PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

Graphical models of protein–protein interaction specificity from correlated mutations and interaction data

John Thomas,[1] Naren Ramakrishnan,[2]* and Chris Bailey-Kellogg[1]*

[1] Department of Computer Science, Dartmouth College, Hanover, New Hampshire 03755
[2] Department of Computer Science, Virginia Tech, Blacksburg, Virginia 24061

ABSTRACT

Protein–protein interactions are mediated by complementary amino acids defining complementary surfaces. Typically not all members of a family of related proteins interact equally well with all members of a partner family; thus analysis of the sequence record can reveal the complementary amino acid partners that confer interaction specificity. This article develops methods for learning and using probabilistic graphical models of such residue "cross-coupling" constraints between interacting protein families, based on multiple sequence alignments and information about which pairs of proteins are known to interact. Our models generalize traditional consensus sequence binding motifs, and provide a probabilistic semantics enabling sound evaluation of the plausibility of new possible interactions. Furthermore, predictions made by the models can be explained in terms of the underlying residue interactions. Our approach supports different levels of prior knowledge regarding interactions, including both one-to-one (e.g., pairs of proteins from the same organism) and many-to-many (e.g., experimentally identified interactions), and we present a technique to account for possible bias in the represented interactions. We apply our approach in studies of PDZ domains and their ligands, fundamental building blocks in a number of protein assemblies. Our algorithms are able to identify biologically interesting cross-coupling constraints, to successfully identify known interactions, and to make explainable predictions about novel interactions.

Proteins 2009; 76:911–929.

Key words: residue coupling; co-evolution; probabilistic model; protein–protein interactions; PDZ domains; molecular recognition.

INTRODUCTION

Extant sequences in a protein family provide evidence for constraints on choices of amino acids. Some residues may be strictly conserved, allowing only a single amino acid type in order to preserve proper structure and function. Other residues may tolerate some mutations, and in some of these cases, proper structure and function may require compensating mutations for other residues. We call such co-evolving residue pairs *coupled*. Conservation and coupling arise from various sources, including internal properties of the protein (maintaining overall stability and functionality[1–3]), environmental forces (e.g., adaptations for thermal extremes[4]), and interactions with other proteins (e.g., forming complementary binding regions[5,6]). We focus here on co-evolution of residues in interacting protein families, and call co-evolving residue pairs (one from each family) *cross-coupled*. Key tasks in cross-coupling studies include identifying cross-coupled residues, abstracting the cross-coupling information into a model, and using the model predictively.

One type of approach to identifying cross-coupled residues is to draw on the extensive literature for studying coupling within individual protein families,[7–12] and employ metrics such as correlation or mutual information between amino acid types at a pair of positions over the protein family/families.[6,13–15] The basic idea is to infer coupling from correlation—if a pair of (cross-family) residues has correlated amino acid types, then that could be due to a compensatory evolutionary process maintaining and adjusting the protein–protein interaction. Alternatively, phylogeny can be explicitly incorporated by testing whether an independent model or a dependent model best explains observed amino acid types for residue pairs.[16,17]

Cross-coupled residues identified by correlated mutations have been used directly as predictors of contact, on the assumption that compensating mutations tend to happen between physically interacting residues and that those tend to be in contact. While one study found that cross-coupled positions tend to be close in space and thus useful for

Additional Supporting Information may be found in the online version of this article.
Grant sponsor: US NSF; Grant numbers: IIS-0444544, IBN-0219332, EIA-0103660.
*Correspondence to: Naren Ramakrishnan, 2050 Torgersen Hall, Blacksburg, VA 24061.
E-mail: naren@cs.vt.edu or Chris Bailey-Kellogg, 6211 Sudikoff Laboratory, Hanover, NH 03755.
E-mail: cbk@cs.dartmouth.edu
Received 2 October 2008; Revised 7 January 2009; Accepted 26 January 2009
Published online 2 February 2009 in Wiley InterScience (www.interscience.wiley.com).
DOI: 10.1002/prot.22398

guiding docking,[13] another study tested a number of typical coupling metrics and found them not to be indicative of intermolecular contacts.[14] A recent large-scale study did find co-evolving pairs to be in general closer than other pairs.[17] Another indirect manifestation of cross-coupling is in differential conservation, which has been used to identify binding sites, for example, by evolutionary trace,[18] ConSurf,[19] and phylogenetic motifs.[20]

One key difference between coupling and cross-coupling studies is that in coupling studies, correlation is evaluated between pairs of amino acids from the same sequence, whereas in cross-coupling studies, correlations are between pairs from different sequences, and it may not be clear which sequences actually interact (and thus should contribute to correlation analysis). This issue is easily resolved if we assume a one-to-one relationship, for example, when we only consider interactions between proteins from the same organism, and only assess correlated mutations in such pairs. In that case, we can form a "super-sequence" by concatenating the partner sequences and employ traditional coupling metrics.[6,14] However, these assumptions break down with multiple proteins from an organism, or cross-organism interactions.

In moving to a more general model of possible interactions (many-to-many, rather than one-to-one), cross-coupling can provide insights into interaction *specificity*, when not all members of one family interact equally well with all members of the other family. Specificity due to cross-coupling is often modeled indirectly. For example, consensus sequence motifs[21] for different classes of ligands recognized by a family (e.g., the "class I" and "class II" motifs for ligands of PDZs) capture the cross-coupling between sets of sequences. Such motifs can then be used predictively—the presence of a particular motif in a given ligand suggests that it will be recognized by corresponding members of the recognition family.[22–25] Such coarse representations, however, lose information; consequently, there is often debate as to how many classes are appropriate and how to partition the classes.[25,26]

SPOT[27,28] employs a more refined model by breaking recognition modules (e.g., SH3, PDZ, WW) into classes by recognized ligand, and gathering position-specific amino acid type statistics within the separate classes. Additional experimental data regarding interactions of the seven PDZ domains on hINADL against a combinatorial peptide library was gathered in order to improve SPOT statistics.[29] As with consensus sequence binding motifs, SPOT models can be used to predict interactions; SPOT was able to successfully identify many of the natural binding partners of SH3 domains from a scan of the SWISSPROT database.[27]

This article develops a more general and powerful model of the cross-coupling basis for interaction specificity, by integrating sequence information and available interaction data in a probabilistic graphical model. By assessing frequencies of cross-coupled amino acid types, our approach provides more refined insights into the interactions conferring specificity than can be provided by simple consensus sequence motifs. By factorizing cross-coupling statistics into a formal probabilistic model rather than treating each cross-coupled pair as independent, our approach enables sound evaluation of interaction likelihoods. By representing constraints in a graphical model, our approach supports prediction of sequence interactions in an explainable manner, justifying predictions in terms of the underlying residue-level interactions. Finally, by utilizing interaction data and making explicit the assumptions regarding its completeness, our approach can handle one-to-one (e.g., pairs of proteins from the same organism) as well as many-to-many (e.g., experimentally identified interacting pairs) cases and can attempt to factor out the bias that may be present in the many-to-many case. We assume that the sequences are evolutionarily diverse, though it remains interesting future work to account for potential phylogenetic artifacts.

We demonstrate the use and effectiveness of our models in studies of PDZ domains and their ligands. Our models uncover a number of cross-coupling constraints supported by the literature and by structural evidence (even though we did not use structural information in identifying constraints). Experimentally validated PDZ/ligand interactions tend to have high likelihoods under our model, giving us confidence in making predictions for additional PDZ/ligand pairs. Simulation studies show that we can achieve relatively high predictive ability even with a sparse amount of interaction data. Finally, we show that our methods outperform the state-of-the-art existing method in predicting PDZ/ligand interactions.

METHODS

As input, we are given multiple sequence alignments (MSAs) $\mathcal{S}$ and $\mathcal{S}'$ for two protein families. We are also given an interaction table, $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{S}'$, containing information about which members of the protein families interact (see the left part of Fig. 1). Each column (residue) in the MSAs can be thought of as a random variable, taking on values from the set of possible amino acid types. Throughout, we use capital letters for random variables and lowercase for values, bold for sets, and prime marks ($'$) to distinguish the second family from the first. Thus, $\mathbf{V} = \{V_1, V_2, \ldots, V_n\}$ and $\mathbf{V}' = \{V'_1, V'_2, \ldots, V'_m\}$ are the random variables for the columns in $\mathcal{S}$ and $\mathcal{S}'$, respectively; V_i is the random variable for the ith column in $\mathcal{S}$ and V'_j is the random variable for the jth column in $\mathcal{S}'$. Furthermore, $\mathbf{a} = \{\text{ala,arg,asp}, \ldots, -\}$ is the set of amino acid types (note that we treat the gap character "–" as an amino acid type), $p(V_i = \text{ala})$ is the probability that column i in $\mathcal{S}$ is an alanine, and an entire sequence $\mathbf{v} = \{v_1, v_2, \ldots, v_n\}$ is an assignment of amino acids for all the random variables.

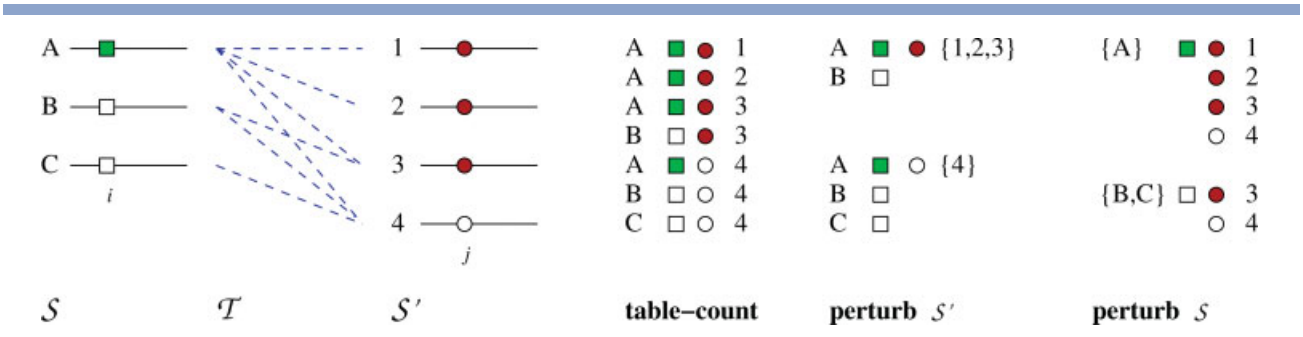

Figure 1

Alternative approaches for computing mutual information for residue cross-coupling. We are given two MSAs $\mathcal{S}$ and $\mathcal{S}'$ and a table $\mathcal{T}$ of interacting partners. We desire to assess the mutual information between column i in the left alignment (denoted by squares) and column j in the right alignment (circles). The "table-count" approach proceeds as if interacting sequences were concatenated. All amino acid combinations listed in the table are counted separately. In this case, when column j is filled, column i tends to be as well. The "perturbation" approach defines subsets over one alignment and joins through $\mathcal{T}$ to identify distributions of residues in the other alignment. For instance, in perturbing $\mathcal{S}'$ to the sequences with filled residues (1, 2, and 3), the set of partners contains sequences A and B, which yield one filled and one unfilled (square) residue, respectively. In perturbing $\mathcal{S}'$ to the sequence with unfilled residues (4), the set of partners contains sequences A, B, and C. In contrast to the table-count approach, these perturbations together make the two columns appear relatively independent. Similarly, we can define subsets in the left alignment and track distributions in the right alignment. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

In presenting our methods, we first discuss how to detect residues that are cross-coupled, drawing analogies to the detection of coupled residues within a single protein family. We then present our approach to learning a graphical model that provides a compact probabilistic representation for cross-coupling information. Finally, we show how to use such graphical models to score possible new interactions, thereby providing a mechanism for predicting whether or not a pair of proteins will interact, based on satisfaction of cross-coupling constraints.

