Spectral Projected Gradients (SPG), as shown in [25]. Thus, we used the algorithm from [25] to solve this problem in our experiments. Methods based on projected gradients are guaranteed to converge to a stationary point [7], and convexity ensures that this stationary point is globally optimal.

4 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 [2, 17] and to perform fold recognition. The pioneering work of [22] used an approach to determine probabilistic dependencies they call SCA and observed that analyzing such patterns could provide insights into the allosteric behavior of the proteins and be used to design new sequences [28]. Others have since developed similar methods [11, 13, 14]. 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, [30] developed

an algorithm that determine conditional independencies to learn a Markov Random Field over sequences. Their constraint-based algorithm proceeds by determining 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 [31] and also develop a sampling algorithm for protein design using such models [32]. More recently, [38] 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. Also, their algorithm does not regularize the model and can 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.

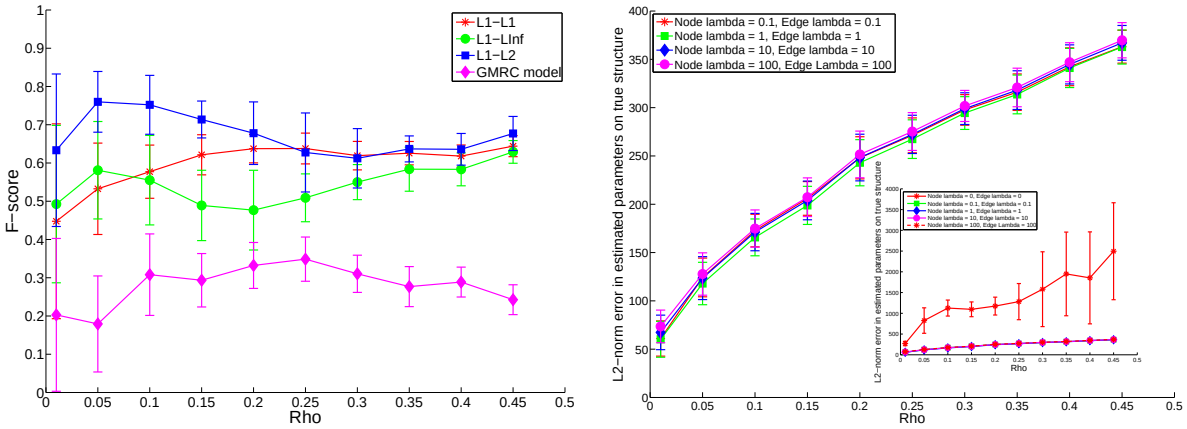
Block regularization is most similar in spirit to the group Lasso [40] and the multi-task Lasso [3]. Lasso [34] is the problem of finding a linear predictor, by minimizing the squared loss of the predictor with an L_1 penalty. It is well known that the shrinkage properties of the L_1 penalty lead to sparse predictors. The group Lasso extends this idea by grouping the weights of some features of the predictor using an L_2 norm, [40] 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.

[21] introduced structure learning in MRFs with a pure L_1 penalty, i.e. $q = 1$, but do not go further to explore the cases when $q = \{2, \infty\}$. They also use a different approximation to the likelihood term, using Loopy Belief Propagation. [25] apply block-regularized structure learning to the problem of detecting abnormalities in heart motion. Particularly they develop an efficient algorithm for tractably solving the convex structure learning problem, based on projected gradients. We use their algorithm in this paper.

5 Results

Given the probabilistic framework defined in Sec. 2, and the optimization objectives and algorithms defined in Sec. 3, we are now in a position to learn a graphical model given the sequence record of a protein family. The optimization framework has two major parameters that can be varied: the norm of the block-regularizer (L_1,L_2,L_∞) and the penalty parameters (λ_v, λ_e). To understand the effects of these parameters, we first evaluated our method 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 our framework 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 using the three types of norms; (ii) the greedy algorithm of [33, 30] ("GMRC method"); and (iii) a simpler greedy algorithm that uses the metric suggested in [22] ("$\Delta\Delta G^{stat}$"). We also compare our performance with the Profile



(A) (B)

Figure 2: (A) Edge occurrence probability ρ versus F-score for the structure learning methods we propose, and the method proposed in [30]. (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

