Improved Protein Residue–Residue Contact Prediction Using Image Denoising Methods

Amelia Villegas-Morcillo, Juan A. Morales-Cordovilla, Angel M. Gomez, Victoria Sanchez
Dept. of Signal Theory, Telematics and Communications, University of Granada, Spain
E-mail: melyvm@correo.ugr.es, {jamc,amgg,victoria}@ugr.es

***Abstract*—A protein contact map is a simplified matrix representation of the protein structure, where the spatial proximity of two amino acid residues is reflected. Although the accurate prediction of protein inter-residue contacts from the amino acid sequence is an open problem, considerable progress has been made in recent years. This progress has been driven by the development of contact predictors that identify the co-evolutionary events occurring in a protein multiple sequence alignment (MSA). However, it has been shown that these methods introduce Gaussian noise in the estimated contact map, making its reduction necessary. In this paper, we propose the use of two different Gaussian denoising approximations in order to enhance the protein contact estimation. These approaches are based on (i) sparse representations over learned dictionaries, and (ii) deep residual convolutional neural networks. The results highlight that the residual learning strategy allows a better reconstruction of the contact map, thus improving contact predictions.**

***Index Terms*—Protein Contact Map, Evolutionary Coupling, Image Denoising, Sparse Representations, Dictionary Learning, Deep Convolutional Neural Networks, Residual Learning**

I. Introduction

Proteins are life essential macromolecules composed of long chains that contain 20 different types of amino acid residues. Protein folding is one of the most challenging problems found in bioinformatics, which still remains unsolved. It refers to the spatial arrangement of the amino acid sequence in three-dimensional (3D) space, which is closely related to the biological function performed by the protein.

Several experimental techniques, such as X-ray crystallography, nuclear magnetic resonance spectroscopy and electron microscopy, are widely employed to determine protein structure. However, these methods are not cost-efficient compared to fast deoxyribonucleic acid (DNA) sequencing processes, which are continuously generating a huge quantity of amino acid sequences with unsolved structure [1]. This fact has led to a great development of computational methods that attempt to predict the protein structure from its amino acid sequence. These computational methods can be divided into template-based and template-free modeling. Template-free modeling methods, also known as *de novo* or *ab initio* prediction methods, are suitable for proteins without structural homologs [2]. The complexity of this approach is usually reduced by using a two-dimensional (2D) representation known as protein contact map. Two amino acid residues within a protein are said to form a contact when they share a spatial proximity that is sufficient for a molecular interaction to occur. Thus, the true contact map for a protein with L residues consists of an $L \times L$ symmetrical matrix, where each element specifies whether two inter-residues are in contact under a certain Euclidean distance threshold. By contrast, the matrix elements within an estimated contact map indicate the contact probability of two residues. Recent research has shown that even a low proportion of correctly-predicted contacts can be enough for accurately modeling the protein structure [3].

Current state-of-the-art contact predictors can be classified into two main categories, based on evolutionary coupling analysis (EC) and supervised machine learning (ML), respectively [4]. In the first group we can find methods such as PSICOV [5] and CCMpred [6], which aim to identify co-evolved residues from the protein multiple sequence alignment (MSA). Co-evolution is related to correlated mutations occurring in proteins. This means that if one of the two residues in contact has mutated during evolution, its counterpart has to change as well in order for the 3D structure to remain stable [7]. On the other hand, supervised machine learning-based predictors use a set of input features derived from the MSA to estimate the contact map, including position-specific scoring matrices (PSSM), secondary structure predictions or solvent accessibility information [8]. ML-based methods learn from these features using several algorithms, such as support vector machines, neural networks and random forests.

Generally, the performance of evolutionary coupling methods relies on having a sufficient number of effective homologous sequences to construct the protein MSA. In order to improve the precision of estimated contacts, methods such as MetaPSICOV [9] wisely combine the EC-based approach with machine learning. Additionally, the EC-based methods were designed to reduce the number of misleading indirect coupling pairs, i.e. those that show a high degree of correlated mutation without being close in the 3D space. This is caused by the transitive relationship between residue pairs, resulting in a transitive noise within the contact map [4].

