Using Physicochemical Properties of Amino Acids to induce Graphical Models of Residue Couplings

K. S. M. Tozammel Hossain[†], Chris Bailey-Kellogg[‡], Alan M. Friedman[§],
Michael J. Bradley[*], Nathan Baker[**], and Naren Ramakrishnan[†]

[†]Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA
[‡]Department of Computer Science, Dartmouth College, Hanover, NH 03755, USA
[§]Department of Biological Sciences, Purdue University, West Lafayette, IN 47907, USA
[*] Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, CT 06520, USA
[**]Pacific Northwest National Laboratory, Richland, WA 99352, USA

tozammel@vt.edu, cbk@cs.dartmouth.edu, afried@purdue.edu,
michael.bradley@yale.edu, nathan.baker@pnl.gov, naren@cs.vt.edu

ABSTRACT

Residue coupling in protein families is an important indicator for structural and functional conservation. Two residues are coupled if changes of amino acid at one residue location are correlated with changes in the other. Many algorithmic techniques have been proposed to discover couplings in protein families. These approaches discover couplings over amino acid combinations but do not yield mechanistic or other explanations for such couplings. We propose to study couplings in terms of amino acid classes such as polarity, hydrophobicity, size, and reactivity, and present two algorithms for learning probabilistic graphical models of amino acid class-based residue couplings. Our probabilistic graphical models provide a sound basis for predictive, diagnostic, and abductive reasoning. Further, our methods can take optional structural priors into account for building graphical models. The resulting models are useful in assessing the likelihood of a new protein to be a member of a family and for designing new protein sequences by sampling from the graphical model. We apply our approaches to understand couplings in two protein families: Nickel-responsive transription factors (NikR) and G-protein coupled receptors (GPCRs). The results demonstrate that our graphcial models based on sequences, physicochemical properties, and protein structure are capable of detecting amino acid class-based couplings between important residues that play roles in activities of these two families.

Keywords

Residue coupling, graphical models, amino acid classes, evolutionary co-variation.

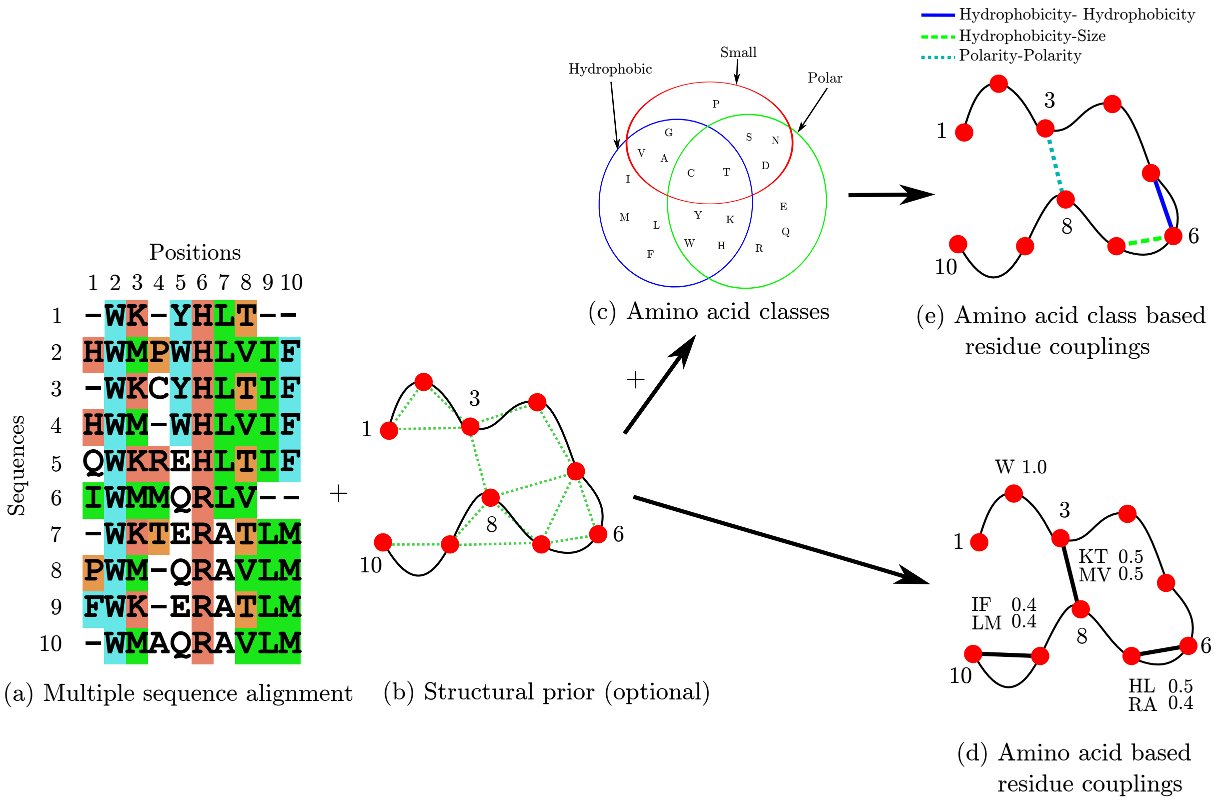
BIOKDD 2011 August 2011, San Diego, CA, USA
Copyright 2011 ACM 978-1-4503-0839-7 ...$10.00.

1. INTRODUCTION

Proteins are grouped into families based on similarity of function and structure. It is generally assumed that evolutionary pressures in protein families to maintain structure and function manifest in the underlying sequences. Two well-known types of constraints are conservation and coupling. The most widely studied constraint is conservation of individual residues. Within a protein family, a particular residue position is *conserved* if a particular amino acid occurs at that residue position for most of the members in the family [3]. Conservation of residues usually occurs at functionally and/or structurally important sites within a protein fold (shared by the protein family). For example in Figure 1(a), a multiple sequence alignment (MSA) of 10 sequences, the second residue is 100% conserved with occurrence of amino acid "W".

A variety of recent studies have used MSAs to calculate correlations in mutations at several positions within an alignment and between alignments [15, 10, 19, 14]. These correlations have been hypothesized to result from structural/functional coupling between these positions within the protein [8]. Two residues are *coupled* if certain amino acid combinations occur at these positions in the MSA more frequently than others [15, 7]. For example, residues 3 and 8 are coupled in Fig. 1(d) because the presence of "K" (or "M") at the third residue co-occurs with "T" (or "V") at the eighth residue position. Going beyond sequence conservation, couplings provide additional information about potentially important structural/functional connections between residues within a protein family. Previous studies [15, 8, 10] show that residue couplings play key roles in transducing signals in cellular systems.