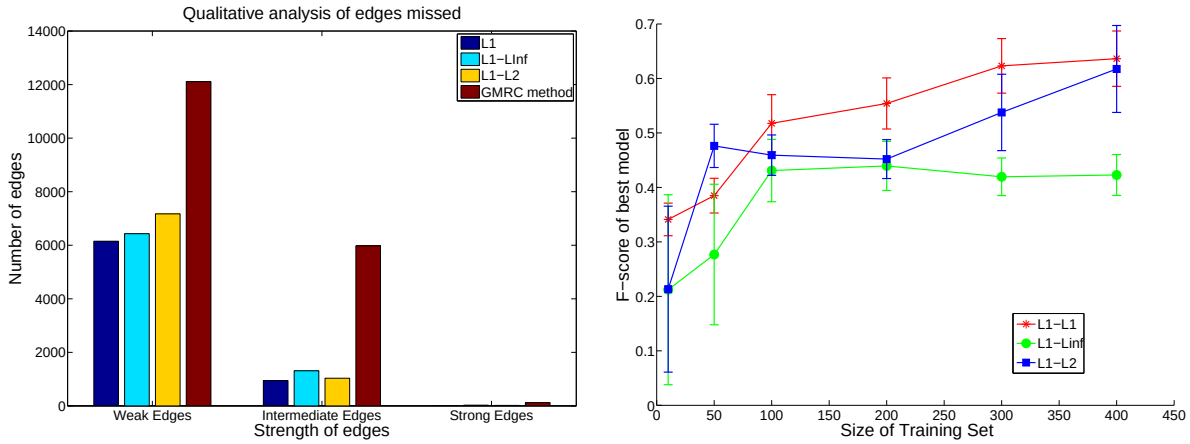
Hidden Markov Models [10] used by [4].

We note that the GMRC method only considers edges that meet certain coupling criteria (see [33, 30] for details). In particular, it is only capable of learning sparse graphs (fewer than 100 edges), regardless of choice of run-time parameters. In contrast, our method returns a full spectrum from disconnected to completely connected graphs. In our experiments, we use our parameter estimation code on their graphs, and compare ourselves to the best graph they return.

In what follows, we demonstrate that the three block regularizers we consider consistently perform comparably. Moreover, we find all our methods significantly out-perform each of the other algorithms, in a variety of scenarios. We also applied our approach to two real protein families, the PDZ and WW domains. By comparing the goodness-of-fit of various models to test sets, we demonstrate that MRFs significantly out-perform profile HMM-based models. Additionally, our models achieve higher goodness of fit to the test set than the GMRC method, even when we learn models of similar sparsity. Our results demonstrate that the use of block-regularized structure learning algorithms can result in higher-quality MRFs than those learnt by the GMRC method.

5.1 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



(A) (B)

Figure 3: (A) Qualitative grouping of edges missed by our methods and the GMRC method (B) Sensitivity of structure learning to size of training set

to have many small weights and a few large ones.

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 each of the three block regularization norms.

We first test our accuracy on structure learning. 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. 2-A shows the global comparison across the three regularizers as a function of ρ. The accuracy is measured using the F-score (the harmonic mean of precision and recall) of the edge set. In each case we use the best model learnt across the different values of λ_e (we consider the F-scores as a function of λ_e for each method in the appendix). Figure 2-A also compares our structure learning method with the algorithm in [30]. We evaluate their method over a wide range of parameter settings and select the best model. Figure 2-A shows that our methods significantly out-perform their method for *all* values of ρ. We see that over all settings our best model has an average F-score of *at least* 0.6. We conclude that we are able to infer fairly accurate structures given the proper choice of settings.

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

5.2 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. On the other hand, we note that there is a distinct "knee" in the graphs of the number of edges versus the imputation error (see Fig. 5 and Fig. 6). We evaluated modified AIC (Akaike Information Criteria) and BIC (Bayesian Information Criteria) for model selection due to their theoretically appealing properties. In the finite sample case we find that BIC performs well when the true graph is sparse, while AIC performs well when the true graph is dense. However, neither method performs as well over the entire range of graphs as selecting a model at the knee of the imputation error curve. We find this method is able to optimally trade-off accuracy and sparsity over a wider range of settings due to which we use it in our experiments. We discuss the information criteria in the appendix, and provide some general suggestions for their use.