Detecting residue cross-coupling

In the case of a single MSA, many statistical and information-theoretic measures have been employed to evaluate coupling between columns.[10] For instance, we can quantify the *mutual information* between columns i and j in $\mathcal{S}$, in terms of the column random variables V_i and V_j, as

$$MI(V_i, V_j) = \sum_{a \in \mathbf{a}} \sum_{b \in \mathbf{a}} p(V_i = a, V_j = b) \cdot \log \frac{p(V_i = a, V_j = b)}{p(V_i = a)\ p(V_j = b)} \quad (1)$$

A high mutual information between two residues indicates that knowing the amino acid type for one strongly restricts what the amino acid type for the other can be. A low value of mutual information, conversely, implies that the residues are quite independent of each other.

To extend mutual information to work with interacting families with a pair of MSAs, the key question is how to utilize the interaction table. We develop here two approaches with differing assumptions about the completeness of the data in the interaction table.

Our first approach concatenates sequences from the first family with their interaction partners from the second family, creating a single MSA suitable for traditional coupling analysis. The "table-count" panel in Figure 1 shows the resulting combinations for one pair of residues. Observe that sequences with many identified partners have more influence on the coupling. That is, if one protein is involved in many interactions while another is involved in only a few, the first protein will be overly represented in the merged MSA. If this overrepresentation is an artifact of how $\mathcal{T}$ was collected, then the identified couplings may be spurious. That is, if one protein in the family was extensively studied and has many known interactions while another has equally many interactions but they have not been experimentally identified yet, then the better-studied protein will have a larger contribution in the concatenated MSA leading to artificially created cross-coupling relationships. If, on the other hand, this overrepresentation is truly indicative of the underlying biological relationships, then the cross-coupling will be correctly identified.

To address the concern of possible overrepresentation, we develop an alternative "perturbation" approach (Fig. 1, right panels). This approach assesses how selection for a particular amino acid type (i.e., a perturbation, or conditioning context) in one family changes the distribution of amino acid types at another position in the partner family, for those sequences that interact with the selected ones. In the perturbation approach, multiple partners that have the same amino acid type at a particular position are treated as a single instance when considering that position. In the example shown in Figure 1, this approach actually concludes (near) independence of columns i and j while the table-count approach appears more dependent. This is a desirable feature when the interaction table is heavily biased toward several

experimentally well-studied proteins that have a large number of interactions. On the other hand, it downplays the impact of truly promiscuous members and may miss some cross-coupling constraints.

We employ different techniques to estimate probabilities under the table-count vs. perturbation approaches. In computing mutual information [Eq. (1)], we can factor the joint probability, $p(V_i = a, V'_j = b)$, into the product of a conditional probability, $p(V_i = a \mid V'_j = b)$, times the marginal probability, $p(V'_j = b)$. The marginal probability is straightforward to compute as the number of sequences in $\mathcal{S}'$ that have amino acid type b in column j, or more formally, $|\{\mathbf{s}' \in \mathcal{S}' : s'_j = b\}|$. Then for the conditional probability, suppose we are selecting from $\mathcal{S}'$ those sequences with amino acid type b at column j, and computing the change in $\mathcal{S}$ of the distribution of amino acid type at column i. In the table-count approach, we compute, of those entries in the interaction table with b at column j in $\mathcal{S}'$, the fraction that also have a at column i in $\mathcal{S}$:

$$p(V_i = a|V'_j = b) = \frac{|\{(\mathbf{s}, \mathbf{s}') \in \mathcal{T} : s_i = a, s'_j = b\}|}{|\{(\mathbf{s}, \mathbf{s}') \in \mathcal{T} : s'_j = b\}|} \quad (2)$$

On the other hand, in the perturbation approach we compute, of the sequences in $\mathcal{S}$ that have partners in $\mathcal{S}'$ with b at column j, the fraction that also have a at column i:

$$p(V_i = a|V'_j = b) = \frac{|\{\mathbf{s} : s_i = a \wedge \exists \mathbf{s}' \in \mathcal{S}' \text{ s.t. } s'_j = b \wedge (\mathbf{s}, \mathbf{s}') \in \mathcal{T}\}|}{|\{\mathbf{s} : \exists \mathbf{s}' \in \mathcal{S}' \text{ s.t. } s'_j = b \wedge (\mathbf{s}, \mathbf{s}') \in \mathcal{T}\}|} \quad (3)$$

[In this and other equations, we use "s.t." to denote "such that" before the predicate(s) in an existential quantification, and a colon to do so in a set comprehension. We use logical operators ($\wedge$ and $\neg$) to denote boolean connectives ("and" and "not," respectively).]

Similar expressions follow if we use the other type of perturbation, computing $p(V'_j = b \mid V_i = a)$. As we will see later, the formulas we employ in our graphical models avoid the asymmetry in the two types of perturbations.

Table-count and perturbation represent the only two informative approaches to computing mutual information with our formalization of families and interaction table. Using database terminology, there are two components to computing mutual information (see Fig. 1): projecting onto the columns of interest and joining via the interaction table. Projection maps sequences to single amino acids, removing duplicates; joining creates pairs (of either sequences or amino acids) of interacting partners. Table-count does the join first; perturbation projects onto a column in one family, joins, and then projects onto a column in the other family. Doing both projections first would only produce a list of observed amino acid pairs, losing any frequency information. Another way of seeing the difference in the two approaches is to consider how they count "votes" for coupling. In table-count, each interaction casts a vote, whereas in perturbation, each sequence from the family opposite the perturbation casts a vote. Thus in perturbation, a sequence interacting with multiple sequences that have the same amino acid type in the column of interest essentially divides its vote among them. This voting perspective suggests other possible ways to assess coupling, for example, by also considering sequence similarity or evolutionary relationships within a family when tallying votes (as ClustalW[30] does for multiple alignment). Because of limited data for assessing all the possibilities, we stick with the distinct approaches of table-count and perturbation.

Learning graphical models of residue cross-coupling

While the approaches presented in the previous section are sufficient to identify potentially cross-coupled residues, our goal is to encapsulate cross-coupling constraints in a model that can be used predictively ("is it likely that these two new proteins from these families interact?") and that provides a sound probabilistic semantics for such predictions. To do that, we develop in this section an approach to learning what we call "graphical models of residue cross-coupling" (GMRCCs); the next section shows how to use a model to predict interactions.

A GMRCC $G = (\mathbf{V}, \mathbf{V}', E)$ is a bipartite graph where vertices $\mathbf{V} = \{V_1, \ldots, V_n\}$ and $\mathbf{V}' = \{V'_1, \ldots, V'_m\}$ denote the random variables for the columns in the two MSAs and edges $E \subseteq \mathbf{V} \times \mathbf{V}'$ represent dependence and independence of the random variables. The traditional semantics of undirected graphical models focuses on probabilistic independence: two vertices are conditionally independent given their neighbors in the graph. To see this, and to make the jump from identifying cross-coupled residues to constructing such a model that appropriately factorizes them, it is important to recognize that a simple list of cross-coupled residues might be redundant. For example (examples like this arise in the PDZ/ligand study in the Results section), suppose that two residues in one family are either HV or QA, and that the sequences with HV interact with sequences with a T while those with QA interact with sequences with a Y. Then we would detect both H–T vs. Q–Y cross-coupling and V–T vs. A–Y cross-coupling. However, note that these two cross-couplings are in fact redundant; once we have H–T, we also have V–T, and vice-versa. If we were to evaluate a new pair of sequences for probability of interaction by combining scores separately provided by

each cross-coupled pair (testing if the amino acid types in the new sequences are consistent with the expected correlated amino acid types), we might artificially be viewing these pairs of couplings as independently confirming the cross-coupling hypothesis whereas there is really only one piece of evidence available. The crux of the issue is that couplings are themselves dependent on each other and such information has to be carefully factored out.

In order to properly account for dependence and independence, we must be able to assess whether a pair of residues is highly cross-coupled in a context where we already know another residue (or some other residues). To do so, we move from mutual information to conditional mutual information. Consider a pair of potentially cross-coupled residues V_i and V_j, in the context of some other residue V_k. Then the conditional mutual information, written $MI(V_i, V_j \mid V_k)$, is a low value when, if we know the value of V_k, then knowing the value of V_i does not provide much more information about the value of V_j. Thus even if two residues have high pairwise mutual information, they may have low mutual information conditioned on another residue. This is the case in the example in the preceding paragraph, with the residues from the one family alone or conditioned on the residue from the other family.

We employ a perturbation subsetting method to estimate conditional mutual information. This method is inspired by the statistical coupling analysis method of Lockless and Ranganathan[8] but preserves symmetry; we have previously utilized it for identifying single-family coupling.[12] The conditional mutual information between columns i and j, given column k, is

$$MI(V_i, V_j|V_k) = \sum_{c\in\mathbf{a}^*} p(V_k = c)\left[\sum_{a\in\mathbf{a}}\sum_{b\in\mathbf{a}} p(V_i = a, V_j = b|V_k = c) \cdot \log\frac{p(V_i = a, V_j = b|V_k = c)}{p(V_i = a|V_k = c)\;\; p(V_j = b|V_k = c)}\right] \quad (4)$$

The "perturbations" here are the selections of particular amino acid types for column k, which might or might not change the distributions at columns i and j. Since this article focuses strictly on cross-coupling constraints, we assess conditional mutual information with V_i and V_j in one family and V_k in the other. As mentioned at the end of the previous section, this also avoids the potential asymmetry in selecting which type of perturbation to perform. The probability distributions in the equation are estimated from the MSAs; we give further details below. Note that we compute the conditionals by conditioning column k only to "frequent-enough" amino acid types $\mathbf{a}^* \subset \mathbf{a}$ (we use only those in at least 15% of the sequences). This is a typical approach to ensure that we maintain fidelity to the original MSA.[8]

To learn a GMRCC, we can aim to apply Eq. (4) to find edges that decouple other residues. Unfortunately, this is fraught with the typical difficulties of estimating conditional mutual information reliably. For example, it is quite unrealistic to expect the conditional mutual information to be exactly 0, due to noise, finite sample size, etc. Instead, we define a score for a GMRCC that measures the amount of "residual" cross-coupling that remains, given some edges E:

$$\text{Score}(\mathbf{V}, \mathbf{V}', E) = \sum_{V_i\in\mathbf{V}}\sum_{V_j\in\mathbf{V}} MI(V_i, V_j|N(V_i)) + \sum_{V'_i\in\mathbf{V}'}\sum_{V'_j\in\mathbf{V}'} MI(V'_i, V'_j|N(V'_i)) \quad (5)$$

where $N(\cdot)$ is set of the neighbors of the vertex according to the edges. We can use this score to measure the overall reduction in mutual information that results from the addition of a single edge e, by comparing the score with E vs. the score with $E \cup \{e\}$.

This leads to a sequential algorithm to build a GMRCC (Algorithm 1): begin with a graph that has no edges; pick an edge that most improves the score; conditional on the context provided by this edge, pick a second edge; and so on. This greedy procedure is analogous to the procedure we have previously developed to capture constraints on a single protein family[12] and is similar in spirit to the "sparse candidate" algorithm of Friedman et al.[31] This algorithm has the advantage of being applicable to both the table-count and perturbation views of the interaction table. In addition, it can employ a prior on the set of edges under consideration (D in the algorithm), for example a structural prior limiting the edges to residue pairs that are near each other in the complex. When D includes all possible edges, we call it the "uninformative prior."

