Protein Contact-Map Prediction using Variational Autoencoder

Anowarul Kabir
Computer Science
George Mason University
akabir4@gmu.edu

Abstract

Protein structure prediction is an important topic in Structural Bioinformatics. It aids in understanding attributes and functionalities which enables novel drug and antibody design. Contact-map is an intermediate representation of the tertiary structure of a protein, which has been proven to be significant to improve structure prediction [1]. In this study, we present an approach for predicting contact-map using variational autoencoder as a generative model. This model takes two sub-sequences of a protein chain of same size and outputs contact-map representing closeness of two residues of corresponding sub-sequences. We use binary cross entropy loss as reconstruction loss and with KL-divergence loss, and we find out reconstruction loss performs better. Finally, we evaluate our model on test dataset using precision-recall with ground truth dataset and analyse the results. The precision and recall score gained by our best model are 65% and 62% respectively.

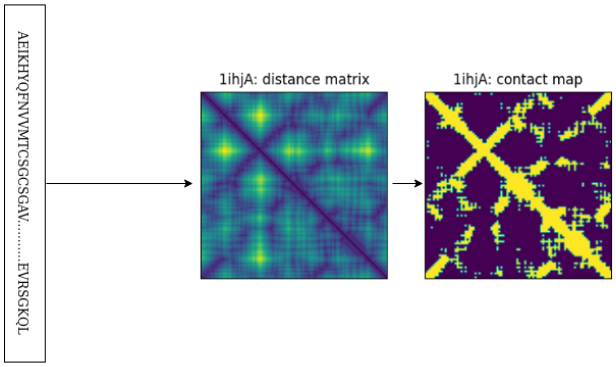

Figure 1. Sequence to contact-map generation from distance matrix. Distances are computed between $C_\beta - C_\beta$ atoms with threshold $6-12$Å (C_α is used for Glycine). Here we use 12Å throughout the work. First four character "1ihj" is the corresponding protein id and fifth one "A" corresponds to chain id of that protein.

1. Introduction and Related work

Understanding protein structure is critical for understanding protein functionalities. Generally proteins with same 3D structure functions in the same way. Therefore, if we can find out tertiary structure and similar proteins with that structure, we can predict the characteristics of the subject protein.

Advancements in protein design aids to developing new therapies, discovering small-molecule binding, designing novel enzymes and anti-bodies and so on [2]. Protein structure prediction task involves predicting tertiary or quaternary structure given primary structure sequence in addition with other features. An intermediate representation, contact-map, has recently been shown that the residue-residue contact information can be used to improve tertiary structure prediction [1]. The Critical Assessment of Techniques for Protein Structure Prediction (CASP) runs competition over residue-residue contact prediction in addition with direct tertiary structure prediction. Note that residue indicates amino-acids in the literature.

Contact-map prediction can be seen as an intermediate step for protein structure prediction. Beyond the scope of tertiary structure prediction, this work only involves predicting contact-map from protein sequence. Existing works can be broadly categorized as template based and sequence based contact prediction. Contacts are predicted from primary sequence or computed features from primary sequence in sequence based contact prediction [3, 4]. On the contrary, in template based contact prediction, a template structure is first found out using multiple sequence alignment using given protein sequence and that template is used for final structure prediction [5, 6].

The information of how a protein will be folded is contained by its primary sequence (for instance amino-acids, also known as residues), according to Anfinsen's thermodynamic hypothesis [7, 8]. As a result, a bunch of energy-driven approach has been proposed for finding stable protein structure by exploiting energy search space. In addition to that, [4] uses neural network for contact-map prediction,

[2] uses generative adverserial network, and [9] predicts novel protein design using pariwise distance of residues as an intermediate step.

Our work is a sequence based residue-residue contact-map prediction using variational autoencoder [10] given primary sequence of the respective protein. First we take two sub-sequences of a given primary sequence. The encoded version of the sub-sequences are passed through the network. Before using basic variational autoencoder [10] model, we put a fully connected layer of some fixed size. Then the output of the fully connected layer is passed through the netwrok. We do an ablation study on loss function between reconstruction loss and KL-divergence (KLD) loss with reconstruction loss. We found reconstruction loss works better in generalizing.