Besides this transitive noise, it has been recently shown that Gaussian noise also exists in contact maps derived from evolutionary couplings. To reduce its effect, the $\mathrm{R_2C}$ system [10] included a post-processing noise filter, yielding higher contact precision. This denoising step was implemented using the non-local means with optimal weights algorithm specified in [11].

This work has been supported by the Spanish MINECO/FEDER Project TEC2016-80141-P and the associated FPI grant BES-2017-079792. We also acknowledge the support of NVIDIA Corporation with the donation of a Titan X GPU.

In this work, we aim to improve the estimated protein contact map by further suppressing the inherent Gaussian noise. To accomplish that, we have followed two approaches related to image processing: sparse representations over learned dictionaries, and deep residual convolutional neural networks.

First, we consider the K-SVD algorithm [12], widely used to design overcomplete dictionaries for sparse representations of signals. This method has proven to be successful for image super-resolution, compression and denoising [13]. Sparse representations exploit image redundancies, trying to find the minimum number of dictionary entries needed to properly reconstruct the image. Although finding the sparsest solution is an NP-hard problem, approximate solutions can be efficiently found by using pursuit algorithms.

Then, we explore a deep neural network approach, as there is a growing interest in deep learning for proteomics (see [14], [15]). In particular, we analyze a deep convolutional neural network (DCNN) specially designed for image denoising [16]. Convolutional neural networks have the ability to extract structural motifs, which makes them suitable for image processing. Moreover, it is possible to reach a high level of abstraction with very deep architectures and residual learning.

The rest of the paper is structured as follows. In Section II, we explain the contact map estimation process. Section III describes the proposed methods for contact map denoising. In Section IV, we present the main results obtained from the experiments. Finally, Section V summarizes the conclusions derived from this research and its possible extensions.

II. EC-Based Contact Map Estimation

The overall approach proposed in this work is outlined in Fig. 1. Our objective is to improve the precision of estimated contact maps. True contacts are defined in terms of spatial molecular interaction. The residue spatial proximity is measured as the Euclidean distance between their beta carbon (C_β) atoms (C_α in Glycine). This results in a distance map, which is converted to a binary contact map (a symmetrical matrix) by considering a threshold (in this work we used a distance less than 8 Å). Thus, the extraction of true contacts requires knowledge of the protein's 3D experimental structure.

In this work, we have used one of the most popular EC-based methods, CCMpred [6], to estimate the contact map. CCMpred implements the approach based on Markov random field pseudo-likelihood maximization. Its election was motivated by the high contact precision reached in comparison to other EC-based methods [4]. In addition, it can be run on a GPU card. The first step of this method is to build a multiple sequence alignment (MSA) for each query protein, using the HHblits software [17] to search for homologous sequences in the `uniprot20_2016_02` database [18]. Then, the MSA is used as input to CCMpred, which estimates correlated mutations between each pair of residues, generating a symmetrical contact likelihood matrix as a result. All parameters in both tools were set to default values, except for the number of HHblits iterations that was increased to five.

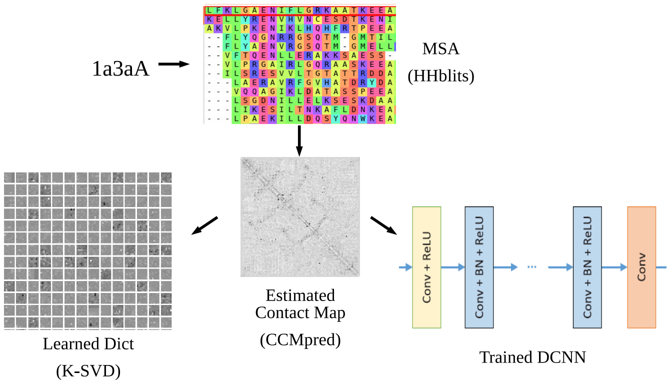

Fig. 1. Description of the overall protein contact map denoising procedures.