Algorithm 1 LearnGMRCC(D, $\mathcal{S}$, $\mathcal{S}'$, $\mathcal{T}$)

Input: a set of possible cross-coupling edges D, two multiple sequence alignments $\mathcal{S}$ and $\mathcal{S}'$, and an interaction table $\mathcal{T}$

Output: a GMRCC $G = (\mathbf{V}, \mathbf{V}', E)$

```
V = {V_1, ..., V_n}; V' = {V'_1, ..., V'_m}; E ← ∅
s ← Score(V, V', E)
for all e ∈ D do
    C(e) ← s − Score(V, V', {e})
end for
repeat
    e ← arg max_{e∈D−E} C(e)
    if e is significant then
        E ← E ∪ {e}
        s ← s − C(e)
        for all e' ∈ D − E s.t. e and e' share a vertex do
            C(e') ← s − Score(V, V', E ∪ {e'})
        end for
    end if
until stopping criterion satisfied
```

The algorithm terminates when none of the available edges are statistically significant. We focus here on assessing individual significances of edges rather than the significance of the entire model. This allows us to ensure that every cross-coupled residue pair is statistically meaningful. After identifying the edge that would best reduce the Score, the algorithm ensures that the edge is significant before adding it to the growing model. We employ a *P*-value approach to test the hypothesis that two vertices are truly independent rather than cross-coupled; smaller *P*-values indicate stronger confidence in dependent relationships. A χ^2 formulation compares the observed number of times that a particular amino acid pair appears in interacting pairs of sequences, against the expected number of such observations:

$$\chi^2 = \sum_{a \in \mathbf{a}_i} \sum_{b \in \mathbf{a}_j} \frac{(f_{i,j}(a,b) - g_{i,j}(a,b))^2}{g_{i,j}(a,b)} \tag{6}$$

Here $\mathbf{a}_i$, $\mathbf{a}_j$ are the amino acid types at column i in one MSA and j in the other, $f_{i,j}(a,b)$ the observed number of co-occurrences of a at column i and b at column j in sequences that interact in $\mathcal{T}$, and $g_{i,j}(a,b)$ the expected number of such co-occurrences if the positions were independent (see below). We note that this χ^2 formulation naturally corresponds to the counting assumptions underlying the table-count method; it would be interesting to develop an analogous formulation for the assumptions underlying the perturbation method. To compute a *P*-value, we determine the probability of getting a χ^2 value, with $(|\mathbf{a}_i| - 1)(|\mathbf{a}_j| - 1)$ degrees of freedom, having a magnitude at least as large as obtained under this formula. To account for multiple hypothesis testing, a simple Bonferroni correction could be applied, scaling the *P*-value threshold by the number of tests performed, $n \cdot m$, where n and m are the numbers of columns in the two MSAs.

It is straightforward to tabulate $f_{i,j}(a, b)$, the observed number of co-occurrences of amino acid pairs with respect to a given interaction table and pair of alignments. To compute the expected number of co-occurrences, $g_{i,j}(a, b)$, while ensuring fidelity to the given interaction table, we employ a randomization approach. We permute the alignments column-wise, thereby rendering the residues independent, and then do the co-occurrence tabulation on the permuted alignments. Performing a large number (say 100,000) of permutations allows us to compute an appropriate statistic for the expected number of co-occurrences.

The runtime of the algorithm depends on $n + m$, the size of $\mathbf{V} \cup \mathbf{V}'$, along with d, the maximum degree of nodes in the prior. A naïve implementation would require $O(d(n + m)^3)$ mutual information computations per iteration, but we can bring this down to $O(d(n + m))$ per iteration by caching mutual information estimates (C in Algorithm 1) and revising these estimates only for those pairs of residues whose conditioning contexts have changed (lines 9–11).

Predicting protein–protein interactions

The probabilistic semantics of a graphical model enables the evaluation of the joint probability of a set of values for its random variables. For a GMRCC, the values specify two amino acid sequences, and the joint probability captures how likely it is that they interact. Given a GMRCC $G = (\mathbf{V}, \mathbf{V}', E)$ and a new pair of sequences (one from each family) $\mathbf{v} = \{v_1, \ldots, v_n\}$ and $\mathbf{v}' = \{v'_1, \ldots, v'_m\}$, we write the probability of interaction as $p_G(I(\mathbf{v}, \mathbf{v}'))$, where I is shorthand for "interact."

Using the standard semantics for an undirected graphical model,[32] this joint probability is computed as

$$p_{G=(\mathbf{V},\mathbf{V}',E)}(I(\mathbf{v},\mathbf{v}')) = \frac{1}{Z} \cdot \frac{\prod_{e=(V_i,V'_j) \in E} p(I(\mathbf{v},\mathbf{v}')|V_i = v_i, V'_j = v'_j)}{\prod_{V_i \in \mathbf{V}:\deg(V_i)>1} p(I(\mathbf{v},\mathbf{v}')|V_i = v_i)^{\deg(V_i)-1} \prod_{V'_j \in \mathbf{V}':\deg(V'_j)>1} p(I(\mathbf{v},\mathbf{v}')|V'_j = v'_j)^{\deg(V'_j)-1}} \tag{7}$$

Here, $\deg(\cdot)$ indicates the degree of the vertex and Z normalizes the products into a probability measure (the equation for Z is given in the Supporting information). The joint probability is given by the product of likelihoods defined over the edges divided by those defined over the edge adjacencies. The bipartite form of a GMRCC leads to the ability to compute exact likelihoods efficiently. The edge adjacencies are simply the vertices, and they contribute one fewer term than their degree (e.g., a vertex with two edges has a single contribution).

We must provide distributions for the terms contributed to Eq. (7) from the edges and vertices. We focus on the edge contributions; vertex contributions are derived similarly (and are given in the Supporting information). The contribution of one edge can be rewritten using Bayes theorem:

$$p(I(\mathbf{v},\mathbf{v}')|V_i = v_i, V'_j = v'_j) = \frac{p(I(\mathbf{v},\mathbf{v}')) \cdot p(V'_j = v'_j|I(\mathbf{v},\mathbf{v}')) \cdot p(V_i = v_i|V'_j = v'_j, I(\mathbf{v},\mathbf{v}'))}{p(V_i = v_i, V'_j = v'_j)} \tag{8}$$

For the prior probability of interaction, $p(I(\mathbf{v}, \mathbf{v}'))$, we simply use the number of observed interactions relative to the number possible:

$$p(I(\mathbf{v}, \mathbf{v}')) = \frac{|\mathcal{T}|}{|\mathcal{S}||\mathcal{S}'|} \tag{9}$$

The prior probability of not interacting is then $1 - p(I(\mathbf{v}, \mathbf{v}'))$.

The remaining terms depend on the connection of sequences via the interaction table, and thus have different estimators for the table-count and perturbation methods.

Table-count-based estimators

Let us first consider $p(V'_j = v'_j \mid I(\mathbf{v}, \mathbf{v}'))$. In the table-count approach, every interaction is treated equivalently, regardless of which members are interacting. Therefore, this probability can be estimated as the fraction of the observed interactions that have amino acid v'_j in column j:

$$p(V'_j = v'_j | I(\mathbf{v}, \mathbf{v}')) = \frac{|\{(\mathbf{s}, \mathbf{s}') \in \mathcal{T} : s'_j = v'_j\}|}{|\mathcal{T}|} \tag{10}$$

The next term, $p(V_i = v_i \mid V'_j = v'_j, I(\mathbf{v}, \mathbf{v}'))$, is computed in the table-count approach as the fraction of those interactions that have amino acid v'_j in column j, that also have amino acid v_i in column i:

$$p(V_i = v_i | V'_j = v'_j, I(\mathbf{v}, \mathbf{v}')) = \frac{|\{(\mathbf{s}, \mathbf{s}') \in \mathcal{T} : s_i = v_i, s'_j = v'_j\}|}{|\{(\mathbf{s}, \mathbf{s}') \in \mathcal{T} : s'_j = v'_j|\}} \tag{11}$$

Since the joint probability $p(V_i = v_i, V'_j = v'_j)$ depends on whether or not the sequences interact, we marginalize over those possibilities:

$$\begin{aligned} p(V_i = v_i, V'_j = v'_j) &= p(I(\mathbf{v}, \mathbf{v}')) \cdot p(V_i = v_i, V'_j = v'_j | I(\mathbf{v}, \mathbf{v}')) + p(\neg I(\mathbf{v}, \mathbf{v}')) \cdot p(V_i = v_i, V'_j = v'_j | \neg I(\mathbf{v}, \mathbf{v}')) \\ &= \frac{|\mathcal{T}|}{|\mathcal{S}||\mathcal{S}'|} \cdot \frac{|\{(\mathbf{s}, \mathbf{s}') \in \mathcal{T} : s_i = v_i, s'_j = v'_j\}|}{|\mathcal{T}|} + \frac{|\mathcal{S}||\mathcal{S}'| - |\mathcal{T}|}{|\mathcal{S}||\mathcal{S}'|} \cdot \frac{|\{(\mathbf{s}, \mathbf{s}') \notin \mathcal{T} : s_i = v_i, s'_j = v'_j\}|}{|\mathcal{S}||\mathcal{S}'| - |\mathcal{T}|} \\ &= \frac{|\{(\mathbf{s}, \mathbf{s}') \in \mathcal{T} : s_i = v_i, s'_j = v'_j\}| + |\{(\mathbf{s}, \mathbf{s}') \notin \mathcal{T} : s_i = v_i, s'_j = v'_j\}|}{|\mathcal{S}||\mathcal{S}'|} \\ &= \frac{f_i(v_i) f_j(v'_j)}{|\mathcal{S}||\mathcal{S}'|} \end{aligned} \tag{12}$$

Recall from Eq. (6) that f measures the frequency of given amino acids or amino acid combinations in specific positions across a MSA. The final reduction occurs because every sequence in $\mathcal{S}$ that has a v_i in column i either interacts or does not interact with every sequence in $\mathcal{S}'$ that has a v'_j in column j.

Perturbation-based estimators

While the table-count method estimates probabilities by considering each interaction separately, the perturbation method considers each sequence independently, regardless of how many interactions it is involved in. In this interpretation, $p(V'_j = v'_j \mid I(\mathbf{v}, \mathbf{v}'))$ is the fraction of the interacting sequences from $\mathcal{S}'$ that have amino acid v'_j in column j:

$$p(V'_j = v'_j | I(\mathbf{v}, \mathbf{v}')) = \frac{|\{\mathbf{s}' : s'_j = v'_j \wedge \exists \mathbf{s} \in \mathcal{S} \text{ s.t.} (\mathbf{s}, \mathbf{s}') \in \mathcal{T}\}|}{|\{\mathbf{s}' : \exists \mathbf{s} \in \mathcal{S} \text{ s.t.} (\mathbf{s}, \mathbf{s}') \in \mathcal{T}\}|} \tag{13}$$

In contrast with Eq. (10), note that rather than counting the number of interactions that have a sequence with v'_j in column j, we instead count the number of sequences involved in interactions that meet this criterion. Thus a sequence with numerous interactions receives the same representation as a sequence with only one interaction.

