Structure Learning Constrained by Node-Specific Degree Distribution

Jianzhu Ma[1], **Qingming Tang**[1], **Sheng Wang**, **Feng Zhao**, **Jinbo Xu**
Toyota Technological Institute at Chicago
Chicago, IL - 60637
{majianzhu, qmtang, wangsheng, fzhao, j3xu}@ttic.edu

Abstract

We consider the problem of learning the structure of a Markov Random Field (MRF) when a node-specific degree distribution is provided. The problem setting is inspired by protein contact map (i.e., graph) prediction in which the contact number (i.e., degree) of an individual residue (i.e., node) can be predicted without knowing the contact graph. We formulate this problem using maximum pseudo-likelihood plus a node-specific ℓ_1 regularization derived from the predicted degree distribution. Intuitively, when a node have k predicted edges, we dynamically reduce the regularization coefficients of the k most possible edges to promote their occurrence. We then optimize the objective function using ADMM and an Iterative Maximum Cost Bipartite Matching algorithm. Our experimental results show that using degree distribution as a constraint may lead to significant performance gain when the predicted degree has good accuracy.

1. INTRODUCTION

Structure learning of a Markov Random Field (MRF) is an important problem and has been applied to many real-world problems which require study of conditional independence between a set of objects. For example, structure learning has been used to derive gene expression or regulatory network from gene expression levels [1, 6, 32, 36] and predict the contact map of a protein from a multiple sequence alignment of a protein family [9, 15, 16, 23, 24]. Two major approaches and their variants have been studied to learn the structure of a graphical model from data: Gaussian Graphical Model (GGM) [11] and maximum pseudo-likelihood [29]. Since many real-world structures are usually sparse, ℓ_1 regularization is usually added to the objective function to generate a sparse structure. Empirical studies indicate that the pseudo-likelihood approach may have better prediction accuracy and is also more efficient than GGM, by dropping the Gaussian distribution assumption.

In real-world applications, the underlying structure (or graph) usually has some special properties and must satisfy some topological constraints. For example, a gene expression network is scale-free. A protein contact graph must satisfy some geometric constraints, e.g., the degree of each node is upper bounded by a constant and also depends on the properties of its corresponding amino acid. Only a few structure-learning algorithms take into consideration topological constraints of the underlying graph, which can be used to reduce the feasible solution set of the problem. From another perspective, predicted graphs without considering these constraints might contain conflicts and are physically infeasible. A predicted contact graph violating the above-mentioned geometric constraints may not correspond to a feasible protein structure. Several papers [12, 17, 30, 35] have considered some very general topological constraints describing the global properties of a graph to improve structure learning. However, these non-specific topological constraints do not help very much in practice. The reason may be that they are too loose for some nodes (graphs) and too restrictive for others and thus, the overall performance gain is limited.

The problem addressed by this paper is inspired by protein contact graph prediction. A protein sequence consists of a string of amino acids (also called residues). In nature, a protein sequence folds into a specific 3D shape to function properly. Two residues are defined to form a contact if they are close (distance≤8 Å) in the 3D space. Therefore, we can use a contact map to model a protein 3D structure. Predicting inter-residue contacts from sequence is an important and challenging problem. Recent studies [22, 24, 25, 27] indicate that predicted inter-residue contacts could be used as a valuable constraint to improve the folding of some proteins. Baker group [16] shows that one correct long-range contact for every 12 amino acids (AAs) in a

[1] The first two authors contribute equally to the paper.

protein allows for accurate topology-level protein folding. Recent breakthroughs [9, 15, 24] apply Gaussian Graphical Model and maximum pseudo-likelihood to formulate protein contact prediction as a structure learning problem. In these formulations, a protein sequence is viewed as a sample generated from a Markov Random Field (MRF), in which an MRF node represents one AA (also called residue) and an edge indicates a contact (i.e., strong interaction) between two AAs.

Without knowing the actual contact graph of a protein, we can use a supervised learning method to predict the number of contacts (i.e., degree) of an AA from sequence information. In particular, we use $2L$ different linear-chain 2^{nd}-order Conditional Neural Fields (CNFs) [28] to predict the degree distribution for a protein of length L. A CNF is an integration of neural networks and Conditional Random Field (CRF) [20]. CNF models the relationship between the label at each node and input features by neural networks and also correlation among neighboring labels. Therefore, CNF can capture the complex relationship between node labels and features as well as the dependency between node labels. The predicted node degree distribution is then used as a regularization to help improve individual contact prediction.

2. RELATED WORK

To our best knowledge, there are very few published work that uses node-specific degree distribution to help with structure learning of MRFs. Motivated by the observation that many social and biological networks follow a power-law degree distribution [2, 5, 14], [21] proposed a novel non-convex reweighted ℓ_1 regularization by using a log surrogate to approximate the power-law distribution. The basic idea is to reduce the regularization coefficients for hub nodes (i.e., nodes with a large degree) to promote their occurrence. A convex variant of this work was developed in [7], resulting in further performance improvement. This work modeled the structure learning problem as a set-function optimization problem and approximated it by Lovasz extension [4]. The resultant objective function is another kind of reweighed ℓ_1 regularization. Although these methods result in a graph following a power-law degree distribution, their accuracy of the predicted edges is not much better than the simple ℓ_1 regularization. A very recent work [37] obtained much better accuracy by making use of a reweighted ℓ_1 regularization accounting for not only global degree distribution, but also the estimated degree of an individual node and the relative strength of all the edges of the same node.