III. Proposed Methods for Contact Map Denoising

The estimated contact map derived from CCMpred can be viewed as an image $\mathbf{X}$, which is a noisy version of the true contact map $\mathbf{Y}$ corrupted by an additive Gaussian noise $\mathbf{Z}$:

$$\mathbf{X} = \mathbf{Y} + \mathbf{Z}. \tag{1}$$

As indicated in the introduction, in order to address the contact map denoising problem, we propose the use of two noise removal techniques: sparse representations over learned dictionaries and deep residual convolutional neural networks.

A. Dictionary Learning for Sparse Representations

The overall K-SVD method for image denoising [13] can be seen as a three-stage iterative process: (i) sparse coding, (ii) dictionary updating, and (iii) final averaging.

In the sparse coding stage, a fixed dictionary $\mathbf{D}$ with K entries (also known as atoms) and the noisy image $\mathbf{X}$ divided into N overlapping patches of size $\sqrt{n} \times \sqrt{n}$ are considered. Thus, the sparse coding vector $\boldsymbol{\alpha}_i$ associated to the i-th patch ($i = 1, 2, ..., N$) can be computed by solving the following optimization problem:

$$\hat{\boldsymbol{\alpha}}_i = \arg\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{\alpha}_i\|_0 \ \text{ subject to } \ \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \le \epsilon^2 \tag{2}$$

where $\mathbf{x}_i$ is the i-th flattened patch (column vector) of size n, $\mathbf{D}$ is the dictionary of size $n \times K$, ϵ is the error tolerance that depends on the noisy image SNR, and $\|\cdot\|_p$ represents the l^p-norm. This problem can be addressed with the greedy orthogonal matching pursuit (OMP) algorithm [19], which takes the closest atom of the dictionary at a time to update the sparse coding vector.

Once the N sparse coding vectors are computed, a new version of the patches $\hat{\mathbf{x}}_i = \mathbf{D}\hat{\boldsymbol{\alpha}}_i$ is obtained to update the dictionary (the initial dictionary $\mathbf{D}_0$ is usually built from the overcomplete discrete cosine transform (DCT) or with randomly taken patches from the image). For each atom k in the dictionary, the K-SVD algorithm [12] locates the set of patches that use this atom ($\hat{\boldsymbol{\alpha}}_i(k) \neq 0$), to perform a singular value decomposition (SVD) operation on the representation errors $\mathbf{e}_i^k$, which are calculated as

$$\mathbf{e}_i^k = \mathbf{x}_i - \sum_{l \neq k} \mathbf{d}_l \hat{\boldsymbol{\alpha}}_i(l) \tag{3}$$

where $\mathbf{d}_l$ are the flattened dictionary atoms. These two stages are repeated a fixed number of iterations, resulting in a learned dictionary $\mathbf{D}$ and sparse coding vectors $\hat{\boldsymbol{\alpha}}_i$ associated to each patch. Finally, the reconstructed image $\hat{\mathbf{X}}$ is computed as an average of the N overlapping denoised patches $\hat{\mathbf{x}}_i$ with the original noisy image $\mathbf{X}$:

$$\hat{\mathbf{X}} = \frac{\lambda \mathbf{X} + \sum_i \mathbf{R}_i^T \mathbf{D} \hat{\boldsymbol{\alpha}}_i}{\lambda \mathbf{I} + \sum_i \mathbf{R}_i^T \mathbf{R}_i} \tag{4}$$

where $\mathbf{R}_i$ is a mask matrix that extracts the i-th patch from the image ($\mathbf{x}_i = \mathbf{R}_i \mathbf{X}$) and λ is an hyper-parameter dependent on the noisy image SNR.

In our study, we evaluated the K-SVD method to denoise our contact map images. To do so, we adopted the efficient implementation available in [20], which is based on approximated K-SVD and batch-OMP. In order to exploit the symmetry in contact maps, we trained the dictionary only with overlapping patches from the upper triangular part of the image. We only considered this reconstructed part to evaluate the denoised contacts.

