Improving Inter-Helix Contact Prediction With Local 2D Topological Information

Jiefu Li, Aman Sawhney, Jung-Youn Lee, and Li Liao *

Abstract—Inter-helix contact prediction is to identify residue contact across different helices in α-helical integral membrane proteins. Despite the progress made by various computational methods, contact prediction remains as a challenging task, and there is no method to our knowledge that directly tap into the contact map in an alignment free manner. We build 2D contact models from an independent dataset to capture the topological patterns in the neighborhood of a residue pair depending it is a contact or not, and apply the models to the state-of-art method's predictions to extract the features reflecting 2D inter-helix contact patterns. A secondary classifier is trained on such features. Realizing that the achievable improvement is intrinsically hinged on the quality of original predictions, we devise a mechanism to deal with the issue by introducing, 1) hybrid-cutoffs, which partially discretizing original predictions to leverage the usefulness of the existing information 2) fuzzy score, which assesses the quality of the original prediction, and selecting the residue pairs where improvement is more achievable. The cross-validation results show that the prediction from our method outperforms other methods including the state-of-the-art method (DeepHelicon) by a notable degree even without using the refinement selection scheme. By applying the refinement selection scheme, our method outperforms the state-of-the-art method significantly in these selected sequences.

Index Terms—Inter-helix contact prediction, Refinement selection, 2D contact model, Fuzzy score.

1 Introduction

Inter-helix contact prediction is to identify residue contact across different helices in α-helical integral membrane proteins. Knowing the residue contact is an important step toward better understanding the topology of membrane proteins and their cellular functions [1], [2], [3]. However, membrane proteins' structures are very difficult to be verified through X-ray crystallography [4]. As reported in [5], only 2% to 3% integral membrane proteins have been verified experimentally, which has motivated development of computational methods for predicting residue contact. Currently, the state-of-art method is DeepHelicon [6]. Like AlphaFold2 [7], which is an end-to-end learning and predicts accurate 3D models of globular proteins, DeepHelicon is also a deep learning based method that takes co-evolutionary features [8], [9], [10] as input to the neural network and outputs a 2D contact matrix of a membrane protein. While deep learning based methods have achieved remarkable success in many bioinformatics applications [11], not limited to protein structure prediction, the computational costs of deep learning have also increased dramatically [12]. As reported in [13], disregarding the model's training, making accurate predictions takes roughly $1K per sequence for AlphaFold. With all these success using deep learning techniques for protein structure prediction, computational cost will become a future challenge at some point, and the computational barrier for the research groups with limited computational power has already been noticed [13].

To mitigate the computational barrier issue, we here proposed a method to improve residue contact prediction by leveraging the existing state-of-art method's predictions and exploiting features that are not fully captured by the existing method with a simple hybrid-cutoffs technology and a novel refinement selection scheme. While DeepHelicon and other methods [14], [15], [16], [17] gather contact patterns based on sequence alignments to extract useful information, such as co-evolutionary features and evolutionary couplings, there is no method to our knowledge that directly tap into the contact map in an alignment free manner, despite it is obvious that, because of the periodic nature of helix turn, the neighborhood of a residue pair in a contact map can provide strong clue regarding whether the pair is a contact pair or not. So we proposed to exploit the 2D topological patterns in the neighborhood of any residue pair in order to improve the contact prediction. Specifically, using the ground truth, we build 2D contact models to extract the features reflecting 2D inter-helix contact patterns in the neighborhood of any residue pair. Concatenating these features with the prediction from an existing methods (we use DeepHelicon) as input to a secondary classifier (we use Random Forest), significant improvement is achieved (up to 14 percentage points in the top-L precision) over DeepHelicon's, indicating that the 2D contact models indeed capture valuable information for accurately predicting residue contact. Of course, in real world applications, the ground truth of residue contact is not available, instead it is exactly what the prediction is aimed at. To overcome this chicken-egg issue, we proposed to use other methods' prediction with applying partially discretization techqniue as surrogate to the ground truth,

- *J. Li is with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China, E-mail: lijiefu@usst.edu.cn*
- *A. Sawhney and L. Liao are with the Department of Computer and Information Science, University of Delaware, Newark,DE,19716. E-mail: asawhney@udel.edu , lliao@cis.udel.edu*
- *JY. Lee is with the Department of Plant and Soil Sciences, University of Delaware, Newark,DE,19716. E-mail: jylee@udel.edu*

and the results showed that we still outperformed these methods, though significantly less than using the ground truth contact in the neighborhood. Note that these other methods may be based on MSA, although the 2D contact models are alignment-free. Realizing that the achievable improvement is intrinsically hinged on how well the ground truth can be approximated by the surrogate, we devise a mechanism to deal with the issue by introducing i) hybrid-cutoffs, which partially discretize prediction scores to 0 or 1 or remain unchanged, and ii) fuzzy score, which can assess the quality of the current prediction, and selecting the residue pairs where improvement is more achievable. The cross-validation results show that the prediction from our method outperforms the state-of-the-art method DeepHelicon and other two methods by a notable degree even without using the refinement selection scheme. By applying our refinement selection scheme, which selects a subset of sequences to refine, our method outperforms the base method significantly in these selected sequences.

The rest of this paper is organized as the following. First, the proposed method will be described in details by introducing (i) hybrid-cutoffs technology, (ii) 2D contact model with its feature calculations, (iii) improve inter-helix prediction with its refinement selection scheme. Second, with the understanding of the proposed method, results from three experiments will be given. The first is a validation experiment which shows the usefulness of 2D contact models' feature for the task of inter-helix prediction refinement. The second experiment examines the idea of fuzzy score as a tool to assess original prediction's quality. The third experiment will demonstrate the refinement of prediction from DeepHelicon [6], DeepMetaPSICOV [18], and MetaPSICOV [16] with and without refinement selection scheme, and comparing them with only using original predictions' simple neighbourhood feature. Last, discussion and conclusion are given at the end.

2 METHOD

The proposed method is a framework designed to improve inter-helix contact predictions from some other existing method in order to achieve better performance as measured by the widely accepted top-L metric, via 2D contact models, hybrid-cutoffs, and refinement selection scheme. As such, the success of this method relies on three key ideas: (1) A prediction score partially discretizing method, which give upper and lower cutoffs to original prediction scores to round them into 1 or 0 or unchanged. (2) The usage of 2D contact models, which are used to generate useful refined features to capture inter-helix contact patterns. (3) A selection scheme based on the idea of fuzzy score, which assesses the quality of current predictions and helps select proper residue pairs/sequences most likely to benefit from the refined features.

Before getting into the details of the method, it is helpful to define the major notations used throughout this section. Our dataset contains N sequences. For the k^{th} sequence in the dataset, we use a square binary matrix C_k to denote the contact matrix of the sequence. Each element $C_k(i,j)$ can be either 0 representing non-contact or 1 representing contact for the residue pair $< i,j >$ in the folding of sequence k. Note that C_k and $C_k(i,j)$ are always referred to the ground truth information, i.e., their values are either 0 or 1, based on experimental data. In contrast, we use $\widetilde{C_k}$ and $\widetilde{C_k}(i,j)$ to refer to the contact prediction made by inter-helix contact prediction methods, and their values are a real number from 0 to 1.

Now, for each inter-helix residue pair $< i,j >$ in sequence k, we extract a $(2n+1) \times (2n+1)$ neighbor block from the matrix C_k centered at position $< i,j >$, where n defines the farthest neighboring residue from the center on either side, and we denote this block as $F_k^n(i,j)$, whose elements are $C_k(a,b)$, where $|i-a| \leq n$ and $|j-b| \leq n$. In the cases where a or b out of the boundary or $< a,b >$ is not an inter-helix pair, 0 values are assigned. Again, since $F_k^n(i,j)$ is extracted from C_k, it is a binary matrix. Similarly, such a matrix can be extracted for each position $< i,j >$ from a predicted contact map, and we denote it as $\widetilde{F_k^n}(i,j)$. When building the 2D structure model, calculating its features, and applying the refinement selection scheme, the center of $F_k^n(i,j)$ and $\widetilde{F_k^n}(i,j)$ needs to be excluded, and we use $E_k^n(i,j)$ and $\widetilde{E_k^n}(i,j)$ to denote such center excluded neighbor block in the ground truth and predicted contact matrix respectively. When used in features calculation or as feature directly, these matrices are flattened row by row into an array of size $(2n+1)^2-1$. A graphical representation with $n = 1$ (i.e., 3 x 3 neighborhood block) of above procedure is shown in Figure 1. For readers reference, the summary of the notations introduced above and several others appeared later are shown in Table 1.

2.1 Hybrid Cutoffs

First, it is worthy to mention that the simple neighbor feature, i.e. $\widetilde{E_k^S}(i,j)$ has already contained useful information; a classifier trained on such simple feature can make improvements over original inter-helix contact prediction in most cases already. The improvement can be further enhanced by partially discretizing original predictions as follows:

$$\begin{aligned} CT(score, lower, upper) =& 1 \text{ if } score > upper \\ =& 0 \text{ if } score < lower \\ =& score \text{ otherwise} \end{aligned}$$

With $CT(score, lower, upper)$ defined as such, finding the optimal lower and upper cutoffs is simply by grid search in defined cutoff searching space $(0 \leq lower \leq upper \leq 1)$, and can be expressed by the following mathematical formula with original predictions denoted by $\widetilde{Y}$, and ground truth label denoted by Y in training set.

$$(lower^*, upper^*) = \text{argmax } corr(CT(\widetilde{Y}, lower, upper), Y)$$

After finding the optimal cutoffs via a training set, applying $CT(score, lower^*, upper^*)$ to $\widetilde{E_k^S}(i,j)$ to have a hybrid-discretized neighbor features, such feature is denoted by $\ulcorner \widetilde{E_k^S}(i,j) \lrcorner$. These hybrid-discretized features will be part of our whole feature, and we will show that such partially discretized feature alone will better than simply applying $\widetilde{E_k^S}(i,j)$.

