STRUCTURED SPARSITY LEARNING FOR COEVOLUTION-BASED PROTEIN CONTACT PREDICTION

A Thesis

by

DI WU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	I-Hong Hou
Co-Chair of Committee,	Yang Shen
Committee Members,	Jiang Hu
	Xia Hu
Head of Department,	Miroslav Begovic

December 2018

Major Subject: Computer Engineering

ABSTRACT

Residue coevolution refers to a biological assumption that residue pairs covary during evolution if they form a contact within a protein or across a protein-protein interface. Under this assumption, such covariance can be used to predict residue contacts within or between protein sequences. The increasing availability of protein sequence data allows for wider applicability and also demand more accurate approaches.

Current methods are modeling sequence data in Markov random fields and use maximum likelihood estimations to infer residue contacts. They mainly target the accuracy of contact prediction under the promise that more accurate 2D contact prediction helps to get a better 3D structure. This is correct but not the whole picture since patterns of predicted 2D contacts also play a significant impact on 3D structure reconstruction. For example, contacts between long-distance residue pairs in general help more than adjacent residue pairs do. Moreover, current methods always get predictions that focus on certain area.

To directly target 3D structure predictions, we introduce a new method which exploits more types of data, such as secondary structure data and folds type information, to characterize the desired sparsity patterns of contact prediction in a biologically meaningful way. It then uses multiple structured sparsity regularization models, including group LASSO and group dispersive sparsity, to enforce such sparsity patterns. This method benefits from the consideration and promotion of structured sparsity, which contributes to improvement of 3D structure prediction.

DEDICATION

To my parents, my advisor Dr. Shen and Dr. Hou, and all my friends who have supported and helped me during this trip.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a thesis committee consisting of Professor Yang Shen, Professor I-Hong Hou and Professor Jiang Hu of the Department of Electrical and Computer Engineering and Professor Xia Hu of the Department of Computer Science and Engineering. Part of the work was supported by the National Institute of General Medical Sciences of the National Institutes of Health (R35GM124952 to YS). Computing time was in part provided by the Texas A&M High Performance Research Computing.

All other works conducted for the thesis were completed by the student independently.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Background

Protein is an important component of every cell in the body and support large range of function as enzyme, antibody and etc. It is made up of smaller units, the amino acids, which are attached to one another to form a long chain. There are 20 different types of standard amino acids and they can be combined in a linear polypeptide chain to make a protein.

A long protein chain can be described using three levels of structures: primary, secondary, and tertiary structures. The simplest level of protein structure is the primary structure, which can be simply regraded as the sequence of amino acids[1][2]. Those amino acids assemble in particular order to make up of polypeptide chains. The sequence of a protein is determined by DNA of the gene that encodes the protein. Secondary structure refers to regions in which protein chains are organized into local 3D structures such as alpha-helices, beta-pleated sheets and random coil[3], shown as Figure 1.1. Those secondary structures are stabilized by hydrogen bonds within individual structures. In an alpha-helix, the protein chain is coiled like a spring and each turn of this loose spring always has 3 complete amino acid residues and also two atoms from above and below. For a beta sheet, the chains are folded as parallel lines alongside each other. There are more detailed classification methods for secondary structures but we can just focus on this simple one. No matter in any methods, random coil is always used to present secondary structures which are not in any of the above conformations. Based on the secondary structure, protein folds itself into a 3D shape and holds this connection between side chains by several types of interactions, such as ionic interaction, hydrogen bond, sulphur bridge and van der Waals dispersion forces. The unique 3-dimensional structure and its specific function are determined by its amino acid sequence. Distant relationships between proteins sometimes can be revealed by the secondary structure[4].

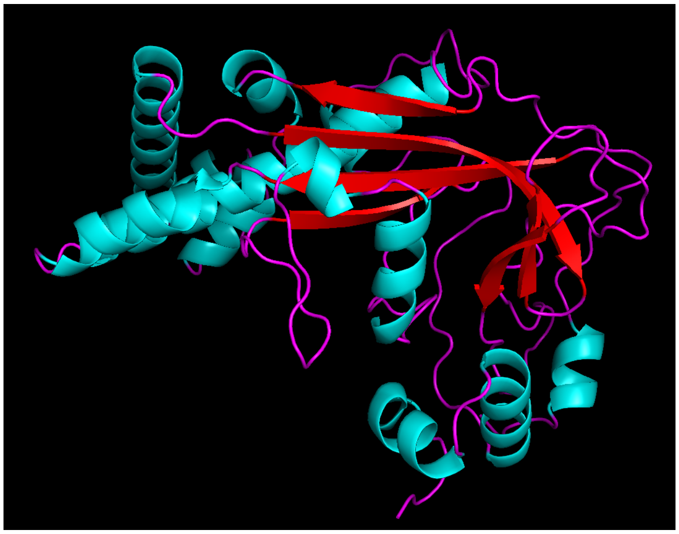

Figure 1.1: A cartoon representation of a 3D protein structure (Protein Data Bank ID: 1MSK) that contains alpha-helices (in cyan), beta-sheets (in red) and coils (in magenta)

Therefore, learning structures of proteins is an important way to understand their functions but structure detection is so time-consuming and it's hard to imply it extensively. The protein structure prediction technique rises in response to that demand. With the growth of the sequence databases, comparative modeling methods which are based on homologs of known structures have become increasingly powerful compared to de novo models[5]. However, with few protein families which can not be gotten homologs accurately, it will lead to a predicament for many methods.

Formation of 3D structure in proteins is dependent on the establishment of close through space contacts and inter residue interactions impose constrains on evolutionary dynamics. Therefore, mutations at contact pairs are expected to be coupled in evolutionary procedure. For this reason, there has been long-standing interest in the prediction of residue-residue contacts based on residue coevolution that can be inferred from multiple sequence alignment (MSA). With the increase of known protein sequences in recent years, such approaches have demonstrated considerable promise, for example, Direct Coupling Analysis[6] and Protein Sparse InverseCOVariance (PSICOV)[7]. GREMLIN gets a further step achievement of better predictions robustness by incorporating prior information on pairs likely to be in contact[5]. Recent studies[8][9] also have shown that the predicted contacts are sufficiently accurate to predict the 3D structures of proteins with deep alignments.

1.2 Challenge and Idea

Even with high accuracy of contact prediction, accurate 3D structure prediction is not always guaranteed when the data is limited. As we known, short-range residue pairs, close neighbors in 1D sequence, are more likely to form contacts since they are closely separated by a few covalent bonds, and they indeed get higher scores in the final result. Moreover, it can not avoid a situation that a lot of predictions huddle in several areas, which does not help global structure prediction. Those predictions provides more information about intra instead of inter secondary structure elements, but the latter ones are more important when we build 3D global structures. Furthermore, we are also interested in how the changes of pattern and contact prediction accuracy affect building 3D structure. In addition, we try to compare the effect of sparsity within and cross groups for 3D structure reconstruction. Based on the correlation of sparsity and 3D structure result, we want to understand if and how sparsity affects the final structure. However, the method of sparsity measurement is also hard to determine. We need first understand what is ideal distribution and then Figure out which one is deterministic between the group level sparsity and the whole matrix sparsity.

In this study, we propose a method which focuses on integrating structural data with sequence information to enforce predictions to possess 2D contact patterns that can improve 3D structure reconstruction by providing more contact information across secondary structures. By embedding with distance information, our method shows that how the sparsity level will change in different areas. The secondary structures data helps to further spread predictions toward desired sparsity pattern and we will use cross validation to train the hyper-parameters to balance various patterns and improve 3D structure prediction. Importantly, we would like to answer three significant questions:

- Do we improve dispersion of prediciton result by introducing various structured sparsity regularization model?