For $p(V_i = v_i \mid V'_j = v'_j, I(\mathbf{v}, \mathbf{v}'))$, we likewise compute, of those sequences that interact with a sequence with amino acid v'_j in column j, the fraction that also have amino acid v_i in column i:

$$p(V_i = v_i | V'_j = v'_j, I(\mathbf{v}, \mathbf{v}')) = \frac{|\{\mathbf{s} : s_i = v_i \wedge \exists \mathbf{s}' \in \mathcal{S}' \text{ s.t. } s'_j = v'_j \wedge (\mathbf{s}, \mathbf{s}') \in \mathcal{T}\}|}{|\{\mathbf{s} : \exists \mathbf{s}' \in \mathcal{S}' \text{ s.t. } s'_j = v'_j \wedge (\mathbf{s}, \mathbf{s}') \in \mathcal{T}\}|} \tag{14}$$

Finally, we again compute the joint probability $p(V_i = v_i, V'_j = v'_j)$ via marginalization:

$$
\begin{aligned}
p(V_i = v_i, V'_j = v'_j) &= p(I(\mathbf{v},\mathbf{v}')) \cdot p(V_i = v_i, V'_j = v'_j | I(\mathbf{v},\mathbf{v}')) \\
&\quad + p(\neg I(\mathbf{v},\mathbf{v}')) \cdot p(V_i = v_i, V'_j = v'_j | \neg I(\mathbf{v},\mathbf{v}')) \\
&= p(I(\mathbf{v},\mathbf{v}')) \cdot p(V_i = v_i | V'_j = v'_j, I(\mathbf{v},\mathbf{v}')) \cdot p(V'_j = v'_j | I(\mathbf{v},\mathbf{v}')) \\
&\quad + p(\neg I(\mathbf{v},\mathbf{v}')) \cdot p(V_i = v_i | V'_j = v'_j, \neg I(\mathbf{v},\mathbf{v}')) \cdot p(V'_j = v'_j | \neg I(\mathbf{v},\mathbf{v}')) \\
&= p(I(\mathbf{v},\mathbf{v}')) \cdot p(V_i = v_i | V'_j = v'_j, I(\mathbf{v},\mathbf{v}')) \cdot p(V'_j = v'_j | I(\mathbf{v},\mathbf{v}')) \\
&\quad + p(\neg I(\mathbf{v},\mathbf{v}')) \cdot \frac{|\{\mathbf{s} : s_i = v_i \wedge \nexists \mathbf{s}' \in \mathcal{S}' \text{ s.t. } s'_j = v'_j \wedge (\mathbf{s},\mathbf{s}') \in \mathcal{T}\}|}{\{\mathbf{s} : \nexists \mathbf{s}' \in \mathcal{S}' \text{s.t. } s'_j = v'_j \wedge (\mathbf{s},\mathbf{s}') \in \mathcal{T}\}|} \\
&\quad \cdot \frac{|\{\mathbf{s}' : s'_j = v'_j \wedge \nexists \mathbf{s} \in \mathcal{S} \text{ s.t. } (\mathbf{s},\mathbf{s}') \in \mathcal{T}\}|}{|\{\mathbf{s}' : \nexists \mathbf{s} \in \mathcal{S} \text{ s.t. } (\mathbf{s},\mathbf{s}') \in \mathcal{T}\}|}
\end{aligned}
\tag{15}
$$

Unlike in the table-count approach, there is no way in general to further reduce the combined terms. However, there are two important special cases in which they reduce to the formulas as the table-count approach: when each sequence in $\mathcal{S}$ interacts with exactly one sequence in $\mathcal{S}'$ and vice versa, and when every sequence in $\mathcal{S}$ interacts with every sequence in $\mathcal{S}'$.

Adding pseudocounts

It is likely that both the MSAs and the interaction table are somewhat incomplete—there are other proteins in the family, and there are other interactions that are possible. To ensure that such missing data do not cause the probability of a protein–protein interaction to be 0 (i.e., to satisfy the Hammersley-Clifford theorem[32]), we add "pseudocounts" to the likelihood. Pseudocounts are used routinely in sequence profiles and hidden Markov models[33] to allow for unobserved data, by adding a small count to each possible observation, so that something that has not (yet) been seen has a small but nonzero score. Here, we include two types of pseudocounts: "interaction pseudocounts," which account for potentially missing entries in the interaction table, and "residue pseudocounts," which account for missing data in the protein families.

Figure 2 shows the parameterization of these pseudocounts. Interaction pseudocounts give every possible interaction an a priori weight of ρ. Observed interactions (in the interaction table) then have weight of 1 + ρ, whereas the rest have a weight of ρ. Setting ρ = 0 is equivalent to assuming to that the interaction table represents ground truth, while higher values of ρ put less weight on the observed interactions. Similarly, residue pseudocounts give every possible amino acid type at every possible position an a priori weight of α. This is equivalent to adding 21 α (20 amino acids and a gap character) additional "pseudosequences" to the end of each protein family. By setting α to 0, we assume that the only members of a protein family are those that are in our MSA, while higher values of α allow for unobserved sequences.

It is straightforward but verbose to revise our estimators to include pseudocounts; the Supporting information provides the formulas.

RESULTS

PDZ domains are protein–protein interaction domains that are involved in a wide variety of biological processes. One role of PDZ domains is assisting in the formation of protein complexes by binding to the C-termini of certain ligands.[34,35] Figure 3 shows a representative three dimensional structure of a PDZ domain interacting with a ligand. The structure of the complex reveals that the interaction is localized to a small area: the last four residues of the ligand interact directly with the beta strand βB, the alpha helix αB, and a carboxylate binding loop (CBL) between beta strands βA and βB.[37]

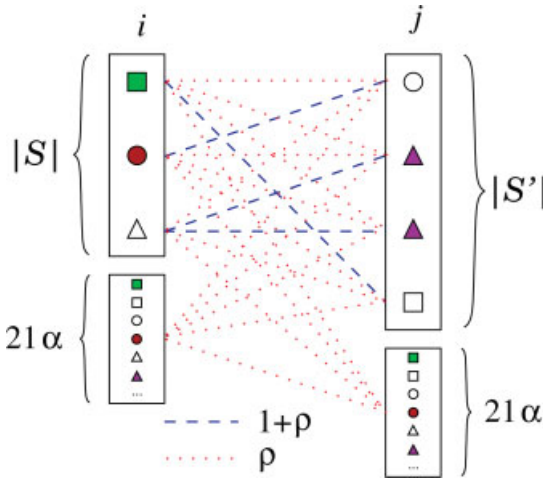

Figure 2

The structure of the two different types of pseudocounts used when scoring a protein–protein interaction under a GMRCC. Interaction pseudocounts, parameterized by ρ, allow for missing interactions in the interaction table. Residue pseudocounts, parameterized by α, allows for unobserved sequences in the MSAs. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

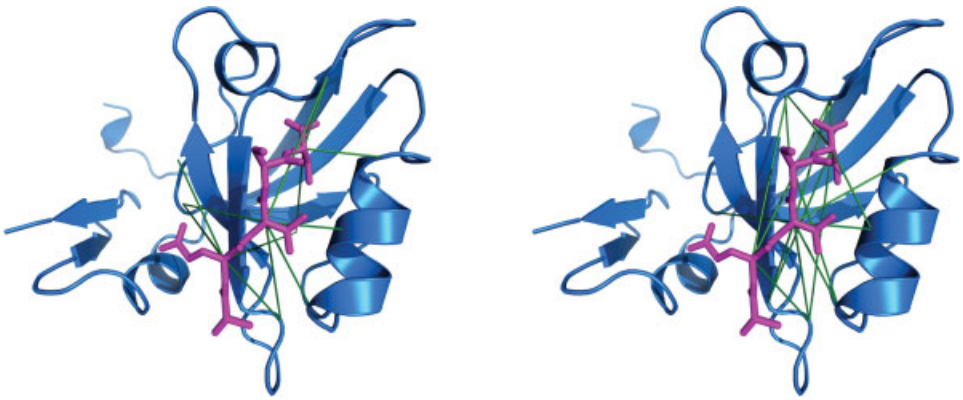

Figure 3

Cartoon representation of a PDZ domain (blue) interacting with a ligand (magenta), pdb id 1BE9, rendered via Pymol.[36] Four C-terminal residues of the ligand have direct interactions with PDZ residues in the beta strand βB, the alpha helix αB, and the loop between beta strands βA and βB. (Left) The first 15 edges (green) identified by table-count approach. (Right) The 17 edges (green) identified by the perturbation approach. Many of the edges involve the physically interacting residues.

To develop a PDZ/ligand GMRCC, we gathered a training set consisting of 167 PDZ domains and 230 ligands, along with 374 experimentally validated interactions derived from a literature search (Dr. Shireen Vali, IBAB, personal communication). We profile-aligned the PDZ domains to an existing PDZ domain alignment[29] using ClustalW,[30] yielding an alignment of 94 PDZ residues. For visualization and interpretation purposes, we matched these residues to the crystal structure of the third PDZ domain of synaptic protein PSD-95 in complex with a peptide derived from CRIPT (pdb id 1BE9[37]); the PDZ residues correspond to 82 residues in the range of P308 to Y392, and the 4 ligand residues to the range of Q6 to V9. Both the PDZ domains and ligands were filtered for uniqueness (arbitrarily eliminating one of each identical pair of sequences), yielding 147 unique PDZs and 164 unique ligands involved in 327 unique interactions. On average, each PDZ domain in the training set interacts with 2.2 ligands. The minimum number of interactions is 1 (69 PDZs) while the maximum is 16 (the PDZ with UniProt[38] id. P31016). Figure 4 shows the distribution of the PDZ interactions for our training set.

In addition to this training set, we established a test set of 169 experimentally validated PDZ/ligand interactions that are not included in the training set. These interactions (along with 63 present in our training set) are part of a dataset used in the development of the iSPOT tool.[28] Since the list of PDZs in our training set is quite extensive, the PDZs in the testing set necessarily occur in the training set. Their ligands, however, are quite different. For an interacting PDZ-ligand pair in the testing set, the closest interacting ligand for that PDZ in the training set is identical in on average only 1.2 (out of 4) positions.

We ran our GMRCC learning algorithms on our training set, using 100,000 randomizations to compute the *P*-value, and choosing edges with a *P*-value threshold of 0.005 or better. This *P*-value choice could readily be adjusted as desired to account for multiple hypothesis testing. For example, employing a simple Bonferroni correction for an uncorrected *P*-value of 0.05, when considering 376 possible edges, would result in a corrected threshold of $\sim 10^{-5}$. Most of the edges identified by our algorithms meet even this far more stringent threshold.

We employed the uninformative prior, considering all 376 possible edges rather than restricting them to contacting residue pairs. The algorithm selected 51 and 17

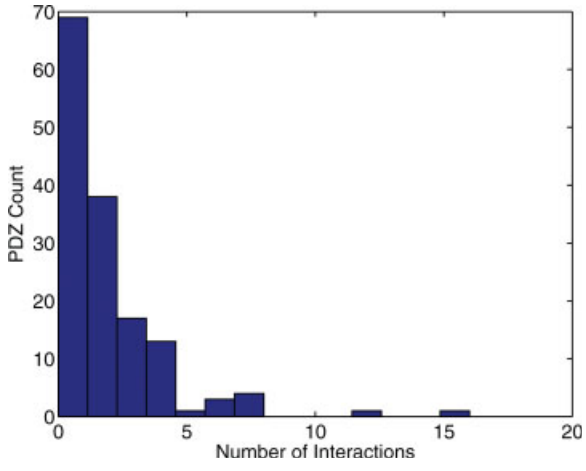

Figure 4

Histogram for the number of ligands with which each PDZ domain interacts in our training set. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Table I
Cross-Coupling Edges Identified by Our Algorithm

PDZ residue[a]		Ligand residue[a]		C^{α} dist[b,c]	P-*value*
By table count[d]					
372	αB1	7	−2	8.2	$< 1 \times 10^{-24}$
n/a		6	−3	n/a	4.66×10^{-15}
n/a		9	0	n/a	1.14×10^{-4}
322	CBL	7	−2	11.5	1.19×10^{-10}
330		6	−3	9.8	6.77×10^{-6}
380	αB9	7	−2	10.3	1.33×10^{-9}
339	βC4	6	−3	8.3	2.43×10^{-13}
339	βC4	7	−2	9.5	2.49×10^{-9}
330		7	−2	11.0	2.55×10^{-13}
322	CBL	9	0	5.5	3.85×10^{-10}
376	αB5	7	−2	6.8	4.05×10^{-12}
340	βC5	6	−3	9.9	9.97×10^{-12}
360	βD4	6	−3	17.2	5.26×10^{-10}
360	βD4	9	0	15.6	$< 1 \times 10^{-24}$
n/a		6	−3	n/a	2.12×10^{-12}
By perturbation[e]					
372	αB1	7	−2	8.2	$< 1 \times 10^{-24}$
376	αB5	7	−2	6.8	4.05×10^{-12}
372	αB1	8	−1	11.3	9.86×10^{-4}
356		7	−2	15.0	2.35×10^{-3}
347	αA2	7	−2	14.8	7.02×10^{-5}
347	αA2	9	0	10.0	1.41×10^{-8}
324	CBL	6	−3	13.2	1.35×10^{-4}
330		7	−2	11.0	2.55×10^{-13}
330		6	−3	9.8	6.77×10^{-6}
330		9	0	14.5	1.28×10^{-7}
380	αB9	7	−2	10.3	9.91×10^{-10}
323	CBL	7	−2	11.5	1.11×10^{-16}
323	CBL	9	0	5.5	$< 1 \times 10^{-24}$
323	CBL	6	−3	14.5	4.67×10^{-7}
376	αB5	9	0	6.7	8.63×10^{-4}
329	βB5	7	−2	9.7	1.55×10^{-15}
329	βB5	9	0	13.8	6.83×10^{-8}

[a]Residues are numbered according to their position in the appropriate chain of the 1BE9 crystal structure (with "n/a" where that sequence has gaps in the MSA) and under the numbering system used by Songyang *et al.*[39]
[b]Distances, in Å, are according to the 1BE9 crystal structure, for cases where both residues are represented.
[c]Rows in dark gray are known to have interacting side chains while those in light gray are likely to. Unhighlighted rows are not known to have direct interactions but are within a single residue of those that do.
[d]First 15 of 51 by table-count.
[e]All 17 by perturbation.

edges for the table-count and perturbation methods, respectively. The Supporting information includes these edge lists, indexed to the columns in the MSAs.