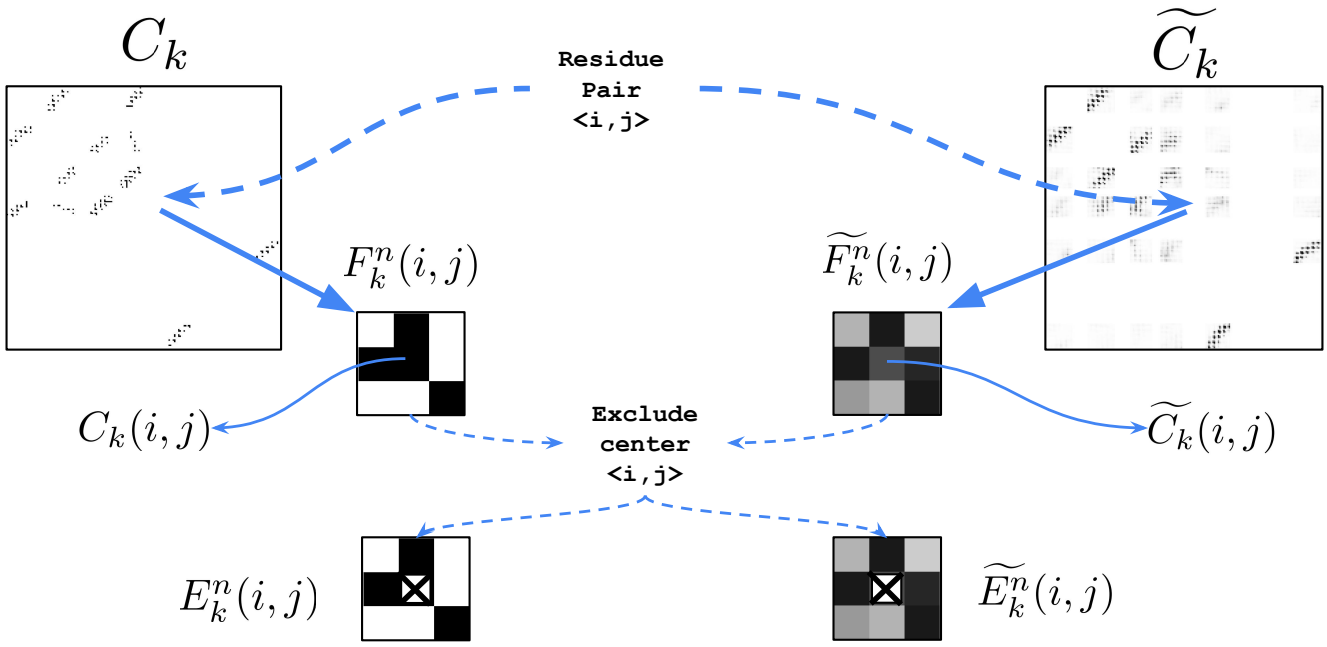


Fig. 1. Diagram of $F_k^n(i,j)$, $\widetilde{F_k^n}(i,j)$, $E_k^n(i,j)$, $\widetilde{E_k^n}(i,j)$ extraction from C_k (contact map from ground truth) and $\widetilde{C_k}$ (contact map from prediction) with $n = 1$

TABLE 1
Notations

Symbols	Explanations
C_k	A square binary matrix represents the k^{th} sequence's 2D contact map
$C_k(i,j)$	C_k's element, which represents contact/non-contact of residue pair $< i,j >$ in sequence k.
$\widetilde{C_k}$	A square matrix represents the k^{th} sequence's predicted 2D contact map, values are from 0 to 1.
$\widetilde{C_k}(i,j)$	$\widetilde{C_k}$'s element, which represents contact estimation of residue pair $< i,j >$ in sequence k.
$F_k^n(i,j)$	$(2n+1) \times (2n+1)$ matrix extracted from C_k, centered at $< i,j >$.
$E_k^n(i,j)$	A flattened vector in size of $(2n+1)^2 - 1$ from $F_k^n(i,j)$ by excluding the center.
$\widetilde{F_k^n}(i,j)$	$(2n+1) \times (2n+1)$ matrix extracted from $\widetilde{C_k}$, centered at $< i,j >$.
$\widetilde{E_k^n}(i,j)$	A flattened vector in size of $(2n+1)^2 - 1$ from $\widetilde{F_k^n}(i,j)$ by excluding the center.
$M_n(+/-)$	2D contact model with parameter n.
$fzScore(*)$	Fuzzy score, which defines the fuzzy level of $*$.
$feature_k^{n+/-}(i,j)$	2D contact model's feature of sequence k residue pair $< i,j >$ with parameter n.

2.2 2D Contact Models

To build 2D contact models, a separate training set with ground truth label is needed. Given N number of training sequences, $F_k^n(i,j)$ can be extracted for all inter-helix residue pairs $< i,j >$ of sequence k from C_k for all $k \in N$. A 2D contact model has one parameter n, which determines the size of the neighborhood block centering at any residue pair in the contact map: $(2n+1) \times (2n+1)$. The key idea of the 2D contact model is to capture the general contact patterns in the contact map surrounding any given residue pair. To avoid sequence alignment and other details such as residue identities, we just calculate the probability of each position within the $(2n+1) \times (2n+1)$ neighborhood block of any residue pair to be a contact. As we believe such a probability distribution across the neighborhood of a contact residue pair is likely different from that of a non-contact residue pair, we build two 2D contact models: one for inter-helix contact residue pairs, and the other one for inter-helix non-contact residue pairs, and are denoted by $M_n(+)$ and $M_n(-)$ respectively. The 2D contact models can be built by approximating the probability with frequency using the training data. To build $M_n(+)$, which is a $(2n+1) \times (2n+1)$ matrix, we go through the 2D ground truth contact map of training sequence k, for each contact point $< i,j >$ in sequence k, namely $C_k(i,j) = 1$, we collect its neighborhood block $E_k^n(i,j)$ and sum these $(2n+1) \times (2n+1)$ matrix blocks, element-wise, over all contact points in sequence k, and then over all N training sequences in the dateset. This way, each element in the summed matrix block gives the count of contact points at the corresponding position in the neighborhood of contact residue pairs, which, after normalized by the total number of contact residue pairs, gives the frequency (thus probability) of the corresponding neighbor of a contact residue pair also to be a contact point. In a similar way, we can build $M_n(-)$, which is a $(2n+1) \times (2n+1)$ matrix, each element is the probability of the corresponding neighbor of a non-contact residue pair to be a contact point. The mathematical formulas for

calculating the 2D contact structure models $M_n(+)$ and $M_n(-)$ are given as follows.

$$M_n(+) = \frac{\sum_{k=1}^{N} \sum_{i,j} E_k^n(i,j) \cdot C_k(i,j)}{\sum_{k=1}^{N} \sum_{i,j} C_k(i,j)}$$
$$M_n(-) = \frac{\sum_{k=1}^{K} \sum_{i,j} E_k^n(i,j) \cdot (1 - C_k(i,j))}{\sum_{k=1}^{K} \sum_{i,j} (1 - C_k(i,j))} \quad (1)$$

Note that $M_n(+)$, $M_n(-)$, and $E_k^n(i,j)$ in the above equations can be viewed as $(2n+1) \times (2n+1)$ matrices with empty center as a whole (so that there are $(2n+1)^2 - 1$ total elements), the summation is element-wise matrix addition, and $\cdot$ represents that the matrix is multiplied by a single value $C_k(i,j)$. It is critical to note that, $E_k^n(i,j)$ instead of $F_k^n(i,j)$ is being used because to build $M_n(+)$ and $M_n(-)$, the corresponding centers of $F_k^n(i,j)$ are either all contact or non-contact, which have been captured by contact or non-contact models already. In sum, the 2D contact models $M_n(+)$ and $M_n(-)$ have the same dimension as E_n^k and consist of contact frequencies position-wise in the $(2n+1) \times (2n+1)$ neighborhood centering a contact or non-contact residue pair respectively.

We hypothesize that, the 2D contact models built from a decently large training set capture general patterns and therefore can be generalized to other protein sequences, not only those present in the training set, but also those in a separate test set. In a test sequence k, for each inter-helix residue pair $< i,j >$, we compare the ground truth contact in the neighborhood to the 2D contact models; intuitively, if its neighborhood $E_k^n(i,j)$ is more similar to the $M_n(+)$ than to $M_n(-)$, then $< i,j >$ is more likely to be a contact point. Consider them respectively as probability distribution of contacts in the neighborhood of $< i,j >$ and collectively in the training data, the similarity can be measured by using the KL divergence. Instead of relying on one neighborhood of a big size, which can give just two KL divergence scores, we use a set of neighborhoods with increasing size $n = 1, .., S$ to get $2S$ KL divergence scores as features to train a secondary classifier. Because a smaller neighborhood is enclosed in a bigger neighborhood, to avoid redundant information due to overlap, for each neighborhood we calculate the KL divergence between the perimeters of the neighborhood block $E_k^n(i,j)$ and contact model block $M_n(+/-)^T$. Here, we denote the perimeter of $E_k^n(i,j)$ and of $M_n(+/-)^T$ 1 by $\overline{E_k^n(i,j)}$ and $\overline{M_n(+/-)^T}$ respectively, and with normalization so that all elements on the perimeter sum to be 1.

$$feature_k^{n+}(i,j) = \sum \overline{E_k^n(i,j)} log(\overline{E_k^n(i,j)} / \overline{M_n(+)^T})$$
$$feature_k^{n-}(i,j) = \sum \overline{E_k^n(i,j)} log(\overline{E_k^n(i,j)} / \overline{M_n(-)^T}) \quad (2)$$

For above computation, all operations are element-wise. Therefore, the final results of these vectors generates a single number, $feature_k^{n+}(i,j)$ or $feature_k^{n-}(i,j)$. The values of $feature_k^{n+}(i,j)$ and $feature_k^{n-}(i,j)$ can be interpreted as a distance score, which indicates how likely $C_k(i,j) = 1$ or 0 respectively, given the perimeters of 2D contact models $M_n(+/-)$ and inter-helix residue pair $< i,j >$'s neighbours contact $E_k^n(i,j)$.