2. Methodology

2.1. Datasets and features

Following [4], the dataset is downloaded from Protein Data Bank using advance search by 30% sequence similarity, X-ray resolution of 0-2 Å and 40-300 sequence length. Since the length of the downloaded protein id list is more than 10000, we keep in total 1500 protein ids. Then this datasets is divided into 60% training, 20% validation and 20% test datasets.

Although previous work, such as DNCON [4], uses predicted secondary structure, solvent accessibility, several statistical potential abilities as their input feature space, they never mentioned of using primary sequences as input. In our work, we only use 1-hot encoding of primary sequences as input features and the ground truth contact-map for evaluating the prediction if corresponding contact-map of the protein structure is known using X-ray crystallography wet-lab approach to train our model.

The sequence lengths of proteins are not fixed, so we extract subset of amino-acids of $window\_size$ in incremental order. We also set $window\_stride$ for moving current window in the next position as $t_i = t_{i-1} + window\_stride$ if the previous position of the window was at t_{i-1}. Default $window\_stride$ is the fourth of the $window\_size$. We use $window\_size$ as 32 throughout the work.

In this study, we use contact-map definition as two amino-acid residues are in contact if the distance between their C_β atoms (C_α is used for Glycine) is < 12Å [4]. Note that, [11] says various contact definitions have been proposed: The distance between the $C_\alpha - C_\alpha$ atom with threshold $6 - 12$Å; distance between $C_\beta - C_\beta$ atoms with threshold $6 - 12$Å (C_α is used for Glycine); and distance between the side-chain centers of mass. To generate ground truth contact-map from the given primary sequence, we download the protein information from protein data bank. We compute the distance between $C_\beta - C_\beta$ atoms (except for Glycine). Figure 1 shows the general idea of ground truth contact-map generation process from primary protein sequence. Finally we map the computed distance matrix into closeness matrix, where *1* indicates two residues locates in close and *0* indicates they are in distant relationship. We use 12Å as threshold for computing comtact-map from distance matrix.

2.2. Method and Training

We use variational auencoder model architecture as our model. We took the basic building block code from here [12]. We extract two input sub-sequences s_1 and s_2 from a protein's primary sequence. For reducing less computation overhead during training, we compute the 1-hot encoding of the protein sequence before starting training. Since the number of amino-acids are 20, each s_1 and s_2 are matrix of size $window\_size$ x 20. Then we put a fully connected layer $fc1$ where input size is $2 * window\_size$ and output size is 20. The reason of choosing 20 is to make the output as $20X20$ which is considered as residue-residue relationship in between those two sub-sequences. In general, the output of $fc1$ represents some arbitrary linear combination of s_1 and s_2.

The output of the $fc1$ is fed to encoder. Encoder has several building block. In each block the depth of the output is increased as shown in 2. Each building block has three components, such as $Conv2d, BatchNorm2d, LeakyReLU$. The input and output channels in first, second, third and forth block are 1:64, 64:128, 128:256 and 256:1024 respectively.

Then we ravel the output of the encoder and fed to a fully connected layer $fc2$. The size of the output sample is 512. From here, we generate mean and variance of size 50 of each from where we sampled our bottlenect layer z of size 50. Then we use two fully connected layers $fc4$ and $fc5$ to generate an output feature of size 1024 which we fed into decoder layer.

In the decoder layer, the size of the output channels are decreased. Like encoder, the decoder building block consists of three components, such as $ConvTranspose2d, BatchNorm2d, ReLU$. The input and output channels of first, second, third and forth blocks are 1024:512, 512:256, 256:128, 128:1. Finally the output is of size $32X32$, shown in 2 as red square block. Each component of the output matrix represents the closeness value between 0 and 1 inclusive. In the next step, we compute loss between predicted contact-map and ground truth contact-map.