- Does 2D contact prediction with improved dispersion patterns help improve 3D structure reconstruction?

- What kind of sparsity pattern and corresponding regularization models for 2D contacts can optimize 3D structure reconstruction?

1.3 Organization of Thesis

- Section I Introduction: This section introduces background knowledge about protein and its different levels of structures. Then it states what problems we are facing, where these problems come from and why it is worthwhile putting more efforts on. Moreover, a detailed discussion about the current research on this field is presented in this section.

- Section II Methods: This section focuses more on the detailed methodologies and utilized materials. It includes the motivation and path of developing our methods.

- Section III Result and Analysis: We include results from each step of the methods here. Then based on all those results, we will provide the corresponding answers to questions proposed in the first section.

- Section IV Conclusion and Future Directions: We will conclude what we have found derived from the results and discussion above and also propose some points for the next step research.

2. METHODS

2.1 Contact definitions

Residue-residue contacts (or simply "contacts") in protein 3D structures are pairs of spatially close residues. A 3D structure of a protein is expressed as three dimensions coordinates of amino acid atoms in the form of PDB (Protein Data Bank) file[10]. Contact is always defined by a threshold based on its physical distance. Realizing that the contacting residues which are far apart in the protein sequence but close together in the 3-D space are important for protein folding[11]. Generally, we will say that a pair of amino acids are in contact if the distance between specific atoms, mostly using alpha carbon or beta carbon, is less than than 8Å apart[12]. Also, a short-range contact is defined between residues i and j with $5 \leq |i-j| \leq 23$ where i and j are their sequence indices, respectively. If the index difference is less than 5, the corresponding contact is regraded trivial thus ignored. There are three main types of secondary structure as α-helix, β-sheet and Coil. By combining with those data, we define six groups of inter-SSE contacts: three of them formed between residue pairs from the same type of SSE and another three residue pairs from two different types of SSE's. Protein contacts contain key information for the understanding of protein structure and function, so contact prediction is an important problem[13].

2.2 Generation of MSA and secondary structures

Generating input alignments and secondary structure data is a crucial step for further statistical inference. To ensure the consistency with CCMpred, we generate multiple aligned sequences by HHBlits according to following parameters[14][15]:

- HHBlits version: 3.0-beta.3 (14-07-2017)
- Database: Uniprot20_2016_02
- Iterations: 3
- E-Value cutoff: 10^{-3}

- Minimum Coverage with master sequence: 60%

- Maximum Pairwise Sequence Identity: 90%

- Maxfilt: 100000

Secondary structure prediction usually has two different types, Q3 or Q8, which refers to 3-state or 8-state secondary structure.[16]. For our case, the 3-state definition is better for further grouping since its prediction has a higher accuracy and it is also enough for structured contact patterns. We generate secondary structure data by SSpro, which combines machine learning methods with evolutionary information and fragment libraries extracted from the Protein Data Band (PDB) and maintain a high accuracy around 80% [17].

2.3 Dataset

To test our method and compare it with a state-of-the-art method (CCMPred), we use a data set comprised of 41 proteins[14] which are selected from Astral 2.05 database (PDB SCOPe 40% ID)[18]. This dataset contains at least one member of 1,668 distinct SCOP folds. To eliminate the redundancy and ensure the diversity of our test set, we distribute selected cases to four different classes of folds which are all α, all β, $\alpha + \beta$ and α/β. In addition, those sequences are also evenly spread in terms of length ranging from 54 to 504 residues. Care was also taken to select proteins with single-chain, well resolved (resolution less than 2.5 Å) and belonging to different Pfam families[14].

In addition, we should notice that this dataset does not provide the whole sequence for each protein. Instead, it uses domain segment of the sequence to replace the original one as a protein chain can have multiple spatially-separated and functionally-independent domains that could adopt different folds.

2.4 Machine Learning Model

The objective function to optimize parameters of our machine learning model is comprised of two parts: pseudo log-likelihood $pll(v, w|D)$ modeled by a Markov random field and regularization

$R(v, w)$:

$$O(v, w) = pll(v, w|D) - R(v, w)$$

By maximizing the objective function, we can determine the parameters v and w and then to predict the probability of contact for each pair of positions. More details are discussed below.

2.4.1 Overview of Computational Method

Our method uses a generative model based on GREMLIN[5] with the following function form:

$$P(X = x) = \frac{1}{Z} exp(\sum_{i=1}^{L} [v_i(x_i) + \sum_{j>i}^{L} w_{i,j}(x_i, x_j)])$$

Learned from MSA of proteins, the variables X_i represent the amino acid composition at index i, v_i is a set of parameters of individual propensity at position i for each amino acid, and $w_{i,j}$ means 21*21 parameter matrix modeling the statistical coupling between position i and j. The Z is partition function to normalize the sum of probabilities to 1. Given a set of aligned protein sequences, if we can match the distribution of this model to observation exactly, the learned parameters tend to give the true result. However, it's computationally intractable to get the exact solution. GREMLIN uses log-pseudo-likelihood to get an approximation of v, w, which is expressed below:

$$pll(v, w|D) = \sum_{n=1}^{N} \sum_{i=1}^{L} logP(x_i^n | x_{-i}^n, v, w)$$

x_i^n is the probability of observed amino acid at position i in the n^{th} sequence in the context of amino acid at other positions x_{-i}^n and it depends on parameters v, w as following:

$$P(x_i^n | x_{-i}^n, v, w) = \frac{1}{Z_i} exp(v_i(x_i^n) + \sum_{j=1, j \neq i}^{L} w_{i,j}(x_i^n, x_j^n))$$

The pseudo-likelihood model uses the local partition function Z_i which is trivial and makes the function computationally tractable. Moreover, it's easy to optimize due to its concavity in v, w.

2.4.2 Group LASSO

Our final target is pursuing a sparsity across the whole matrix instead of just part of groups. However,group lasso achieves group sparsity which is opposite to our rationale. This part will show how the group LASSO method controls the pattern and how its pattern affect the 3D reconstruction as is a comparative study. It is not a method we propose for the desired dispersion.

Regularization helps to avoid over-fitting of model parameters. Originally, GREMLIN uses $l2$-norm for parameters w[5], but later in CCMpred paper, they modified it to $l2$-square since it will make no significant difference in the results and is easier to optimize[19]. The corresponding regularization function is shown as:

$$R(v, w) = \lambda_v ||v||_2^2 + \sum_{i,j} \lambda_w ||w_{i,j}||_2^2$$

LASSO, which is short for least absolute shrinkage and selection operator, is a regression analysis method for variable selection and regularization in order to promote prediction accuracy and interpretability of statistical models. It was developed by Robert Tibshirani in 1996 [20]. The basic LASSO was shown below:

$$(\hat{\alpha}, \hat{\beta}) = arg\ min \left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\}, \quad subject\ to \sum_j |\beta_j| \leq t.$$

in which β is parameters to be estimated and t is a prespecified free parameter to restrain the amount of regularization. Due to the $l1$-norm restraint above, LASSO can help to find solution with few nonzero entries in β, which can not be done by ridge regression. We are able to apply sparse structure to our method by simple LASSO, but this kind of sparsity will be forced 'randomly'. In our case, short distance residue pairs are more likely to have contact and they will force more coefficients of long distance area to be zeros. However, we hope that each area has similar level of sparsity.