Other work such as [10] takes into consideration eigenvector centrality constraints and triangle-shaped local motifs of the graph, which are properties of gene regulatory networks and protein-protein interaction networks. [33] presented a convex formulation that uses a group-sparsity penalty on the rows and columns of the data precision matrix, effectively selecting which nodes connect with all the other nodes or no nodes at all. This formulation also results in a graph following a power-law distribution.

3. METHOD

3.1 NOTATIONS AND PRELIMINARIES

Given a protein sequence, we can run PSI-BLAST [3] to find its sequence homologs (i.e., proteins in the same family) and build a multiple sequence alignment (MSA) of homologs. By examining this MSA, we can identify evolution and co-evolution patterns in a protein family. By co-evolution, we mean the evolution of one AA is strongly impacted by the other. As shown in Figure 1, the AAs in the two red MSA columns are co-evolved. It has been observed that two co-evolved residues are likely to form a contact in the 3D space since they strongly interact with each other.

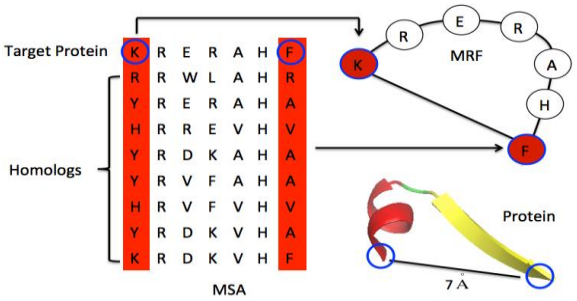

Figure 1. Two coevolved AAs (in red columns) may form a contact in 3D space.

We can use Markov Random Fields (MRF) to model the MSA and infer inter-residue contacts by structure learning of the MRF. In the MRF model, a node represents one MSA column and an edge represents correlation between two MSA columns. Let $X = \{X_1, X_2, \dots, X_L\}$ be a protein sequence where X_i represents amino acid type (or gap) at column i. Let R denote the number of protein sequences (or rows) in the MSA. Let X_{ir} denote the amino acid type observed at row r $(1 \le r \le R)$ and column i $(1 \le i \le L)$. The probability of observing X can be defined as follows.

$$P(X) = \prod_{r=1}^{R} \frac{\exp(\sum_{i=1}^{L} b_i(X_{ir}) + \sum_{i<j} w_{ij}(X_{ir}, X_{jr}))}{Z} \quad (1)$$

Here b_i and w_{ij} denote the unary and binary potential functions for nodes i and j, respectively. Z is the partition function, summing over all the possible label combinations. If nodes i and j share an edge in the graph, they are correlated given all the other nodes, indicating that their corresponding AAs form a contact and interact with each other in the 3D space. Therefore, the contact number of one AA corresponds to the node degree in the graph. Both training and inference by maximizing (1) over a general graph are NP-hard. Pseudo-likelihood approximation [9, 29] is proposed to

deal with this. Substituting the original likelihood function, we have,

$$P(X_i) = \prod_{r=1}^{R} P(X_{i,r}|X_{\setminus i,r}) \quad (2)$$

$$= \prod_{r=1}^{R} \frac{\exp(b_i(X_{i,r}) + \sum_{j=1,j\neq i}^{L} w_{i,j}(X_{i,r}, X_{j,r}))}{\sum_{X_{i,r}} \exp(b_i(X_{i,r}) + \sum_{j=1,j\neq i}^{L} w_{i,j}(X_{i,r}, X_{j,r}))}$$

Each binary potential function w_{ij} is a 21×21 matrix. We can estimate all w_{ij} by maximizing (2). We can use $\sum_{a,b=1}^{20} |w_{i,j}(a,b)|$ to measure the interaction strength between two nodes i and j. A pair of nodes with strong interaction is predicted to share an edge or form a contact.

3.2 NODE-SPECIFIC DEGREE REGULARIZATION

In this section we introduce how to add a node-specific degree distribution as a prior to the above pseudo-likelihood function. Let $P_i(k)$ be the predicted probability of node i having k contacts. Let W_{ij}=$\sum_{a,b=1}^{20} |w_{i,j}(a,b)|$, which indicates the interaction strength between two AAs i and j. Given a i, we exclude W_{ii} and denote the *t-th* largest W_{ij} as $W_{i,(t)}$. We use the following penalty term Ω_i for AA i.

$$\Omega_i = \sum_{k=1}^{L-1} P_i(k)\left(-\sum_{t=1}^{k} W_{i,(t)} + \sum_{t=k+1}^{L-1} W_{i,(t)}\right) \quad (3)$$

Eq. (3) implies that if the degree of AA i is k, its k largest $W_{i,(1)}, W_{i,(2)}, \ldots, W_{i,(k)}$ shall be big and the remaining $W_{i,(k+1)}, W_{i,(k+2)}, \ldots$ shall be very small. The outer summation ranges from 1 to $L-1$ since the contact number is less than L. Regrouping Eq. (3) by each $W_{i,(k)}$ we have,

$$\Omega_i = \sum_{k=1}^{L-1} g_{i,k} W_{i,(k)} \quad (4)$$

where $g_{i,k} = -\sum_{t=k}^{L-1} P_i(t) + \sum_{t=1}^{k-1} P_i(t)$. Notice that the coefficient $g_{i,k}$ for $W_{i,(k)}$ can be negative, which will lead the optimization problem to be unbounded, so empirically we add a constant β (0.2 by default) to each $g_{i,k}$ to make it positive. Figure 2 shows two examples. For node i, its degree is most likely to be either 1 or 3, so the coefficient for the largest interaction strength $W_{i,(1)}$ should be reduced. For node j, its degree is most likely to be zero, so all the coefficients are increased to drive its interaction strengths to zero.