By stacking a set of neighborhoods with increasing size $n = 1$ to S, using the perimeter of each, we effectively cover the neighborhood of size S with more details (i.e., 2S KL scores) than a single neighborhood of size S can provide (i.e., just 2 KL scores). In the next subsection, we show how to use such $2S$ KL scores as features to train a secondary classifier to predict residue contact, and the results in Table 2 show great improvement as compared to DeepHelicon, and hence strongly support our hypothesis about the 2D contact models.

However, the above definition of $feature_k^{n+}(i,j)$ or $feature_k^{n-}(i,j)$ requires knowing the ground truth contact in the neighborhood of $< i,j >$, which is not available but rather what a prediction method is aimed at in a real world scenario. To overcome this "chicken-egg" issue, we propose to use the predicted contact map from a method with good performance (e.g., DeepHelicon) as the surrogate to the ground truth contact map. That is, we can use 2D contact models to calculate KL divergence from other method's prediction about the inter-helix of residue pairs, and treat these divergence as refined features for a secondary classifier. Specifically, the calculation is done as follows:

$$\widetilde{feature_k}^{n+}(i,j) = \sum \overline{\widetilde{E_k^n}(i,j)} log(\overline{\widetilde{E_k^n}(i,j)} / \overline{M_n(+)^T})$$
$$\widetilde{feature_k}^{n-}(i,j) = \sum \overline{\widetilde{E_k^n}(i,j)} log(\overline{\widetilde{E_k^n}(i,j)} / \overline{M_n(-)^T}) \quad (3)$$

Here, $\widetilde{E_k^n}(i,j)$ is built from the predicated contact map $\widetilde{C}_k$ for test sequence k by an existing computational method. In our case, such existing methods include DeepHelicon, DeepMetaPSICOV, and MetaPSICOV.

2.3 Inter-helix Prediction

Now we are ready to describe the procedure for improved inter-helix prediction with a secondary classifier (Random Forest), consisting the following steps: applying hybrid-cutoffs to simple neighbourhood feature, contact model feature preparation, model training and testing, and refinement selection.

Second, using the an independent training dataset and the formulas in Eq 1, we build multiple 2D contact models $M_n(+)$ and $M_n(-)$ with different sizes: $n = 1...S$, and then use Eq 2 to generate the refined features for inter-helix residue pair$< i,j >$ in sequence k, denoted as $RF\_feature_k^S(i,j)$ in the following form:

$$\begin{aligned} RF\_feature_k^S(i,j) = concat[& \\ & feature_k^{1+}(i,j), feature_k^{1-}(i,j), \\ & feature_k^{2+}(i,j), feature_k^{2-}(i,j), \\ & ..... \\ & feature_k^{S+}(i,j), feature_k^{S-}(i,j), \\ & \ulcorner \widetilde{E_k^S}(i,j) \urcorner, \widetilde{C_k}(i,j)] \end{aligned} \quad (4)$$

Note that, this refined feature set consists of scores calculated according to Eq 2 with 2D contact models of different sizes from 1 to S, in order to capture hidden contact patterns in a wide range of neighborhood. This is more advantageous than using just models of size S, because although neighborhood of size S encompasses all

neighborhoods of size smaller than S, a single score over a large neighborhood may hide detailed variations within it. On the other hand, by stacking features calculated from 2D contact models of various sizes we can "expose" the detailed and nuanced patterns for the secondary classifier to learn.

With $RF\_feature_k^S(i,j)$ ready for each inter-helix residue pair of each sequence, the training and testing procedures in a cross-validation setting are standard: we randomly split the dataset into 10 subsets, merge 9 subsets to train the random forest classifier and test the trained classifier on the reserved subset, and rotate to use each of the 10 subset once as test set. Since the proposed method is designed to improve predictions made by some existing computational methods, it is reasonable to assume that the predictions are significantly better than random tossup. Therefore, in practice, users may choose to only improve top ranked inter-helix residue pairs, instead of all inter-helix residue pairs, to save computational cost. In our experiments, we chose to improve top $3L$ ranked residue pairs from DeepHelicon, where L is the total length of the helices in a particular sequence. The results, listed in Table 2, show that our method achieves significant improvement (e.g., up to 14% in top-L precision and 9% in top-L recall) over DeepHelicon's. This is clearly a very strong support of our hypothesis that contact patterns in the neighborhood of a residue pair contain valuable information that can help, when properly used, significantly improve contact prediction.

Of course, the construction of the refined features in Eq 2 requires knowing the ground truth contact map, which is not available in a real world scenario, actually the contact map is what the prediction method is aimed at predicting. To overcome this "chicken-egg" issue, we propose to use the predicted contact map from a decently performing method (e.g., DeepHelicon) as the surrogate to the ground truth contact map. Specifically,

$$\begin{aligned} RF\_feature_k^S(i,j) = concat[\\ & \widetilde{feature}_k^{1+}(i,j), \widetilde{feature}_k^{1-}(i,j), \\ & \widetilde{feature}_k^{2+}(i,j), \widetilde{feature}_k^{2-}(i,j), \\ & ..... \\ & \widetilde{feature}_k^{S+}(i,j), \widetilde{feature}_k^{S-}(i,j), \\ & \ulcorner \widetilde{E_k^S}(i,j) \lrcorner, \widetilde{C_k}(i,j)] \end{aligned} \quad (5)$$

Here everything is the same as Eq 4, except that $feature_k^{1-}(i,j)$ is replaced with $\widetilde{feature}_k^{1-}(i,j)$.

Due to the use of predicted contact map in calculating the refined features, it is expected that the achievable improvement is intrinsically hinged on how well the ground truth can be approximated by the surrogate. Even though we select a state-of-the-art method with a good overall performance, the quality of prediction in the predicted contact map can vary from position to position. We devise a mechanism to deal with this issue by introducing a fuzzy score, which can assess the quality of the current prediction, and selecting the residue pairs/sequences where improvement from the refined features is more achievable.

Let us assume that the prediction by the existing method (e.g., DeepHelicon) for a residue pair is a score in the range from 0 to 1, which can be interpreted as the likelihood that the pair is a contact, with 0.5 being a tossup, or most uncertain prediction. The fuzzy score for a given position $< i,j >$ in a predicted contact map $\widetilde{C_*(i,j)}$ is calculated as follows:

$$fzScore(\widetilde{C_*(i,j)}) = \min(1 - \widetilde{C_*(i,j)}, \widetilde{C_*(i,j)}) \quad (6)$$

When $\widetilde{C_*(i,j)} = 0.5$, then $fzScore(\widetilde{C_*(i,j)}) = 0.5$, indicating the highest fuzzy (uncertain) level. When $\widetilde{C_*(i,j)} = 1$ or 0, $fzScore(\widetilde{C_*(i,j)}) = 0$, indicating the lowest fuzzy (uncertain) level.

Our rationale for using such fuzzy scores is that, given predictions of a residue pair $\widetilde{C_k}(i,j)$ in sequence k and its neighborhood $\widetilde{E_k^n}(i,j)$, only when the fuzzy score of $\widetilde{E_k^n}(i,j)$ is on average less than the fuzzy score of $\widetilde{C_k}(i,j)$ by a certain degree, then the prediction of neighborhood contact is significantly better off than random tossup and hence can be used as good surrogate to the ground truth so that the refined feature $feature_k^{n+}(i,j)$ can be better approximated as $\widetilde{feature_k}^{n+}(i,j)$. So, based on this rationale, we define the fuzzy score of $\widetilde{E_k^n}(i,j)$ as just the average of each element's fuzzy score as follows.

$$fzScore(\widetilde{E_k^n}(i,j)) = \frac{\sum_{0<|a-i|\le n, 0<|b-j|\le n} fzScore(\widetilde{C_k}(a,b))}{(2n-1)^2 - 1}$$

For each inter-helix residue pair $< i,j >$ and its neighbours bounded by a $(2n+1) \times (2n+1)$ box, we have a pair fuzzy scores that measure the fuzzy level of $\widetilde{C_k}(i,j)$ and $\widetilde{E_k^n}(i,j)$. Their difference is denoted by $\partial fzScore_k^n(i,j)$ and calculated as:

$$\partial fzScore_k^n(i,j) = fzScore(\widetilde{C_k}(i,j)) - fzScore(\widetilde{E_k^n}(i,j))$$