Then we introduce group LASSO to apply sparsity in group level[21]:

$$min_{\beta}\frac{1}{2}\left\|y-\sum_{l=1}^{m}X^{(l)}\beta^{(l)}\right\|_2^2+\lambda\sum_{l=1}^{m}\sqrt{p_l}\|\beta^{(l)}\|_2$$

where $X^{(l)}$ is the submatrix of X with columns corresponding to the predictors in group l, $\beta^{(l)}$ the coefficient vector of that group. For our case, the matrix can be grouped by SSE prior since each type of secondary structure can be regarded as a unity to interact with other parts. To spread the contact prediction distribution, we apply mixed group LASSO to the regularization part as:

$$R(v,w)=\lambda_v||v||_2^2+\lambda_w[\sum_{k=1}^{K}\alpha_k(\sum_{g_k=1}^{G_k}\beta_{g_k}\sqrt{\rho_{g_k}}||w^{(g_k)}||_2)]$$

$$\sum_k\alpha_k=1,\sum_{g_k}\beta_{g_k}=1$$

where k means the k^{th} grouping strategy (k=1,2, $\cdots$, K) and g_k indicates group index in the k-th grouping strategy. Since a larger group is regularized less to be sparse, we use ρ_{g_k}, which is the size of group, to balance this effect so that each group will have similar probability to maintain a same sparsity level. The hyper parameters α_k and β_{g_k} are also used to balance the relative weight for different grouping strategies and different groups defined by the same grouping strategy. The sum of each equal to one maintain the convexity of function. They can be determined from prior knowledge or trained through data.

2.4.3 Group-Dispersive Sparsity

As we discussed above, group LASSO does provide some sparsity on group level, but it also treated a whole group as a variable and used only a few of them. It means that we may loss the information for the whole group. In certain cases, some of groups are supposed to be no contact. This situation is seldom for real protein but always happens for group LASSO. In our application, it is desirable to also enforce sparsity within groups and select at least one variable per group. We thus consider an extension of group LASSO, dispersive sparsity models[22]:

$$\hat{\beta} = argmin\|y - X\beta\|_2^2 + \frac{\lambda}{2}\sum_{g\in\mathcal{G}}\|\beta_g\|_1^2$$

For each group g, the penalty takes $l1$-norm of parameters, β_g, and then take square for the vector of norms. If all coefficients are in the same group, the group-dispersive sparsity method is actually equal to squaring the $l1$-norm. In contract, if each coefficient is grouped by itself, this formula will turn to be ridge regression. Generally speaking, it performs selection within groups by applying LASSO selection to each group separately and it also protects the whole group from going to exact zero as ridge regression. In other words, it uses $l1$-norm within groups to enforce sparsity and uses $l2$-norm between groups to make sure the density, which is opposite with group LASSO.

In our case, the formula is:

$$R(v,w) = \lambda_v||v||_2^2 + \lambda_w\left[\sum_{k=1}^{K}\alpha_k\left(\sum_{g_k=1}^{G_k}\beta_{g_k}\sqrt{\rho_{g_k}}||w^{(g_k)}||_1^2\right)\right]$$

If we only introduce secondary structure as the grouping criteria and treat interactions within one type of structure and between two types of structures separately, then our matrix can be grouped based on secondary structure and classified into 9 different types of group:

$$R(v,w) = \lambda_v||v||_2^2 + \lambda_w\left((1+\alpha_k)\sum_{g_1=1}^{3}\beta_{g_1}\sqrt{\rho_{g_1}}||w^{(g_1)}||_1^2 + (1-\alpha_k)\sum_{g_2=1}^{6}\beta_{g_2}\sqrt{\rho_{g_2}}||w^{(g_2)}||_1^2\right)$$

where g1 means intra groups which are for interactions within each type of secondary structure and g2 means inter groups which contain interactions between two types of structures. Moreover, we use $1 \pm \alpha_k$ instead of just α_k, which α_k ranges from -1 to 1, to enhance or recede one side of groups not just balance them. To be noticed, $l1$-norm is a non-smooth function so the optimization problem has no closed form solution. Therefore, when we try to solve it by modifying a conjugate gradient method, we use subgradient instead of gradient to calculate in each step.

2.5 Evaluation Method

To judge the quality of our result, we need to use some criterion for each step.

2.5.1 Dispersion Level

Before assessing dispersion level, it is important to define the ideal dispersion pattern. As our discussion above, our focus is not limited to the accuracy for the intermediate result. Instead, we want to analyze the relation of dispersion and final result. Therefore, as a stepping stone, we can set the uniform distribution as the reference. As we known, uniform distribution has equal possibility for each position, which means it spreads the "predictions" randomly. A more informative prior distribution of contact dispersion pattern than uniform distribution can be introduced later by learning from known structure data.

Now we have the reference for our 2D result and then we need to determine how to judge its difference. The simplest way is Manhattan distance. The problem is that the range of Manhattan distance is uncertain, so the value will be affected by the size of matrix and number of samples. We can't compare the similarity over difference cases. Therefore, we introduce Jensen-Shannon Divergence (JSD). The JSD is a symmetrized and smoothed measurement and it utilizes the Kullback-Leibler Divergence $D(P||D)$ to calculate the similarity. It is defined as below:

$$JSD(P\|Q) = \frac{1}{2}D(P\|M) + \frac{1}{2}D(Q\|M)$$

where $M = \frac{1}{2}(P+Q)$. The Jensen-Shannon divergence score ranges from 0 to 1 for two probability distributions if we use the base 2 logarithm[23].

$$0 \leq JSD(P,Q) \leq 1$$

Then we use Jensen-Shannon distance, which is defined by the square root of the JSD, and it provides a legitimate distance metric. We will use it as the criteria of dispersion level.

2.5.2 3D Structure Reconstruction and Assessment

For 3D structure assessment, the reference is easy to determine, which is the PDB file. Protein Data Bank (PDB) format is a standard for files to contain atomic coordinates. It is generated based on experiment result so that it can be regarded as the ground truth.

From the 2D contact to 3D structure, we use CoinFold as the predictor[24]. It is original a web server for protein contact prediction and contact-assisted de novo structure prediction. They use Group Graphical Lasso (GGL) to predict contacts by joint EC analysis[25] and improve the accuracy by implementing consistency of co-evolution pattern . Then the CoinFold also generates the secondary structure by DeepCNF[26] and then 3D structure by feeding those two to Crystallography & NMR System (CNS) package without the template[27][28]. For our case, we only need use the last step of it to predict the 3D structure. Also, to maintain the consistency, we will provide our 2D structure result to CoinFold for reconstruction.

To compare two protein structures, Template Modeling Score (TM-score) is a good measurement of similarity. It is more accurate than RMSD and GDT when we are trying to measure the protein structures quality. The TM-score indicates the difference by score between (0,1], where 1 means full accord. Generally, we take score less than 0.2 as a randomly chosen unrelated proteins and larger than 0.5 as a much the same structure. Proteins with TM-score equal to 0.5 have a 37% posterior probability in the same CATH topology family and 13% in the same SCOP fold family according to a quantitative study. [29].

The TM-score is calculated as below:

$$\text{TM-score} = max\left[\frac{1}{L_{target}}\sum_{i}^{L_{aligned}}\frac{1}{1+\left(\frac{d_i}{d_0(L_{target})}\right)^2}\right]$$

where L_{target} is the length of target protein and $L_{aligned}$ means the aligned region. d_i is the distance between the ith residue pair and also is used to normalize distance is:

$$d_0(L_{target}) = 1.24\sqrt[3]{L_{target}-15}-1.8$$

2.6 Pipeline

As shown in Figure 2.1, the main idea is to train hyper-parameters first by training set and then use test set to assess the model.