The reweighted ℓ_1 regularized pseudo-likelihood function is defined as,

$$\min_W L(W) + \sum_{i=1}^{L} \Omega_i(W) \quad (5)$$

Where $L(W)$ is the negative log of Eq. (2) and Ω_i is a special ℓ_1 penalty defined in Eq. (4). To optimize (5), we use Alternating Direction Method of Multipliers (ADMM) [13] to separate the pseudo-likelihood function and the ℓ_1 penalty term. ADMM alternatively solves the following three sub-problems.

$$Z^{n+1} = argmin_Z L(Z) + \frac{\rho}{2}\|Z - W^n + U^n\|_2^2 \quad (6)$$

$$W^{n+1} = argmin_W \sum_{i=1}^{L} \Omega_i(W) + \frac{\rho}{2}\|Z^n - W + U^n\|_2^2 \quad (7)$$

$$U^{n+1} = U^n + Z^{n+1} - W^{n+1} \quad (8)$$

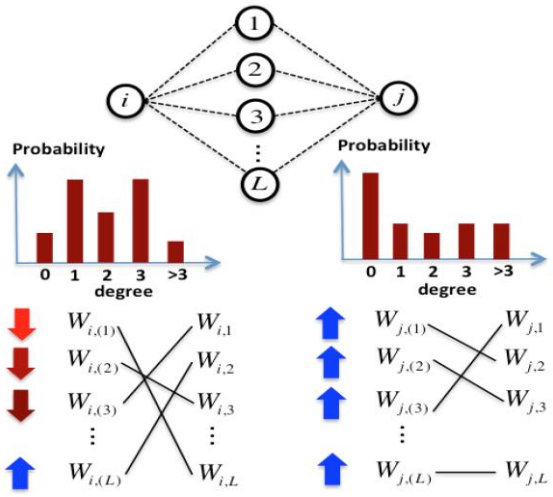

Figure 2. Node-specific degree-based ℓ_1 penalty. This figure shows how the regularizers are different based on the predicted degree distribution. The X-axis is the probability and the Y-axis is the degree. The colorful arrows represent the change of coefficient for the ℓ_1-norm. The warmer the color the smaller the value is. Down arrows represent negative value and up arrows represent positive value.

where $\rho > 0$ is a fixed step-size parameter (we used $\rho = 0.01$) and U is the dual variable passing information between sub-problems (6) and (7). Problem (6) can be solved using conjugate gradient decent. Since the order of $W_{i,*}$ for each i is unknown in problem (7), it is challenging to solve (7). We need to consider their order so that $g_{i,k}$ can be used to weight the k^{th} largest $W_{i,(k)}$. Let $M = Z + U$ and $g_{i,k} = g_{i,k}/\rho$. We may further divide problem (7) into L sub-problems; the i^{th} sub-problem is as follows.

$$min_{W_i} \frac{1}{2}\| W_i - |M_i| \|_2^2 + \Omega_i(W_i) \quad (9)$$

To solve problem (9), we need a mapping between original $\{W_{i,k}\}$ and $\{g_{i,k}\}$, as different mappings lead to different optimization problems. For a given mapping, we need to minimize the corresponding function subject to the constraints provided by the mapping. We want the mapping with the smallest optimal value. However, there are an exponential number of mappings between $\{W_{i,k}\}$ and $\{g_{i,k}\}$, which makes it impossible to enumerate all the possible mappings. To make it easy, we assume $W_{i,(k)} >$

$W_{i,(k+1)}$ for all k. That is, we only want to find a solution satisfying this condition. Next we will introduce how to use an Iterative Maximum Cost Bipartite Matching algorithm to solve the relaxation problem.

3.2.1 Iterative Maximum Cost Bipartite Matching Algorithm

Theorem 1. Let $W_{i,(1)} > W_{i,(1)} > \cdots > W_{i,(L-1)}$ be the ranking *of* $\{W_{i,k}\}$. *The optimal solution* $W^*_{i,(k)} = \sum_{a,b=1}^{20} |w^*_{i,(k)}(a,b)|$ *of problem (9) always has the form,*

$$\left|w^*_{i,(k)}(a,b)\right| = max\{\left|M_{i,(k)}(a,b)\right| - g_{i,k}, 0\} \quad (10)$$

Proof. By taking the sub-gradient with respect to each $\left|w_{i,(k)}(a,b)\right|$ and setting it to zero we obtain Eq. (10). If the optimal solution $|w^*_{i,(k)}(\mathrm{a,b})|$ of (9) does not satisfy Eq. (10), we can always decrease the objective function by adding or subtracting a small constant to (10) so that $W_{i,(k-1)} > W_{i,(k)} > W_{i,(k+1)}$ still holds. This contradicts with our assumption that $\left|w^*_{i,(k)}(a,b)\right|$ is the optimal solution. □

Based on Theorem 1, given a ranking of $\{W_{i,k}\}$, we can substitute $\{W_{i,k}\}$ of (9) by Eq. (10) to obtain the below equation.

$$\sum_{k=1}^{L-1}\sum_{a,b=1}^{20} M_{i,(k)}(a,b)^2 - w^*_{i,(k)}(a,b)^2 \quad (11)$$

Now we need to minimize (11). Notice that the summation of all the $M_{i,(k)}(a,b)^2$ is a constant, so minimizing (11) is equivalent to the following optimization problem.

$$max \sum_{k=1}^{L-1}\sum_{a,b=1}^{20} w^*_{i,(k)}(a,b)^2 \quad (12)$$

Substituting (9) into (12) and considering the constraints of the mapping, we have the following optimization problem.

