



# Improved sequence-based prediction of interaction sites in $\alpha$ -helical transmembrane proteins by deep learning

Jianfeng Sun, Dmitrij Frishman\*

Department of Bioinformatics, Wissenschaftszentrum Weihenstephan, Technical University of Munich, Maximus-von-Imhof-Forum 3, 85354 Freising, Germany



## ARTICLE INFO

### Article history:

Received 13 December 2020  
Received in revised form 2 March 2021  
Accepted 2 March 2021  
Available online 09 March 2021

### Keywords:

Protein–protein interactions  
Protein structure  
Protein function  
Molecular evolution  
Sequence annotation  
Deep learning

## ABSTRACT

Interactions between transmembrane (TM) proteins are fundamental for a wide spectrum of cellular functions, but precise molecular details of these interactions remain largely unknown due to the scarcity of experimentally determined three-dimensional complex structures. Computational techniques are therefore required for a large-scale annotation of interaction sites in TM proteins. Here, we present a novel deep-learning approach, DeepTMInter, for sequence-based prediction of interaction sites in  $\alpha$ -helical TM proteins based on their topological, physiochemical, and evolutionary properties. Using a combination of ultra-deep residual neural networks with a stacked generalization ensemble technique DeepTMInter significantly outperforms existing methods, achieving the AUC/AUCPR values of 0.689/0.598. Across the main functional families of human transmembrane proteins, the percentage of amino acid sites predicted to be involved in interactions typically ranges between 10% and 25%, and up to 30% in ion channels. DeepTMInter is available as a standalone package at <https://github.com/2003100127/deeptminter>. The training and benchmarking datasets are available at <https://data.mendeley.com/datasets/2t8kgwzp35>.

© 2021 The Authors. Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Transmembrane (TM) proteins are crucial for a large variety of biological functions [1], acting as transporters, channels, receptors and enzymes [2], and many of these functions are mediated by protein–protein interactions (PPI). For example, a large-scale affinity purification assay of yeast proteins [3] yielded 13,343 interactions between 2875 proteins, of which 1726 interactions involve at least one TM protein. The membrane-linked interactome of *Arabidopsis thaliana* was reported to involve 12,102 interactions [4]. According to the BioGRID database [5] TM proteins are involved in nearly a quarter of all confirmed human interactions, and an even higher percentage (almost 40%) was identified based on the most recent human interactome map [6]. Precise molecular details of the majority of membrane-bound protein complexes remain poorly characterized due to the scarcity of high-resolution structures. Although TM proteins constitute 25–30% of the cellular proteomes [7], they account for less than 2% of all experimentally determined structures in PDB and, based on the UniProt annotation [8], we estimate that less than 15% of all

human protein complexes with a known structure involve TM proteins. Thus, knowledge about the location of protein binding sites in the amino acid sequences of TM proteins remains limited.

There are several computational tools for predicting protein interaction sites, including DLPred [9] and DELPHI [10] and, to our knowledge, there are only two methods specifically trained on membrane proteins. One of them, described in [11] and trained on 128 protein chains, relies on amino acid composition and evolutionary conservation while a more recent tool, developed in our group (MBPred [12]) and trained on 171 chains, also leverages co-evolutionary information. Both methods are implemented using the same conventional machine learning approach – random forest [13]. The recent advent of deep learning techniques has led to a surge in prediction performance in many areas of structural bioinformatics and allowed to reduce its dependence on subjective selection of informative features by domain experts [14]. Among these deep-learning architectures, deep residual neural networks (ResNets), which allow for a fast training with a very large number of layers [15], have enabled considerable progress in predicting secondary structures [16], residue contacts [17,18] as well as 3D protein structures [19]. An additional path to enhancing the accuracy lies in the application of deep learning model ensembles, as highlighted in a recent review [20]. Exponential growth in the number of known amino acid sequences makes it possible to

\* Corresponding author.

E-mail address: [d.frishman@wzw.tum.de](mailto:d.frishman@wzw.tum.de) (D. Frishman).

obtain deep multiple sequence alignments (MSAs) required for efficient training of deep learning models for a growing number of protein families.