The following sections test these models in a number of different ways. First, we examine their edges, and show that many edges are supported by the literature or by structural studies. Second, we demonstrate that the models are able to predict interaction among the experimentally tested PDZ/ligand pairs. Third, we study the effect of data sparsity on the predictive ability of the models, and show that they maintain good predictive ability with even sparser data. Finally, we study the effects of the assumptions underlying the SPOT approach to predicting interaction, and show that the graphical model approach outperforms SPOT, and why.

Identified constraints

Figure 3 graphically illustrates the cross-coupling edges identified by our algorithm for both table-count (left) and perturbation (right). For clarity, we only show the first 15 edges learned by the table-count approach. The figures show that many of the edges occur between a ligand residue and a residue in the PDZ domain near the interaction site, even though we did not use structural information to learn the model. Table I lists the edges identified by our algorithm, as well as the $C^{\alpha}-C^{\alpha}$ distances of the residues in the 1BE9 structure. For the first fifteen edges identified by the table-count approach, all but two of the edges that can be mapped to the 1BE9 structure are within 12 Å. This is true also for the perturbation approach, where eleven of the seventeen identified cross-coupling constraints connect residues with C^{α} distance less than 12 Å. We further compared the distances between all residue pairs vs. those for identified edges (Fig. 5). The mean distance between all residue pairs is 14.5 Å, while that for pairs in the 51 table-count edges is 11.6 Å and that between pairs in the 17 perturbation edges is 11.0 Å. The differences between the all-pairs distribution and each edge distribution are statistically significant, by *t*-test ($P = 10^{-5}$ for all-pairs vs. either table-count or perturbation).

One cannot directly compare graphical models by looking at their edge lists, since factorization of a joint probability is not unique, and thus edge lists that differ could actually represent exactly the same probabilistic model. Nonetheless, we note that all but four edges (the 4th, 5th, 10th, and 14th) that are included in the pertur-

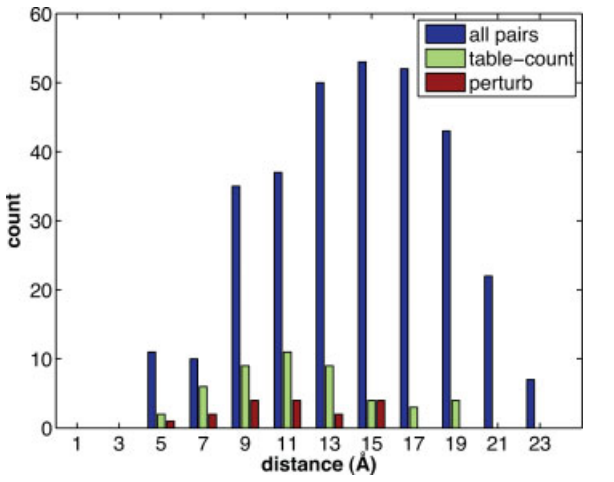

Figure 5
Distributions of $C^{\alpha}-C^{\alpha}$ distances, with respect to the 1BE9 crystal structure, of all PDZ-ligand pairs (blue), and those participating in edges in the table-count (green) and perturbation (red) models. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

bation model are also included in the table-count model. Thus the models are making use of many of the same residue relationships, although arriving at the choices from different perspectives.

Many of the edges identified by our models involve pairs of residues that have been previously determined to play important roles. Several of the edges occur between residues whose side chains are predicted to directly interact during binding.[39] For example, both of our models identify a cross-coupling relationship between αB1 and ligand residue −2, which is predicted to have a side chain interaction. Also, αB5 is predicted to have side chain interactions with both 0 and −2 but not −1 or −3. Both our models identify both of those predicted interactions (the table-count approach adds it as its 25th edge) and neither identifies either of the noninteracting residue pairs. In total, 4 of the first 15 edges identified by the table-count approach and 3 of the 17 edges identified by the perturbation approach are predicted to have interacting side chains (dark gray in Table I). If we allow interactions to include adjacent side chains (accounting for modest flexibility in complex formation), an additional two edges in both models are predicted to have side chain interactions (light gray in Table I). Another cross-coupling edge identified by our algorithms is between a residue in CBL in the PDZ domain and the last residue of the ligand (position 0), which form hydrogen bonds during peptide binding in the crystal structure of PDZ-3[37] but are not predicted to have side chain interactions with the peptide in other cases.[39] In total, the table-count methods identifies six residue pairs that are in close physical proximity in the complex structure and have evidence of direct interaction during the binding process while the perturbation approach identifies five.

The biological significance of the remaining edges is unknown. Both models identify strong cross-coupling between residue 330 and the ligand. Residue 330 is the first residue after β strand B, which forms a portion of the binding pocket. Though it does not fill the binding pocket, this strong cross-coupling in both methods suggests that it does have a role in ligand binding. With a few exceptions, the remaining edges tend to be further in three-dimensional space than the edges known to be involved in ligand binding. Since there is very little conformational change between the bound and unbound structure of the PDZ domain,[37] it is unlikely any of these residues plays an indirect role in ligand binding. Nonetheless, the sequence record and interaction data shows a statistically significant level of cross-coupling involving these residues.

Predicting interaction

The cross-coupling constraints capture strong residue relationships mediating interaction, suggesting that the models may be useful in predicting PDZ/ligand interactions. To test the predictive ability of our model, we considered the interactions in our test set between PSD95-2 and six ligands. Recall that interactions in our test set were gathered separately and are not part of our training set. Using Eq. (7), we evaluated the likelihood of PSD95-2 interacting with these six ligands. We also evaluated its likelihood of interaction with the 164 unique ligands from our training set, eight of which are known interactions (i.e., are in the training set), while 156 are not (i.e., the training set includes interactions between the ligands and other PDZs). For pseudocounts, we employed an interaction pseudocount (ρ) of 0.001 and a residue pseudocount (α) of 0.01; we have found our results to be largely insensitive to these parameters (see the "Comparison to SPOT" section).

Figure 6 shows, for both the table-count and perturbation methods, the 25 ligands with the highest likelihood scores for predicted interaction with PSD95-2. Of the eight previously known PSD95-2 interactions in our training set (green squares), six occur in the top 25 for the table-count approach, while seven appear in the top 25 for the perturbation approach. The remaining training ligands are ranked 33rd and 46th in the table-count approach and 28th in the perturbation approach. Three of the six interactions from the test set score in the top 15 of the interactions (red triangles) in the table-count. The remaining are ranked 26th (LTDV), 29th (FTDV), and 64th (QSLV), respectively. For the perturbation approach, three of the six are again ranked in the top 15 while the others are ranked 33rd (FTDV), 63rd (LTDV), and 109th (QSLV), respectively. Note that FTDV scores highly under both methods (although not in the top 25) while LTDV scores highly under the table-count approach but not as highly in the perturbation approach. Finally, QSLV scores poorly under both methods, suggesting that some classes of interactions are simply not represented in our training set and therefore are not predicted well under our models.

Some of the predicted interactions have not yet been experimentally tested, but serve as interesting hypotheses proposed by our methods. For instance, four untested ligands (ETHV, ETLV, ETPV, and ETQV) rank near the top of interactions against PSD95-2 for both methods. In addition, several other ligands (ESLV, ESYV, and ESKV) have higher likelihoods for both methods than do ligands from our training set. Finally, some predicted interactions are scored quite differently under the two methods (e.g., NTVV is the highest ranking under table-count, but 27th under perturbation), and would help provide insight into the relative quality of the models.

The choice of 25 ligands is somewhat arbitrary; we are effectively selecting a threshold and predicting that ligands scoring above that threshold interact, while those below do not. In the case of Figure 6, we are essentially using a threshold of −67 for the table-count approach

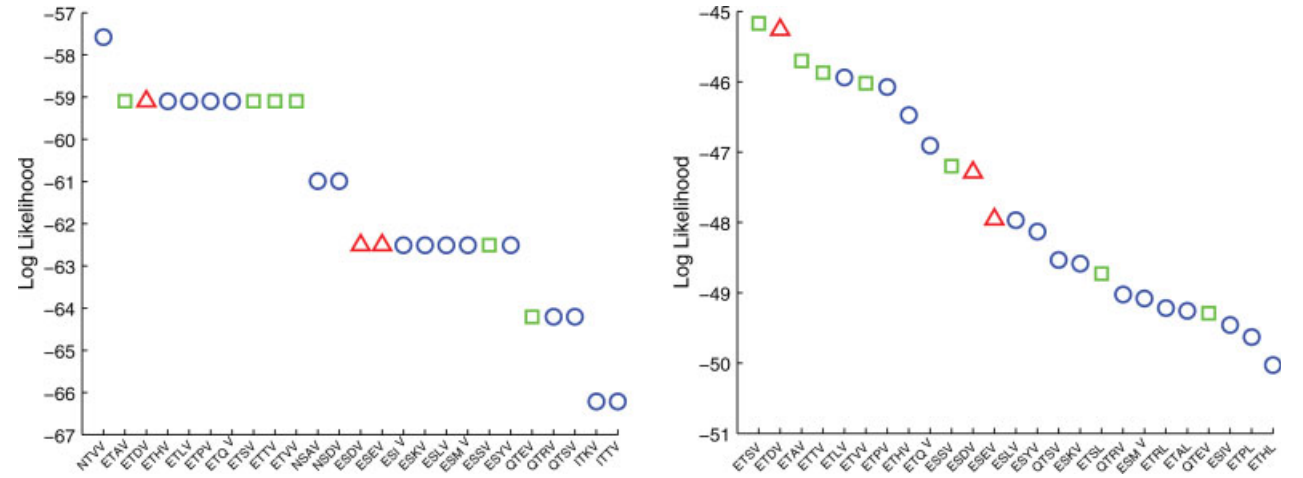

Figure 6

The 25 ligands with the highest likelihood of interacting with PSD95-2 according to the table-count (left) and perturbation (right) approaches. *x*-axis: ligand sequence; *y*-axis: log likelihood score. Green squares are interactions in the training set, red triangles are known interactions from the test set, and blue circles are ligands that have not (yet) been observed to interact with PSD95-2. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

and −51 for the perturbation approach. However, these thresholds also have an effect on the predicted interactions of other PDZ/ligand pairs. To study the effect of the threshold, we scored all interacting pairs in the training set, all interacting pairs in the test set, and all possible pairs from the training set (147 PDZs × 164 ligands). Figure 7 shows the fraction of these interactions scoring higher than each possible threshold. For instance, using the table-count approach, by selecting a threshold of −92, our model predicts all the interacting pairs in the training set and 61% of those in the test set. At this threshold, it predicts interaction between 24% of all possible pairs. This is in contrast to the perturbation approach, which requires a threshold of −87 to predict all the training interactions, while also predicting 94% of the test interactions. At this threshold, 92% of all possible pairs are predicted to interact. Lower thresholds yield fewer predicted interactions overall, but also miss more