In the beginning, based on the dataset, we select 2 cases from each all α, $\alpha + \beta$, all β and α/β type and make sure that two cases in one type have divergent sequence length to compose the test set. Before training and testing, we use HHBlits and SSpro to generate MSA and SSE data respectively. Then we set α_k as hyper-parameter and train it from -1 to 1. Later, we also try to train λ_w since this hyper-parameter is optimized by CCMpred but may be not suitable for our model. In terms of generating 3D structure from contact prediction, we use the CoinFold as standard generation approach for both baseline method and our algorithm. To evaluate and compare results, we need to define an assessment metric. Since the 3D structure is our target, all of our validation process will base on the dissimilarity between predicted 3D structure and ground truth PDB file. Specifically, we use TM-score as the metric to measure the quality of the 3D structure prediction. It belongs to [0,1], which indicates from the worst similarity to the best respectively.

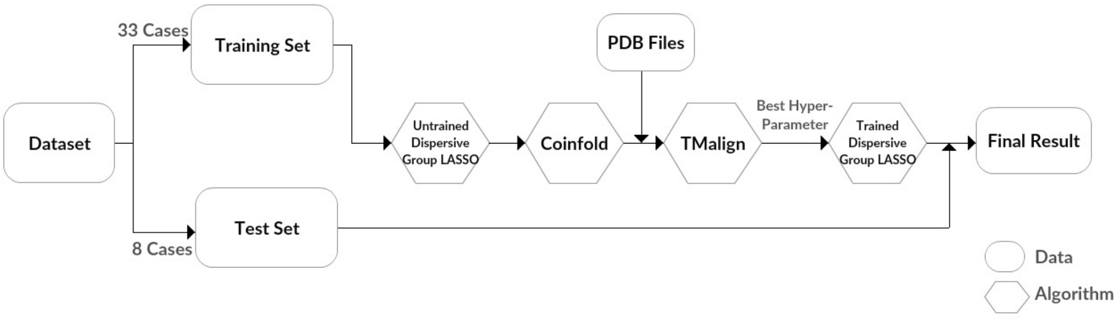

Figure 2.1: Pipeline

3. RESULT AND ANALYSIS

3.1 Contact Prediction of Group LASSO

To balance the weight across groups and make them have equal chance of prediction, we use $\sqrt{\rho_g}/\sqrt{w_{size}}$ to replace $\sqrt{\rho_g}$, where w_{size} is the size of whole group. Then our group LASSO formula is:

$$R(v,w) = \lambda_v||v||_2^2 + \lambda_w \left((1+\alpha_k) \sum_{g_1=1}^{3} \beta_{g_1} \frac{\sqrt{\rho_{g_1}}}{\sqrt{w_{size}}} ||w^{(g_1)}||_1^2 + (1-\alpha_k) \sum_{g_2=1}^{6} \beta_{g_2} \frac{\sqrt{\rho_{g_1}}}{\sqrt{w_{size}}} ||w^{(g_2)}||_1^2 \right)$$

3.1.1 Pattern Analysis

Before learning hyper parameters and parameters of our model, we are going to show how our method will affect the distribution of predictions by applying different grouping strategies.

As we mentioned before, current methods mainly focus on accuracy. Sometimes their results will ignore some area which may provide contact information across SSEs that can be crucial to structure predictions. Therefore, we hope that we can enforce predictions to appear in certain area of matrix. Our first step is to compare contact maps resulting from different models.

Table 3.1 shows how the regularization function will affect the pattern of predictions. All groups are based on the distance in sequence index for residue pairs. The group LASSO method uses grouping strategies which divide predictions into short, medium and long-range groups whose residue pairs' distances are less than 7, 7 to 12 and larger than 12 respectively. The first three methods use the same parameters and the fourth one modified the β for medium-range and long-range group from $\frac{1}{3}$ to $\frac{5}{12}$ and from $\frac{1}{3}$ to $\frac{1}{4}$,respectively. Since the short range is not important for building 3D global structures, we will only discuss medium and long range groups. As shown in Table 3.1, result of CCMpred, which uses $l2$-norm square, is similar with $l2$-norm CCMpred, just as what we mentioned before. And for group LASSO method, we notice that a lot of predictions are "forced" to the medium group and the percentage of predictions in each group are more balanced than original method. The fourth row shows that when we modify parameters, we can force the pattern to change in different directions.

Method	Medium Range Prediction	Long Range Prediction
CCMpred	33(2.55%)	136(1.09%)
$l2$-norm CCMpred	35(2.71%)	134(1.08%)
Group LASSO	75(5.80%)	94(0.76%)
Group LASSO with modified parameters	27(2.09%)	142(1.14%)

Table 3.1: Prediction distribution of methods in different ranges (UniProt ID:P0A6A3)

According to Figure 3.1, we see the changes of prediction distribution. Numbers in brackets are sparsity level of each group (measured by the number of predictions in that group over the group size). From $l2$-norm square to $l2$-norm, they look pretty similar. However, it shows a great difference when we introduce the group LASSO to our regularization function. It looks like that some points are pushed into medium groups. After enlargement of medium range, that effect is weaker than before but it still balance the distribution between groups.