In this paper, we study residue couplings that manifest at the level of amino acid classes rather than just the occurrence of particular letters within an MSA. Our underlying hypothesis is that if structural and functional behaviors are the underlying cause of residue couplings within MSAs, then couplings are more naturally studied at the level of amino acid properties. We are motivated by the prior work of Thomas et al. [9, 10] which proposes probabilistic graphical models for capturing couplings in a protein family in terms of amino acids. Graphical models are useful for support-



(a) Multiple sequence alignment (b) Structural prior (optional) (c) Amino acid classes (e) Amino acid class based residue couplings (d) Amino acid based residue couplings

Figure 1: Inferring graphical models from an MSA of a protein family: (a)-(c) illustrate input to our models and (d),(e) illustrate two different residue coupling networks.

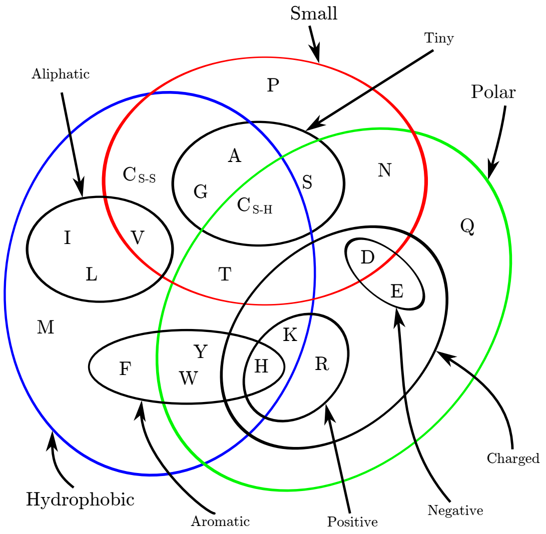


Figure 2: Taylor's classification: a Venn diagram depicting classes of amino acids based on physicochemical properties. Figure redrawn from [17].

ing better investigation, characterization, and design of proteins. The above works infer an undirected graphical model for couplings given an MSA where each node (variable) in the graph corresponds to a residue (column) in the MSA and an edge between two residues represents significant correlation between them. Figure 1(a),(b) illustrates the typical input (an MSA and a structural prior) and Figure 1(d) is an output (undirected graphical model) of the procedure of Thomas et al. In the output model (see Fig. 1(d)), three residue pairs—(3,8), (6,7), and (9,10)—are coupled.

Evolution is the key factor determining the functions and structures of proteins. It is assumed that the type of amino acid at each residue position within a protein structure is (at least somewhat) constrained by its surrounding residues. Therefore, explaining the couplings in terms of amino acid classes is desirable. To achieve this, we consider amino acid classes based on physicochemical properties (see Fig. 2).

Graphical models can be made more expressive if we represent the couplings (edges in the graphs) in terms of underlying physicochemical properties. Figure 1(c) is a Venn diagram of three amino acid classes–polarity, hydrophobicity, and size. Figure 1(e) illustrates three couplings in terms of amino acid classes. For example, residue 3 and residue 8 are coupled in term of "polarity-polarity", which means correlated changes of polarities occur at these two positions – a change from polar to nonpolar amino acids at residue 3, for instance, induces concomitant change from polar to nonpolar amino acid at residue 8. Similarly, residue 6 and residue 7 are also correlated since a change from hydrophobic to hydrophilic amino acids at residue 6 induces a change from big to small amino acids at residue 7. There is no edge between residue 5 and residue 7, however, because they are independent given residue 6. Hence, the coupling between residue 5 and residue 7 is explained via couplings (5,6) and (6,7). This is one of the key features of undirected graphical models as they help distinguish direct couplings from indirect couplings. Note that the coupling between residue 9 and residue 10 (originally present in Fig. 1(d)) does not occur in Figure 1(e) due to class conservation in residues 9 and 10. Also note that the coupling between residue 5 and residue 6 in Figure 1(e) is not apparent in Figure 1(d). Class-based representations of couplings hence recognize a different set of relationships than amino acid value-based couplings. We show how the class-based representation leads to more explainable models and suggest alternative criteria for protein design.

The key contributions of this paper are as follows:

1. We investigate whether residue couplings manifest at the level of amino acid classes and answer this question in the affirmative for the two protein families studied here.

2. We design new probabilistic graphical models for capturing residue coupling in terms of amino acid classes. Like the work of Thomas et al. [10] our models are precise and give explainable representations of couplings in a protein family. They can be used to assess the likelihood of a protein to be in a family and thus constitute the driver for protein design.

3. We demonstrate successful applications to the NikR and GPCR protein families, two key demonstrators for protein constraint modeling.

The rest of the paper is organized as follows. We review related literature in Section 2. Methodologies for inferring graphical models are described in Section 3. Experimental results are provided in Section 4 followed by a discussion in Section 5.

2. LITERATURE REVIEW

Early research on correlated amino acids was conducted by Lockless and Ranganathan [15]. Through statistical analysis they quantified correlated amino acid positions in a protein family from its MSA. Their work is based on two hypotheses, which are derived from empirical observation of sequence evolution. First, the distribution of amino acids at a position should approach their mean abundance in all proteins if there is a lack of evolutionary constraint at that position; deviance from mean values would, therefore, indicate evolutionary pressure to prefer particular amino acid(s). Second, if two positions are functionally coupled, then there should be mutually constrained evolution at the two positions even if they are distantly positioned in the protein structure. The authors developed two statistical parameters for conservation and coupling based on the above hypothesis, and use these parameters to discover conserved and correlated amino acid positions. In their SCA method, a residue position in an MSA of the family is set to its most frequent amino acid, and the distribution of amino acids at another position (with deviant sequence at the first position removed) is observed. If the observed distribution of amino acids at the other position is significantly different from the distribution in the original MSA, then these two positions are considered to be coupled. Application of their method on the PDZ protein family successfully determined correlated amino acids that form a protein-protein binding site.

Valdar surveyed different methods for scoring residue conservation [17]. Quantitative assessment of conservation is important because it sets a baseline for determining coupling. In particular, many algorithms for detecting correlated residues run into trouble when there is an 'in between' level of conservation at a residue position. In this survey, the author investigates about 20 conservation measures and evaluates their strengths and weaknesses.

Fodor and Aldrich reviewed four broad categories of measures for detecting correlation in amino acids [11]. These categories are: 1) Observed Minus Expected Squared Covariance Algorithm (OMES), 2) Mutual Information Covariance Algorithm (MI), 3) Statistical Coupling Analysis Covariance Algorithm (SCA; mentioned above), and 4) McLachlan Based Substitution Correlation (McBASC). They applied these four measures on synthetic as well as real datasets and reported a general lack of agreement among the measures. One of the reasons for the discrepancy is sensitivity to conservation among the methods, in particular, when they try to correlate residues of intermediate-level conservation. The sensitivity to conservation shows a clear trend with algorithms favoring the order McBASC > OMES > SCA > MI.