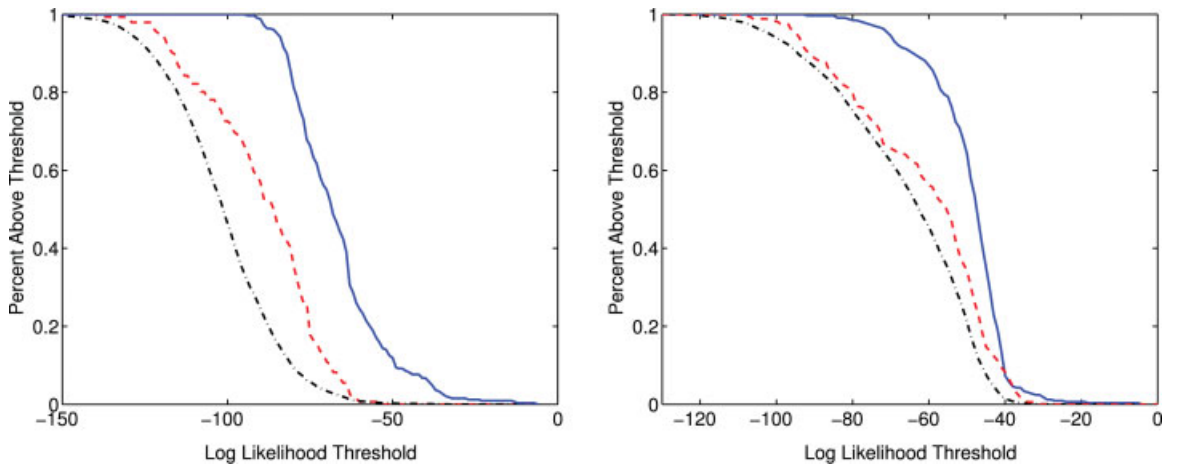

Figure 7

The fraction of interactions which score above a given log likelihood threshold according to the table-count (left) and perturbation (right) approaches. *x*-axis: log likelihood threshold; *y*-axis: fraction of interactions scoring above that threshold. The blue line is for interactions from our training set, the red dashed lines is for those in our test set, and the black dot-dashed line is for all possible PDZ/ligand interactions. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

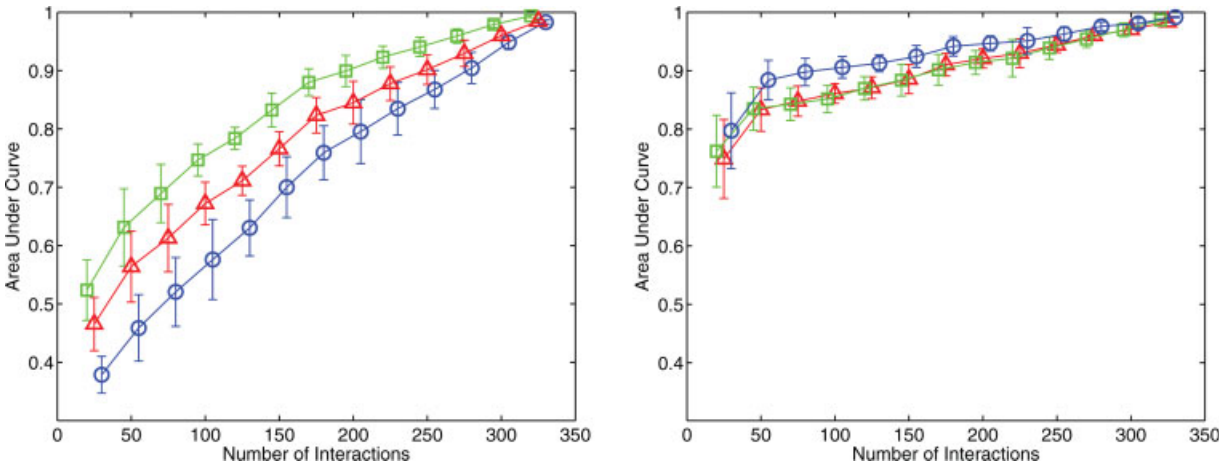

Figure 8

Evaluation of interaction predictions under increasing (to the left) sparsity in the interaction table for the table-count (left) and perturbation (right) methods. The *x*-axis indicates the number of interactions in the table; the *y*-axis indicates the area under the ROC curve, or AUC, which assesses the overall predictive ability of the resulting model. Blue circles corresponds to a 70% complete model, while red triangles and green squares correspond 30% and 10% complete, respectively. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

known interactions; higher thresholds result in many of the possible pairs being predicted to interact.

Effect of interaction table sparsity

Our approach explicitly incorporates interaction data in the interaction table $\mathcal{T}$ and assumptions about its completeness in the choice of table-count vs. perturbation-based probability models. To assess the effects of the relative incompleteness, or sparsity, of the observed interaction table, we conducted a simulation study. The basic idea is to learn a model using only a random portion of the interactions, and compare interaction predictions from the sparse-data model with those from the original "complete" model.

A key question is how to compare interaction predictions, since we do not know the ground truth for all 24108 PDZ/ligand pairs that could be scored. We employed a classification-based approach, in which a score threshold under the "complete" model establishes whether a PDZ/ligand pair is classified as interacting or noninteracting. We tried three different thresholds for the complete model, such that it would classify as interacting 10, 30, or 70% of the PDZ/ligand pairs. We treated these classifications as the truth, and evaluated how well the classifications by the sparse models agreed. In this classification setting, we employed ROC curves to evaluate the quality of the predictions under the sparse models. To generate an ROC curve, we varied the score threshold that separates the interacting class from the noninteracting class. For each such threshold, we measured the true positive rate (the fraction of "true" interactions, under the complete model, that the sparse model correctly classified as interacting according to the threshold) and the false positive rate (the fraction of noninteractions that it incorrectly classified as interacting). An ROC curve plots the true positive rate against the false positive rate, over the choices of thresholds. We then evaluated the predictive ability of each approach by the area under the ROC curve, or AUC; an AUC of 1 is perfect, while random guessing would have an AUC of 0.5.

In order to generate a sparse model, we randomly selected 25 starting interactions. After learning and evaluating both the table-count and perturbation models with these edges, we randomly added 25 of the remaining known interactions. We repeated this process until all the known interactions are included. At each level of sparsity we compared the sparse models to the complete model. We repeated this experiment 10 times, using a different random 25 starting interactions each time. For each level of sparsity we report the mean and standard deviations of these 10 simulations.

The results (Fig. 8) clearly demonstrate the impact of sparsity on both methods for the three different thresholds. As we would expect, the more data provided, the greater the ability of the models to predict the "true" interactions. However, the effect of sparsity on both methods is very different. In general, the table-count approach does very poorly with sparse data (almost random guessing with a few interactions provided) and gradually increases as more information is added. The perturbation approach, on the other hand, starts much better and rapidly improves with only a little more interaction data. After quickly improving to a high AUC

(≈ 0.9), it then gradually improves as more interaction data are added. Since each interaction in the table-count approach is given equal weight, when fewer interactions are provided for training, the method can deviate from the underlying edge scores and produce a poorer predictor. The perturbation approach, on the other hand, is able to quickly learn from sparse data because the more redundant interactions are not counted as much. Thus, for sparser interaction data, the edge weights are closer to the true model, allowing the perturbation approach to outperform the table-count approach.

Another interesting observation from the simulation study is that the threshold for the complete model (defining the "true" interactions) has a different effect on the two methods. In the table-count approach, the more "true" interactions, the worse the predictions are. One possible explanation is that the few "true" interactions do interact strongly, and as such, always look better than the other possible interactions, regardless of the level of sparsity. As the threshold is increased, however, more noninteractions are included in the "true" interactions and are thus harder to predict with sparse data. The exact opposite is observed with the perturbation approach—the more "true" interactions, the better the predictions are. This suggests that unlike the table-count approach, the perturbation approach may score many interactions highly. When only a few of these high scoring sequences are included in the "true" interactions, the sparse models have a harder time predicting which were included and which were not. As more of these interactions are included, the predictive ability of the sparse models improves.

Comparison to SPOT

SPOT,[27] like our method, uses cross-coupling to predict PDZ/ligand interactions, based on the consistency of residue pairs in a test sequence pair with residue pairs in the training sequence pairs. In our terminology, SPOT essentially employs a structural prior but uses all contacting residue pairs as edges, and scores according to a table-count like score. That is, the SPOT score for one amino acid pair (edge) is determined by the number of training interactions that have that amino acid pair. While using a structural prior provides for a mechanistic explanation of the observed cross-coupling, it requires solved structures and assumes that the structures are sufficiently representative of all interacting pairs. Furthermore, as we showed above, our method discovers many cross-coupled pairs that are close in space across the interface, even without this prior restriction. The other significant difference between our method and SPOT is that, as discussed in the "Methods" Section, we provide a probabilistic semantics for evaluating likelihood, avoiding "double counting" of edges that are themselves dependent. Thus we don't use all edges, but instead factorize them to eliminate redundant information.

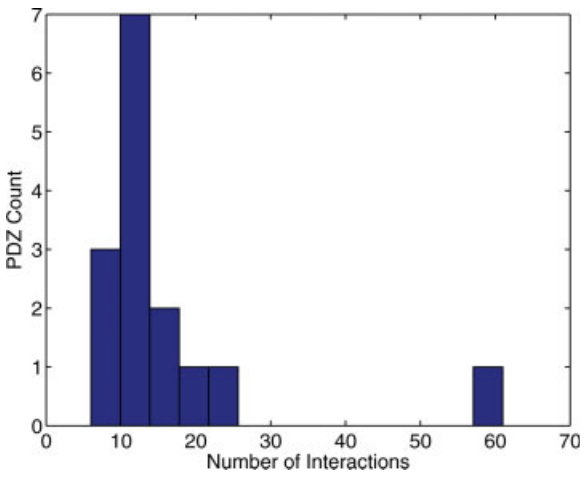

Figure 9

Distribution of PDZ domain interactions in the SPOT training set. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

To assess the impact of the special case assumptions of SPOT (structural prior, all edges, and table-count), we trained our methods with the same dataset used to train SPOT.[27] The SPOT training set contains 15 PDZ domains, multiply aligned to 94 residues, as well as 229 ligands, each consisting of 4 residues. Based on three different PDZ/ligand structures (pdb ids. 1QAV, 1KWA, and 1BE9), SPOT restricts cross-coupling statistics to 43 contacting residue pairs. Each PDZ domain in the SPOT dataset interacts with 15.5 ligands on average (232 interactions total); the minimum number of ligand interactions is 6 (psd95-2, MAGI-2) and the maximum number is 61 (nNos). Figure 9 provides a histogram for the complete distribution. Comparing to our training set (see Fig. 4), the SPOT training set contains many fewer PDZ domains sequences than ours does and each PDZ domain is involved in many more interactions. Further, with the exception of a single PDZ domain, most PDZ domains in the SPOT dataset interact with approximately the same number of ligands.

For a test set, we use the results of a solid phase PDZ/ligand immunoassay, created to improve SPOT statistics[29] but not included in the training set. In this experiment, 14 ligands were screened against 7 PDZ domains. Of the 98 possible interactions, 27 were experimentally determined to interact while the other 71 were determined not to interact.