$$max \sum_{k=1}^{L-1}\sum_{a,b=1}^{20} max\{\left|M_{i,(k)}(a,b)\right| - g_{i,k}, 0\}^2 \quad (13)$$
$$s.t. \quad \forall k \quad W^*_{i,(k)} > W^*_{i,(k+1)}$$

where $W^*_{i,(k)} = \sum_{a,b=1}^{20} \max\{\left|M_{i,(k)}(a,b)\right| - g_{i,k}, 0\}$. Note that in this problem we are looking for a one-to-one matching between $M_{i,(k)}$ and $g_{i,k}$, which can be modeled as an Integer Linear Problem (ILP) defined on variable $E = \{e_{k,l}\}$ as follows,

$$max \sum_{k,l} \theta_{k,l} e_{k,l} + \sum_{k\neq q,l} \theta_{k,q,l,l+1}\, e_{k,l} e_{q,l+1} \quad (14)$$
$$s.t. \quad \forall k,l \quad \sum_k e_{k,l} = 1\,, \sum_l e_{k,l} = 1$$

Here $e_{k,l} = 1$ if $M_{i,k}$ is assigned to $g_{i,l}$; otherwise 0. Let $\Lambda_{i,k,l}(a,b)$=$max\{|M_{i,k}(a,b)| - g(i,l), 0\}$, then each $\theta_{k,l}$ and $\theta_{k,q,l,l+1}$ can be computed as follows.

$$\theta_{k,l} = \sum_{a,b=1}^{20} \Lambda_{i,k,l}(a,b)^{\,2} \quad (15)$$

$$\theta_{k,q,l,l+1} = \begin{cases} 0 & \sum_{a,b=1}^{20} \Lambda_{i,k,l}(a,b) > \sum_{a,b=1}^{20} \Lambda_{i,q,l+1}(a,b) \} \\ -\infty & otherwise \end{cases}$$

Each $\theta_{k,l}$ reflects the preference of mapping $M_{i,k}$ to $g_{i,l}$ while $\theta_{k,q,l,l+1}$reflects the constraints to be satisfied. With these definitions, we can use the ADMM algorithm by introducing an auxiliary variable $v_{k,l}$ for each $e_{k,l}$ and solving (14) using the following iterative procedure,

$$V^{n+1} = argmin_V \sum_{q,l} C_{q,l} v_{q,l} \quad (16)$$

$$E^{n+1} = argmin_E \sum_{k,l} D_{k,l} e_{k,l} \quad (17)$$

$$\eta^{n+1} = \eta^{n+1} + (E^{n+1} - V^{n+1}) \quad (18)$$

Each $C_{q,l}$ and $D_{k,l}$ can be computed as,

$$C_{q,l} = -\eta_{q,l} + \left(\sum_{k\{k\neq q\}} \theta_{k,q,l-1,l}\, e_{k,l-1}\right) + \gamma e_{q,l} \quad (19)$$

$$D_{k,l} = \theta_{k,l} + \sum_{q\{q\neq k\}} \theta_{k,q,l,l+1} v_{q,l+1} + \eta_{k,l} + \gamma v_{k,l} \quad (20)$$

Here $\gamma > 0$ is a fixed step-size parameter (we used $\gamma = 0.5$) and η (we used 0.1) is the dual variable passing information between sub-problems (16) and (17). Both (16) and (17) can be viewed as a bipartite matching problem, which can be solved by the Hungarian algorithm [19].

Using the above algorithm (steps 16-18), we can find a permutation of $\{W_{i,k}\}$; $\left|w^*_{i,(k)}(a,b)\right|$ is then given by (10). Notice that in order to minimize (9), if $\left|w^*_{i,(k)}(a,b)\right| \neq 0$, then $w^*_{i,(k)}(a,b)$ and $M_{i,k}(a,b)$ should have the same sign. The final solution of $w^*_{i,(k)}(a,b)$ is therefore given by,

$$w^*_{i,(k)}(a,b) \quad (21)$$
$$= \begin{cases} M_{i,(k)}(a,b) - sign\left(M_{i,(k)}(a,b)\right) g_{i,k} & \left|M_{i,(k)}(a,b)\right| > g_{i,k} \\ 0 & otherwise \end{cases}$$

3.3 ESTIMATE NODE-SPECIFIC DEGREE DISTRIBUTION

Here we introduce how to predict the degree distribution of each node. Notice that in a protein sequentially-adjacent AAs must be close in the 3D space so their corresponding MRF nodes are connected. We introduce a concept called Partial Contact Number (PCN) denoted by $B_{i,j}$ for each node pair i and j, which is defined as the number of edges formed by node i with nodes $i+1, i+2, \ldots, j$ (if $j > i$) or nodes $i-1, i-2, \ldots, j$ (if $j < i$). Each $B_{i,j}$ has 15 labels indicating the degree from 0 to 13 and ≥ 14. We set the maximum degree to 14 since the contact number of an AA is upper bounded by a small constant. Since $B_{i,j}$ is correlated with its nearest neighbors $B_{i,j-1}$ and $B_{i,j+1}$, we apply a Conditional Neural Field (CNF) to predict $B_{i,j}$. Each node i is associated with two 2nd-order CNFs, as

shown in Figure 3. A protein with L nodes (AAs) has $2L$ different 2nd-order CNFs.