2.3. Loss function

Now we have computed contact-map y and ground truth contact-map y'. From Figure 2 we can see, the predicted contact-map does not represent full contact-map. So we

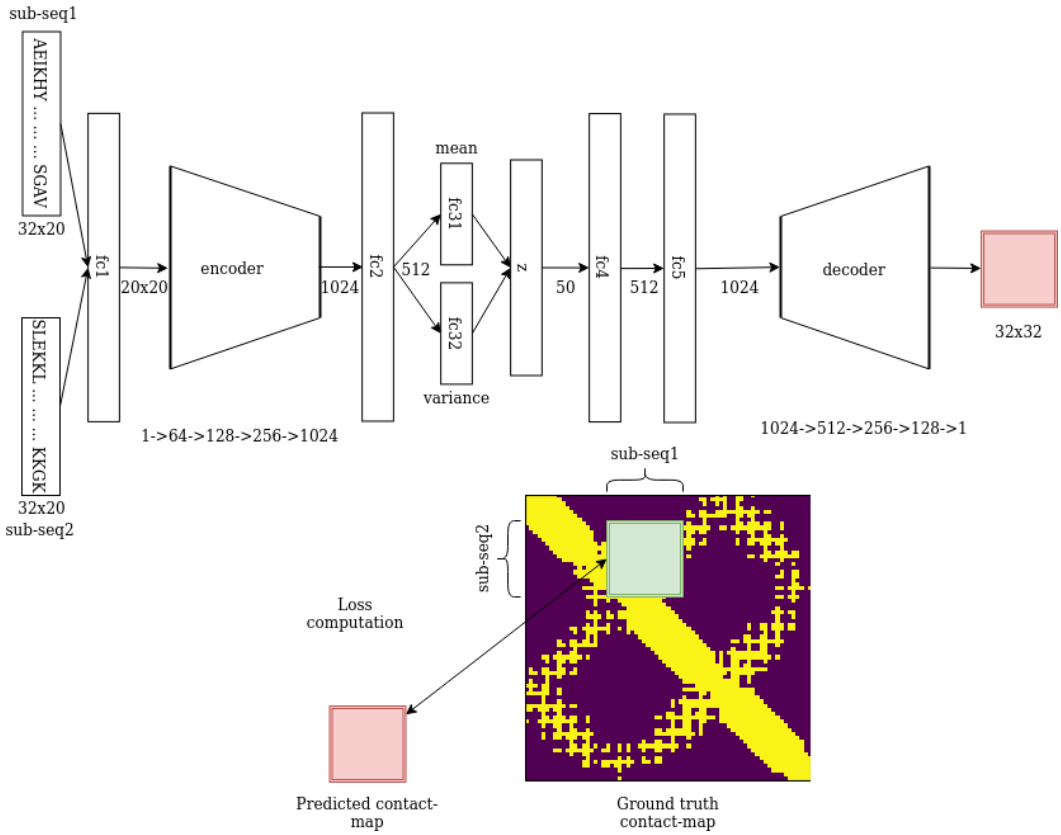

Figure 2. Model architecture and Loss computation

extract corresponding contact-map y' respect to $s1$ and s_2 from ground truth contact-map. We trained our model using two loss function. 1) binary cross entropy loss or reconstruction loss (BCE) and 2) binary cross entropy loss with KL-divergence loss (BCE+KLD).

$$BCE = L(y, y') = \sum_{i=0}^{N} (y_i * log(y'_i) + (1 - y_i) * log(1 - y'_i))$$

$$KLD = L(\mu, \sigma) = -.5 * \sum_{i=0}^{N} (1 + log(\sigma_i) - \mu_i^2 - \sigma_i^2)$$

$$loss = BCE \; or \; loss = BCE + KLD$$

Here μ and σ correspond to mean and variance while predicting y'. In the next section, we will discuss the quality evaluation of the models with test datasets.