B. DCNN Training with Residual Learning

The DCNN model implements a very deep architecture with 2D convolutional layers. This architecture consists of d layers that apply 64 convolution filters of size 3×3. As our contact map images are in gray-scale, we have one channel at the input layer and one filter at the output layer. The output image size is kept by applying zero-padding before convolutions. At each layer (except for the output one), rectified linear units (ReLU) non-linearities are used as activation function. In addition, if we want that all the pixels within a patch (of size $\sqrt{n} \times \sqrt{n}$) contribute to each output after the cascade of convolutional layers (with $\sqrt{n_f} \times \sqrt{n_f}$ filter size), network depth d must be set to a minimum value given as:

$$d = \left(\sqrt{n} - 1\right) / \left(\sqrt{n_f} - 1\right). \tag{5}$$

On the other hand, the residual learning strategy aims to extract the differences between the network inputs and outputs. In other words, the DCNN trains a residual mapping of the input image regarding the noise $\mathcal{R}(\mathbf{X}) \approx \mathbf{Z}$ and then calculates the reconstructed image as $\hat{\mathbf{X}} = \mathbf{X} - \mathcal{R}(\mathbf{X})$. Thus, the loss function $l(\boldsymbol{\Theta})$ can be formulated as:

$$l(\boldsymbol{\Theta}) = \frac{1}{2B} \sum_{i=1}^{B} \|\mathcal{R}(\mathbf{x}_i; \boldsymbol{\Theta}) - (\mathbf{x}_i - \mathbf{y}_i)\|_F^2 \tag{6}$$

where B is the number of noisy/clean training patches in a batch, and $\boldsymbol{\Theta}$ are the trainable parameters. This is equivalent to minimizing the mean squared error between each clean patch $\mathbf{y}_i$ and the reconstructed patch $\hat{\mathbf{x}}_i$. Besides residual learning, an optimization strategy based on mini-batches with Adam algorithm for gradient-descend and batch-normalization was adopted as in [16].

Unlike the K-SVD method, this approximation is able to train the model without knowing the noisy image SNR. We adopted the TensorFlow GPU implementation of [16] to train a noise level-independent model with our contact map images. As the output matrix is not assured to be symmetrical, we computed the mean value of the two triangular parts (upper and lower) to evaluate the contacts.

IV. Experimental Framework and Results

In this section, we first describe the protein datasets used in the experiments. Next, we specify the criteria followed for contact evaluation. Finally, we present the results obtained from the experiments carried out and we discuss them.

A. Training and Test Datasets

During the evaluation stage, we considered three test datasets. The first one comprises 150 Pfam proteins from PSICOV [5], which have been widely used for contact map evaluations. The other two sets were obtained from two CASP (Critical Assessment of Structure Prediction) competitions. In particular, 116 protein domains from CASP10 (2012) and 103 from CASP11 (2014) were used as in [10]. Moreover, none of the test sequences contained less than 50 residues or more than 519 residues.

For DCNN training purposes, a set with 3760 protein domains [15] was taken. We further modified this set in order to guarantee its independence from the testing sets. To do so, we run the CD-HIT [21] tool to exclude those proteins that shared more than 40% of sequence identity with any protein in the test set. A total number of 3427 training proteins were kept, with sequence lengths varying from 28 to 597 residues. From these, we randomly selected 300 proteins to validate the DCNN and we used the remaining ones in the training stage.

B. Evaluation Criteria

Contact evaluation was accomplished by comparing the estimated contact map with the true contact map extracted from the protein data bank (PDB) [1]. The widely used evaluation strategy [4] is based on dividing contacts into three groups. These groups are dependent on the amino acid sequence positions, i.e. short-range (residue separation between 6 and 11 positions), medium-range (from 12 to 23) or long-range (greater than 23). For each group, prediction accuracy is obtained by computing the precision of the L/k contacts with the highest probability, where L is the sequence length and $k = \{10, 5, 2, 1\}$ controls the ratio of contacts to be evaluated.