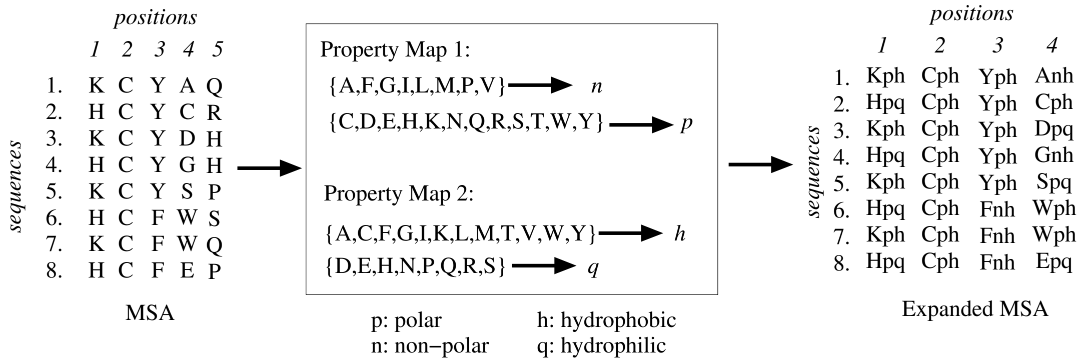
Although current research is successful in discovering conserved and correlated amino acids, they fail to give a formal probabilistic model. Thomas *et al.* [10] is a notable expection. This paper differentiates between direct and indirect correlations which previous methods did not. Moreover, the models discovered by this work can be extended into *differential* graphical models which can be applied to protein families with different functional classes and can be used to discover subfamily-specific constraints (conservation and coupling) as opposed to family-wide constraints.

The above research on coupling and conservation do not aim to model evolutionary processes directly. Yeang and Haussler, in contrast, suggest a new model of correlation in and across protein families employing evolution [19]. They refer to their model as a *coevolutionary model* and their key claims are: coevolving protein domains are functionally coupled, coevolving positions are spatially coupled, and coevolving positions are at functionally important sites. The authors give a probabilistic formulation for the model employing a phylogenetic tree for detecting correlated residues.

A more recent work, by Little and Chen [14], studies correlated residues using mutual information to uncover evolutionary constraints. The authors show that mutual information not only captures coevolutionary information but also non-coevolutionary information such as conservation. One of the strong non-coevolutionary biases is stochastic bias. By first calculating mutual information between two residues which have evolved randomly (referred to as random mutual information), the authors then study relationships with other mutual information quantities to detect the presence of non-coevolutionary biases.

3. METHODS

A multiple sequence alignment $\mathcal{S}$ allows us to summarize each residue position in terms of the probabilities of encoun-



MSA

sequences	1	2	3	4	5
1.	K	C	Y	A	Q
2.	H	C	Y	C	R
3.	K	C	Y	D	H
4.	H	C	Y	G	H
5.	K	C	Y	S	P
6.	H	C	F	W	S
7.	K	C	F	W	Q
8.	H	C	F	E	P

Property Map 1:
{A,F,G,I,L,M,P,V} → n
{C,D,E,H,K,N,Q,R,S,T,W,Y} → p

Property Map 2:
{A,C,F,G,I,K,L,M,T,V,W,Y} → h
{D,E,H,N,P,Q,R,S} → q

p: polar h: hydrophobic
n: non−polar q: hydrophilic

Expanded MSA

sequences	1	2	3	4
1.	Kph	Cph	Yph	Anh
2.	Hpq	Cph	Yph	Cph
3.	Kph	Cph	Yph	Dpq
4.	Hpq	Cph	Yph	Gnh
5.	Kph	Cph	Yph	Spq
6.	Hpq	Cph	Fnh	Wph
7.	Kph	Cph	Fnh	Wph
8.	Hpq	Cph	Fnh	Epq

Figure 3: Expansion of a multiple sequence alignment into an 'inflated MSA'. Two classes—polarity and hydrophobicity—are used for illustration. Each column in the MSA is mapped to three columns in the expanded MSA.

tering each of the 20 amino acids (or a gap) in that position. Let $V = \{v_1, \ldots, v_n\}$ be a set of random variables, one for each residue position. The MSA then gives a distribution of amino acids for each random variable. We present two different classes of probabilistic graphical models to detect couplings. These inferred graphical models capture conditional dependence and independence among residues, as revealed by the MSA. The first approach uses an undirected graphical model (UGM), also known as a Markov random field. The second method employs a specific hierarchical latent class model (HLCM) which is a two-layered Bayesian network.

3.1 UGMs from Inflated MSAs

This approach can be viewed as an extension of the work of Thomas et al. [10]. It induces an undirected graphical model, $G = (V, E)$, where each node, $v \in V$, corresponds to a random variable and each edge, $(u, v) \in E$, represents a direct relationship between random variables u and v. In our problem setting, a node of G corresponds to a residue position (a column of the given MSA) and each edge represents a coupling between two residues. In this method, we redefine the approach of Thomas et al. [10] to discover MSA residue position couplings in terms of amino acid classes rather than residue values.

3.1.1 Inflated MSA

We augment the MSA $\mathcal{S}$ of a protein family by introducing extra 'columns' for each residue. Let l be the number of amino acid classes and $\mathcal{A}_i$ be the alphabet for the ith class where $1 \leq i \leq l$. Legal vocabularies for the classes can be constructed with the help of Taylor's diagram (see Fig. 2). For example, possible classes are polarity, hydrophobicity, size, charge, and aromaticity. Moreover, we may consider the amino acid sequence of a column as a "amino acid name" class. These classes take different values; e.g., the polarity class takes two values: polar and non-polar. Each column of $\mathcal{S}$ is mapped to l subcolumns to obtain an inflated MSA $\mathcal{S}_e$ where the extra columns (referred to as subcolumns) encode the corresponding class values. We use v_{ik} to denote the kth subcolumn of residue v_i. Figure 3 illustrates the above procedure for obtaining an inflated alignment $\mathcal{S}_e$. (A gap character in $\mathcal{S}$ is mapped to a gap character in $\mathcal{S}_e$.)

3.1.2 Detecting Coupled Residues

Couplings between residues can be quantified by many statistical and information-theoretic metrics [11]. In our model, we use conditional mutual information because it allows us to separate direct from indirect correlations. Recall that the *mutual information* (MI), $I(v_i, v_j)$, between residues v_i and v_j is given by:

$$I(v_i, v_j) = \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{A}} P(v_i = a, v_j = b) \cdot \log \frac{P(v_i = a, v_j = b)}{P(v_i = a) P(v_j = b)} \tag{1}$$

where the probabilities are all assessed from $\mathcal{S}$. If $I(v_i, v_j)$ is non-zero, then they are dependent, and each residue position (v_i or v_j) encodes information that can be used to predict the other. In the original *graphical models of residue coupling* (GMRC) model [10], Thomas et al. use conditional mutual information:

$$I(v_i, v_j | v_k) = \sum_{c \in \mathcal{A}^*} \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{A}} P(v_i = a, v_j = b | v_k = c) \cdot \log \frac{P(v_i = a, v_j = b | v_k = c)}{P(v_i = a | v_k = c) P(v_j = b | v_k = c)} \tag{2}$$

to construct edges, where the conditionals are estimated by subsetting residue k to its most frequently occurring amino acid types ($\mathcal{A}^* \subset \mathcal{A}$). The most frequently occurring amino acid types are those that appear in at least 15% of the original sequences in the subset. As discussed [15], such a bound is required in order to ensure sufficient fidelity to the original MSA and allow for evolutionary exploration.