5.3 A generative model for the WW domain

The WW domain family (Pfam id: PF00397 [4]) 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 [28] and [24], 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 (highlighted in yellow in Fig. 4), 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 vivo*. 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 90% 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

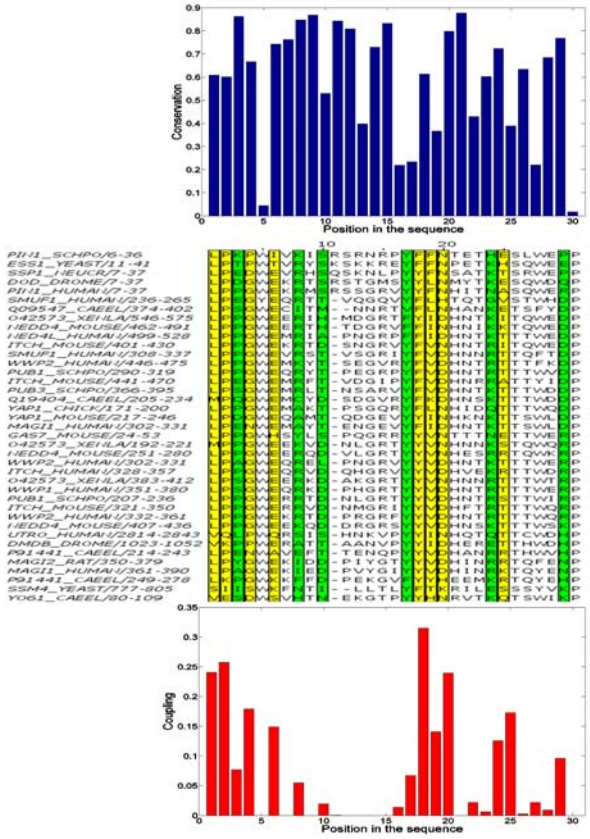
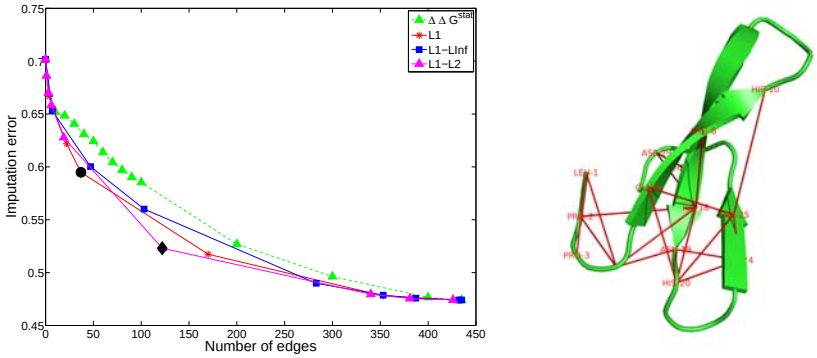


Figure 4: Part of the WW MSA we used. In yellow are positions identified by [24] as being critical to folding. Positions we additionally identify are in green. The conservation profile (top) shows the entropy (scaled) at each position. The coupling profile is shown below the MSA.



(A) (B)

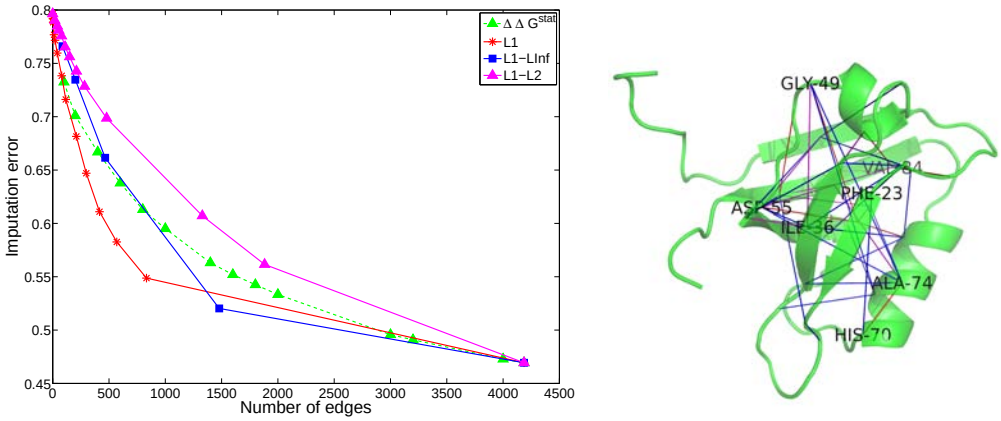
Figure 5: (A) Number of edges versus imputation error for the WW domain. The model at the knee was a model that minimized the L_1-L_2 norm and had 122 edges (shown with a black diamond). The model we used for AUC comparisons is shown with a black circle. In comparison, the GMRC method returned a graph with 21 edges, which had an imputation error of 0.6522, the Profile HMM had an imputation error of 0.7034. Our best model of comparable sparsity (28 edges) had a much lower imputation error of 0.622. The graph also shows the imputation error for a method based on statistical coupling ($\Delta\Delta G^{stat}$). (B) The edges of the model used in the discriminative task, overlaid on the structure of the WW domain of a ubiquitin protein ligase(PDB id: 1I5H)[5]