The key idea of using the fuzzy score is: when $\partial fzScore_k^n(i,j)$ is higher than a threshold, it indicates that $\widetilde{C_k}(i,j)$ is more fuzzy than $\widetilde{E_k^n}(i,j)$, and applying the 2D contact model onto the neighbours of $< i,j >$ to produce refined features is more likely to give rise to better prediction at $< i,j >$. In practice, an empirical procedure can be used to optimize the threshold of $\partial fzScore_k^n(i,j)$, by maximizing the performance gain from the refined features, using an independent dataset. In this dataset, each residue pair $< i,j >$'s fuzzy score difference $\partial fzScore(i,j)$ is computed, here the sequence index k and the neighborhood size n are omitted for simplicity. We like to know which residue pair's fuzzy score difference, if used as a threshold, can lead to maximal performance gain. Let's say $< i,j >$ is that residue pair. Then for all other residue pairs $< i',j' >$ whose fuzzy score difference larger than $\partial fzScore(i,j)$ we will assess the collective performance difference from applying the refinement versus not applying the refinement as $L1(i,j) = \sum_{i'j': \partial fzScore(i',j') > \partial fzScore(i,j)} [err_1(i',j') - err_2(i',j')$, where $err_1(i',j')$ is the prediction error when

refined features are used, and $err_2(i', j')$ the prediction error when refined features are not used for each residue pair $< i', j' >$. The smaller the $L1$ is, the more benefit we get from applying the refinement. Similarly, for all other residue pairs $< i", j" >$ whose fuzzy score difference smaller than $\partial fzScore(i, j)$ we also assess the performance difference from applying the refinement versus not applying the refinement $L2(i, j) = \sum_{i"j": \partial fzScore(i", j") < \partial fzScore(i,j)} [err_1(i", j") - err_2(i", j")]$, where $err_1(i", j")$ is the prediction error when refined features are used, and $err_2(i", j")$ the prediction error when refined features are not used for residue pair $< i", j" >$. The smaller the L2 is, the more benefit we get from applying the refinement; reversely the larger the L2 is, the less benefit we get from applying the refinement. So, $L(i, j) = L1(i, j) - L2(i, j)$ tells us the net effect of using residue pair $< i, j >$'s fuzzy score difference as the cutoff: the lower the L is, the more benefit we get from applying the refinement to these $< i', j' >$ and not applying refinement to these $< i", j" >$. We calculate $L(i, j)$ for all residue pairs $< i, j >$ and choose the one that has the minimal L.

Specifically,

1) Calculate the prediction error: $err_1(i, j)$ when refined features are used, and the prediction error: $err_2(i, j)$ when refined features are not used for each residue pair $< i, j >$.
2) Calculate the difference between these two errors: $\partial err(i, j) = err_1(i, j) - err_2(i, j)$ for each residue pair.
3) With a given fuzzy score cutoff (arbitrarily indexed by $< *, * >$) $\partial fzScore(*, *)$, calculate the corresponding refinement loss defined as: $L(*, *) = \sum_{i'j': \partial fzScore(i', j') > \partial fzScore(*,*)} \partial err(i', j') - \sum_{i"j": \partial fzScore(i", j") < \partial fzScore(*,*)} \partial err(i", j")$.
4) Repeat above step for all possible fuzzy score cutoffs.
5) Find the minimal value of L, and its corresponds fuzzy score ∂fz_{min} is the final threshold to determine when refinement method applied.

It is optional to do selection more aggressively, i.e. select the ones with larger level of improvements, by only a small modification of the 2^{nd} step: $\partial err(i, j) = err_1(i, j) - [err_2(i, j) - C]$, where C is a positive constant to make the original prediction error $err_2(i, j)$ artificially smaller and hence only when the refinement error is even smaller , i.e., $err_1(i, j) < err_2(i, j) - C$, then $\partial err(i, j) < 0$ is satisfied.

Note that, the specifics of this procedure may vary, mainly for lower computational costs and/or better performance, via alternative metric, such as per residue pair or per sequence. Validation experiment of this procedure is demonstrated in Results section.

3 Results

In this section, we first describe in details the datasets used for this study, and the results from four experiments. The first experiment is served to demonstrate the utility of 2D contact structure model and establish the upper bound of this refinement method. **The second experiment is to validate the hypothesis that 2D contact model without the ground truth can still lead to improved prediction, with the help of fuzzy score and refinement selection scheme, as compared with DeepHelicon's results**. The third experiment is to compare the results of using 2D contact model, with and without refinement selection technique with three different methods: DeepHelicon, DeepMetaPSICOV, and MetaPSICOV. Finally, last experiment is to compare the improvements over original predictions of three different features: i) simple neighbor feature alone ii) simple neighbor feature with applying hybrid-cutoffs, and iii) the main feature used here as shown in Eq 5, which consists contact models feature, simple neighbor feature with applying hybrid-cutoffs, to show the usefulness of hybrid-cutoffs technique and the contact model features.

3.1 Dataset

Since our method mainly refines the predictions from DeepHelicon [6], the same datasets are adopted. There are three different datasets: TRAIN, TEST, and PREVIOUS. They contain 165, 57, and 44 bitopic or polytopic α-helical transmembrane proteins respectively. TRAIN and TEST datasets were extracted from PDBTM database [19] by DeepHelicon group. PREVIOUS is a combined dataset from TMhhcp [20] and MemConP [21].

Here, we use the TRAIN as an independent dataset to build 2D contact structure model with different values of the parameter n. And combining the TEST and PREVIOUS datasets (101 sequences in total) to evaluate the proposed method by 10 fold cross validation. The combined dataset is called DATA(101) hereafter.

3.2 2D Contact Model Validation

In this experiment, the main goal is to demonstrate the usefulness of the 2D contact structure model and its features in task of inter-helix prediction. In addition to the purpose of validation, this experiment also computes the upper bound with the given data.

This experiment and later two used TRAIN dataset to build the 2D contact models, and DATA(101) to perform 10-fold cross validation. The classifier used is random forests with 500 trees. The hyper-parameter S in Eq 4 is chosen to be 7.

The results are shown in Table 2. The performance of inter-helix predictions is evaluated using the metric of top-L precision and recall, adopted from [6] and commonly used in the community. The term L refers to the total length of a particular sequence's α-helix. Instead of setting a particular cutoff on the prediction score, top L/1, L/2, L/5, or L/10 in the list of residue pairs ranked by the prediction score are chosen as decision boundary's cutoff, namely predicted as positive. For example, with a given bitopic or polytopic transmembrane protein, the total length of its α-helices is X, the top-L/5 evaluation means ranking prediction scores in descending order, and considering the top $X/5$ predictions as predicted contact inter-helix residue pairs, remaining as predicted non-contact inter-helix residue pairs. For comparison purpose, we adopt top-L evaluation with L/1, L2, L/5, L/10 to evaluate precision, recall, F_1 score, $F_{0.35}$ score, and Matthews correlation coefficient (MCC). With multiple sequences in the testing set, average of precision, recall,

TABLE 2
Top-L Comparisons of the refinement method denoted as RM(with 2D contact model features calculated from ground truth contact map) and DeepHelicon (DH) without refinement selection on DATA(101)

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	RM	**74.36%**	**53.13%**	**59.70%**	**70.15%**	**60.80%**
	DH	62.55%	44.49%	50.13%	58.99%	50.68%
L/2	RM	**89.60%**	**33.81%**	**47.05%**	**73.73%**	**53.14%**
	DH	76.30%	27.92%	39.39%	62.49%	44.41%
L/5	RM	**96.65%**	**15.10%**	**25.21%**	**57.82%**	**36.70%**
	DH	85.40%	12.75%	21.63%	50.57%	31.76%
L/10	RM	**98.28%**	**7.83%**	**14.14%**	**40.96%**	**26.57%**
	DH	89.09%	6.68%	12.25%	36.43%	23.50%

$F1$-scores, and MCC are computed as final results. The definition of F-scores and MCC is given as follows.

$$F_\beta = (1+\beta)^2 \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

$$\text{MCC} = \frac{tp \cdot tn + fp \cdot fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}$$

where tp is the number of true positive, tn is the number of true negative, fp is the number of false positive, and fn is the number of false negative.

It is clear from Table 2 that, with 2D contact structure model's feature calculated from the ground truth contact map, all top-L performances are strictly better than the original results across different L's. This validates our hypothesis and the utility of applying 2D contact topological model in the neighborhood of a residue pair for improved contact prediction. This ground-truth based 2D model established the upper bound for performance gain from leveraging the neighborhood information in our proposed method. In real world applications, the ground truth contact map is not available and only an estimation of it is given. In such cases, as shown later, the performance drops from the upper bound but is still better off than the original inter-helix contact prediction made by DeepHelicon.

3.3 Fuzzy Score Validation

The goal of this second experiment is to evaluate the fuzzy score and test the 5-step procedure for optimizing a fuzzy score cutoff used by the refinement selection scheme. The overall setting of this experiment is the same as the previous one expect that the evaluation criterion is absolute error, which measure performance per residue pair, instead of top-L, which measures performance per sequence, and the features used here are all calculated from DeepHelicon's predictions.

Let X denote the collection of original prediction results from DeepHelicon, X_r denote the collection of inter-helix prediction refinement results, and Y denote the ground truth label. X, X_r, Y are arrays with length equal to total inter-helix residue pairs in DATA(101). Then, define:

$$Error = |X - Y|$$
$$Error_r = |X_r - Y|$$
$$\partial Error = Error - Error_r$$

Let ∂e be an element of $\partial Error$. When ∂e is greater than 0, original error is greater than the refinement's, which is our favourable case. Otherwise, it indicates refinement model should not be used for the residue pair. Equivalent to the 5-step procedure in refinement selection scheme, we plot the curve for cutoff $\partial fzScore$ vs cumulative error with sorting $\partial fzScore$ in ascending order, and the result is shown in Figure 2. The curve is almost perfectly convex, and the red "X" indicates the minimum point of the curve, which is to be used as the threshold or cutoff of $\partial fzScore$ by the refinement selection scheme to decide whether the refined features from applying 2D contact model should be used or not. The left region of the threshold goes down almost monotonically indicating majority ∂e is less than 0. The right region of the threshold goes up almost monotonically indicating majority ∂e is greater than 0. This result validates our hypothesis, and supports the 5-step procedure for finding a cutoff used by the refinement selection scheme. It is worthy to note that $\partial fzScore$ is plotting against the cumulative error not individual errors, in order to identify a cutoff on the $L(\partial fz_{min})$ computed per the 5-step procedure as the minimum point of the cumulative error curve.