For modeling residue position couplings in terms of amino acid classes, we use Eq. 2. As each residue in $\mathcal{S}_e$ has l columns, we consider all $O(l^2)$ pairs of columns for estimating mutual information between two residues. For calculating conditional mutual information in an inflated MSA, we condition a residue to its most appropriate class. The most appropriate class is the one that reduces the overall network score the most. The modified equation for conditional mutual information is as follows:

$$I_e(v_i, v_j | v_{kr}) = \sum_{p=1}^{l} \sum_{q=1}^{l} I_e(v_{ip}, v_{jq} | v_{kr}) \tag{3}$$

where

$$I_e(v_{ip}, v_{jq}|v_{kr}) = \sum_{c \in \mathcal{A}_r^*} \sum_{a \in \mathcal{A}_p} \sum_{b \in \mathcal{A}_q} P(v_{ip} = a, v_{jq} = b|v_{kr} = c) \cdot \log \frac{P(v_{ip} = a, v_{jq} = b|v_{kr} = c)}{P(v_{ip} = a|v_{kr} = c)P(v_{jq} = b|v_{kr} = c)} \quad (4)$$

Here $\mathcal{A}_i$ denote the alphabet of the ith amino acid class where $1 \leq i \leq l$. The conditional variable v_k is set to the rth class. If $I_e(v_i, v_j|v_{kr}) = 0$, then it implies that residue v_i and v_j are independent conditioned on the rth class of v_k. Observe that we can subset the residue v_k to any class out of l classes. We take the minimum of $I_e(v_i, v_j|v_{kr})$ for $1 \leq r \leq l$ to obtain the final mutual information between v_i and v_j.

3.1.3 Normalized Mutual Information

In an inflated MSA, the subcolumns corresponding to a residue take values from different alphabets of different sizes. Let v_{ip} and v_{jq} be two subcolumns that take values from alphabets $\mathcal{A}_p$ and $\mathcal{A}_q$ respectively. To understand the effect of the sizes of alphabets in mutual information score, we calculate pairwise mutual information of subcolumns for every residue pair and produce a scatter plot (see Fig. 4(a)).

In Fig. 4(a), we see that $MI(A, A)$ is dominating over $MI(P, P)$, $MI(H, H)$, and $MI(S, S)$. This is expected, because amino acids are of 21 types whereas polarity, hydrophobicity, and size have 3 types. We adopt the following equation to normalize mutual information scores proposed by Yao [18]:

$$I_{norm}(v_{ip}, v_{jq}|v_{kr}) = \frac{I(v_{ip}, v_{jq}|v_{kr})}{\min(H(v_{ip}|v_{kr}), H(v_{jq}|v_{kr})} \quad (5)$$

where $H(v_{ip}|v_{kr})$ and $H(v_{jq}|v_{kr})$ denote the conditional entropy.

3.1.4 Learning UGMs

Given an expanded MSA $\mathcal{S}_e$, we infer a graphical model by finding *decouplers* which are sets of variables that makes other variables independent. If two residues v_i and v_j are independent given v_k, then v_k is a decoupler for v_i and v_j. In this case, we add edges (v_i, v_k) and (v_j, v_k) to the graph. Thus the relationship between v_i and v_j is explained transitively by edges (v_i, v_k) and (v_j, v_k). Moreover, we can consider a prior that can be calculated from a contact graph of a representative member of the family. A prior gives a set of edges between residues which are close in three-dimensional structure. When a residue contact network is given as a prior, we consider each edge of the residue contact network as a potential candidate for couplings. Without a prior, we consider all pairwise residues for coupling. Algorithm 1 gives the formal details for inferring a graphical model.

Our algorithm builds the graph in a greedy manner. At each step, the algorithm chooses the edge from a set of possible couplings which scores best with respect to the current graph. The score of the graph is given by:

$$\mathrm{S}_{UGM}(G = (V, E)) = \sum_{v_i \in V} \sum_{v_j \notin \mathrm{N}(v_i)} I_e(v_i, v_j|\mathrm{N}(v_i)) \quad (6)$$

where $\mathrm{N}(v_i)$ is the set neighbors of v_i.

Algorithm 1 GMRC-Inf($\mathcal{S}, P$)

Input: $\mathcal{S}$ (multiple sequence alignment), P (possible edges)
Output: G (a graph that captures couplings in $\mathcal{S}$)

```
$V = \{v_1, v_2, \ldots, v_n\}$
$E \leftarrow \phi$
$s \leftarrow \mathrm{S}_{UGM}(G = (V, E))$
for all $e = (v_i, v_j) \in P$ do
    $C_e \leftarrow s - \mathrm{S}_{UGM}(G = (V, \{e\}))$
while stopping criterion is not satisfied do
    $e \leftarrow \arg\max_{e \in P-E} C_e$
    if $e$ is significant then
        $E \leftarrow E \cup \{e\}$
        label $e$ based on the score
        $s \leftarrow s - C_e$
        for all $e' \in P - E$ s.t $e$ and $e'$ share a vertex do
            $C_{e'} \leftarrow s - \mathrm{S}_{UGM}(G = (V, E \cup \{e'\}))$
return $G = (V, E)$
```

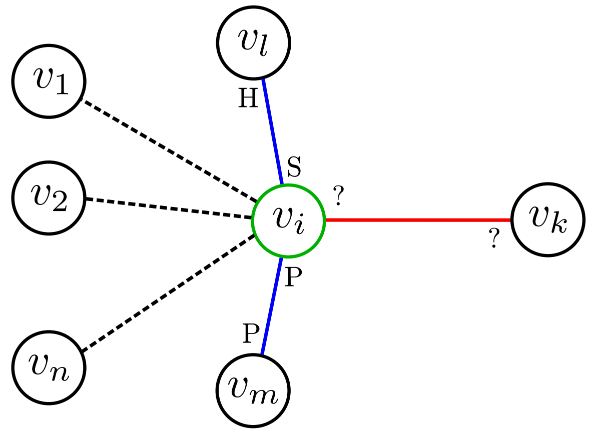


Figure 5: Class labeling of coupled edges. The blue edges are already added to the network and dashed edges are not. The red edge is under consideration for addition in the current iteration of the algorithm. The "?" takes any of the four classes: polarity (P), hydrophobicity (H), size (S), or the default amino acid values (A).