For our models, we ran our GMRCC algorithm on the new training set, again employing 100,000 randomizations to compute the *P*-value and only choosing edges with a *P*-value of .005 or better. We generated a model with a structural prior, considering the 43 edges used by SPOT, as well as a model with an uninformative prior, considering all 376 (94 × 4) possible edges.

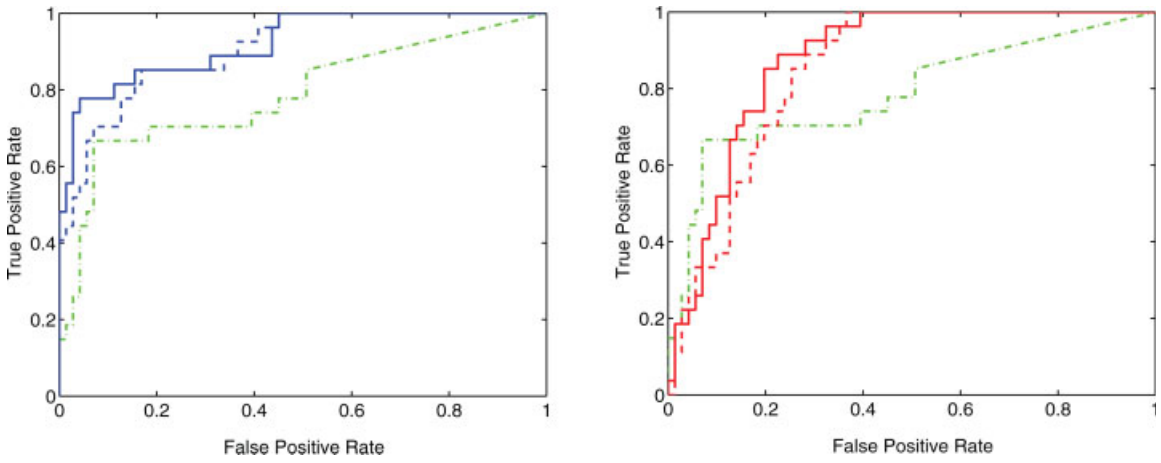

Figure 10

ROC curves for table-count (blue, left), perturbation (red, right), and SPOT (green dot-dashed, both). The table-count and perturbation methods outperform SPOT with either a structural (dotted) or uninformative prior (dashed). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

We scored all test sequence pairs under both our models and SPOT (using the iSPOT webserver[28]). For our models, we applied Eq. (7) for each method (table-count and perturbation) and prior (structural and uninformative), using as above an interaction pseudocount (ρ) of .001 and a residue pseudocount (α) of .01.

Figure 10 shows the ROC curves for the different approaches. The AUC values for the variations of our approach are .92 for table-count/uninformative, .91 for table-count/structural, .88 for perturbation/uninformative, .85 for perturbation/structural. Since iSPOT only returns scores for interactions scoring more than .5 (interactions scoring lower than .5 are assumed not to interact), we were unable to score 4 true interactions and 35 false interactions. If we evaluated SPOT using only those 59 interactions whose scores were reported, the AUC value would be .77. We can bound the value for the complete set by noting that in the best case, the 4 true interactions would score the highest among those scoring under .5. If this were the case, the value would be at most .83. It could be worse, as low as .75, if the 4 true interactions scored lowest. Thus, even when employing a structural prior, our methods outperform SPOT in predictive ability. By using only informative cross-coupled pairs, scoring them in a probabilistic setting, and factorizing them appropriately, our models do not overfit our the data and are able to better predict interactions. This test shows that the structural restriction does reduce the useful information, slightly weakening the predictive ability.

To further illustrate the predictive ability of the models, Figure 11 shows the precision-recall (PR) curves for the different approaches. Each point in the curve corresponds to a score threshold where interactions with scores above the threshold are predicted to interact while those with scores below it are predicted not to interact. The precision at a threshold is given by the fraction of true interactions above the threshold divided by the total number of interactions above it. The recall is the fraction of true interactions above the threshold. The PR curve for a perfect classifier would go through the point (1, 1). Notice that for all levels of recall, the precision of the table-count approach outperforms the SPOT approach. The perturbation approach outperforms the SPOT approach for nearly all values of recall. The performance can be characterized by the maximum F-score (the harmonic mean of precision and recall) along the curve. The maximum F-scores are .8708, .8670, and .6352 for the table-count, perturbation, and SPOT approaches, respectively.

In this study, the table-count approach slightly outperforms the perturbation approach. There are two key factors at play here. First, the distribution of the interactions is nearly uniform (see Fig. 9), so that all sequences are equally represented in the interaction table, without the kind of bias alleviated by the perturbation method. Furthermore, recall that the perturbation approach deals with unique sequences rather than all interactions; since there are only 15 PDZs, it faces the problem of generalizing from a very small training set.

To test the robustness of our approach to the selection of pseudocount parameters, we scored the test sequence pairs with a range of different settings. Figure 12 shows the AUC values for different settings of the parameters. In general, smaller pseudocounts tend to yield better values for both methods. However, even over a broad range of magnitudes the values are still very high and outperform SPOT. Only when the pseudocounts overwhelm the data (for example, setting $\rho = .5$, so that each unobserved interaction is worth half an observed interaction)

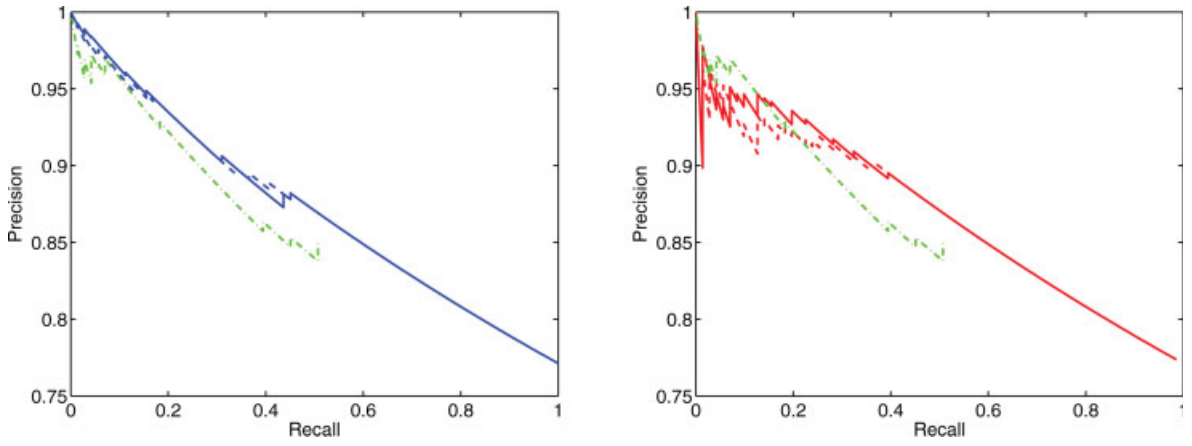

Figure 11

Precision-recall curves for table-count (blue, left), perturbation (red, right), and SPOT (green dot-dashed, both). The table-count and perturbation methods outperform SPOT with either a structural (dotted) or uninformative prior (dashed). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

do our methods have lower AUC values than SPOT. Finally, we note that the residue pseudocounts appear to have relatively little impact as compared to the interaction pseudocounts. This suggests that the predictions are determined more by the observed interactions than by the observed amino acids.

High-throughput experimental interaction data

Recently, a high-throughput experimental study was published in which 96 human and 72 *C. elegans* PDZ domains were tested for binding against over 10 billion random peptides in a phage-displayed combinatorial library.[25] This resulted in the identification of about 10,000 interactions, between 3100 peptide ligands and 82 PDZs. We used a profile hidden Markov model method provided in the Matlab bioinformatics toolbox to align the PDZs against our training set. Of the 82 PDZs in this testing set, none occur in our training set. Since our models consider only the four C-terminal residues of the ligands, we filtered the 3,100 peptide ligands down to 1044 unique tetra-peptide ligands, 34 of which occur in our training set.

We scored each of the possible 85,608 interactions (82 PDZs * 1044 ligands) under both table-count and pertur-

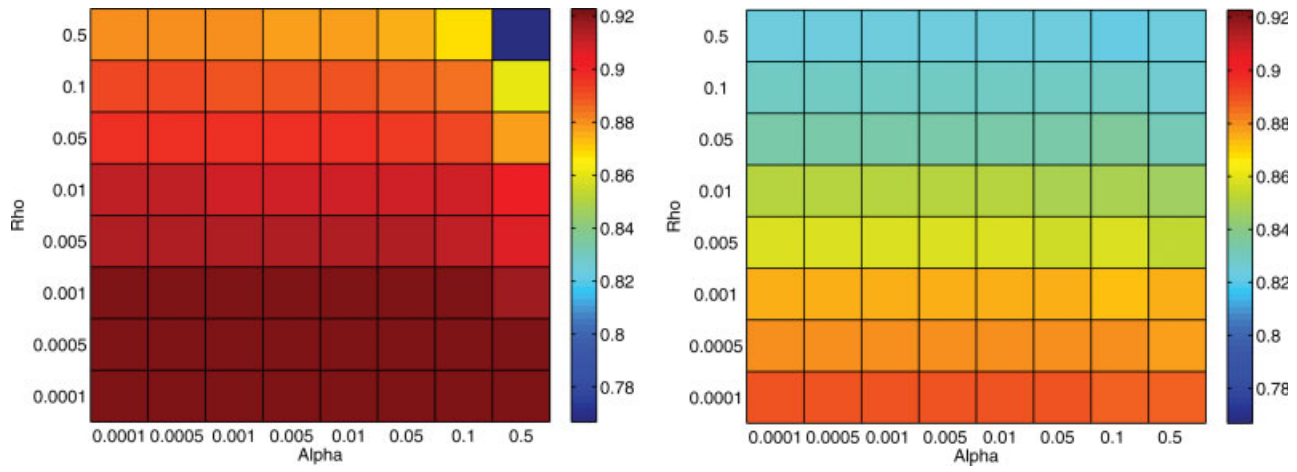

Figure 12

Robustness of the models under different choices of pseudocounts for table-count (left) and perturbation (right). The *x*-axis shows the amino acid pseudocount, α, while the *y*-axis shows the interaction pseudocount, ρ. Colors indicate AUC values. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

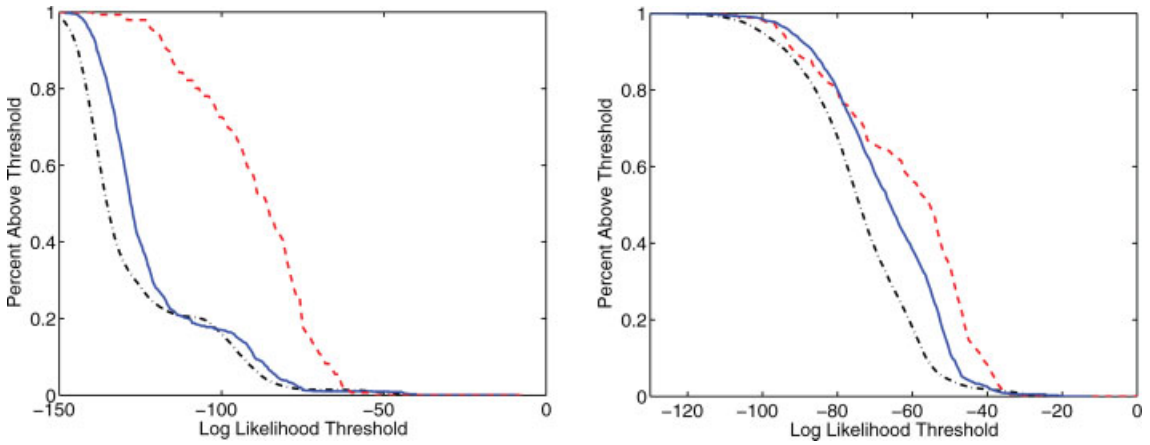

Figure 13

The fraction of interactions scoring above a given log likelihood threshold according to the table-count (left) and perturbation (right) approaches. *x*-axis: log likelihood threshold; *y*-axis: fraction of interactions scoring above that threshold. The blue lines are for the high-throughput interactions experimentally found to interact. The dashed red lines are for those in our testing set (see Fig. 7), while the black dot-dashed lines are for the "unidentified" interactions from the high-throughput experiments. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

bation methods with our original model (i.e., learned from our training data). Of the 85,608 interactions, 1476 were experimentally identified to interact, while 84,132 "unidentified" ones were not found in the phage display library (but are not necessarily non-interacting). We expect all of the identified interactions to have good scores overall, but only some of the unidentified ones to score well (and possibly to interact). Thus when we select a score threshold and predict that those above the threshold interact and those below the threshold do not interact, we expect the above-threshold set to include more of the identified interactions. Figure 13 shows the fraction of sequences from the original testing set (dashed red line in Fig. 7), along with the experimentally identified (solid blue) and unidentified (dot-dashed black) interactions from the high-throughput experiment for both the table-count (left) and perturbation (right) approaches. It is the case that the identified interactions are better represented among those above threshold, for most threshold. Note that in the case of perturbation, the threshold from the earlier training set can be used to achieve a similar fraction for the identified interactions from the high-throughput experiment. This is not true for table-count, where the absence of similar training data causes the predictive ability to drop significantly.