C. K-SVD Parameter Setting

In order to evaluate the K-SVD method, we followed a dictionary learning strategy based on training one dictionary $\mathbf{D}$ for each noisy contact map. Thus, we first divided the upper triangular part of the zero-padded image in overlapping patches. As the contact map has size $L \times L$ and protein lengths are variable, the number of training patches also varies in each

TABLE I
CONTACT PRECISION VALUES FOR SHORT-, MEDIUM- AND LONG-RANGE FOR THE EVALUATED METHODS ON THE TEST DATASETS

Test dataset	Method	Short-range				Medium-range				Long-range			
		L/10	*L/5*	*L/2*	*L*	*L/10*	*L/5*	*L/2*	*L*	*L/10*	*L/5*	*L/2*	*L*
150 Pfam proteins	Baseline	56.1	40.2	23.0	15.5	64.2	49.8	29.0	18.4	78.1	71.0	50.5	33.7
	R_2C filter	51.3	36.8	22.0	15.2	64.5	49.4	29.9	19.2	78.9	70.1	51.2	35.8
	K-SVD OMP	55.7	39.7	24.1	16.3	67.2	53.0	32.2	20.8	79.8	73.0	54.0	38.4
	DCNN	**77.6**	**65.2**	**40.9**	**25.0**	**80.5**	**71.0**	**48.3**	**30.3**	**89.6**	**85.3**	**72.1**	**54.5**
116 CASP10 proteins	Baseline	41.5	31.2	19.4	13.5	53.1	41.9	26.3	18.1	53.7	47.8	34.4	23.1
	R_2C filter	41.3	30.6	19.8	14.3	54.0	42.5	27.8	19.1	57.1	51.1	37.3	26.4
	K-SVD OMP	43.1	32.2	20.6	14.7	55.6	43.8	29.8	20.4	56.3	49.7	38.2	27.1
	DCNN	**58.4**	**48.5**	**31.9**	**20.7**	**65.8**	**58.1**	**43.0**	**29.8**	**67.3**	**63.2**	**50.6**	**37.6**
103 CASP11 proteins	Baseline	32.9	23.9	15.3	11.3	38.0	28.5	17.8	12.5	47.4	40.2	28.9	20.1
	R_2C filter	31.4	22.8	14.6	11.5	39.7	29.6	19.2	13.8	50.0	42.5	30.7	22.3
	K-SVD OMP	32.8	23.4	15.4	12.1	40.4	31.7	20.6	14.3	48.5	42.4	31.4	22.7
	DCNN	**48.1**	**38.8**	**26.1**	**17.6**	**53.7**	**46.6**	**31.9**	**21.2**	**56.5**	**53.2**	**43.0**	**32.6**

case. To initialize each dictionary, we used the overcomplete DCT with K atoms. Then, we executed 10 iterations of the K-SVD algorithm, using OMP for sparse coding with maximum error $\epsilon = 1.15\sqrt{n}\sigma$, where the noise standard deviation σ was estimated following the fast approach cited in [10]. Finally, the upper triangular matrix was reconstructed using (4), with $\lambda = 0.5/\sigma$. We chose low values for λ in order to give less importance to the original noisy contact map than to the denoised one.

For K-SVD parameter selection purposes, we conducted initial experiments on the 150 Pfam proteins, testing two patch sizes (5×5 and 7×7) and two different number of atoms K (625 and 900). The results were compared with the baseline, i.e. contact precision values from CCMpred estimations. All combinations of patch/dictionary sizes yielded similar contact results. However, slightly better results were achieved when considering 5×5 patches and $K = 900$ atoms, so we have selected this configuration for a complete evaluation on the rest of datasets.

D. DCNN Parameter Setting

The denoising task with DCNN was carried out by adopting the architecture described in Section III-B. In order to apply residual learning, we extracted the true and estimated contact maps from the 3127 training proteins mentioned in Section IV-A. All training contact maps were divided into patches of size 35×35 (with a stride of 10 pixels), which provided us with more than a million noisy/clean samples to train the DCNN. Unlike the previous method, we extracted patches from the entire image and not only from the upper triangle (the denoising procedure was also applied on the entire image). According to (5), the depth of the network was set to $d = 17$. Training contact maps that are smaller than the patch size (35×35) were filled with zeros at the input and multiplied by a binary mask before computing the loss function in (6).