The calculation of conditional mutual information and labeling of edges with different properties is illustrated in Fig. 5. In Fig. 5, we consider edge (v_i, v_k) for addition to the graph where v_i already has two neighbors v_l and v_m. The edge (v_i, v_l) has the label S-H which means the coupling models v_i with respect to size and v_l with respect to hydrophobicity. Similarly, the edge (v_i, v_m) has the label P-P which means the coupling between v_i and v_m can be described with respect to their polarities. To evaluate the edge (v_i, v_k), we condition on v_m and v_l first and then condition v_k on any of the properties. We then sum up all $I_e(v_i, v_j)$, where $v_j \notin \{v_l, v_m, v_k\}$. The subsetting class of v_k for which we obtain a maximum for $\sum I_e(v_i, v_j)$ is the label that we finally assign to v_k (the question mark in Fig. 5) if the edge (v_i, v_k) is added. Similarly, we do the same calculation for v_k while subsetting only v_i, as the residue v_k does not have any neighbors in the current network.

Algorithm 1 can incorporate various stopping criteria: 1)

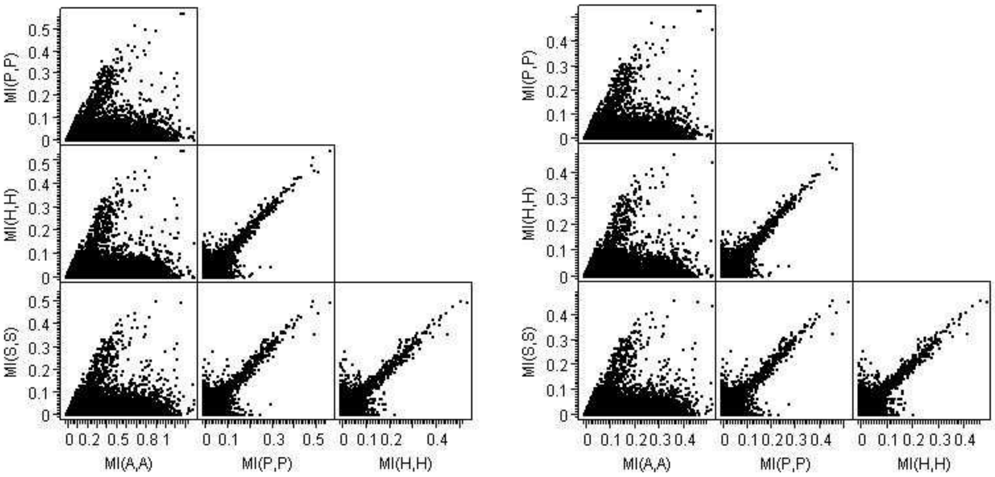


Figure 4: Effect of alphabet length on mutual information. Here, A,P,H,S denote amino acid, polarity, hydrophobicity, and size column respectively. (a) Scatter plot of mutual information for every residue pair without normalization. (b) Scatter plot of mutual information for every residue pair with normalization. Notice the different scales of plots between (a) and (b).

stop when a newly added edge does not contribute much to the score reduction of the graph, 2) stop when a designated number of edges have been added, and 3) stop when the likelihood of the model is within acceptable bounds. We use the first criterion in our model. With naive implementation of Algorithm 1 the running time is $O(dn^2)$ where n is the number of residues in a family and d is the maximum degree of nodes in the prior. By caching and preprocessing the complexity can be reduced to $O(dn)$.

3.2 Hierarchical Latent Class Models

A *latent class model* (LCM) is a hidden-variable model which consists of a hidden (class) variable and a set of observed variables [13]. The semantics of an LCM are that the observed variables are independent given a value of the class variable. Let u and v be two observed variables. The latent class model of u and v introduces a latent variable z, so that

$$P(u,v) = \sum_k P(z=k)P(u|z=k)P(v|z=k) \qquad (7)$$

When the number of observed variables increases, the LCM model performs poorly due to the strong assumption of local independence. To improve the model, Zhang et al. proposed a richer, tree-structured, latent variable model [20]. Our hierarchical model is a restricted case of the model proposed by Zhang et al. We propose a two-layered binary hierarchical latent class model where the lower layer consists all the observed variables and the upper layer consists of hidden class variables. In our problem setting, observed variables correspond to residues and the hidden class variables take values from all possible permutations of pairwise amino acid classes. Figure 6 illustrates a hypothetical hierarchical latent class model.

Let Z be the set of all hidden variables and V be the set of observed variables. The joint probability distribution of the model is as follows:

$$P(Z)\prod_{i=1}^{n} P(v_i|\mathrm{Pa}(v_i)) \qquad (8)$$

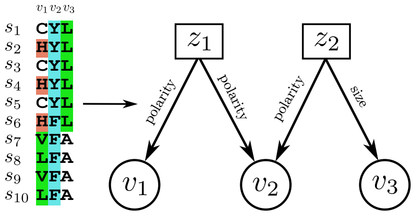


Figure 6: A hypothetical residue coupling in terms of amino acid classes using a two-layered Bayesian network.

where $\mathrm{Pa}(v_i)$ denotes the set of parents of v_i.

3.2.1 Learning a HLCM

We learn this model in a greedy fashion as before. We define the following scoring function:

$$\mathrm{S}_{\mathrm{HLCM}}(G=(\{V,Z\},E)) = \sum_{v_i\in V}\sum_{v_j\notin \mathrm{Pa}(v_i)} I_e(v_i,v_j|\mathrm{Pa}(v_i)) \qquad (9)$$

where $\mathrm{Pa}(v_i)$ is the set neighbors of v_i. When we condition on the parent nodes, we use a 35% support threshold for the sequences. This support threshold is required in order to ensure sufficient fidelity to the original MSA and allow for evolutionary exploration. From extensive experiments with this parameter (data not shown), we found that while there is some variation in the edges with changes of this parameter from 15% to 60%, many of the best edges are retained when support threshold is 35%. Moreover, the model has less number of couplings when support threshold is 35% which is an indication in the reduction of the overfitting effect. Besides, we use a parameter *minsupport* which is set to 2; minsupport is used to avoid class conservation between sequences. The value of minsupport for two residue positions is the number of class-values combinations for which the number of sequences in each subset is greater

Algorithm 2 HLCM($\mathcal{S}, P$)