3.4 Inter-Helix Prediction Refinement and Refinement Selection Scheme

This third experiment contains four parts. The first part is to test the performance of contact prediction using the refined features indiscriminatively for all sequences. The second part is to test the performance of contact prediction when the refined features are used selectively by both the normal and aggressive settings of refinement selection scheme. In aggressive setting, the constant C is picked as the average of $err_2(i) - err_1(i)$, which is a positive value. The third part is the complement results of the second part, namely, the performance of the sequences that are not selected in the second part. The last part is a comparison of the improvements in first and second part. This comparison highlights our method's overall performance gain, which is enhanced with the refinement selection scheme. As mentioned, we tested this refinement method on three different other methods. Detailed results are only shown for DeepHelicon; for other two methods DeepMetaPSICOV and MetaPSICOV, only the third part is shown, which is enough for demonstrating the performance of refinement and the selection scheme.

For our dataset, with normal selection setting, the scheme selects 92, 100 out of 101 sequences to refine for DeepHelicon and MetaPSICOV respectively. For DeepMetaPSICOV, 56 out of 89 are selected as it failed to output results for the remaining 12 sequences. With our aggressive setting of selection, 33, 5, 40 are selected for DeepHelicon, MetaPSICOV, and DeepMetaPSICOV correspondingly.

The experiments used random forests with 500 tress, training and testing in 10 fold cross validation fashion. Here, the performance metric described in the refinement selection scheme is the AUC-ROC. Moreover, as we found there is no direct link from refinement per residue pair to refinement per sequence, empirical, in 5-step procedures, $\partial fzScore_k^n(i,j)$ is replaced by $fzScore(\widetilde{E_k^n}(i,j))$. Since this experiment is more complex than the early two, the overall pipeline is shown in Figure 3 to readers' convenience.

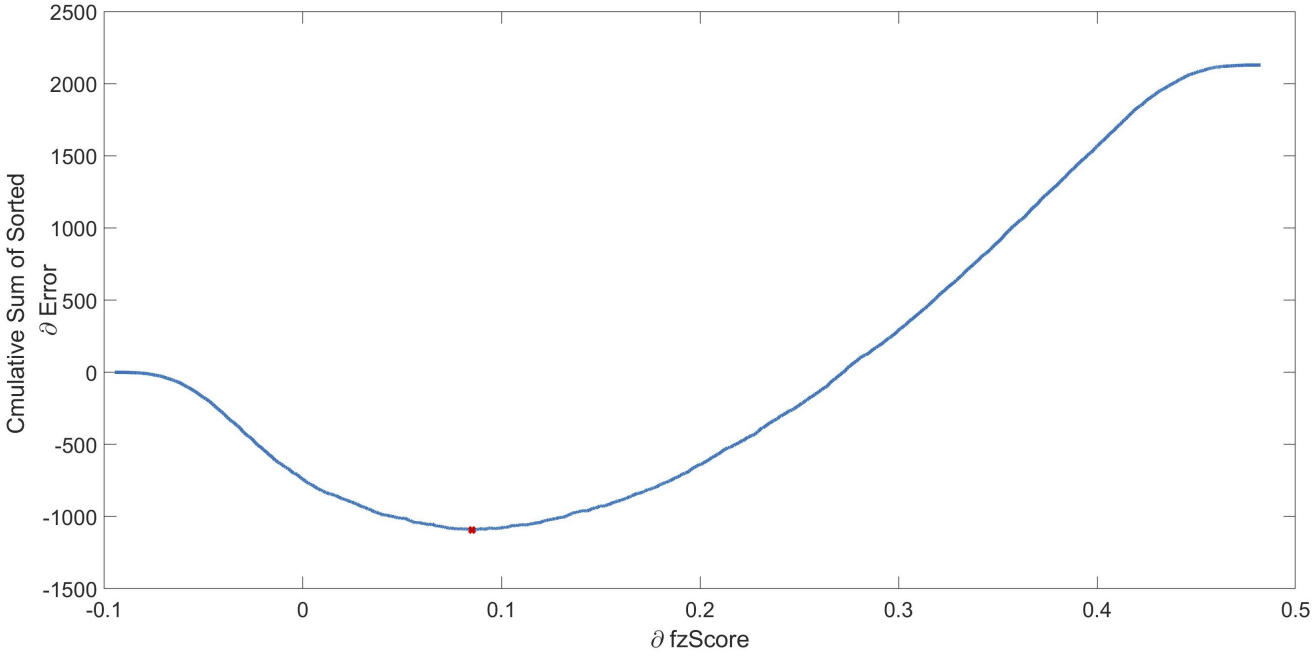


Fig. 2. Fuzzy Score Validation.

TABLE 3
Top-L comparisons of the refinement method (RM) and DeepHelicon (DH) without refinement selection on DATA(101)

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	RM	**64.45%**	**45.67%**	**51.53%**	**60.75%**	**52.17%**
	DH	62.55%	44.49%	50.13%	58.99%	50.68%
L/2	RM	**77.93%**	**28.92%**	**40.48%**	**63.92%**	**45.63%**
	DH	76.30%	27.92%	39.39%	62.49%	44.41%
L/5	RM	**87.94%**	**13.69%**	**22.83%**	**52.47%**	**33.21%**
	DH	85.40%	12.75%	21.63%	50.57%	31.76%
L/10	RM	**90.54%**	**6.90%**	**12.63%**	**37.32%**	**24.07%**
	DH	89.09%	6.68%	12.25%	36.43%	23.50%

TABLE 4
Top-L comparisons of the refinement method (RM) and DeepHelicon (DH) with refinement selection (92 out of 101) on DATA(101).

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	RM	**65.09%**	**45.03%**	**51.38%**	**61.17%**	**52.09%**
	DH	62.99%	43.75%	49.84%	59.22%	50.46%
L/2	RM	**77.91%**	**28.20%**	**39.74%**	**63.54%**	**45.04%**
	DH	76.17%	27.12%	38.58%	62.01%	43.74%
L/5	RM	**88.13%**	**13.46%**	**22.47%**	**52.09%**	**32.94%**
	DH	85.46%	12.44%	21.19%	50.09%	31.39%
L/10	RM	**90.68%**	**6.76%**	**12.39%**	**36.89%**	**23.85%**
	DH	88.78%	6.49%	11.92%	35.78%	23.13%

TABLE 5
Top-L comparisons of the refinement method (RM) and DeepHelicon (DH) with aggressive refinement selection (33 out of 101) on DATA(101).

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	RM	**60.83%**	**45.53%**	**50.19%**	**57.79%**	**50.35%**
	DH	57.49%	44.02%	47.82%	54.71%	47.91%
L/2	RM	**75.32%**	**29.49%**	**40.69%**	**62.61%**	**45.16%**
	DH	70.52%	27.36%	37.86%	58.51%	41.97%
L/5	RM	**87.31%**	**14.65%**	**24.19%**	**53.72%**	**34.17%**
	DH	82.19%	13.34%	22.19%	50.04%	31.59%
L/10	RM	**91.88%**	**7.81%**	**14.10%**	**39.99%**	**25.64%**
	DH	87.49%	7.27%	13.15%	37.59%	24.10%

The results for all four parts are shown in Table 3 to Table 10 respectively. In Table 3, our method gains around $1\%-3\%$ improvements in precision and around 1% improvements in recall, without using the refinement selection scheme. In Table 4 and Table 5, for the selected sequences, the improvements of our method are higher. Moreover, even for the sequences unlisted by the refinement scheme, our method perform almost as same as DeepHelicon as shown in Table 6 and Table 7. In Table 8 to Table 10, the comparison results highlight that both first and second parts of this experiment gain improvements in precision, recall, F_1, $F_{0.35}$, MCC cross different Ls, and show the differences of improvements between the true improvements and diluted improvements of our method for all three different methods considered.

3.5 Feature Comparison

In this experiment, we will compare the performance, measured by improvements over original prediction (in this case, DeepHelicon), of three different features: i) the simple neighbor feature: $\widetilde{F_k^S}(i,j) = [\widetilde{E_k^S}(i,j), \widetilde{C_k}](i,j)$, ii) simple neighbor feature with applying hybrid-cutoffs: $[\ulcorner \widetilde{E_k^S}(i,j) \lrcorner, \widetilde{C_k}(i,j))]$, and iii), main feature used here as shown in Eq 5. The experiments settings are identical as in section 3.4, and the results are shown in Table 11 to Table 13.