Thereafter, we trained the DCNN for 50 epochs in a mini-batch mode. At each epoch, we randomly scrambled all patches and iteratively fed the network with subsets (batches) of 256 noisy/clean samples. When an epoch was finished, we used the current model to denoise and evaluate the 300 validation contact maps. Fig. 2 shows the precision curves for long-range contact predictions on the validation set. As can be observed, the evaluation values in epoch 0 (before the first gradient-descend step) are similar to those from the baseline (CCMpred), which can be explained by the residual learning strategy. Although the precision curves show a considerable performance increase in the very first epochs, saturation appears rapidly. In this study, we followed an early stopping strategy to prevent DCNN overfitting. Thus, we selected the model at epoch 31 to denoise the three test datasets, as it yields the best evaluation results on the validation dataset.

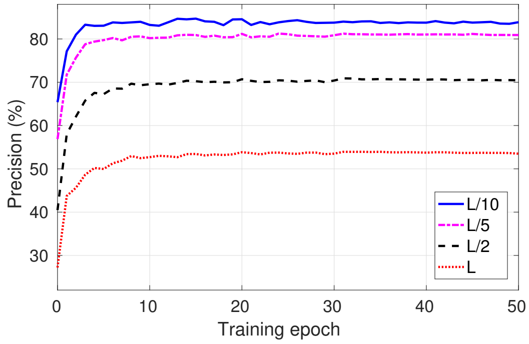

Fig. 2. Long-range contact prediction performance on 300 validation protein domains along DCNN training epochs.

E. Comparison of Methods

Finally, we present the contact precision results for the three test datasets (i.e. 150 Pfam proteins, 116 CASP10 protein domains and 103 CASP11 protein domains), achieved by the baseline (CCMpred) and the proposed post-processing denoising methods K-SVD and DCNN. These results have been also compared to the R_2C noise filter [10]. In this case, we used the implementation in [11], setting the main parameters to 9×9 for neighboring size and 13×13 for patch size, as in [10].

Table I summarizes all the prediction results for short-, medium- and long-range contacts. For all contact ranges in each test dataset, we marked in red those results that are worse than the baseline and, in boldface, the best results. As we can

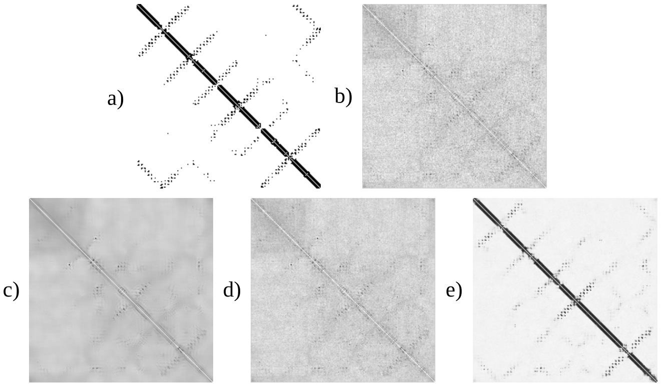

Fig. 3. Contact maps obtained for protein domain T0682-D1 in the CASP10 dataset. a) True from PDB. b) Estimated from CCMpred. c) Denoised using R_2C. d) Denoised using K-SVD OMP. e) Denoised using DCNN.

see, both the R_2C filter and the K-SVD method do not perform very well in some cases (specially for short-range contacts). The proposed K-SVD method performs slightly better than the R_2C filter, whereas the highest overall prediction enhancement is achieved with the residual DCNN approach.