Let $F_{i,j}$ denote the feature vector extracted from two AAs i and j, we use one CNF is to estimate $P(B_{i,i+6}{\sim}B_{i,L}|F_{i,i+6}{\sim}F_{i,L})$ and the other for $P(B_{i,1}{\sim}B_{i,i-6}|F_{i,i+6}{\sim}F_{i,L})$. We ignore very short-range contacts (sequence distance<6) as they are less informative for structure prediction. We train CNFs by maximum likelihood. We use a ℓ_2 regularization to avoid over-fitting and 5-fold cross validation to choose the hyper parameters. Since CNF is non-convex, we train it starting from 5 different initial solutions and pick the best one. See [28] for more details of CNF.

After training the CNF models, we calculate the marginal probabilities $P(B_{i,j})$ using the standard forward-backward algorithm [20] independently on each CNF. Finally, we calculate the probability of node i having degree K as follows.

$$\sum_{K1+K2=K}(P(B_{i,1}=K1)+P(B_{i,L}=K2)) \quad (22)$$

4. EXPERIMENTS

4.1 TRAINING AND TEST DATA

We use a subset of the PDB25 dataset, generated by the PISCES server [34], to train and validate our CNF models. Any two proteins in this dataset share <25% sequence identity. In total we used 3118 proteins with length between 40 and 500, among which 3/4 are randomly chosen for training and the remaining 1/4 for validation. To test the performance, we evaluate our results on CASP10 [18] and CASP11 [26] datasets. We rule out short proteins with fewer than 70 amino acids since they have relatively low contact number prediction accuracy. This leads to 109 test proteins in the CASP10 set and 99 proteins in the CASP11 dataset. We use the CASP official domain boundary definition for each test protein. For each test protein, we run PSI-BLAST [3] with 5 iterations and E-value 0.001 to generate sequence profile, from which we extract $F_{i,j}$. All the native structures of our training and validation proteins are solved before CASP10 and CASP11 and do not share high sequence identify with the CASP test proteins.

4.2 EVALUATION CRITERIA AND PROGRAMS TO COMPARE

Depending on the sequence distance (i.e., the number of AAs between the two ends of a contact along the protein sequence), we divide contacts into 3 categories: [6,12) for short-range contacts, [12,24) for medium-range contacts and ≥ 24 for long-range contacts. Generally speaking, medium- and long-range contacts are more important for structure prediction, but more challenging to predict. We evaluate only top $L/5$, $L/10$, and $L/2$ predicted contacts. The accuracy is calculated as the percentage of the correctly predicted contacts. The ground truth is calculated from the experimental structure. When more predicted contacts are evaluated, the difference among methods becomes smaller since it is more likely to pick a native contact by chance. We compare our method to three other structure learning methods: PSICOV [15], plmDCA [9], and CCMpred [31]. PSICOV uses Graphical Lasso for contact prediction while plmDCA and CCMpred use maximum pseudo-likelihood with ℓ_2 regularization. These programs are run with their default parameters. When node-specific degree distribution is no used, our method is exactly the same as CCMpred, so we can calculate performance gain by examining the improvement of our method over CCMpred.

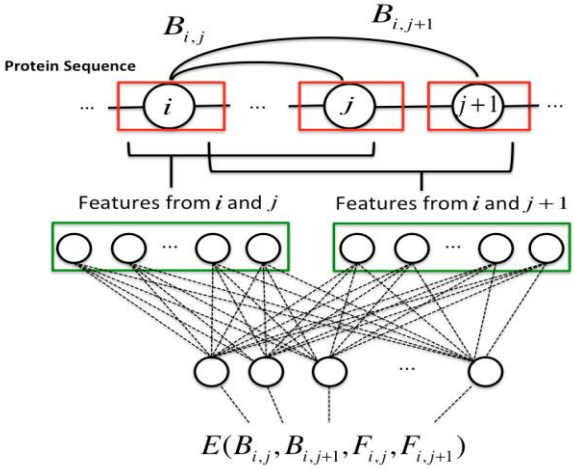

Figure 3. A Conditional Neural Field model for the prediction of Partial Contact Numbers.

4.3 PRE-PROCESSING AND POST-PROCESSING

We employ the same pre- and post-processing procedures as plmDCA and CCMpred to ensure our comparison with them is fair. To reduce the impact of redundant sequences, we apply the same sequence weighting method as plmDCA. In particular, duplicate sequences are removed and MSA columns containing more than 90% of gaps are deleted. The sequence is weighted using a threshold of 62% sequence identity. Similar to plmDCA and CCMpred, average-product correction (APC) [8] is applied to post-process predicted contacts.

4.4 PERFORMANCE

4.4.1 Overall Performance

As shown in Tables 1 and 2, on both CASP10 and CASP11 test proteins, our method significantly outperforms the others in terms of the accuracy of the top *L/10*, *L/5* and *L/2* predicted contacts. plmDCA and CCMpred achieve better

results than PSICOV because they drop the Gaussian distribution assumption. Our method differs from plmDCA and CCMpred in that we use a separate ℓ_1 regularization term for every pair of AAs instead of a universal regularization on all the AA pairs.