Here, we present a fully automated deep-learning tool, DeepTMInter, to predict interaction sites in transmembrane proteins. This method exploits an ultra-deep ResNet containing 27 residual units, 58 convolutional layers and 1 dense layer, followed by stacked generalization [21] for variance error reduction. DeepTMInter was trained on a dataset of 301 chains from 241 unique transmembrane protein assemblies using a wide range of features, including a non-redundant set of physicochemical scales, protein topology, as well as evolutionary and co-evolutionary characteristics [22]. The model hyperparameters (>2 million weight parameters) were determined based on a 5-fold cross validation by the stratified-shuffle method [23]. On an independent dataset of 30 chains (of which more than one third were released between 2019 and 2020) and on two previously released datasets of 137 chains, DeepTMInter significantly outperformed MBpred and DELPHI both within structure-derived and predicted cytoplasmic, transmembrane, and extracellular regions as well as in full amino acid sequences. Furthermore, by reducing MSA sizes using HHfilter [24] at a 90% sequence-identity threshold we were able to speed up the method by an order of magnitude without a loss in accuracy.

## 2. Materials and methods

### 2.1. Datasets of transmembrane proteins with known 3D structure

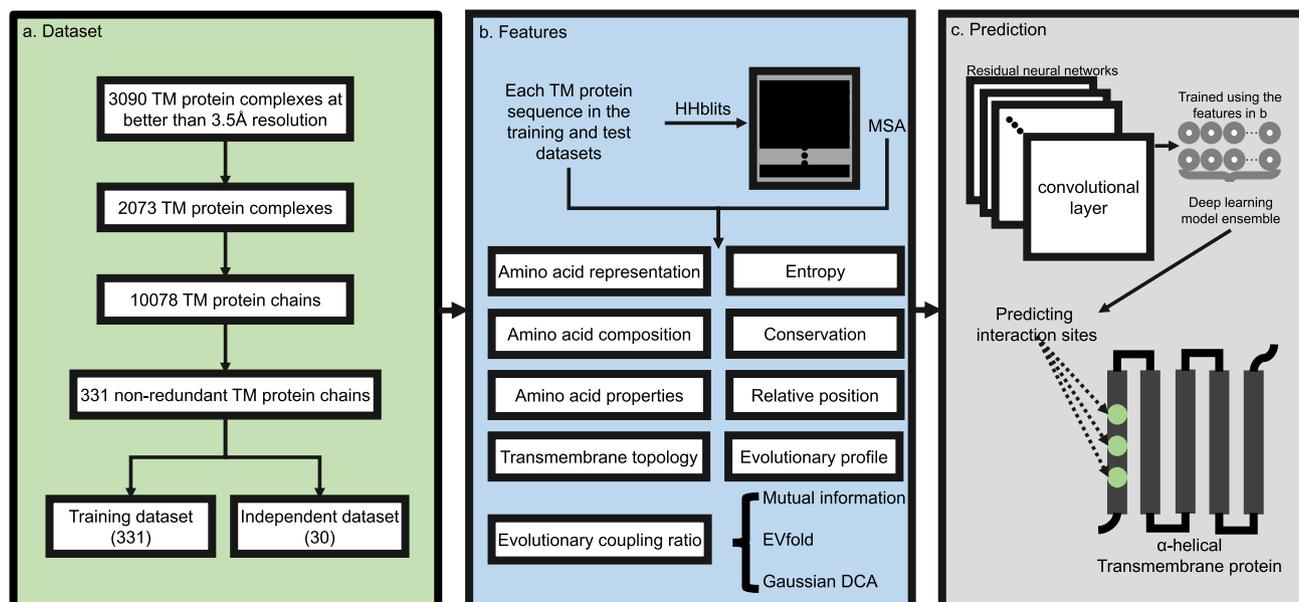
We obtained from the PDBTM database (version: July 2020) [25] a dataset of 3090 three-dimensional structures of  $\alpha$ -helical TM proteins at better than 3.5 Å resolution (Fig. 1a). Their biological oligomer structures were generated using the TMDET algorithm [26,27] based on the PDB BIOMATRIX records. Upon removing structures with non-biological contacts and those with less than two chains we were left with 2073 PDB files containing TM protein complexes. Subsequently, a TM protein chain in any of the 2073 complexes was retained only if it possessed at least one residue contact with any other chain in the same complex,

defined based on the minimal distance between any two non-hydrogen atoms of less than 6 Å (see Section 2.2 for detailed information about interaction site definition). This procedure resulted in 10,194 unique protein chains.