CONCLUSION

Our overarching goal is to construct formal probabilistic models capturing evolutionary coupling in protein families, in order to better support protein investigation, characterization, and design. Such models make explicit the essential constraints underlying a family, and provide compact descriptions of joint amino acid distributions. They generalize traditional motif representations and enable transparent probabilistic reasoning. We have previously developed the basic graphical model approach for a single family and additional techniques that make use of functional class information.[12] The present article builds upon that work in order to analyze and utilize information about co-evolving families of proteins. Our approach proved effective in uncovering, describing, and predicting coupling in PDZ-ligand interactions, and we intend to apply it to other such families, as well as to help others in doing so, by making our software freely available for academic use.

In addition to applications, there are a number of interesting directions for future development. In order to more fully characterize conservation and coupling information within protein families, we will integrate within- and between-family coupling models within a single framework. We will seek to refine the model by incorporating quantitative interaction data (e.g., free energies of association, rather than simple binary indicators). Integration of quantitative data may also enable us to predict free energies of association (generalizing the work mentioned in the introduction, e.g., Ref. 40). Finally, sampling from a cross-coupling model can guide the design of new partners for a given protein or even new pairs of interacting proteins from modeled families, just as coupling information has been demonstrated to enable the design of new, stably folded[1] and functional[2] WW domains. A particularly interesting direction is to design

for specificity (as has been accomplished by other techniques for a number of systems[40–45]), leveraging the demonstrated ability of our model to capture the cross-coupling information underlying specific recognition.

ACKNOWLEDGMENTS

This work was inspired by conversations with Dr. Alan Friedman (Purdue) and Dr. Shireen Vali (IBAB, Bangalore, India). The authors thank Dr. Vali for suggesting the study involving PDZ-ligand binding and helping organize one of the datasets used here. They also thank the anonymous reviewers whose helpful comments led to significant improvements in the article.

REFERENCES

1. Socolich M, Lockless SW, Russ WP, Lee H, Gardner KH, Ranganathan R. Evolutionary information for specifying a protein fold. Nature 2005;437:512–518.
2. Russ WP, Lowery DM, Mishra P, Yaffee MB, Ranganathan R. Natural-like function in artificial WW domains. Nature 2005;437:579–583.
3. Bloom JD, Labthavikul ST, Otey CR, Arnold FH. Protein stability promotes evolvability. Proc Natl Acad Sci 2006;103:5869–5874.
4. Counago R, Chen S, Shamoo Y. In vivo molecular evolution reveals biophysical origins of organismal fitness. Mol Cell 2006;22:441–449.
5. Madaoui H, Guerois R. Coevolution at protein complex interfaces can be detected by the complementarity trace with important impact for predictive docking. Proc Nat Acad Sci 2008;105:7708–7713.
6. Mintseris J, Weng Z. Structure, function, and evolution of transient and obligate protein–protein interactions. Proc Natl Acad Sci 2005;102:10930–10935.
7. Korber BTM, Farber RM, Wolpert DH, Lapedes AS. Covariation of mutations in the V3 loop of HIV Type 1 envelope protein: an information theoretic analysis. Proc Nat Acad Sci 1993;90:7176–7180.
8. Lockless SW, Ranganathan R. Evolutionarily conserved pathways of energetic connectivity in protein families. Science 1999;286:295–299.
9. Olmea O, Rost B, Valencia A. Effective use of sequence correlation and conservation in fold recognition. J Mol Biol 1999;293:1221–1239.
10. Fodor AA, Aldrich RW. Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. Proteins 2004;56:211–221.
11. Ye X, Friedman AM, Bailey-Kellogg C. Hypergraph model of multi-residue interactions in proteins: sequentially-constrained partitioning algorithms for optimization of site-directed protein recombination. J Comput Biol 2007;14:777–790.
12. Thomas J, Ramakrishnan N, Bailey-Kellogg C. Graphical models of residue coupling in protein families. IEEE Trans Comput Biol Bioinformat 2008;5:183–197.
13. Pazos F, Helmer-Citterich M, Ausiello G, Valencia A. Correlated mutations contain information about protein–protein interaction. J Mol Biol 1997;271:511–523.
14. Halperin I, Wolfson H, Nussinov R. Correlated mutations: advances and limitations. A study on fusion proteins and on the cohesin-dockerin families. Proteins 2006;63:832–845.
15. Tillier ERM, Biro L, Li G, Tillo D. Codep: maximizing co-evolutionary interdependencies to discover interacting proteins. Proteins 2006;63:822–831.
16. Pollock DD, Taylor WR, Goldman N. Coevolving protein residues: Maximum likelihood identification and relationship to structure. J Mol Biol 1999;287:187–198.
17. Yeang CH, Haussler D. Detecting coevolution in and among protein domains. PLoS Comput Biol 2007;3:e211.
18. Lichtarge O, Bourne HR, Cohen FE. An evolutionary trace method defines binding surfaces common to protein families. J Mol Biol 1996;257:342–358.
19. Armon A, Graur D, Ben-Tal N. ConSurf: an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. J Mol Biol 2001;307:447–463.
20. La D, Sutch B, Livesay DR. Predicting protein functional sites with phylogenetic motifs. Proteins 2005;58:309–320.
21. Hulo N, Bairoch A, Bulliard V, Cerutti L, Cuche BA, de Castro E, Lachaize C, Langendijk-Genevaux PS, Sigrist CJA. The 20 years of PROSITE. Nucleic Acids Res 2008;36:D245–D249.
22. Wang H, Segal E, Ben-Hur A, Li Q-R, Vidal M, Koller D. InSite: a computational method for identifying protein–protein interaction binding sites on a proteome-wide scale. Genome Biol 2007;8:R192.
23. Guo J, Wu X, Zhang D-Y, Lin K. Genome-wide inference of protein interaction sites: lessons from the yeast high-quality negative protein–protein interaction dataset. Nucleic Acids Res 2008;36:2002–2011.
24. Pitre S, North C, Alamgir M, Jessulat M, Chan A, Luo X, Green JR, Dumontier M, Dehne F, Golshani A. Global investigation of protein–protein interactions in yeast *Saccharomyces cerevisiae* using re-occurring short polypeptide sequences. Nucleic Acids Res 2008;36:4286–4294.
25. Tonikian R, Zhang Y, Sazinsky SL, Currell B, Yeh JH, Reva B, Held HA, Appleton BA, Evangelista M, Wu Y, Xin X, Chan AC, Seshagiri S, Lasky LA, Sander C, Boone C, Bader GD, Sidhu SS. A specificity map for the PDZ domain family. PLoS Biol 2008;6:e239.
26. Beuming T, Skrabanek L, Niv MY, Mukherjee P, Weinstein H. PDZBase: a protein–protein interaction database for PDZ-domains. Bioinformatics 2005;21:827–828.
27. Brannetti B, Via A, Cestra G, Cesareni G, Citterich MH. SH3-SPOT: an algorithm to predict preferred ligands to different members of the SH3 gene family. J Mol Biol 2000;298:313–328.
28. Brannetti B, Helmer-Citterich M. iSPOT: a web tool to infer the interaction specificity of families of protein modules. Nucleic Acids Res 2003;31:3709–3711.
29. Vaccaro P, Brannetti B, Montecchi-Palazzi L, Philipp S, Helmer Citterich M, Cesareni G, Dente L. Distinct binding specificity of the multiple PDZ domains of INADL, a human protein with homology to INAD from *Drosophila melanogaster*. J Biol Chem 2001;276:42122–42130.
30. Chenna R, Sugawara H, Koike T, Lopez R, Gibson TJ, Higgins DG, Thompson JD. Multiple sequence alignment with the Clustal series of programs. Nucleic Acids Res 2003;31:3497–3500.
31. Friedman N, Nachman I, P'eer D. Learning bayesian network structure from massive datasets: the "sparse candidate" algorithm. In: Proceedings of the 15 International Conference on Uncertainty in Artificial Intelligence (UAI), 1999 pp 206–215.
32. Lauritzen SL. Graphical models. Oxford, UK: Oxford University Press; 1996.
33. Durbin R, Eddy SR, Krogh A, Mitchison G. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge, UK: Cambridge University Press; 1998.
34. Harris BZ, Lim WA. Mechanism and role of PDZ domains in signaling complex assembly. J Cell Sci 2001;114:3219–3231.
35. Hung AY, Sheng M. PDZ domains: Structural modules for protein complex assembly. J Biol Chem 2002;277:5699–5702.
36. Delano WL. The PyMOL molecular graphics system. Palo Alto, CA: DeLano Scientific; 2002.
37. Doyle DA, Lee A, Lewis J, Kim E, Sheng M, MacKinnon R. Crystal structures of a complexed and peptide-free membrane protein–

binding domain: molecular basis of peptide recognition by PDZ. Cell 1996;85:1067–1076.
38. The UniProt Consortium. The universal protein resource (UniProt). Nucleic Acids Res 2008;36:D190–D195.
39. Songyang Z, Fanning AS, Fu C, Xu J, Marfatia SM, Chishti AH, Crompton A, Chan AC, Anderson JM, Cantley LC. Recognition of unique carboxyl-terminal motifs by distinct PDZ domains. Science 1997;275:73–77.
40. Li J, Yi Z-P, Laskowski MC, Laskowski M, Jr, Bailey-Kellogg C. Analysis of sequence-reactivity space for protein–protein interactions. Proteins 2005;58:661–671.
41. Reina J, Lacroix E, Hobson SD, Fernandez-Ballester G, Rybin V, Schwab MS, Serrano L, Gonzalez C. Computer-aided design of a PDZ domain to recognise new target sequences. Nat Struct Mol Biol 2002;9:621–627.
42. Shifman JM, Mayo SL. Exploring the origins of binding specficity through the computational redesign of calmodulin. Proc Nat Acad Sci 2003;100:13274–13279.
43. Lilien RH, Stevens BW, Anderson AC, Donald BR. 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 adenlytaion enzyme. J Comput Biol 2005;12:740–761.
44. Kortemme T, Joachimiak LA, Bullock AN, Schuler AD, Stoddard BL, Baker D. Computational redesign of protein–protein interaction specificity. Nat Struct Mol Biol 2004;11:371–379.
45. Joachimiak LA, Kortemme T, Stoddard BL, Baker D. Computational design of a new hydrogen bond network and at least a 300-fold specificity switch at a protein–protein interface. J Mol Biol 2006;361:195–208.