used to learn models for each of the three norms, using multiple settings of λ_e. Given the structure of the graph, parameters were learned using $\lambda_v = 1, \lambda_e = 1$.

Fig. 5-A shows the imputation error of each of the learnt models on the test set. As can be seen, the first few edges contribute to a significant decrease (the first 20 edges contribute to a third of the total decrease in imputation error). This is consistent with [28, 30] who find a small set of vertices and edges to be important. However, the knee of the curve is much further down, at 122 edges. We compare our results to the GMRC method of [30], Profile HMMs [10] and to a method that adds edges in the order of their statistical coupling ($\Delta\Delta G^{stat}$) (statistical coupling is also used by the SCA method). We find that our imputation errors are considerably 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". We construct a shuffled MSA by taking the natural MSA and randomly permuting the amino acids within the same column for each column. 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 strongly constrained in the original MSA. Fig. 4 shows the results of this analysis; we identify 15 positions in the MSA including all 8 positions previously identified by [24].

In addition we also performed a retrospective analysis of the artificial sequences designed by [24]. We attempt to distinguish sequences that folded from those that didn't. To make a fair comparison we select a model of comparable sparsity (with 38 edges) to that in [24]. Although this is a discriminative (folded or not) test of a generative model we achieve a high AUC of 0.883



(A) (B)

Figure 6: (A) Number of edges versus imputation error for the PDZ domain. Comparisons to a method based on statistical coupling ($\Delta\Delta G^{stat}$) are shown. The imputation error of the model returned by the GMRC method (with 41 edges) was 0.7596, and that of the Profile HMM was 0.7801. In comparison, our best model of comparable sparsity to the GMRC method had an imputation error of 0.739. (B) Edges learnt from our models overlaid on the structure of PDZ domain of PSD-95(PDB id:1BE9). Edge colors indicate the strength of the coupling (red being the strongest, and blue being the weakest)

(the ROC curve is shown and described in the appendix). 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 [30] (AUC of 0.82) and the Profile HMM (AUC of 0.8319) we see that we are able to better distinguish artificial sequences that fold from those that don't.

5.4 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 trans-membrane 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 ([22]) and Molecular Dynamics simulations [9], to NMR based experimental studies ([15]).

We use the MSA from [22]. 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, with multiple settings of λ_e in each case and then computed the imputation error on the test set (shown in Fig. 6-A). For further analysis, we selected the model that formed the knee on the curve with the best imputation error. This model had around 700 edges. Fig. 6-B

shows a subset of the edges colored according to their strength(red strongest; blue weakest).

The SCA based approach of [22] 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 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 [22], but has been implicated by NMR experiments [15] as being dynamically linked to the active site. Thus, our method appears to capture a richer set of interactions than those possible using SCA.

6 Discussion and Future Work

In this paper we have proposed a statistical sequence-based approach to modeling the evolutionary pressures on a protein family. Overall, we find that 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. We demonstrate the utility of our method in identifying constraints useful both in protein design and in furthering our understanding of protein function and regulation.

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 [23, 12]. 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 subtrees, 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.

However, 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.

Designing proteins from a generative sequence based model such as ours could be greatly enhanced by incorporating structure based information which explicitly models the physical constraints of the protein. Such information could easily be incorporated either through the use of informative priors (e.g., interaction energies, etc), or by the addition of edge features.