As shown in Table 11, the largest improvement is achieved with Main feature, and the second largest is achieved by SC feature consistently in top-L/10, top-L/5. For other top-L/2 and top-L/1, SC feature achieves the best results. Overall, comparing with using simple neigh-

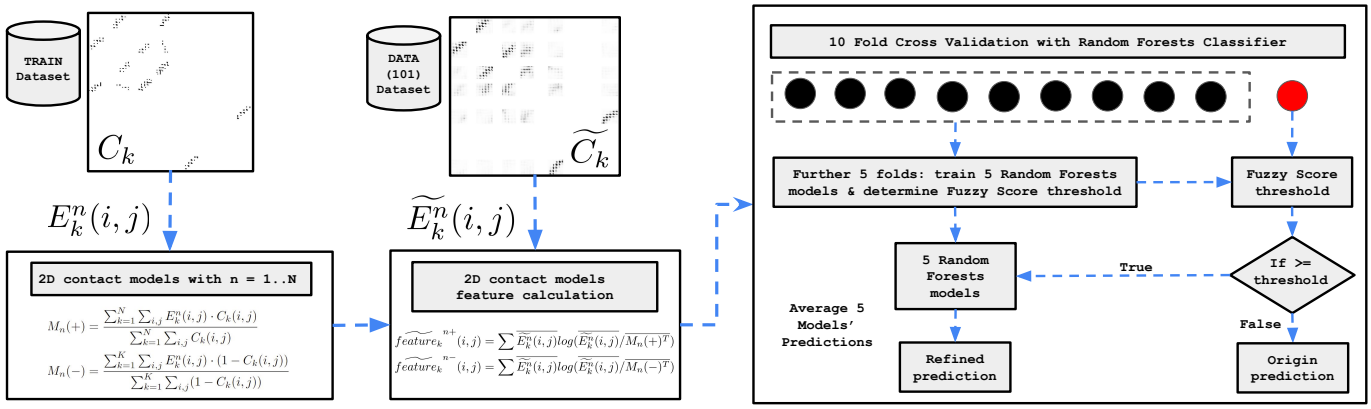


Fig. 3. Pipeline of inter-helix contact prediction refinement. The left is 2D contact models building, the middle is feature extraction from predicted contact map via 2D contact models, and the right contains 10-fold cross validation and fuzzy score threshold estimation.

TABLE 6
Top-L comparisons of the refinement method (RM) and DeepHelicon (DH) of complement to refinement selection (9 out of 101) on DATA(101)

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	RM	57.92%	**52.18%**	**53.08%**	56.47%	52.92%
	DH	**58.07%**	52.07%	53.07%	**56.58%**	52.92%
L/2	RM	**78.08%**	**36.35%**	**48.02%**	**67.84%**	**51.59%**
	DH	77.62%	36.11%	47.66%	67.37%	51.23%
L/5	RM	**85.95%**	**16.06%**	**26.49%**	**56.32%**	**35.95%**
	DH	84.70%	15.90%	26.18%	55.48%	35.45%
L/10	RM	89.16%	8.32%	15.02%	41.68%	26.33%
	DH	**92.24%**	**8.60%**	**15.54%**	**43.14%**	**27.27%**

TABLE 7
Top-L comparisons of the refinement method (RM) and DeepHelicon (DH) of complement to aggressive refinement selection (68 out of 101) on DATA(101)

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	RM	**66.22%**	**45.74%**	**52.18%**	**62.18%**	**53.05%**
	DH	65.01%	44.72%	51.25%	61.06%	52.02%
L/2	RM	**79.19%**	**28.65%**	**40.38%**	**64.56%**	**45.85%**
	DH	79.12%	28.20%	40.14%	64.42%	45.59%
L/5	RM	**88.24%**	**13.22%**	**22.17%**	**51.86%**	**32.74%**
	DH	86.96%	12.45%	21.37%	50.83%	31.83%
L/10	RM	**89.89%**	**6.46%**	**11.91%**	**36.02%**	**23.31%**
	DH	89.87%	6.39%	11.81%	35.87%	23.21%

TABLE 8
DeepHelicon's comparisons of improvements with (W/), with aggressive (W/a) and without (W/O) refinement selection scheme. Corresponding p-values are shown underneath inside parentheses (up to 4 decimal), and improvements are bolded if their p-value < 0.05.

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	W/	**2.10%**	**1.28%**	**1.53%**	**1.94%**	**1.64%**
		(0.0002)	(0.0109)	(0.0004)	(0.0002)	(0.0006)
	W/a	**3.34%**	1.51%	**2.13%**	**2.72%**	**2.21%**
		(0.0044)	(0.1089)	(0.0130)	(0.0052)	(0.0165)
	W/O	**1.90%**	**1.18%**	**2.37%**	**3.09%**	**2.44%**
		(0.0002)	(0.0112)	(0.0006)	(0.0003)	(0.0007)
L/2	W/	**1.74%**	**1.07%**	**1.16%**	**1.53%**	**1.30%**
		(0.0038)	(0.0063)	(0.0014)	(0.0026)	(0.0017)
	W/a	**4.81%**	**2.13%**	**2.83%**	**4.11%**	**3.19%**
		(0.0002)	(0.0002)	(0.0002)	(0.0002)	(0.0002)
	W/O	**1.63%**	**1.00%**	**1.09%**	**1.43%**	**1.22%**
		(0.0037)	(0.0058)	(0.0013)	(0.0025)	(0.0017)
L/5	W/	**2.67%**	**1.02%**	**1.28%**	**2.00%**	**1.55%**
		(0.0042)	(0.0463)	(0.0172)	(0.0054)	(0.0101)
	W/a	**5.13%**	**1.31%**	**2.00%**	**3.69%**	**2.57%**
		(0.0102)	(0.0168)	(0.0148)	(0.0113)	(0.0117)
	W/O	**2.54%**	**0.94%**	**1.20%**	**1.90%**	**1.45%**
		(0.0041)	(0.0441)	(0.0158)	(0.0049)	(0.0091)
L/10	W/	**1.90%**	**0.27%**	**0.47%**	**1.11%**	**0.72%**
		(0.0444)	(0.0185)	(0.0182)	(0.0216)	(0.0230)
	W/a	**4.40%**	**0.54%**	**0.95%**	**2.39%**	**1.54%**
		(0.0267)	(0.0300)	(0.0283)	(0.0254)	(0.0252)
	W/O	1.45%	**0.22%**	**0.38%**	0.88%	0.57%
		(0.1044)	(0.0384)	(0.0396)	(0.0522)	(0.0545)

bor feature alone, it is clear that having hybrid-cutoffs technique and contact model feature is strictly better-off in DeepHelicon dataset. This phenomena is also true in DeepMetaPSICOV case in Table 13. On the other hand, for MetaPSICOV dataset, the situation is mixed as shown in Table 12. We believe this is caused by the relative poor predictions of MetaPSICOV comparing with DeepHelicon and DeepMetaPSICOV. The low prediction quality makes the hybrid-cutoffs technology inject more noise and degrade hybrid-cutoffs' leveraging power in the same time.

4 Discussion

There are several points worthy mentioning. Firstly, the machine learning method used here is not necessary to be random forest. In principle, any classifier should be applicable as long as its outputs can be interpreted as probability. When the classifier's output is not natively interpreted as probability, e.g., SVM, then the prediction score has to be scaled and normalized first. Secondly, for the refinement selection scheme, as mentioned in Results section, the plot of error vs fuzzy score in Figure 2 looks almost perfectly convex, due to the smoothing effect of the plot software. When the plot is not perfectly convex, the choice of optimal cutoff for fuzzy score can be less reliable, i.e., it cannot perfectly separate the cases of refine vs non-refine. One example is the AUC-ROC used here that finds the maximum point instead of the minimum. For such cases, besides smoothing the curve, other metric may also be explored. Thirdly, while DeepHelicon, DeepMetaPSICOV, and MetaPSICOV

TABLE 9
MetaPSICOV's comparisons of improvements with (W/), with aggressive (W/a) and without (W/O) refinement selection scheme. Corresponding p-values are shown underneath inside parentheses (up to 4 decimal), and improvements are bolded if their p-value < 0.05.

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	W/	**6.29%** (0.0)	**4.62%** (0.0)	**5.06%** (0.0)	**5.93%** (0.0)	**5.38%** (0.0)
	W/a	**4.76%** (0.0191)	4.84% (0.0508)	**4.74%** (0.0340)	**4.75%** (0.0222)	**4.94%** (0.0328)
	W/O	**6.24%** (0.0)	**4.59%** (0.0)	**5.03%** (0.0)	**5.89%** (0.0)	**5.34%** (0.0)
L/2	W/	**8.51%** (0.0)	3.45% (0.102)	**4.53%** (0.0)	**7.00%** (0.0)	**5.28%** (0.0)
	W/a	9.14% (0.0569)	4.78% (0.0716)	6.23% (0.0668)	8.27% (0.0600)	6.73% (0.0638)
	W/O	**8.51%** (0.0)	**3.46%** (0.0068)	**4.54%** (0.0)	**7.02%** (0.0)	**5.29%** (0.0)
L/5	W/	**7.93%** (0.0007)	1.35% (0.4026)	2.22% (0.0803)	**4.88%** (0.0031)	**3.22%** (0.0069)
	W/a	**14.69%** (0.0215)	**2.63%** (0.0337)	**4.43%** (0.0321)	**9.68%** (0.0269)	**6.30%** (0.0268)
	W/O	**7.94%** (0.0005)	1.35% (0.2904)	2.23% (0.0558)	**4.90%** (0.0020)	**3.23%** (0.0046)
L/10	W/	**7.90%** (0.0227)	**0.70%** (0.0491)	**1.27%** (0.0316)	**3.54%** (0.0132)	**2.33%** (0.0131)
	W/a	8.96% (0.3001)	1.08% (0.1931)	1.94% (0.1973)	5.09% (0.2211)	3.20% (0.2317)
	W/O	**7.73%** (0.0383)	0.69% (0.0844)	1.24% (0.0558)	**3.46%** (0.0244)	**2.27%** (0.0236)

TABLE 10
DeepMetaPSICOV's comparisons of improvements with (W/), with aggressive (W/a) and without (W/O) refinement selection scheme. Corresponding p-values are shown underneath inside parentheses (up to 4 decimal), and improvements are bolded if their p-value < 0.05.