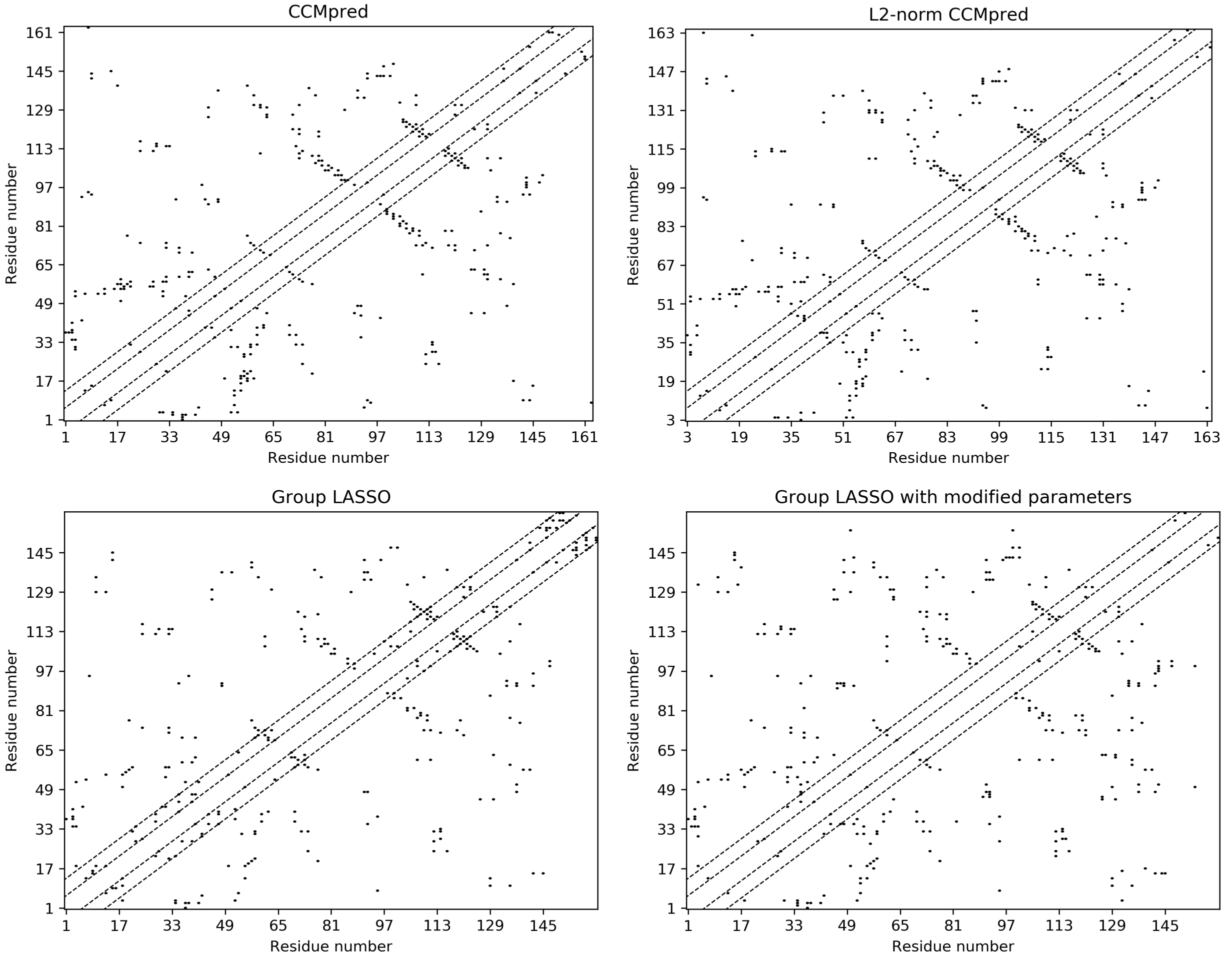

Figure 3.1: Contact maps of top L prediction from four methods

3.1.2 Precision Result

Then we train the hyper-parameter β based on our training set with secondary structure prior and use the best one to run for the test set. The result is shown as Table 3.2. The worst case means that it got the lowest TM-score for both method. The PDB ID of this protein is 1BMG and its length is 98. According to the result, we can see that the group LASSO is really bad at 3D structure reconstruction. The reason may be its abandon of certain groups. However, we can see that the original method also failed to predict the 1BMG. The sequence is not long and also its structure only contains beta sheet and coils. It may be because of the quality of alignments of protein sequence but it also reflects the robustness of methods.

	Average TM-score	Worst Case TM-score
Group LASSO Method	0.37	0.25
Original Method	0.52	0.31

Table 3.2: Test result for our method and the original method

Then let's go back to see the contact map result of this case as shown in Figure 3.2. The left one is reference structure of our method and the right one is for original method. Red nodes mean false positive matches, black means true positive and grey ones are false negative. In each contact map, we have already removed short distance results and choose the top 2L results to plot, where L indicates the length of protein sequence. From the reference map of our method, we notice that it does well on avoiding mismatches at two corners of diagonal but it still loses more points on certain areas which are totally missed or too dense. Compared with original pattern, group LASSO helps to enhance the weight on certain groups so that we get pattern there. However, it is too hard to control and determine where should have contacts. Even though we know which group should be chosen, density in few groups does not help to construct 3D structure of whole protein.

From Figure 3.3, we can see that the original method keeps a relatively higher precision from the beginning to the end. Group LASSO does a fine job at beginning but drops too fast from L/2.

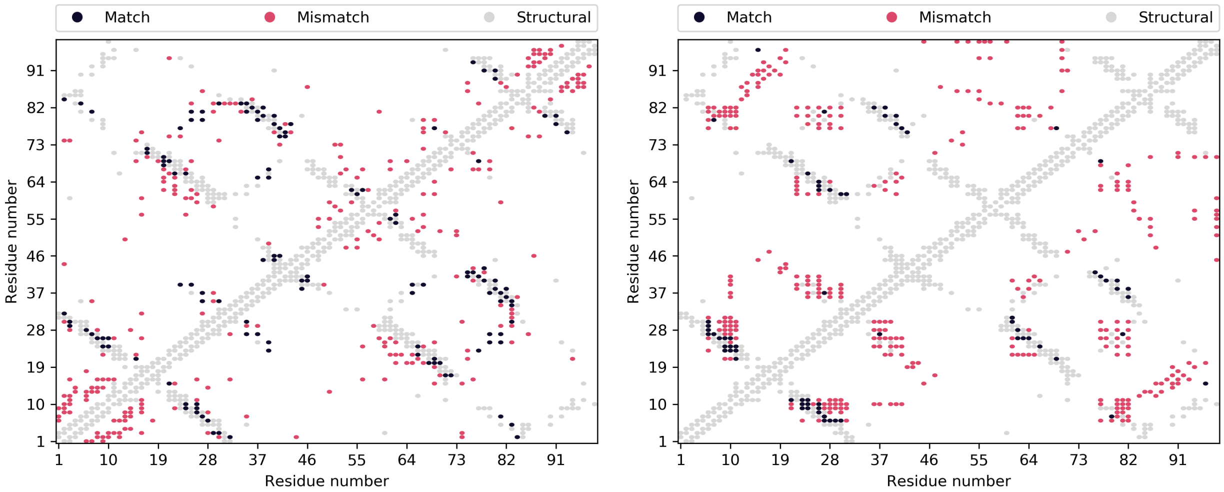

Figure 3.2: Reference contact map of group LASSO method and the original method

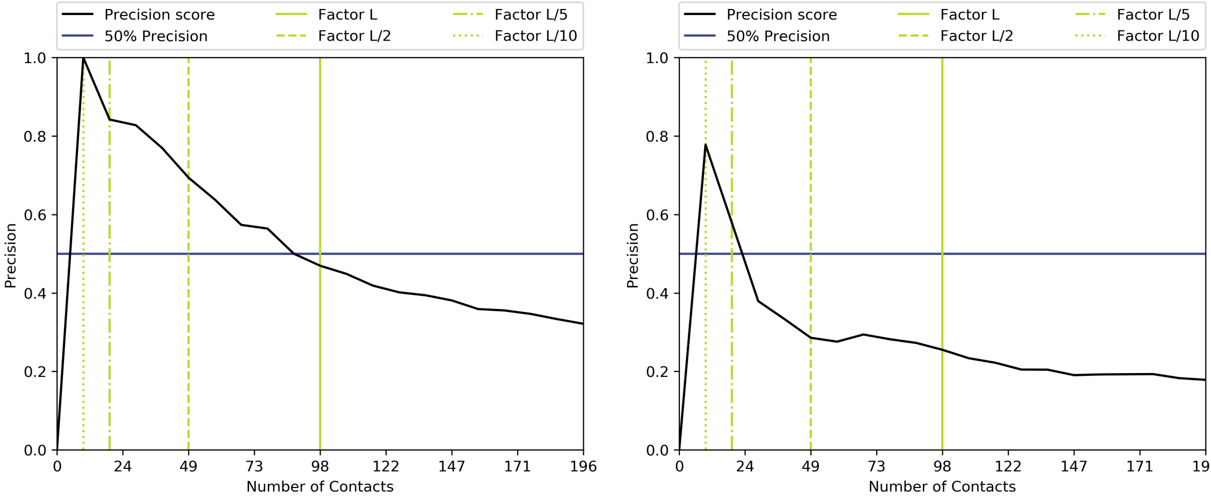

Figure 3.3: Precision score of group LASSO method and the original method

Then we try to modify the λ_w to test if it will affect the accuracy of final TM-score. Previously, the CCMpred uses default λ_w as:

$$\lambda_w = factor * (ncol - 1)$$

where $factor$ is 0.2 and ncol means the length of protein sequence. As Table 3.3 shown, we tried the factor of λ_w as 0.2, 2 and 20. We only choose one direction since the normalization term $\sqrt{w_{size}}$ enforces each group to lower weight. Through the increasing of λ_w, we want to balance this effect. It does help a little bit, but the result is still awful.

factor of λ_w	Average TM-score of Training Set
0.2	0.29
2	0.33
20	0.35

Table 3.3: Train result for our method with different factors of λ_w

3.2 Contact Prediction of Group-Dispersive Sparsity

Based on the formula discussed in section II, we still first train through the training set. In addition, we also treat lambda as hyper-parameter. This time, we choose the $factor$ as 0.02, 0.1, 0.2, 2, 20 to see which direction will benefit our dispersion level and the final result.