Finally, we find that while block-regularization with q = 1 takes only a few hours to learn a single model for the PDZ domain (and a few minutes for the WW domain), while using $q = \{2, \infty\}$ can take as long as a day. We have experimented with several efficiency tricks including warm-starts and pruning edges using a mutual information based cut-off [38]. Using these, we have run experiments on families with upto 800 positions in a few hours. [26] recently proposed a new method specifically to optimize costly functions, using a quasi-Newton algorithm which uses local curvature of the objective to approximate its second derivative. Typically, this leads to much faster convergence, and we expect this to be applicable directly to our methods.

A Appendix

A.1 Comparison of structures learnt at different regularization levels

Fig. 7 shows our performance in predicting the true structure by using L_1-L_2 (Fig. 7-A), L_1-L_∞ (Fig. 7-B), and L_1 (Fig. 7-C). The accuracy is measured using the F-score (the harmonic mean of precision and recall) of the edge set. We observe that for all settings of ρ each of the block regularizers learn fairly accurate graphs at some value of λ_e. We find that $L_1 - L_\infty$ requires higher regularization than the other norms to achieve comparable F-scores and sparsity. This is because the L_∞ norm of a vector is strictly less than its L_2 and L_1 norms.

A.2 Model selection using information criteria

We consider modifications to two widely used model selection strategies. The Bayesian Information Criterion (BIC) [27], is used to select parsimonious models and is known to be asymptotically consistent in selecting the true model. The Akaike Information Criterion (AIC) [1], typically selects denser models than the BIC, but is known to be asymptotically consistent in selecting the model with lowest predictive error. In general, they do not however select the same model and their strengths cannot be shared [39].

We use the following definitions,

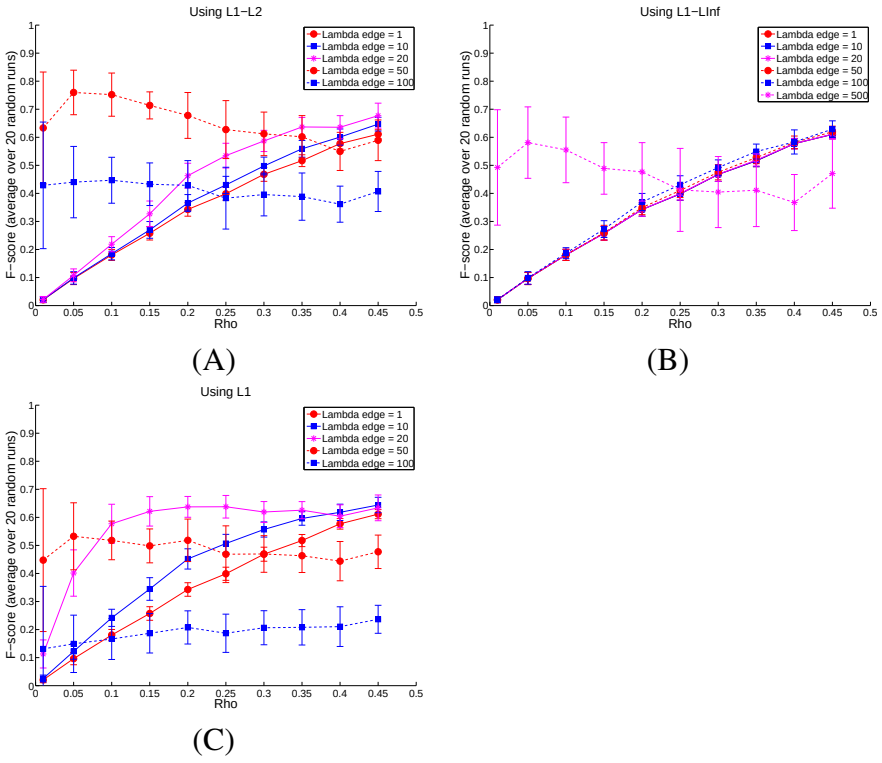


Figure 7: F-scores of structures learnt by using (A) L_1-L_2 norm, (B) L_1-L_∞ norm and (C) L_1 norm. Each figure shows the average and standard deviation of the F-score across 20 different graphs as a function of ρ, the probability of edge-occurrence.