Input: $\mathcal{S}$ (multiple sequence alignment), P (possible pairs of residues)
Output: G (a graph that captures couplings in $\mathcal{S}$)

```
$V = \{v_1, v_2, \ldots, v_n\}$
$Z \leftarrow \phi$        ▷ set of hidden nodes
$E \leftarrow \phi$
$T \leftarrow \phi$        ▷ tabu list of residue pairs
$s \leftarrow \mathrm{S}_{\mathrm{HLCM}}(G = (V, E))$
for all $e = (v_i, v_j) \in P$ do
    $E' \leftarrow \{(h_e, v_i), (h_e, v_j)\}$
            ▷ $h_e$ is a hidden class between $v_i$ and $v_j$
    $C_e \leftarrow s - \mathrm{S}_{\mathrm{HLCM}}(G = (\{V, \{h_{ij}\}\}, E'))$
while stopping criterion is not satisfied do
    $e \leftarrow \arg\max_{e \in P-T} C_e$
    if $e$ is significant for coupling then
        $E \leftarrow E \cup \{(h_e, v_i), (h_e, v_j)\}$
        $Z \leftarrow Z \cup \{h_e\}$
        $T \leftarrow T \cup \{e\}$
        label two edges of $h_e$ based on the score
        $s \leftarrow s - C_e$
        for all $e' = (v_k, v_l) \in P - T$ s.t $e$ and $e'$ share a vertex do
            $E'' \leftarrow \{(h_{e'}, v_k), (h_{e'}, v_l)\}$
            $C_{e'} \leftarrow s - \mathrm{S}_{\mathrm{HLCM}}(G = (\{V, Z\}, E \cup E''))$
return $G = (V, E)$
```

than the support threshold. When minsupport is 1 for two residue positions, we consider that a class conservation has occurred in these residue positions. The algorithm chooses a pair of residues for which introducing a hidden variable reduces the current network score the most. We then add the hidden variable if it is statistically significant. Algorithm 2 gives the formal details for learning HLCMs. We can employ various stopping criteria: 1) stop when a newly added hidden node does not contribute much to the score reduction of the graph, 2) stop when a designated number of hidden nodes have been added, and 3) stop when the likelihood of the model is within acceptable bounds. We use the first criterion in our model.

3.3 Statistical significance

While learning the edges, hidden nodes or factors of the above graphical models, we assess the significance of each coupling imputed. In both algorithms, we perform a statistical significance test on potential pairs of residues before adding an edge or hidden variable to the graph. To compute the significance of the edge, we use p-values to assess the probability that the null hypothesis is true. In this case, the null hypothesis is that two residues are truly independent rather than coupled. We use the χ-squared test on potential edges. If p-value is less than a certain threshold p_θ, we add the edge to the graph. In our experiment, we use $p_\theta = 0.005$.

3.4 Classification

The graphical models learned by algorithm are useful for annotating protein sequences of unknown class membership with functional classes. To demonstrate the classification methodology, we consider HLCM as an example. We adopt Eq. 10 to estimate the parameters of a residue in the HLCM model. The reason for using this estimator is that the MSA may not sufficiently represent every possible amino acid value for each residue position. Therefore, we must consider the possibility that an amino acid value may not occur in the MSA but still be a member of the family. In Eq. 10, $|\mathcal{S}|$ is number of sequences in the MSA and α is a parameter that weights the importance of missing data. We employ a value of .1 for α but tests (data not shown) indicate that results are similar for values in $[0.1, 0.3]$.

$$P(v = a) = \frac{freq(v = a) + \frac{\alpha|\mathcal{S}|}{21}}{|\mathcal{S}|(1 + \alpha)} \tag{10}$$

Given two different graphical models, G_{C_1} and G_{C_2}, say for two different classes, we can classify a new sequence s into either functional class C_1 or C_2 by computing the log likelihood ratio LLR:

$$LLR = \log \frac{\mathcal{L}_{G_{C_1}}}{\mathcal{L}_{G_{C_2}}} \tag{11}$$

If LLR is greater than 0 then, then we classify s to the class C_1; otherwise, we classify it to the class C_2.

4. EXPERIMENTS

In this section, we describe the datasets that we use to evaluate our model and show results that reflect the capabilities of our models. We seek to answer the following questions using our evaluation:

1. How do our graphical models fare compared to other methods? Do our learned models capture important covariation in the protein family? (Section 4.2)

2. Do the learned graphical models have discriminatory power to classify new protein sequences? (Section 4.3)

3. What forms of amino acid class combinations are prevalent in the couplings underlying a family? (Section 4.4)

4.1 Datasets

4.1.1 Nickel receptor protein family

The Nickel receptor protein family (NikR) consists of repressor proteins that bind nickel and recognize a specific DNA sequence when nickel is present, thereby repressing gene transcription. In the *E. coli* bacterium, nickel ions are necessary for the catalytic activity of metalloprotein enzymes under anaerobic conditions; NikABCDE permease acquires Ni^{2+} ions for the bacterium [2]. NikR is one of the two nickel-responsive repressors which control the excessive accumulation of Ni^{2+} ions by repressing the expression of NikABCDE. When Ni^{2+} binds to NikR, it undergoes conformational changes for binding to DNA at the NikABCDE operator region and represses NikABCDE [2].

NikR is a homotetramer consisting of two distinct domains [16]. The N-terminal domain of each chain has 50 amino acids and constitutes a ribbon-helix-helix (RHH) domains that contact the DNA. The C-terminal of each chain consisting of 83 amino acids form a tetramer composed of four ACT domains that together contain the high-affinity Ni^{2+} binding sites [2]. Figure 7 shows a representative NikR structure determined by X-ray crystallography [2].

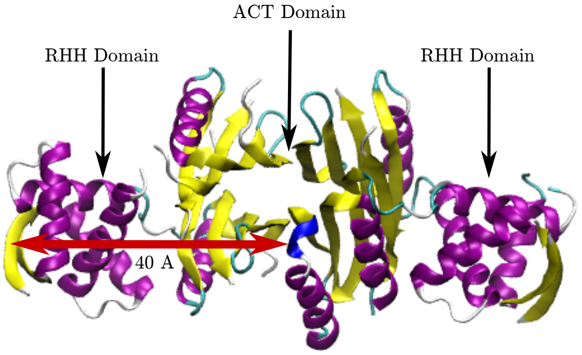


Figure 7: A rendering of NikR protein (PDB id 1Q5V) showing two domains: ACT domain (Nickel binding site) and RHH domain (DNA binding site). The distance between these two domains is 40ÅṪhe molecular image is generated using VMD 1.9 [5].

We organized an MSA of the NikR family that has 82 sequences which are used to study allosteric communication in NikR [2]. Each sequence has 204 residues. For a structural prior, we use Apo-NikR (pdb id 1Q5V) as a representative member of the NikR family and calculate prior edges from its contact map. Residue pairs within 7Å of each other are considered to be in contact which gives us 734 edges as a prior. We use this prior for the analysis to ensure that all identified relationships have direct mechanistic explanations.

4.1.2 *G-protein coupled receptors*

G-protein coupled receptors (GPCRs; see Fig. 8) represent a class of large and diverse protein family and provide an explicit demonstration of allosteric communication. The primary function of this proteins is to transduce extracellular stimuli into intracellular signals [6]. GPCRs are a primary target for drug discovery.