3.2.1 Pattern Analysis

In this part, we still use the 1BMG as an example. To analyze the pattern change across groups, we add grid lines, which is drawn based on secondary structure group, to t,he original reference contact map. As shown in Figure 3.4, left figure is our method result and right one is the original one. In this case, we can see that there are 8 blocks without predictions in our method result and only 2 of them has true contact. In contract, the original method has 12 empty clocks and 5 of them has true contact. Roughly speaking, our method does provide a better dispersion level than the original method.

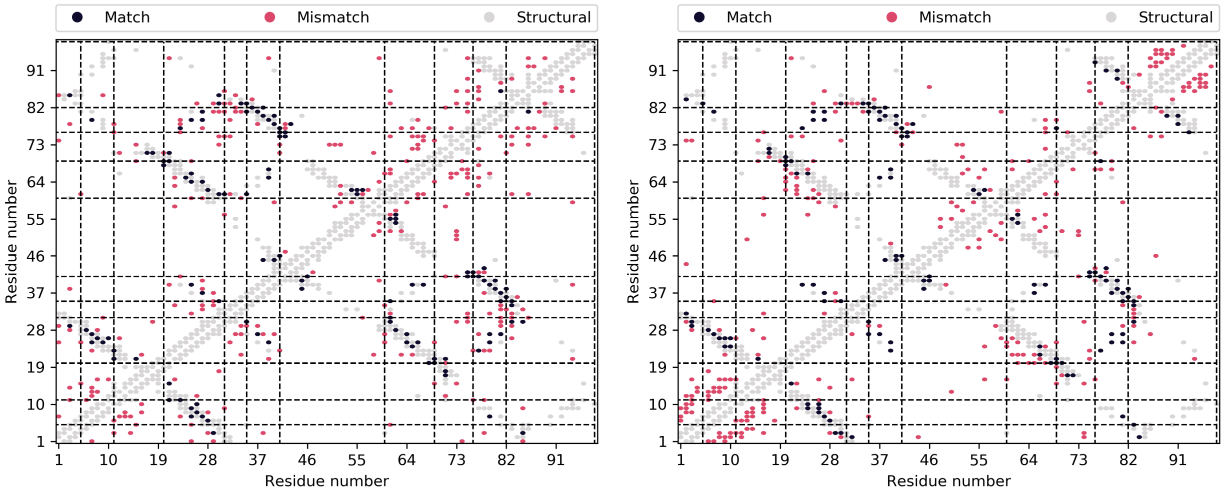

Figure 3.4: Reference contact map of our method and original method with grid lines

To be more specific, we can use Jensen-Shannon distance as a measurement to test if our method does promote the dispersion in general.

3.2.2 Precision Result

Let's still use the 1BMG as an example case. According to Figure 3.5, we can see that even though our method doesn't reach 100% in the beginning, it keeps a relatively high accuracy than before and its precision is pretty similar to the original method.

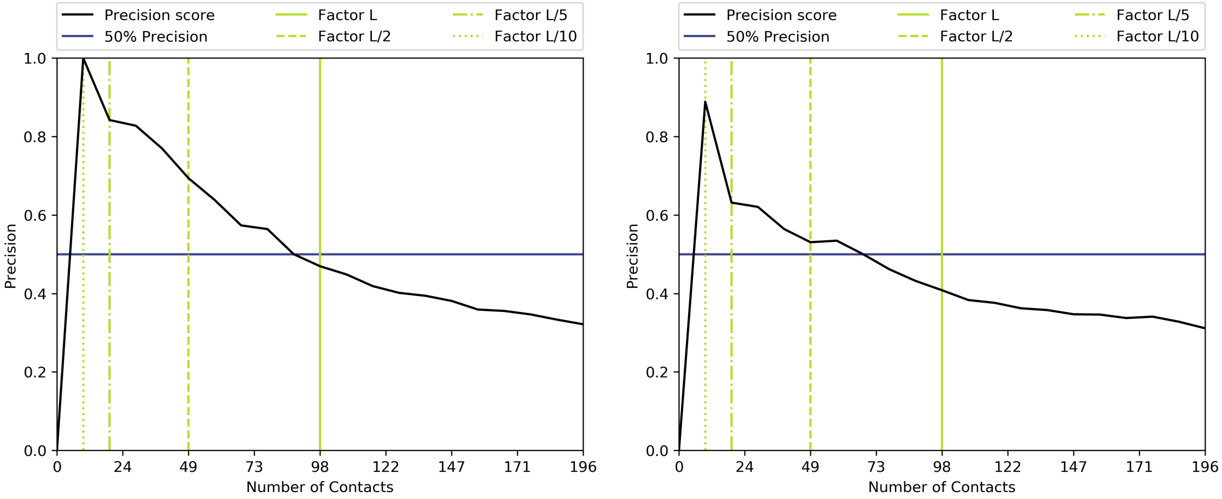

Figure 3.5: Precision score of group-dispersive sparsity method

In general, compared with CCMpred in test set as shown in Table 3.4, our method is a little bit worse. However, it shows some shining point in some cases like 1VIN. Since the hyper-parameters of our method are still not optimized, it is very likely to have a better result when we run the training round more times.

For the test set part, we can see that dispersive LASSO method also improve a lot for each λ_w from Table 3.5. Table 3.4 is the result of best combination hyper-parameter and the original method for training set. According to those two tables, our method roughly reaches the same accuracy level with the original method. What interesting is that dispersive LASSO reaches its peak when factor of λ_w equals to 0.2, but group LASSO reach its peak when factor is 20. It's easy to understand that the dispersive LASSO doesn't use normalization any more and its structure may lead to a situation

Test Case Name	Group-Dispersive Method	CCMpred
1CEW	0.44	0.47
1AYE	0.60	0.61
1NAT	0.80	0.78
1WL7	0.31	0.53
2J9V	0.34	0.35
1BMG	0.31	0.31
1VIN	0.64	0.61
1VFF	0.47	0.50

Table 3.4: Test result for our method and CCMpred

that the λ_w is too large instead of too small. Therefore, we add more value as factor of λ_w for further analysis as .

factor of λ_w	Average TM-score of Training Set
20	0.38
2	0.41
0.2	0.45
0.1	0.45
0.02	0.47
0.01	0.48
CCMpred	0.51

Table 3.5: Train results for our method with different factor of λ_w

3.2.3 Dispersion Level Analysis

In this part, we try to measure the dispersion level by comparing the 'ideal distribution', uniform distribution, with our result and calculate the distance by Jensen-Shannon Distance. Uniform distribution is sampled by random function and excludes those points which are in the short distance area.

In case of measurement approach, we don't use raw points to calculate the distance directly, since two same distribution may have a big distance because of sampling. For example, if two uniform distribution samples like (1,0,1,0) and (0,1,0,1), they are basically the same distribution but in our case the JSD will be 0.69, which is not a desired measure of “dispersion". Instead, we group them first based on SSE and then calculate their dispersion level by Jensen-Shannon Distance. The result is shown as Figure3.6, which uses the distance of CCMpred as the white color. For JS-Distance, value is the lower the better. In this figure, the color is lighter the better. As we can see, the best combination of parameters is one the upper left corner. However, CCMpred is better than our case. There are three possible reasons for that:

- The current optimization method may be not adequate for this high-dimensional and non-smooth problem.
- CCMpred has trained their hyper-parameters to be the optimal. In contrast, our models may not have done so given limited hyper-parameter space explored in limited time.
- The way dispersion is measured is not accurate. For example, the ideal dispersive distribution can be non-uniform. And a better “prior" can be learned from existing data.