Table 1. Contact prediction accuracy on the 109 CASP10 targets

	Short-range			Medium-range			Long-range		
	L/10	L/5	L/2	L/10	L/5	L/2	L/10	L/5	L/2
Our Method	**0.32**	**0.33**	**0.19**	**0.39**	**0.34**	**0.28**	**0.37**	**0.34**	**0.25**
PSICOV	0.23	0.19	0.14	0.31	0.26	0.19	0.28	0.23	0.17
plmDCA	0.26	0.22	0.15	0.34	0.29	0.21	0.33	0.28	0.21
CCMpred	0.28	0.29	0.16	0.36	0.30	0.22	0.33	0.30	0.22

Table 2. Contact prediction accuracy on the 99 CASP11 targets

	Short-range			Medium-range			Long-range		
	L/10	L/5	L/2	L/10	L/5	L/2	L/10	L/5	L/2
Our method	**0.25**	**0.22**	**0.15**	**0.27**	**0.22**	**0.16**	**0.28**	**0.25**	**0.19**
PSICOV	0.19	0.14	0.11	0.20	0.16	0.12	0.20	0.17	0.13
plmDCA	0.19	0.14	0.11	0.21	0.17	0.13	0.23	0.23	0.17
CCMpred	0.21	0.17	0.12	0.24	0.19	0.13	0.24	0.22	0.17

4.4.2 Impact of Predicted Contact Number Distribution

First we evaluate the accuracy of contact number prediction. The contact number is predicted by picking the label with the maximum marginal probability computed by (22). The 15-label accuracy calculated on the CASP10 and CASP11 datasets are both 0.30 while random guess (i.e., predicting all the labels to be the one with the largest background probability) is 0.21. The average Pearson correlations between the ground truth and our prediction are 0.71 and 0.74, respectively. In addition, we can predict small- or large-valued contact number labels very accurately. These two properties help suppress the contacts of those AAs with very few contacts from showing up in the final prediction and thus, decrease the false positives.

Now we evaluate the impact of contact number prediction on individual contact prediction. We compare our method (i.e., predicted contact number used) with CCMpred (i.e., predicted contact number not used) in terms of the accuracy of the top *L/10* predicted long-range contacts. The top $L/10$ predicted contacts cover only a small number of AAs, so for each protein we only calculate the contact number accuracy on the AAs covered by the top $L/10$ predicted contacts. We group the test proteins into seven bins according to their accuracy of predicted contact number. For each bin we calculate the average accuracy improvement of individual contact prediction by our method over CCMpred. As shown in Figure 4, the improvement is positively correlated with the accuracy of contact number prediction on both CASP10 and CASP11 datasets. That is, the more accurately we can predict the contact number, the more performance gain can be obtained for individual contact prediction. In particular, when the accuracy of contact number prediction is low (<0.1), our method cannot improve individual contact prediction accuracy because the predicted contact number has too much noise. When the accuracy of contact number prediction is above 0.5, the performance gain from the predicted contact number information is large ($\geq$0.05). This implies that our method makes a good use of predicted contact number information.

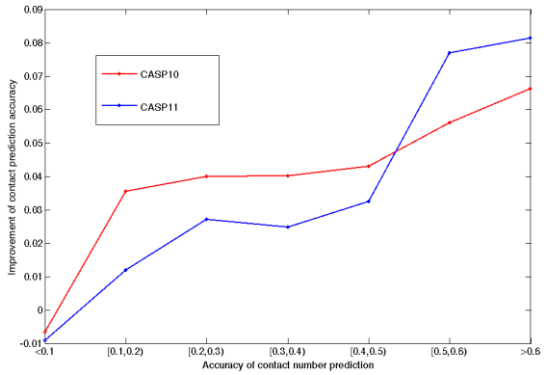

Figure 4. Relationship between top $L/10$ long-range contact prediction accuracy gain and the accuracy of contact number prediction. The accuracy gain is calculated as the performance difference between our method and CCMpred.

4.4.3 Case Study

Here we use two specific examples to further demonstrate the strength of our method. In particular, we want to study how the predicted contact number distribution helps with individual contact prediction. One example is a CASP10 target T0758 (PDB ID 4RM7). The other is a CASP11 target T0813 (PDB ID 4WJI). They have 366 and 302 AAs, respectively, and 9572 and 4177 similar sequences. As shown in Tables 3 and 4, our method significantly outperforms the others by at least 0.2 on the top $L/10$ long-range contact predictions. plmDCA and CCMpred have similar results since they use the same loss function. PSICOV yields relatively low performance on T0813, most likely attributing to the default sparsity setting being too aggressive.

Table 3. Contact prediction accuracy of T0758 (4RM7)

	Short-range			Medium-range			Long-range		
	L/10	L/5	L/2	L/10	L/5	L/2	L/10	L/5	L/2
Our method	**0.62**	**0.39**	**0.26**	**0.67**	**0.55**	**0.28**	**0.76**	**0.64**	**0.50**
PSICOV	0.50	0.31	0.19	0.55	0.41	0.19	0.50	0.47	0.40
plmDCA	0.44	0.30	0.20	0.61	0.42	0.26	0.56	0.56	0.45
CCMpred	0.42	0.32	0.19	0.64	0.56	0.30	0.53	0.48	0.45

Table 4. Contact prediction accuracy of T0813 (4WJI)

	Short-range			Medium-range			Long-range		
	L/10	L/5	L/2	L/10	L/5	L/2	L/10	L/5	L/2

Our method	**0.43**	**0.32**	**0.16**	**0.60**	**0.44**	**0.25**	**0.73**	**0.60**	**0.50**
PSICOV	0.30	0.20	0.11	0.43	0.28	0.19	0.50	0.40	0.31
plmDCA	0.37	0.27	0.14	0.57	0.40	0.20	0.53	0.47	0.40
CCMpred	0.37	0.28	0.13	0.53	0.42	0.23	0.47	0.50	0.45

Now for each target we examine two AA pairs not in contact. Our method can correctly predict that they are not in contact, but the other three methods predict they are in contact. Figures 5 and 6 show the predicted contact number distributions for the eight AAs of the two targets. Most of these AAs are predicted to have very few contacts. Especially the 137th AA of T0758 and the 268th AA of T0813; the predicted probability of the contact number being zero are both over 0.5, which means all the parameters associated with these two AAs shall be forced to zero.