This is due to the fact that, both the R_2C filter and the K-SVD method only consider the estimated contact map itself, while the residual DCNN has access to true contact maps during the training phase. Therefore, the first two approaches have the ability to reduce Gaussian noise but at the cost of losing some contacts due to a smoothing effect. On the contrary, the residual DCNN shows noise reduction potential along with the ability to recover some missing contacts not present in CCMpred estimations. Moreover, one of the main DCNN advantages is that we can train a blind model without knowing the noise standard deviation, so we can avoid using poor estimations of it as in the R_2C filter or the K-SVD method. As an example, Fig. 3 shows the resulting contact maps for protein domain T0682-D1 (included in the CASP10 dataset), produced by the three denoising methods in comparison to the true and baseline contact maps. Once again, we can see that the DCNN provides the best denoised contact map.

V. Conclusions

In this paper, we have evaluated two alternative Gaussian denoising methods for the protein contact map prediction problem. The first one is based on sparse representations over learned dictionaries for image denoising, using K-SVD with the OMP algorithm. In the second one, we train a deep convolutional neural network (DCNN) with residual learning for image noise removal. The experimental results show that better contact precision values can be obtained by these noise reduction techniques. We particularly found that the residual DCNN strategy performs the best in the denoising task, allowing the acquisition of more true contacts. Future work will explore other residual DCNN architectures that can exploit the sparse nature of contact maps, along with the study of how the improved contacts enhance the prediction of the 3D protein structure.

References

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[2] 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.

[3] D. E. Kim, F. DiMaio, R. Y.-R. Wang, Y. Song, and D. Baker, "One contact for every twelve residues allows robust and accurate topology-level protein structure modeling," *Proteins*, vol. 82, no. 0 2, pp. 208–218, 2014.

[4] Q. Wuyun, W. Zheng, Z. Peng, and J. Yang, "A large-scale comparative assessment of methods for residue–residue contact prediction," *Briefings in Bioinformatics*, pp. 1–12, 2016.

[5] D. T. Jones, D. W. A. 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, 2012.

[6] S. Seemayer, M. Gruber, and J. Söding, "CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations," *Bioinformatics*, vol. 30, no. 21, pp. 3128–3130, 2014.

[7] U. Göbel, C. Sander, R. Schneider, and A. Valencia, "Correlated mutations and residue contacts in proteins," *Proteins: Structure, Function, and Bioinformatics*, vol. 18, no. 4, pp. 309–317, 1994.

[8] J. Eickholt and J. Cheng, "Predicting protein residue–residue contacts using deep networks and boosting," *Bioinformatics*, vol. 28, no. 23, pp. 3066–3072, 2012.

[9] D. T. Jones, T. Singh, T. Kosciolek, and S. Tetchner, "MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins," *Bioinformatics*, vol. 31, no. 7, pp. 999–1006, 2015.

[10] J. Yang, Q.-Y. Jin, B. Zhang, and H.-B. Shen, "R_2C: improving *ab initio* residue contact map prediction using dynamic fusion strategy and Gaussian noise filter," *Bioinformatics*, vol. 32, no. 16, pp. 2435–2443, 2016.

[11] Q. Jin, I. Grama, C. Kervrann, and Q. Liu, "Non-local means and optimal weights for noise removal," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1878–1920, 2017.

[12] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[13] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[14] 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.

[15] J. Zhu, H. Zhang, S. C. Li, C. Wang, L. Kong, S. Sun, W.-M. Zheng, and D. Bu, "Improving protein fold recognition by extracting fold-specific features from predicted residue–residue contacts," *Bioinformatics*, vol. 33, no. 23, pp. 3749–3757, 2017.

[16] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. [Online]. Available: https://github.com/crisb-DUT/DnCNN-tensorflow

[17] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment," *Nature Methods*, vol. 9, no. 2, pp. 173–175, 2012.

[18] The UniProt Consortium, "UniProt: the universal protein knowledgebase," *Nucleic Acids Research*, vol. 45, no. D1, pp. D158–D169, 2017.

[19] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *27-th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 40–44, 1993.

[20] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using Batch Orthogonal Matching Pursuit," *Cs Technion*, vol. 40, no. 8, pp. 1–15, 2008.

[21] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.