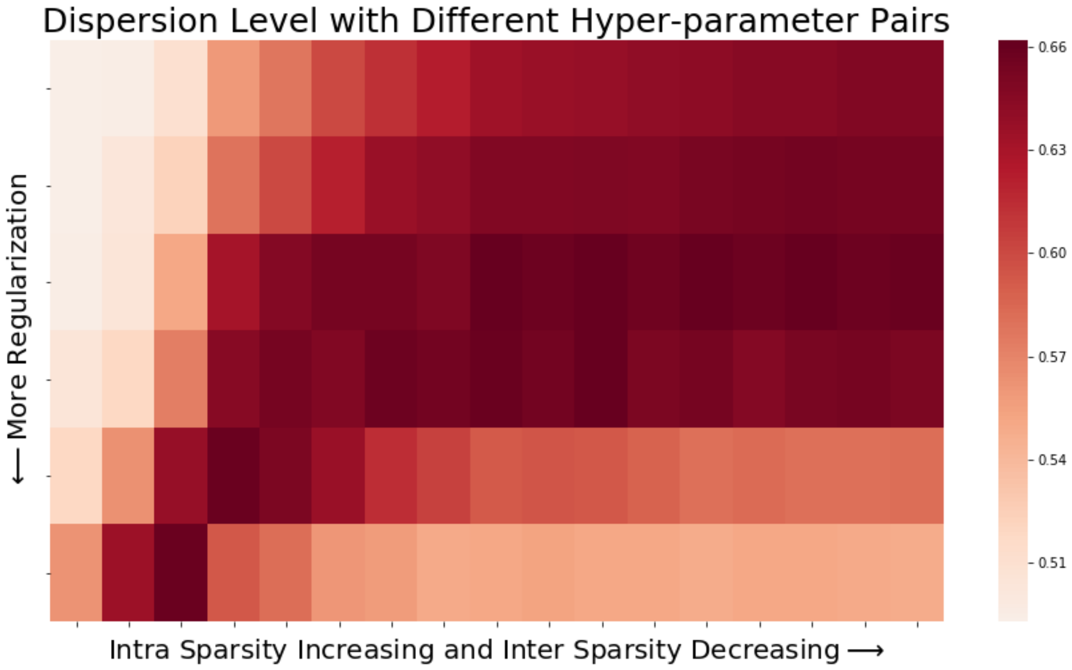

Figure 3.6: Dispersion level distribution

Then we want to answer if the dispersion level is correlated with accuracy of 3D structure reconstruction. Therefore, we introduce the Spearman Correlation, which is a nonparametric measure of rank correlation, to measure the relation between dispersion level vs. TM-score as shown in Figure 3.7. Spearman correlation ranges from -1 to 1. Negative value means negatively correlated and the larger the value is, the stronger the correlation is. In this figure, each block presents the Spearman correlation coefficient of all training set with one pair of hyper-parameter. Still, we use the CCMpred correlation coefficient as the baseline. As we can see, most of blocks are in light blue or red, which means dispersion level does help to promote the 3D structure reconstruction. Also, our method shows a stronger correlation. Therefore, improving the dispersive level, which is a 2D data, may be another path to help to promote or assess the accuracy of 3D structure prediction.

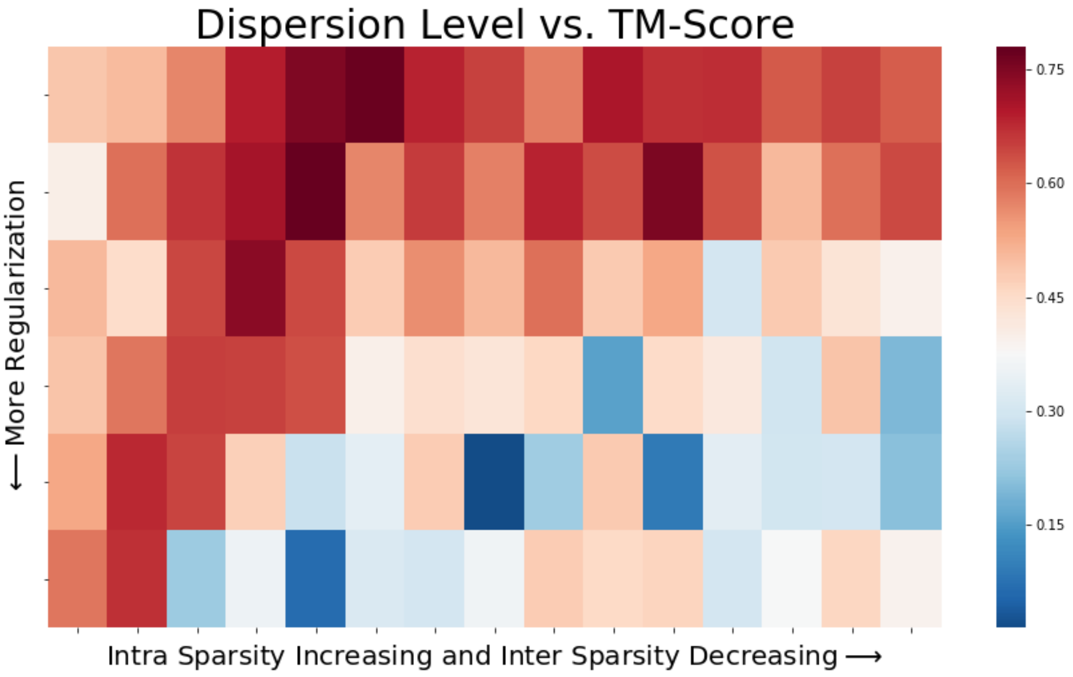

Figure 3.7: Dispersion level vs. TM-score

Taking the best result combination as an example, we draw the scatter and linear regression line as Figure 3.8. Each point represents one case of this hyper-parameter pair and the p-value of T-test is 0.002, which is less than 0.05. Therefore, we can say that dispersion level and 3D structure accuracy are significant linear correlated.

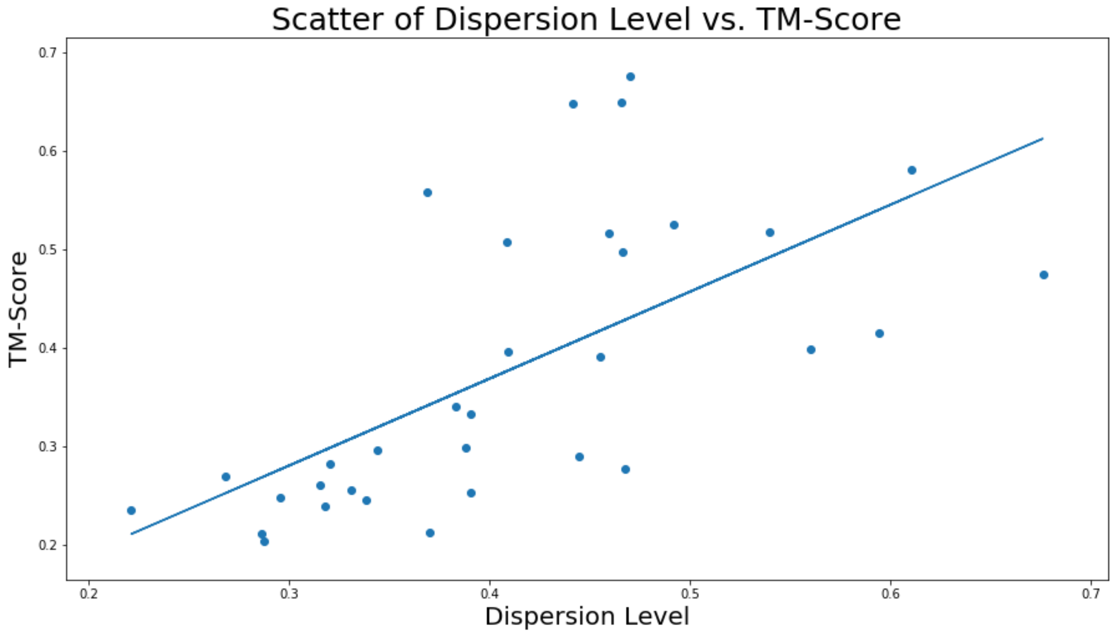

Figure 3.8: Scatter and linear regression for hyper-parameter combination with λ_w=0.01 and β=-0.9999