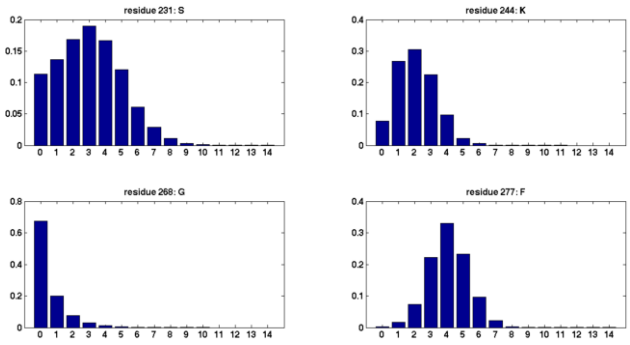

Figure 5. Predicted contact number distributions of 4 AAs of T0758 (4RM7) shown in Figure 7.

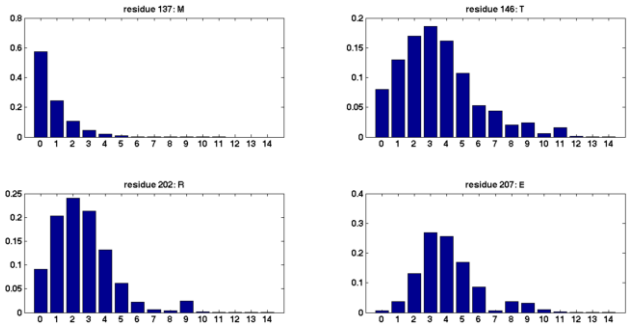

Figure 6. Predicted contact number distributions of 4 AAs of T0813 (4WJI) shown in Figure 8.

As shown in Figures 7 and 8, these two AAs are exposed at the protein surface and they do not form any long-range contacts. Similarly, for the other 6 AAs, the mass of predicted contact number distribution are all concentrated around 3 or 4, which means our model most likely can only allow 3 or 4 contacts for each AA. The top predicted contacts of these AAs are all short- and medium-range. That is why our method does not predict any long-range contacts for these AAs.

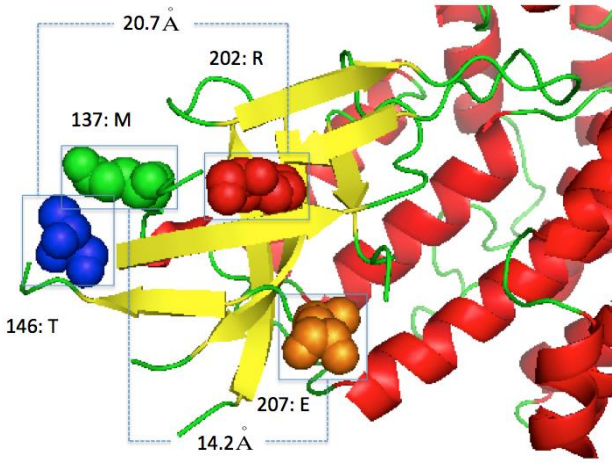

Figure 7. Two long-range false positives predicted by PSICOV, plmDCA and CCMpred for Target T0758 (4RM7): one false contact between the 146th AA and 202nd AA and the other between the 137th AA and 207th AA. Their true distances are 20.7 Å and 14.2 Å, respectively.

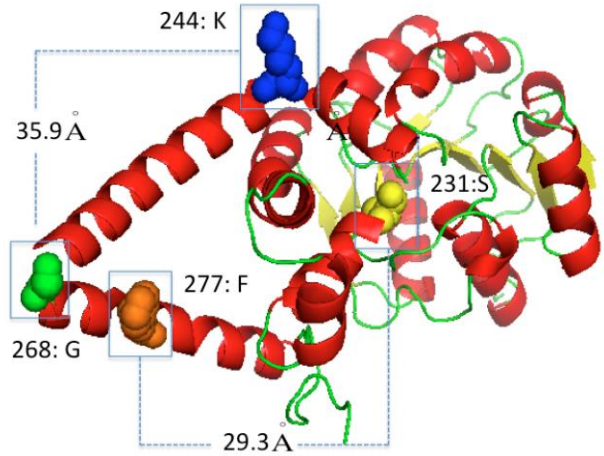

Figure 8. Two false positives predicted by PSICOV, plmDCA and CCMpred for T0813 (4WJI): one false contact between the 244th AA and 268th AA and the other between the 231st AA and 277th AA. Their true distances are 35.9 Å and 29.3 Å, respectively.

5. DISCUSSION AND FUTURE WORK

We have presented a new structure learning method that can make use of the predicted node-specific degree distribution to improve prediction accuracy of edges. The predicted degree distribution is used as a kind of soft topological constraints to restrict the solution space and avoid "unreasonable" predictions. Experimental results show that by using the degree distribution we can significantly improve protein contact prediction over current state-of-the-art structure learning methods.

From a computational perspective, our method provides a new framework to integrate orthogonal information into structure learning. That is, we first use supervised learning to learn node-specific local topological

constraints and then add it as a prior to learn the whole network structure. In many real-world applications, the connections of the graph are more or less influenced by the properties of nodes, so a node-specific degree distribution can be learned from local features without knowing the whole network structure. The contact number prediction is a very challenging supervised learning problem. In this work, we use multiple linear-chain graphical models to circumvent the difficulty of training and inference on loopy graphs. In the future, we will extend CNF by adding a deep learning module to further improve it.

For protein contact prediction, adding AA-specific topological constraint is only our first step. We are considering other AA- and segment-specific topological constraints, such as some geometric constraints imposed by a single secondary structure segment, two correlated secondary structure segments, or even the global structure of a protein.