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	W/	**3.34%** (0.0)	**2.00%** (0.0)	**2.57%** (0.0)	**3.12%** (0.0)	**2.68%** (0.0)
	W/a	**3.35%** (0.0)	**2.01%** (0.0)	**2.52%** (0.0)	**3.13%** (0.0)	**2.66%** (0.0)
	W/O	**2.82%** (0.0)	**1.78%** (0.0)	**2.21%** (0.0)	**2.65%** (0.0)	**2.31%** (0.0)
L/2	W/	**3.61%** (0.0)	**1.11%** (0.0010)	**1.73%** (0.0002)	**2.91%** (0.0)	**2.05%** (0.0001)
	W/a	**3.74%** (0.0)	**1.11%** (0.0010)	**1.74%** (0.0002)	**2.99%** (0.0)	**2.09%** (0.0001)
	W/O	**2.77%** (0.0)	**0.97%** (0.0)	**1.42%** (0.0)	**2.28%** (0.0)	**1.66%** (0.0)
L/5	W/	1.96% (0.0642)	0.30% (0.1426)	0.49% (0.1308)	1.14% (0.0944)	0.75% (0.0969)
	W/a	2.18% (0.0593)	0.28% (0.1021)	0.50% (0.0961)	1.24% (0.0776)	0.79% (0.0778)
	W/O	1.28% (0.0880)	0.22% (0.1660)	0.35% (0.1570)	0.77% (0.1195)	0.51% (0.1213)
L/10	W/	1.10% (0.3240)	0.10% (0.4599)	0.17% (0.4485)	0.46% (0.3976)	0.31% (0.3812)
	W/a	0.78% (0.5643)	0.00% (0.9754)	0.02% (0.9205)	0.19% (0.7464)	0.13% (0.7290)
	W/O	0.61% (0.4308)	0.05% (0.5502)	0.09% (0.5446)	0.25% (0.5053)	0.17% (0.4867)

TABLE 11
Comparisons of improvements over DeepHelicon with simple neighbor feature (SN), with simple neighbor feature with hybrid-cutoffs (SC) and the main feature (Main) in Eq 5. Corresponding p-values are shown underneath inside parentheses (up to 4 decimal), and highest improvements are bolded.

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	SN	1.84% (0.0004)	1.11% (0.0182)	1.33% (0.0011)	1.70% (0.0004)	1.42% (0.0013)
	SC	**1.95%** (0.0002)	**1.21%** (0.0094)	**1.42%** (0.0005)	**1.80%** (0.0002)	**1.52%** (0.0006)
	Main	1.90% (0.0002)	1.18% (0.0112)	1.40% (0.0006)	1.76% (0.0003)	1.49% (0.0007)
L/2	SN	1.75% (0.0027)	1.10% (0.0036)	1.18% (0.0009)	1.54% (0.0018)	1.32% (0.0011)
	SC	**1.87%** (0.0019)	**1.11%** (0.0035)	**1.22%** (0.0008)	**1.63%** (0.0013)	**1.37%** (0.0009)
	Main	1.63% (0.0037)	1.00% (0.0058)	1.09% (0.0013)	1.43% (0.0025)	1.22% (0.0017)
L/5	SN	2.00% (0.0204)	0.78% (0.0861)	0.97% (0.0406)	1.51% (0.0179)	1.17% (0.0280)
	SC	2.37% (0.0054)	0.91% (0.0505)	1.15% (0.0189)	1.79% (0.0058)	1.38% (0.0112)
	Main	**2.54%** (0.0041)	**0.94%** (0.0441)	**1.20%** (0.0158)	**1.90%** (0.0049)	**1.45%** (0.0091)
L/10	SN	0.87% (0.2906)	0.14% (0.1161)	0.24% (0.1224)	0.55% (0.1649)	0.35% (0.1726)
	SC	0.87% (0.2701)	0.13% (0.1128)	0.23% (0.1179)	0.54% (0.1551)	0.35% (0.1622)
	Main	**1.45%** (0.1044)	**0.22%** (0.0384)	**0.38%** (0.0396)	**0.88%** (0.0522)	**0.57%** (0.0545)

TABLE 12
Comparisons of improvements over MetaPSICOV with simple neighbor feature (SN), with simple neighbor feature with hybrid-cutoffs (SC) and the main feature (Main) in Eq 5. Corresponding p-values are shown underneath inside parentheses (up to 4 decimal), and highest improvements are bolded.

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	SN	6.10% (0.0)	4.50% (0.0)	4.91% (0.0)	5.75% (0.0)	5.22% (0.0)
	SC	**6.28%** (0.0)	**4.63%** (0.0)	**5.06%** (0.0)	**5.92%** (0.0)	**5.37%** (0.0)
	Main	6.24% (0.0)	4.59% (0.0)	5.03% (0.0)	5.89% (0.0)	5.34% (0.0)
L/2	SN	8.42% (0.0)	3.28% (0.0)	4.37% (0.0)	6.89% (0.0)	5.14% (0.0)
	SC	8.24% (0.0)	3.23% (0.0)	4.29% (0.0)	6.75% (0.0)	5.04% (0.0)
	Main	**8.51%** (0.0)	**3.46%** (0.0)	**4.54%** (0.0)	**7.02%** (0.0)	**5.29%** (0.0)
L/5	SN	**8.10%** (0.0)	**1.50%** (0.0)	2.38% (0.0)	**5.05%** (0.0)	**3.38%** (0.0)
	SC	8.06% (0.0)	**1.50%** (0.0)	2.37% (0.0)	5.03% (0.0)	3.37% (0.0)
	Main	7.94% (0.0)	1.35% (0.0)	2.23% (0.0)	4.90% (0.0)	3.23% (0.0)
L/10	SN	**8.06%** (0.0)	**0.73%** (0.0)	**1.30%** (0.0)	**3.60%** (0.0)	**2.38%** (0.0)
	SC	7.78% (0.0)	0.65% (0.0)	1.17% (0.0)	3.36% (0.0)	2.22% (0.0)
	Main	7.73% (0.0)	0.69% (0.0)	1.24% (0.0)	3.46% (0.0)	2.27% (0.0)

are used as reference methods, it is reasonable to believe that contact prediction by other deep learning methods can benefit from 2D contact models for improvements as well. Lastly, the usage of refinement selection scheme is highly recommended, as the utility of the refined features hinges on the collective reliability of the neighborhood as measured by the fuzzy score.

In addition, we like to know whether the improvement is affected by the sequence length or the number of transmembrane domains in a sequence. We calculated the correlation of α-helix length and AUC-ROC gain (refined

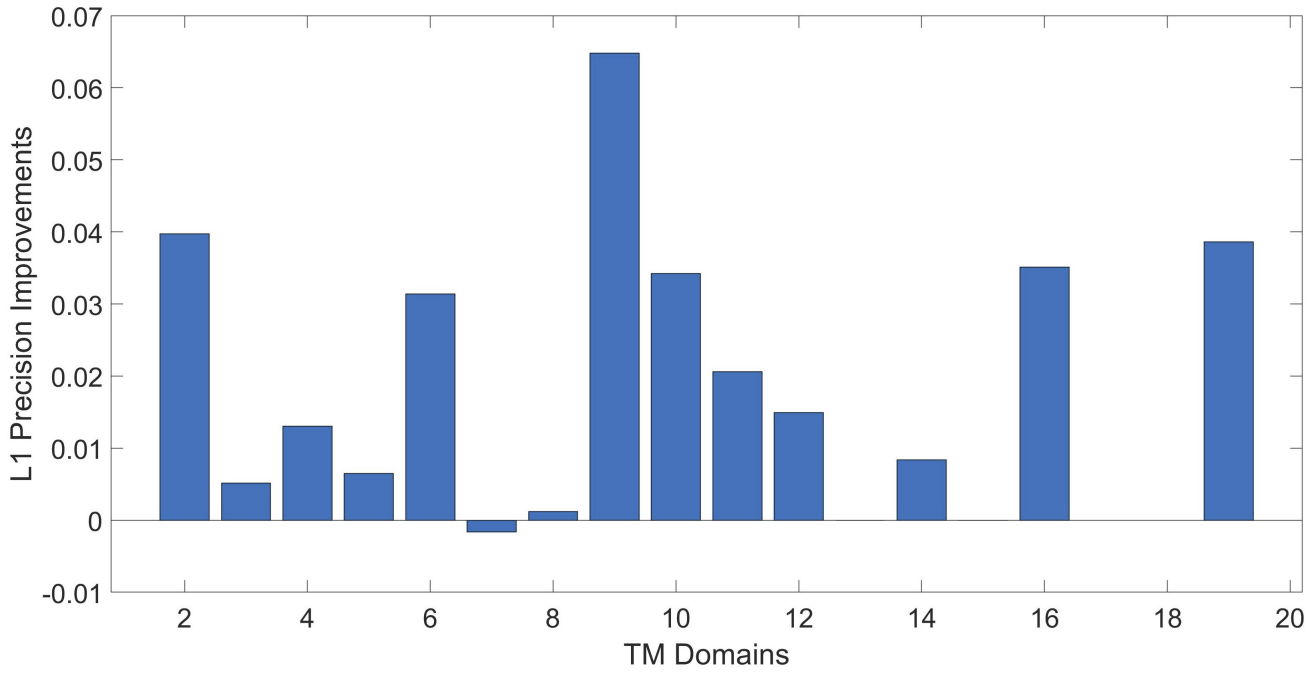


Fig. 4. Histogram of performance improvement versus the number of transmembrane domains

TABLE 13
Comparisons of improvements over DeepMetaPSICOV with simple neighbor feature (SN), with simple neighbor feature with hybrid-cutoffs (SC) and the main feature (Main) in Eq 5. Corresponding p-values are shown underneath inside parentheses (up to 4 decimal), and highest improvements are bolded.