3. Results and Analysis

In total we have 1500 proteins. We split them into 60% training (900 proteins), 20% validation (300 proteins) and 20% test (300 proteins) sets. Since all protein sequence length are not same, we take a fixed window size of 32. So for each protein we extract two subsequences of same length of 32 and we move forward. As a result, the size of our total dataset increases. Using window size of 32, our train and validation dataset size are 308360 and 111784 respectively. However, when we compute the quality of our model over test dataset, we generate full contact-map of the same size as the ground truth contact map.

We train several models using different combinations of hyperparameters. Table 1 shows such models. Model ids correspond to the output-log generated by each model while training. We tried bigger learning rate but they did not work well. All binary cross entropy loss used in the training are used as reduction to sum. Reduction to mean leads to not-a-number value, so we did not experiment further with reduction to mean.

We use argo [13] cluster to train our model. In addition to that, we use 64GB memory and a 16GB GPU worker to run our model. In average, it took about 8-9 hours for a single run for each model. Figure 3 shows the loss values versus number of epochs for different models. We see, M17 learns relatively slow and the loss of the M16 is the lowest. The final loss of the M24, M25 and M26 are ex-

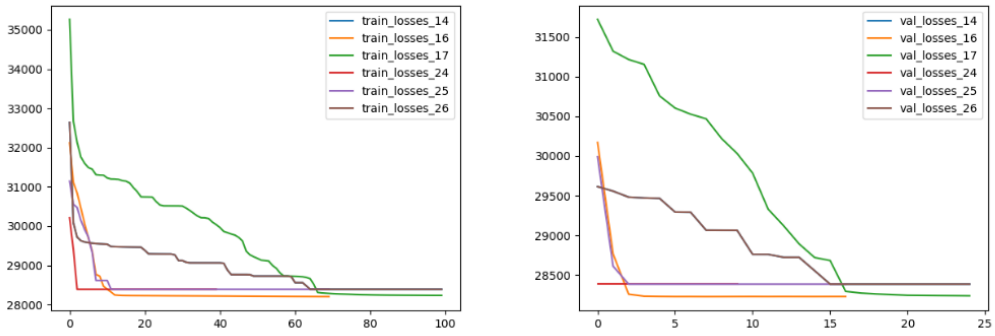

Figure 3. Train and validation losses

Model id	Initial Learning rate	epochs	Batch size	Loss function	Final train loss
M14	1e-5	40	30	BCE	28064.9
M16	1e-6	70	40	BCE	28204.6
M17	1e-7	100	40	BCE	28234.7
M24	1e-5	40	40	BCE + KLD	28391.3
M25	1e-6	70	40	BCE + KLD	28391.3
M26	1e-7	100	40	BCE + KLD	28391.3

Table 1. Model with hyper-parameters

actly same and M24 and M25 falls very sharp compared to M26. The reason behind this is of using KL divergence loss which uses mean and variance while computing loss function. This makes the model to learn so quickly, however when they generate no good results are found while evaluating the model quality.

Before computing the quality we need to map the predicted probability distributions in to 0-1 mapping. For that we use a threshold th, if predicted probability distribution is greater than th, we replace that value as 1, otherwise 0. The value we considered for th is 8e-5. Figure 4 shows some outputs based on th.

To generate the full predicted contact-map we use the following scheme. Suppose, we have a protein sequence of length l. So the ground truth contact-map size should be lxl. We take s_1 and s_2 sub-sequences from l and predict kxk contact-map based on th where $k < l$ and we use $k = 32$. In the next iteration, it will generate kxk matrix and so on. Then we merge this k_ixk_i with k_{i-1}xk_{i-1} where i is the iteration number. The merging procedure is putting one current matrix on top of the previous matrix. The merged matrix component is 1 if any of those two matrices have 1 in that position. Figure 4 shows four such examples. The right most column is the ground truth contact-map of the annotated four proteins. The left two columns are predicted by M14 and M16. However, contact-map predicted by M24 is just empty. From the training we can justify this. M24, M25 and M26 did not learn actually. After very few iterations it reached its lowest loss value because of KL divergence loss value. So we excluded M24, M25 and M26 from further model evaluation process.