We obtained an MSA of 940 GPCR sequences used in the statistical coupling analysis by Ranganathan and colleagues [8]. Each sequence has 348 residues. GPCRs can be organized into five major classes, labeled A through E. The MSA that we obtained is from class A; using the GPCRDB [4], we annotate each sequence with functional class information according to the type of ligand the sequence binds to. The three largest functional classes—Amine, Peptide, and Rhodopsin—have more than 100 sequences. There are 12 other functional classes having less than 45 sequences. There are 66 orphan sequences which do not belong to any family. For prior couplings, we constructed a contact graph network from the 3D structure of a prominent GPCR member, viz. bovine rhodopsin (pdb id 1GZM). We identify 3109 edges as coupling priors using a pairwise distance threshold of 7Å.

4.2 Evaluation of couplings

We evaluate four methods on the NikR and GPCR datasets: the traditional GMRC method proposed by Thomas et al. [10, 9]; GMRC-Inf from this paper; GMRC-Inf* (a variant of GMRC-Inf) where the inflated alignment uses only class-based information; and HLCM. We consider three physicochemical properties—polarity, hydrophobicity, and size—of amino acids as classes. Although GMRC discovers couplings in terms of amino acids, we compare our methods with GMRC with respect to the number of discovered important

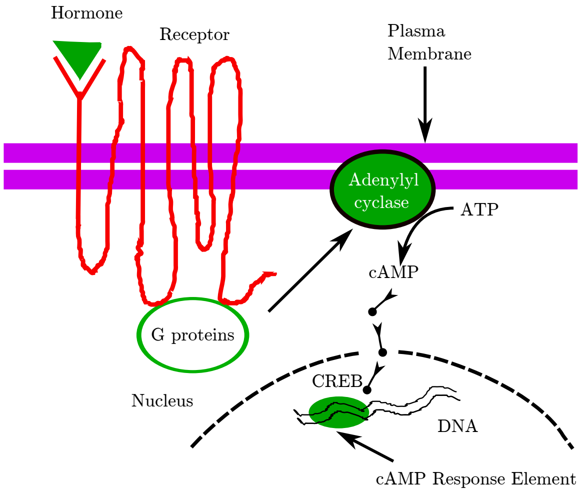


Figure 8: A cartoon describing GPCR functionality. Figure redrawn from [12].

Table 1: Important residues for allosteric activity in NikR collected from [2]. Residues are mapped from indices with respect to Apo Nikr (PDB id 1Q5V) to the indices of NikR MSA column. Important residues having conservation greater than 90% are not shown.

Residue	Sequence Conservation	Significance
3	0.83	Specific_DNA_binding
5	0.62	Specific_DNA_binding
7	0.81	Specific_DNA_binding
9	0.58	Unknown
22	0.45	Unknown
27	0.64	Nonspecific_DNA_contact
30	0.81	Low-affinity_Metal_Site
33	0.87	Nonspecific_DNA_contact
34	0.71	Low-affinity_Metal_Site
37	0.85	Unknown
42	0.41	Unknown
58	0.60	Ni2+_site_H-bond_network
60	0.86	Close_proximity_to_Ni2+_site
62	0.83	Close_proximity_to_Ni2+_site
64	0.38	Nonspecific_DNA_contact
65	0.52	Nonspecific_DNA_contact
69	0.51	Unknown
75	0.74	Ni2+_site_H-bond_network
109	0.49	Unknown
114	0.47	Unknown
116	0.39	Low-affinity_Metal_Site
118	0.45	Low-affinity_Metal_Site
119	0.62	Nonspecific_DNA_contact
121	0.82	Low-affinity_Metal_Site

Table 2: Comparisons of methods for various feature on NikR dataset.

Features	GMRC	GMRC-Inf	GMRC-Inf*	HLCM
Support Threshold (%)	15	15	35	35
Num of couplings	80	65	26	51
Num of important residues (out of 24)	15	11	9	15
Unique residues in the network	81	61	38	74
Num of components	11	6	13	23

residues (we desire to investigate whether our models can recapitulate important residues identified by previous methods). In Table 1, we list 24 important residues for NikR activity from [2] which are not conserved. (We exclude seven important residues for NikR which have a conservation of more than 90%.) Table 2 gives comparisons between methods for these two datasets.

Likewise, we identify 47 important residues for the GPCR family from [8]. The support threshold for GMRC and GMRC-Inf is set to 15%; the support threshold and minsupport for HLCMis set to 35% and 2 respectively. (To be more confident about the quality of the model, the support for HLCMis set to a higher value.)

Bradley et al. [2] identify four residues (Res 9, Res 37, Res 62, and Res 118) as highly connected “hubs”. In our models, Res 9 and Res 118 are present, but Res 37 and Res 62 are not present since these residues are highly conserved. Important residues discovered by four methods are shown in Table 3. We see that GMRC-Inf and GMRC-Inf* are progressively more strict than GMRC in the number of important residues discovered but GMRC-Inf* has a greater ratio of important residues discovered to the total residues in the network. HLCM provides as good performance as the GMRC method in terms of the important residues but compacts them into a smaller set of couplings.

Table 3: Important residues discovered by HLCM, GMRC-Inf, GMRC-Inf*, and GMRC in NikR.

Method	Important Residues
HLCM	3, 7, 9, 27, 30, 34, 42, 60, 97, 109, 114, 116, 118, 119, 121
GMRC-Inf	27, 30, 33, 34, 37, 58, 60, 97, 116, 118, 121
GMRC-Inf*	3, 5, 27, 33, 37, 42, 60, 116, 121
GMRC	3, 7, 9, 27, 30, 33, 34, 37, 58, 60, 97, 116, 118, 119, 121

4.3 Classification performance

Although our goal is to represent amino acid class-based residues couplings in a formal probabilistic model, we demonstrate that our models can also classify protein sequences. We use the GPCR dataset to assess the classification power of our models. The GPCR datasets has 16 subclasses with, as stated earlier, the three major subclasses being amine, peptide, and rhodopsin. We performed a five-fold cross-validation test for these three major classes. A comparison between our HLCM model and the vanilla GMRC is given in Table 4. We see an improved performance for the Amine subclass and a slightly decreased performance for the Rhodopsin subclass.

Table 4: Classification of GPCR subclasses.

Functional Class	Total Sequence	Accuracy (%)	
		GMRC	HLCM
Amine	196	99.5	100
Peptide	333	100	100
Rhodopsin	143	98.6	95.8

Recall that there are 66 orphan sequences in GPCR family which are not assigned to any functional class. We apply our model to classify these orphan sequences to any of the three major classes: Amine, Peptide, and Rhodopsin. Toward this end, we build models for the three classes using HLCM method by considering all of the sequences. Of the 66 sequences, 3 are classified to Amine and the rest are classified to the Peptide class. This result is the same as the GMRC result reported in [10].

4.4 Finding coupling types

We determine the frequency of each class-coupling type for the various models on the NikR dataset. Histograms are shown in Figure 9. We see that there are a significant number of class-based residue coupling relationships discovered, although in the case of GMRC-Inf, there are many value-based couplings as well (as expected). Many of the couplings discovered by GMRC-Inf* and HLCM have polarity as one of the properties, but there are interesting differences as well: HLCM identifies a significant number of P-S couplings whereas GMRC-Inf* finds P-P, P-H, and S-S couplings.