4. CONCLUSION AND FUTURE DIRECTIONS

In this study, we investigate protein contact prediction at two levels: dispersion of 2D contact map and accuracy of 3D structure. At dispersion level, we are eager to obtain a more dispersive structure, which means high sparsity within groups and high density between groups. After combination with the ideas of LASSO and ridge regression, we apply group-dispersive LASSO on CCMpred framework to predict protein contacts. Given secondary structure, our method shows a higher dispersion level than group LASSO and original method. The performance on test set indicates that dispersion level can help to promote the final result in certain area, but it still needs more trains for hyper-parameter and larger test set to assess its robustness.

At the 3D structure level, we also use two steps to show the effect of structured sparsity pattern. Group LASSO shows that such pattern can be controlled and different patterns help to reconstruct 3D structures in different area. However, since group LASSO always loses information in its empty groups, it can not provide a good result for 3D structure. Then we promote the pattern from density in certain groups to sparsity in each group. Because a higher level dispersion is more likely to help 3D structure reconstruction, especially for long distance contacts, the group-dispersive LASSO does a good job in contact prediction. Even though it has not universally outperformed a state-of-the-art method (CCMPred) in 3D structure reconstruction, its ability of pattern control has shown a great potential and will provide more possibility with more prior knowledge added. Furthermore, it still has space to improve. For example, we can treat different types of groups differently, optimize their hyper-parameters, and promote finer patterns.

REFERENCES

[1] F. Sanger and H. Tuppy, “The amino-acid sequence in the phenylalanyl chain of insulin. 1. the identification of lower peptides from partial hydrolysates,” *Biochemical Journal*, vol. 49, no. 4, p. 463, 1951.

[2] F. Sanger, “Chemistry of insulin,” *Science*, vol. 129, no. 3359, pp. 1340–1344, 1959.

[3] L. Pauling, R. B. Corey, and H. R. Branson, “The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain,” *Proceedings of the National Academy of Sciences*, vol. 37, no. 4, pp. 205–211, 1951.

[4] V. Simossis and J. Heringa, “Integrating protein secondary structure prediction and multiple sequence alignment,” *Current Protein and Peptide Science*, vol. 5, no. 4, pp. 249–266, 2004.

[5] H. Kamisetty, S. Ovchinnikov, and D. Baker, “Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era,” *Proceedings of the National Academy of Sciences*, p. 201314045, 2013.

[6] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, “Direct-coupling analysis of residue coevolution captures native contacts across many protein families,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. E1293–E1301, 2011.

[7] D. T. Jones, D. W. Buchan, D. Cozzetto, and M. Pontil, “Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments,” *Bioinformatics*, vol. 28, no. 2, pp. 184–190, 2011.

[8] J. I. Sułkowska, F. Morcos, M. Weigt, T. Hwa, and J. N. Onuchic, “Genomics-aided structure prediction,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 26, pp. 10340–10345, 2012.

[9] T. Nugent and D. T. Jones, “Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 24, pp. E1540–E1547, 2012.

[10] O. Carugo, F. Eisenhaber, and Carugo, *Data mining techniques for the life sciences*, vol. 609. Springer, 2010.

[11] M. Niggemann and B. Steipe, “Exploring local and non-local interactions for protein stability by structural motif engineering1,” *Journal of molecular biology*, vol. 296, no. 1, pp. 181–195, 2000.

[12] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander, “Protein 3d structure computed from evolutionary sequence variation,” *PloS one*, vol. 6, no. 12, p. e28766, 2011.

[13] S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, “Accurate de novo prediction of protein contact map by ultra-deep learning model,” *PLoS computational biology*, vol. 13, no. 1, p. e1005324, 2017.

[14] S. H. P. de Oliveira, J. Shi, and C. M. Deane, “Comparing co-evolution methods and their application to template-free protein structure prediction,” *Bioinformatics*, vol. 33, no. 3, pp. 373–381, 2017.

[15] “Faq · soedinglab/ccmpred wiki · github.” `https://github.com/soedinglab/CCMpred/wiki/FAQ`, 2013.

[16] M. Spencer, J. Eickholt, and J. Cheng, “A deep learning network approach to ab initio protein secondary structure prediction,” *IEEE/ACM transactions on computational biology and bioinformatics (TCBB)*, vol. 12, no. 1, pp. 103–112, 2015.

[17] C. N. Magnan and P. Baldi, “Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity,” *Bioinformatics*, vol. 30, no. 18, pp. 2592–2597, 2014.

[18] J.-M. Chandonia, G. Hon, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner, “The astral compendium in 2004,” *Nucleic acids research*, vol. 32, no. suppl_1, pp. D189–D192, 2004.

[19] S. Seemayer, M. Gruber, and J. Söding, “CcmpredâĂŤfast and precise prediction of protein residue–residue contacts from correlated mutations,” *Bioinformatics*, vol. 30, no. 21, pp. 3128–3130, 2014.

[20] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[21] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[22] F. Campbell, G. I. Allen, *et al.*, “Within group variable selection through the exclusive lasso,” *Electronic Journal of Statistics*, vol. 11, no. 2, pp. 4220–4257, 2017.

[23] J. Lin, “Divergence measures based on the shannon entropy,” *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[24] S. Wang, W. Li, R. Zhang, S. Liu, and J. Xu, “Coinfold: a web server for protein contact prediction and contact-assisted protein folding,” *Nucleic acids research*, vol. 44, no. W1, pp. W361–W366, 2016.

[25] P. Danaher, P. Wang, and D. M. Witten, “The joint graphical lasso for inverse covariance estimation across multiple classes,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 2, pp. 373–397, 2014.

[26] S. Wang, J. Peng, J. Ma, and J. Xu, “Protein secondary structure prediction using deep convolutional neural fields,” *Scientific reports*, vol. 6, p. 18962, 2016.

[27] A. T. Brünger, P. D. Adams, G. M. Clore, W. L. DeLano, P. Gros, R. W. Grosse-Kunstleve, J.-S. Jiang, J. Kuszewski, M. Nilges, N. S. Pannu, *et al.*, “Crystallography & nmr system:

A new software suite for macromolecular structure determination," *Acta Crystallographica Section D*, vol. 54, no. 5, pp. 905–921, 1998.

[28] B. Adhikari, D. Bhattacharya, R. Cao, and J. Cheng, "Confold: residue-residue contact-guided ab initio protein folding," *Proteins: Structure, Function, and Bioinformatics*, vol. 83, no. 8, pp. 1436–1449, 2015.

[29] J. Xu and Y. Zhang, "How significant is a protein structure similarity with tm-score= 0.5?," *Bioinformatics*, vol. 26, no. 7, pp. 889–895, 2010.