L	Method	Precision	Recall	F_1	$F_{0.35}$	MCC
L/1	SN	2.70% (0.0)	1.66% (0.0)	2.10% (0.0)	2.54% (0.0)	2.19% (0.0)
	SC	2.81% (0.0)	1.77% (0.0)	2.20% (0.0)	2.64% (0.0)	2.30% (0.0)
	Main	**2.82%** (0.0)	**1.78%** (0.0)	**2.21%** (0.0)	**2.65%** (0.0)	**2.31%** (0.0)
L/2	SN	2.47% (0.0)	0.86% (0.0001)	1.26% (0.0)	2.03% (0.0)	1.48% (0.0)
	SC	2.72% (0.0)	**0.97%** (0.0)	1.41% (0.0)	2.25% (0.0)	1.64% (0.0)
	Main	**2.77%** (0.0)	**0.97%** (0.0)	**1.42%** (0.0)	**2.28%** (0.0)	**1.66%** (0.0)
L/5	SN	1.00% (0.1482)	0.13% (0.3550)	0.22% (0.3240)	0.55% (0.2251)	0.36% (0.2387)
	SC	**1.69%** (0.0321)	**0.31%** (0.0991)	**0.49%** (0.0853)	**1.05%** (0.0523)	**0.70%** (0.0571)
	Main	1.28% (0.0880)	0.22% (0.1660)	0.35% (0.1570)	0.77% (0.1195)	0.51% (0.1213)
L/10	SN	−0.87% (0.3277)	−0.12% (0.2106)	−0.21% (0.2094)	−0.51% (0.2332)	−0.33% (0.2413)
	SC	0.24% (0.7601)	0.03% (0.7547)	0.05% (0.7679)	0.10% (0.7842)	0.07% (0.7670)
	Main	**0.61%** (0.4308)	**0.05%** (0.5502)	**0.09%** (0.5446)	**0.25%** (0.5053)	**0.17%** (0.4867)

AUC-ROC minus original AUC-ROC), the results are mixed: -0.2575 for DeepHelicon and -0.4236 for MetaPSICOV, which indicate longer α-helix length; on the other hand, this correlation for DeepMetaPSICOV is 0.1407. In Figure 4, for different number of transmembrane domains in our dataset, their average performance improvement as measured by L10 precision is shown. Note that in our dataset, it just happens that no sequence contains 18 transmembrane domains, which is why there is no registered performance improvement for the data point. As shown in Figure 4, no clear pattern is observed, though it is possible that some patterns may emerge with a different dataset or as the size of the dataset grows bigger and hence more statistically stable.

5 Conclusion

In conclusion, we proposed a low computational cost and quite general method for improving inter-helix contact prediction. The proposed method shows notable improvements as measured by the top-L evaluation criterion. The success is achieved by simple but powerful hybrid-cutoff technology, exploiting features that are not fully captured by the current state-of-art methods, and the development of refinement selection scheme via the idea of fuzzy score, which offers a partial solution to the intrinsic challenge of any refinement method.

With this demonstrated success, there are several components of the proposed method that can be improved further. The first component is the 2D contact model. At the current stage, the developed 2D contact model with its feature calculation is still simplistic, which can lead to losing important spatial information of inter-helix patterns. Second, the refinement selection scheme is only an empirical solution, and does not guarantee optimally. To investigate the challenge of refinement further both practically and theoretically, more experiments across different domains are necessary in the future work.

Finally, the code and data are freely available online at: https://www.cis.udel.edu/~lliao/inter-helix-refinement

Acknowledgments

The authors would like to thank National Science Foundation NSF-MCB1820103, which support this research. Also,

the authors would like to thank Dr. Sun, who gave generous help on DeepHelicon's dataset used in this paper.

REFERENCES

[1] O. Olmea, B. Rost, and A. Valencia, "Effective use of sequence correlation and conservation in fold recognition," *Journal of molecular biology*, vol. 293, no. 5, pp. 1221–1239, 1999.

[2] R. Bonneau, I. Ruczinski, J. Tsai, and D. Baker, "Contact order and ab initio protein structure prediction," *Protein Science*, vol. 11, no. 8, pp. 1937–1944, 2002.

[3] J. Cheng and P. Baldi, "A machine learning information retrieval approach to protein fold recognition," *Bioinformatics*, vol. 22, no. 12, pp. 1456–1463, 2006.

[4] Y. Xia, A. W. Fischer, P. Teixeira, B. Weiner, and J. Meiler, "Integrated structural biology for α-helical membrane protein structure determination," *Structure*, vol. 26, no. 4, pp. 657–666, 2018.

[5] R. M. Bill, P. J. Henderson, S. Iwata, E. R. Kunji, H. Michel, R. Neutze, S. Newstead, B. Poolman, C. G. Tate, and H. Vogel, "Overcoming barriers to membrane protein structure determination," *Nature biotechnology*, vol. 29, no. 4, pp. 335–340, 2011.

[6] J. Sun and D. Frishman, "Deephelicon: Accurate prediction of inter-helical residue contacts in transmembrane proteins by residual neural networks," *Journal of Structural Biology*, vol. 212, no. 1, p. 107574, 2020.

[7] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, K. Tunyasuvunakool, O. Ronneberger, R. Bates, A. Žídek, A. Bridgland *et al.*, "High accuracy protein structure prediction using deep learning," *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*, vol. 22, p. 24, 2020.

[8] V. Golkov, M. J. Skwark, A. Golkov, A. Dosovitskiy, T. Brox, J. Meiler, and D. Cremers, "Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[9] 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.

[10] D. T. Jones and S. M. Kandathil, "High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features," *Bioinformatics*, vol. 34, no. 19, pp. 3308–3315, 2018.

[11] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 851–869, 2017.

[12] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," *arXiv preprint arXiv:2007.05558*, 2020.

[13] E. Laine, S. Eismann, A. Elofsson, and S. Grudinin, "Protein sequence-to-structure learning: Is this the end (-to-end revolution)?" *arXiv preprint arXiv:2105.07407*, 2021.

[14] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell, "Improved contact prediction in proteins: using pseudolikelihoods to infer potts models," *Physical Review E*, vol. 87, no. 1, p. 012707, 2013.

[15] 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.

[16] 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.

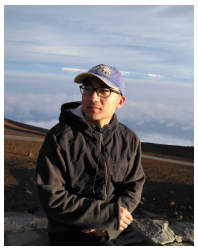
[17] M. Michel, M. J. Skwark, D. Menéndez Hurtado, M. Ekeberg, and A. Elofsson, "Predicting accurate contacts in thousands of pfam domain families using pconsc3," *Bioinformatics*, vol. 33, no. 18, pp. 2859–2866, 2017.

[18] S. M. Kandathil, J. G. Greener, and D. T. Jones, "Prediction of interresidue contacts with deepmetapsicov in casp13," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1092–1099, 2019.

[19] D. Kozma, I. Simon, and G. E. Tusnady, "Pdbtm: Protein data bank of transmembrane proteins after 8 years," *Nucleic acids research*, vol. 41, no. D1, pp. D524–D529, 2012.

[20] X.-F. Wang, Z. Chen, C. Wang, R.-X. Yan, Z. Zhang, and J. Song, "Predicting residue-residue contacts and helix-helix interactions in transmembrane proteins using an integrative feature-based random forest approach," *PloS one*, vol. 6, no. 10, p. e26767, 2011.

[21] P. Hönigschmid and D. Frishman, "Accurate prediction of helix interactions and residue contacts in membrane proteins," *Journal of structural biology*, vol. 194, no. 1, pp. 112–123, 2016.



Jiefu Li is an assistant professor of Computer and Information Science at University of Shanghai for Science and Technology. He received his Ph.D. degree in Computer Science from University of Delaware 2021. His main research interests include pattern recognition, machine learning, and bioinformatics. Previously, he obtained his MS degree in Electrical and Computer Engineering from University of California, Santa Barbara 2016, and BS degree in Electrical Engineering from University of Delaware 2014.

Aman Sawhney is a doctoral student in the Department of Computer and Information Sciences at the University of Delaware. His research is focused in machine learning, network theory and bioinformatics. Previously, he obtained his Master's degree in Computer Science from the University of New Mexico and his Bachelor's degree in Electronics and Communication Engineering from the National Institute of Technology Kurukshetra.

Jung-Youn Lee is a Professor of Plant Biology in the Department of Plant and Soil Sciences, Delaware Biotechnology Institute, at the University of Delaware, U.S.A. She did her postdoctoral research as a Kathrin Esau Postdoctoral Fellow at the University of California, Davis after receiving a Ph.D. degree in Plant Molecular and Cellular Biology from the University of Florida, Gainesville, U.S.A. Prior to this, she graduated from Korea University, Seoul, Korea. Her research interest is on solving the enigma of plasmodesmata, the communication channels fundamental to the development and survival of plants.

Li Liao , who received a Ph.D. degree in theoretical physics from Peking University, is an associate professor of Computer and Information Sciences at the University of Delaware. His current research is in the field of bioinformatics. An author or co-author of more than 80 peer-reviewed publications, he is active in research and serving the bioinformatics community – he has served as a panelist for NSF, program committee member and/or organizer for over 20 conferences and workshops in bioinformatics for the past 5 years, and is currently on the editorial board of three journals. He is a senior member of the ACM.