#	M14	M16	M17
Precision	**0.65637**	0.60495	0.27361
Recall	0.62372	0.69132	**0.80867**
f_score	**0.73700**	0.73691	0.61137

Table 2. Model evaluation

For further study, we consider only M14, M16 and and M17. We compute precision, recall and f_score for the test dataset using these three models. Table 2 shows the comparison of three models using precision, recall and f_score value. We see M14 has the highest precision score of 65%. Although the outputs shown in Figure 4 are not so attractive, the precision score is quite high. Because, all amino-acids are close to itself which indicates 1's in the main diagonal and our model is able to learn this general characteristics. Additionally, almost all the space of the ground truth contact map are just empty and this is the default value of our model. Therefore, even the model is unable to learn outside the main diagonal, the precision is quite good. Moreover, M17 has the best recall score but lowest precision score. Thus, M17 has the lowest f-score and M14 has the highest f-score. Considering both precision and recall, we see that M14 is our best model in which case f-score is 73%.

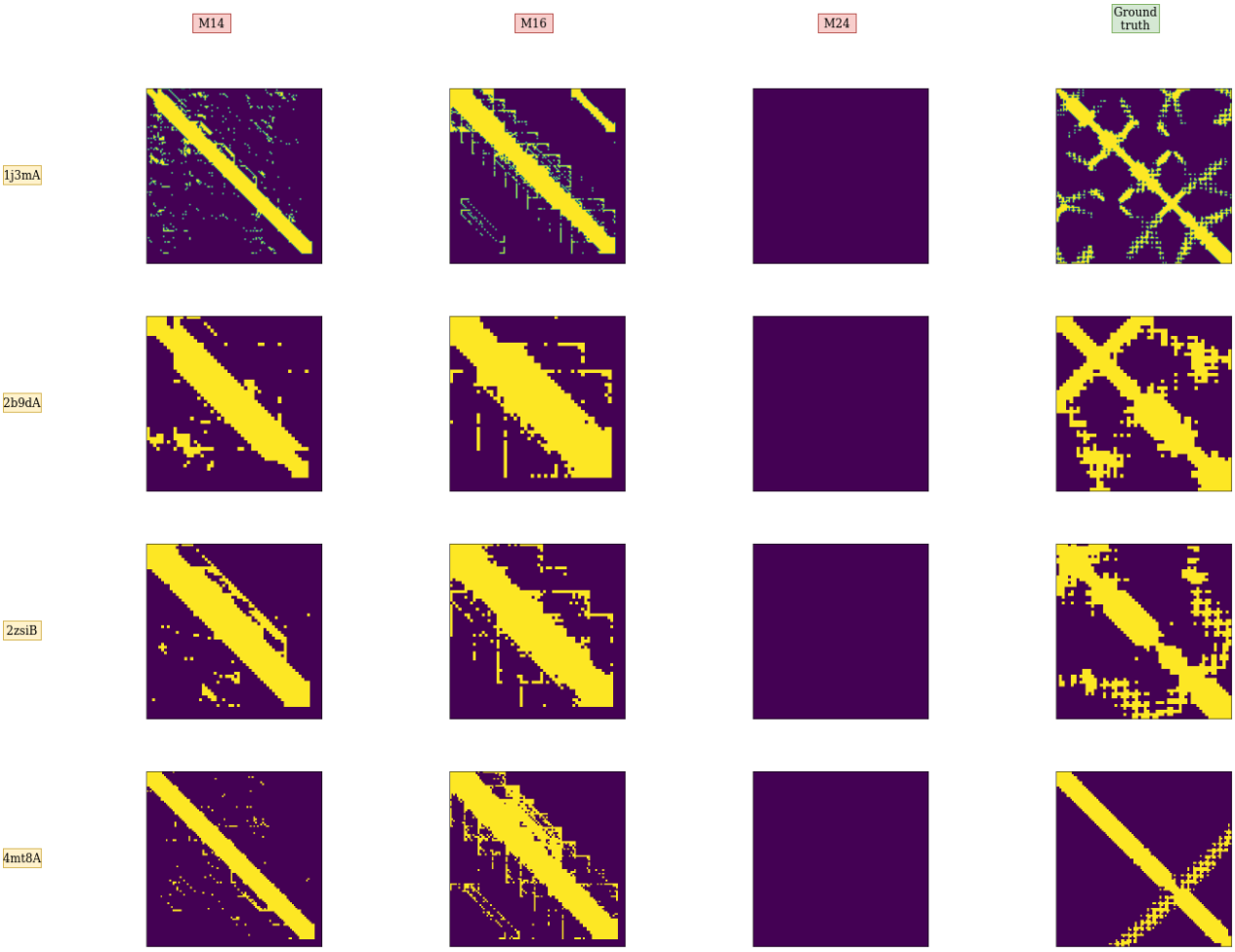

Figure 4. Contact-map generated by model versus ground truth contact-map

3.1. Future Direction

Since the primary sequence length varies for each protein and for that reason we leverage window scheme for input of the network and output for prediction, we could generate fixed sized input and output by interpolation for further learning global distributions of the input sets. The variational autoencoder network model is chosen as trial and error scheme. For further study why KL-divergence loss with BCE not working, we can analyze bottleneck layer as t-SNE visualization. Since the problem is defined as generative problem, we can also go for generative adversarial network for predicting contact-map. From this work, we can see it is unable to learn the distributions on the side of main diagonal, probably only sparse encoding of primary sequence is not enough to learn the contact-map distributions. We need to further study on state-of-the-art literature on this task so that we can find which model architecture and input data structure work best so far. Within our knowledge no one uses variational autoencoder with only encoded primary sequence on this task.

3.2. Miscellaneous