Acknowledgements

We thank Payman Yadollahpour for useful discussions. This work is financially supported by the NIH grant R01GM089753 the NSF grant DBI-1262603 (to J.X.) and the NSF CAREER award CCF-1149811 (to J.X.). The authors are also grateful to the computational resources provided by the University of Chicago RCC.

References

1. Ahmed A, Song L, Xing EP (2008) Time-varying networks: Recovering temporally rewiring genetic networks during the life cycle of drosophila melanogaster. arXiv preprint arXiv:0901.0138
2. Albert R, Barabási A-L (2002) Statistical mechanics of complex networks. Reviews of modern physics 74:47
3. Altschul SF, Madden TL, Schäffer AA et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic acids research 25:3389-3402
4. Bach FR (2010) Structured sparsity-inducing norms through submodular functions. In: Advances in Neural Information Processing Systems. p 118-126
5. Barabási A-L, Albert R (1999) Emergence of scaling in random networks. science 286:509-512
6. Celik S, Logsdon B, Lee S-I (2014) Efficient Dimensionality Reduction for High-Dimensional Network Estimation. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). p 1953-1961
7. Defazio A, Caetano TS (2012) A convex formulation for learning scale-free networks via submodular relaxation. In: Advances in Neural Information Processing Systems. p 1250-1258
8. Dunn SD, Wahl LM, Gloor GB (2008) Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. Bioinformatics 24:333-340
9. Ekeberg M, Lövkvist C, Lan Y et al. (2013) Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. Physical Review E 87:012707
10. Fiori M, Musé P, Sapiro G (2012) Topology constraints in graphical models. In: Advances in Neural Information Processing Systems. p 791-799
11. Friedman J, Hastie T, Tibshirani R (2008) Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9:432-441
12. Hayat S, Elofsson A (2012) BOCTOPUS: improved topology prediction of transmembrane β barrel proteins. Bioinformatics 28:516-522
13. Hestenes MR (1969) Multiplier and gradient methods. Journal of optimization theory and applications 4:303-320
14. Jeong H, Mason SP, Barabási A-L et al. (2001) Lethality and centrality in protein networks. Nature 411:41-42
15. Jones DT, Buchan DW, Cozzetto D et al. (2012) PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics 28:184-190
16. Kamisetty H, Ovchinnikov S, Baker D (2013) Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. Proceedings of the National Academy of Sciences 110:15674-15679
17. Klepeis J, Floudas C (2003) ASTRO-FOLD: a combinatorial and global optimization framework for ab initio prediction of three-dimensional structures of proteins from the amino acid sequence. Biophysical Journal 85:2119-2146
18. Kryshtafovych A, Barbato A, Fidelis K et al. (2014) Assessment of the assessment: evaluation of the model quality estimates in CASP10. Proteins: Structure, Function, and Bioinformatics 82:112-126
19. Kuhn HW (1955) The Hungarian method for the assignment problem. Naval research logistics quarterly 2:83-97
20. Lafferty J, Mccallum A, Pereira FC (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
21. Liu Q, Ihler AT (2011) Learning scale free networks by reweighted l1 regularization. In: International Conference on Artificial Intelligence and Statistics. p 40-48
22. Ma J, Wang S, Wang Z et al. (2014) MRFalign: protein homology detection through alignment of Markov random fields. PLoS computational biology 10:e1003500

23. Ma J, Wang S, Xu J (2013) Protein contact prediction by joint evolutionary coupling analysis across multiple families. arXiv preprint arXiv:1312.2988
24. Marks DS, Colwell LJ, Sheridan R et al. (2011) Protein 3D structure computed from evolutionary sequence variation. PloS one 6:e28766
25. Michel M, Hayat S, Skwark MJ et al. (2014) PconsFold: improved contact predictions improve protein models. Bioinformatics 30:i482-i488
26. Moult J, Fidelis K, Kryshtafovych A et al. (2014) Critical assessment of methods of protein structure prediction (CASP)—round x. Proteins: Structure, Function, and Bioinformatics 82:1-6
27. Nugent T, Jones DT (2012) Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis. Proceedings of the National Academy of Sciences 109:E1540-E1547
28. Peng J, Bo L, Xu J (2009) Conditional neural fields. In: Advances in neural information processing systems. p 1419-1427
29. Ravikumar P, Wainwright MJ, Lafferty JD (2010) High-dimensional Ising model selection using ℓ1-regularized logistic regression. The Annals of Statistics 38:1287-1319
30. Savojardo C, Fariselli P, Martelli PL et al. (2013) BCov: a method for predicting β-sheet topology using sparse inverse covariance estimation and integer programming. Bioinformatics:btt555
31. Seemayer S, Gruber M, Söding J (2014) CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. Bioinformatics 30:3128-3130
32. Sharan R, Ulitsky I, Shamir R (2007) Network‐based prediction of protein function. Molecular systems biology 3
33. Tan KM, London P, Mohan K et al. (2014) Learning graphical models with hubs. The Journal of Machine Learning Research 15:3297-3331
34. Wang G, Dunbrack RL (2003) PISCES: a protein sequence culling server. Bioinformatics 19:1589-1591
35. Wang Z, Xu J (2013) Predicting protein contact map using evolutionary and physical constraints by integer programming. Bioinformatics 29:i266-i273
36. Wei Z, Li H (2007) A Markov random field model for network-based analysis of genomic data. Bioinformatics 23:1537-1544
37. Xu J, Com G Learning Scale-Free Networks by Dynamic Node-Specific Degree Prior.