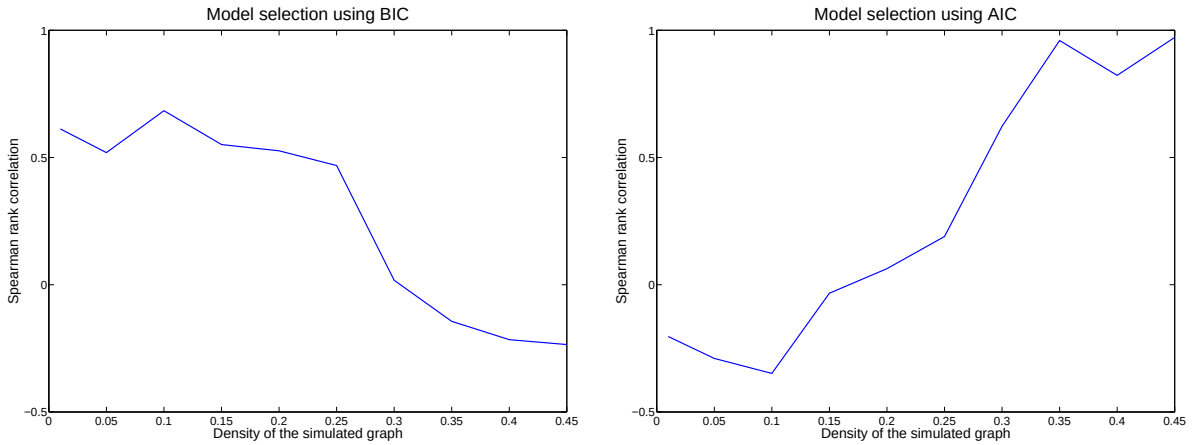


Figure 8: Graph density versus the rank correlation for ranking and selection using (A) BIC (B) AIC

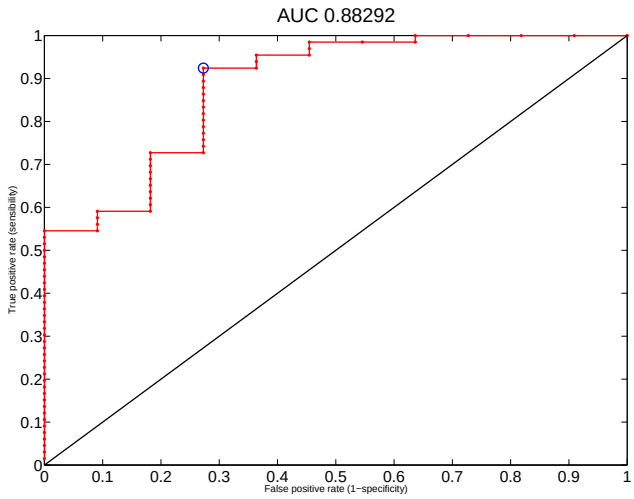


Figure 9: Receiver operating characteristic (ROC) curve of our model for the task of distinguishing artificial WW sequences that fold from those that don't.

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

Where we use the pseudo log-likelihood approximation to the log-likelihood. We evaluate the likelihood on the *training* sample to score the different models. We use the L_1-norm of the learned weight vectors as an estimate of the degrees of freedom (df), and n is the number of training sequences. We typically select the model which has the lowest AIC/BIC score.

Figure 8 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.

A.3 Receiver operating characteristic curve

We consider the task of distinguishing artificial sequences that were found to take the WW fold from those that did not. All sequences and their labels (folded in vivo or not) are from [24]. The ROC curve (Fig. 9) is obtained by varying a threshold on scores (we use the unnormalized likelihood as the score). Sequences above the threshold are predicted to fold. For each threshold we calculate the sensitivity and specificity and show the resulting curve.

References

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

[2] 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.

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

[4] 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.

[5] 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.

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

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

[8] David Maxwell Chickering and Craig Boutilier. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[9] 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.

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

[11] 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.

[12] Joseph Felsenstein. *Inferring Phylogenies*. Sinauer Associates, September 2003.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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, November 2003.

[21] 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, 2007.

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

[23] 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.

[24] 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.

[25] 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.

[26] Mark Schmidt, Ewout van den Berg, Michael P. Friedlander, and Kevin Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. *Proc. of Conf. on Artificial Intelligence and Statistics*, pages 456–463, 2009.

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

[28] 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.

[29] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search (Lecture Notes in Statistics)*. Springer-Verlag, 1993.

[30] 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, 2008.

[31] 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, 2009.

[32] 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, 2009.

[33] 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.

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

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

[36] 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.

[37] 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.

[38] 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.

[39] Yuhong Yang. Can the strengths of aic and bic be shared? *BIOMETRICA*, 92:2003, 2003.

[40] 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.