The code is made publicly available in here https://github.com/akabiraka/protein_project_1 with basic instructions about how to run. We run our code in GMU Argo cluster [13].

4. Conclusion and Future work

In this study we use variational autoencoder model to map from protein primary sequence to contact-map prediction. Contact-map could be thought of as an intermediate steps before generating tertiary or quarternary structure of a protein which conveys closeness map of amino-acids. The output of the model is far less down than its hypothesised position. We assume, with further features such as solvency

accessibility, given secondary structure, statistical potentials and so on, we can achieve comparative result with the existing state of the art methods.

References

[1] S. Wu, A. Szilagyi, and Y. Zhang. Improving protein structure prediction using multiple sequence-based contact predictions. *Structure*, 19(8):1182–1191, Aug 2011.

[2] Namrata Anand and Possu Huang. Generative modeling for protein structures. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7494–7505. Curran Associates, Inc., 2018.

[3] Jianlin Cheng and Pierre Baldi. Improved residue contact prediction using support vector machines and a large feature set. *BMC bioinformatics*, 8:113–113, Apr 2007. 17407573[pmid].

[4] Jesse Eickholt and Jianlin Cheng. Predicting protein residue–residue contacts using deep networks and boosting. *Bioinformatics*, 28(23):3066–3072, 10 2012.

[5] I. Ezkurdia, O. Gra?a, J. M. Izarzugaza, and M. L. Tress. Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8. *Proteins*, 77 Suppl 9:196–209, 2009.

[6] Bohdan Monastyrskyy, Krzysztof Fidelis, Anna Tramontano, and Andriy Kryshtafovych. Evaluation of residue-residue contact predictions in casp9. *Proteins*, 79 Suppl 10(Suppl 10):119–125, 2011. 21928322[pmid].

[7] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, Jul 1973.

[8] C. B. ANFINSEN, E. HABER, M. SELA, and F. H WHITE Jr. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 47(9):1309–1314, Sep 1961. 13683522[pmid].

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[11] Protein contact map. https://en.wikipedia.org/wiki/Protein_contact_map. Accessed: 2020-04-01.

[12] Sergii Kharagorgiev. Github VAE code link. https://github.com/coolvision/vae_conv, 2020. [Online; accessed 15-April-2020].

[13] Argo wiki. Argo Cluster. http://wiki.orc.gmu.edu/index.php/Main_Page, 2020. [Online; accessed 30-April-2020].