5. DISCUSSION

Our results on the NikR dataset demonstrate that employing amino acid types is useful for learning couplings and the underlying properties of those couplings. This approach provides us with a way to build an expressive model for residue couplings. We have shown that our extended graphical model is more powerful than the previous graphical model approach of Thomas et al. [10].

Our use of conditional mutual information as a correlation measure is subject to different biases [14]. Removing possible biases is a direction for future work. A more unifying probabilistic approach for residue couplings would be a factor graph representation since it can capture couplings among more than two residues. A *factor graph* is a bipartite graph that represents how a joint probability distribution of several variables factors into a product of local probability distributions [1]. Let $G = (\{F, V\}, E)$ be a factor graph, where $F = \{f_1, f_2, \ldots, f_m\}$ is a set of factor nodes and $V = \{v_1, \ldots, v_n\}$ is a set of observed variables. A *scope* of a factor f_i is set a set of observed variables. Each factor

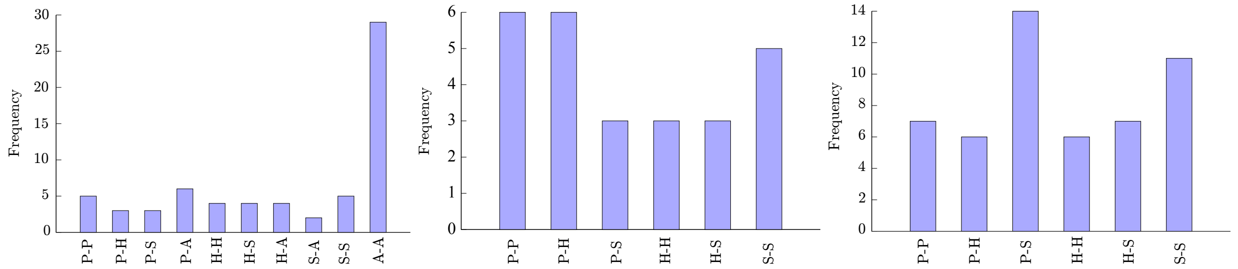


Figure 9: Histograms for class-coupling types on the NikR dataset using three methods: (a) GMRC-Inf (b) GMRC-Inf*, and (c) HLCM.

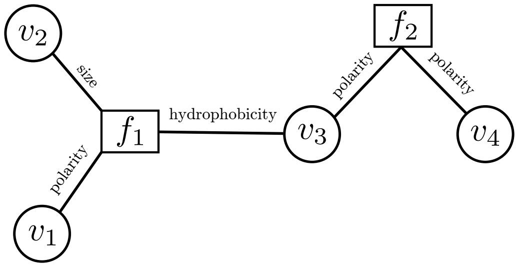


Figure 10: A hypothetical residue coupling in terms of amino acid classes using a factor graph model.

f_i with scope C is a mapping from $\mathrm{Val}(C)$ to $\mathbb{R}^+$. The joint probability distribution of V is as follows:

$$P(v_1, v_2, \dots, v_n) = \frac{1}{Z}\prod_{j=1}^{m} f_j(C_j) \tag{12}$$

where C_j is the scope of the factor f_j and the normalizing constant Z is the partition function. Figure 10 illustrates a hypothetical residue coupling network for four residues with two factors. Observe how such a model can capture couplings involving more than two residues.

While there are polynomial time algorithm for learning factor graphs from polynomial samples [1], such methods require a canonical parameterization which constraints the applicability of factor graphs to learn couplings from an MSA. Canonical parameterizations are defined relative to an arbitrary but fixed set of assignments to the random variable, and it is hard to define such a 'default sequence' for an MSA. Hence, newer algorithms need to be developed.

Acknowledgement

This work is funded by NSF grant IIS-0905313. We convey our thanks to Debprakash Patnaik for his suggestions.

6. REFERENCES

[1] Abbeel et al. Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7:1743–1788, 2006.

[2] Bradley et al. Molecular dynamics simulation of the Escherichia coli NikR protein: equilibrium conformational fluctuations reveal interdomain allosteric communication pathways. *JMB*, 378(5):1155–1173, May 2008.

[3] Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, 1998.

[4] Horn et al. Collecting and harvesting biological data: The GPCRDB and NucleaRDB databases. *Nucleic Acids Research*, 29(1):346–349, 2001.

[5] Humphrey et al. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

[6] Kroeze et al. G-protein-coupled receptors at a glance. *Journal of Cell Science*, 116:4867–4869, 2003.

[7] Lichtarge et al. An evolutionary trace method defines binding surfaces common to protein families. *Journal of Molecular Biology*, 257:342–358, 1996.

[8] Suel et al. Evolutionary conserved networks of residues mediate allosteric communication in proteins. *Nature Structural Biology*, 10(1):59–69, Jan 2003.

[9] Thomas et al. Graphical models of residue coupling in protein families. In *5th ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD)*, pages 1–9, 2005.

[10] Thomas et al. Graphical models of residue coupling in protein families. *IEEE/ACM TCBB*, 5(2):183–97, 2007.

[11] A.A. Fodor and R.W. Aldrich. Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. *Proteins: Structure, Function, and Bioinformatics*, 56:211–221, 2004.

[12] John W. Kimball. Cell signaling, June 2006.

[13] P. F. Lazarsfeld and N. W. Henry. *Latent Structure Analysis*. Boston, Mass.: Houghton Mifflin., 1968.

[14] Little. Identification of coevolving residues and coevolution potentials emphasizing structure, bond formation and catalytic coordination in protein evolution. *PloS One*, 4(3):e4762, January 2009.

[15] S.W. Lockless and R. Ranganathan. Evolutionarily conserved pathways of energetic connectivity in protein families. *Science*, 286(5438):295–299, Oct 1999.

[16] Eric R. Schreiter, Michael D. Sintchak, Yayi Guo, Peter T. Chivers, Robert T. Sauer, and Catherine L Drennan. Crystal structure of the nickel-responsive transcription factor nikr. *Nature Structural and Molecular Biology*, 10:794–799, September 2003.

[17] William S J Valdar. Scoring residue conservation. *Proteins*, 48(2):227–41, August 2002.

[18] Y. Y Yao. Information-theoretic measures for knowledge discovery and data mining. *Entropy Measures, Maximum Entropy Principle and Emerging Applications*, pages 115–136, 2003.

[19] Chen-Hsiang Yeang and David Haussler. Detecting coevolution in and among protein domains. *PLoS Computational Biology*, 3(11):13, 2007.

[20] Nevin L. Zhang and Tomás Kocka. Efficient learning of hierarchical latent class models. *IEEE International Conference on Tools with Artificial Intelligence*, 0:585–593, 2004.