For comparison purposes we also used two additional datasets described in our previous work [12]. Briefly, the CompData dataset (101 TM protein chains, Supplementary Table S1) was derived by imposing a less than 30% sequence identity cutoff on a dataset of 267 TM protein chains benchmarked by Bordner [11]. The TestData dataset (Supplementary Table S2) contains a non-redundant (sequence identity <30%) dataset of 36 protein chains deposited with the PDBTM database between June 2015 and June 2017 and used to test our previously developed MBpred method [12]. The structures of 81 and 35 chains in the CompData and TestData datasets were determined at better than 3.5 Å resolution, respectively. Upon removing these 116 chains from the collection of 10,194 chains described above, we were left with 10,078 chains. Following the common practice in structural bioinformatics [28–30], we subjected this dataset to a stringent redundancy reduction procedure by imposing the requirement that no sequence pair shares a sequence identity above 25%. The resulting 331 chains were then randomly split into a training dataset (301 protein chains, dubbed TrainData) and an independent dataset (30 protein chains, dubbed IndepData) (Supplementary Tables S3 and S4). The detailed description of the interaction partners of the training and testing protein chains is given in Supplementary Tables S31–34.

### 2.2. Definition of interaction sites

Prediction of interaction sites is a class-imbalanced problem as the interacting (minority) class is strongly under-represented compared to the non-interacting (majority) class. As discussed in our earlier publication [12], this problem can be partially alleviated by defining amino acid residue contacts based on a somewhat larger distance threshold, which will result in more residues being assigned to the interacting class. For this reason, out of several alternative residue contact definitions, we selected the one proposed by Hamp and Rost [31], which is based on the distance between any two non-hydrogen atoms of less than 6 Å.



**Fig. 1.** Flowchart of our method to predict interaction sites in TM proteins. (a), (b), and (c) schematically illustrate the dataset generation, input features, and prediction process, respectively.

### 2.3. Protein topology

Extracellular (Extra), transmembrane (TM), and cytoplasmic (Cyto) segments were determined exactly as in our previous publication [12]. First, positions of TM regions were defined according to PDBTM. Since PDBTM does not always contain information about the localization of extramembraneous regions (inside or outside), we used Phobius [32] predictions to verify sequence topology. A non-TM segment as defined by PDBTM was confirmed as cytoplasmic if the overlap between this segment and the cytoplasmic region predicted by Phobius was larger than the overlap between this segment and the predicted extracellular region. The same approach was used for extracellular regions. A combination (Combined) of the three segment types above was also used in benchmarking the performance of predictors. The sizes of the three kinds of regions in the training and test datasets are presented in Supplementary Table S17.

### 2.4. Multiple sequence alignments

We generated multiple sequence alignments (MSAs) for each TM protein sequence by running HHblits searches [24] against the latest release of the specially prepared Uniclust30 database available at [http://wwwuser.gwdg.de/~compbiol/uniclust/2020\\_03/](http://wwwuser.gwdg.de/~compbiol/uniclust/2020_03/). Additionally, we also utilized the Uniprot20 database ([http://wwwuser.gwdg.de/~compbiol/data/hhsuite/databases/hhsuite\\_dbs/old-releases/](http://wwwuser.gwdg.de/~compbiol/data/hhsuite/databases/hhsuite_dbs/old-releases/)), which has been used in several recent contact prediction projects [33–35]. We found that there was no significant difference in the prediction performance of models trained using either of these two databases. All parameters used to run HHblits in this work were according to the recommendations of Seedmayer et al. [36], which can be found at <https://github.com/soedinglab/CCMPred/wiki/FAQ>. In order to keep the CPU and memory requirements for calculating features at a manageable level, HHfilter [24] was applied to only keep sequences sharing <90% sequence identity, which resulted in a significant reduction of MSA depth. In Section 3.1.2 we will discuss the influence of the filtered MSAs on prediction performance.

### 2.5. Input features

For each amino acid and each MSA position we generated a series of sequence-based, physiochemical, and evolutionary characteristics (Fig. 1b), including amino acid representation, amino acid physicochemical scales, amino acid composition, MSA evolutionary profile, Shannon entropy, evolutionary conservation, relative position, protein topology, and residue coevolution.

#### 2.5.1. Amino acid representation

Amino acids in each sequence position were encoded by the one-hot representation. A boolean vector of length 20 was used to indicate the presence (1) or absence (0) of the amino acid X, where X is one of the 20 amino acid symbols arranged in sequential order: A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, and Y.

#### 2.5.2. Amino acid physicochemical scales

The AAanalysis tool (Breimann et al., manuscript in preparation) was used to generate a representative set of amino acid physicochemical scales from the 565 redundant scales curated in the AAindex database [37] and further 69 scales compiled from other references. The redundancy of scales was reduced by applying 2-centroid k-means clustering with the Pearson correlation cutoff of 0.5. The resulting set of non-redundant scales was further clustered into 33 groups and in each group a scale with the highest

Pearson correlation value with the centroid of that group was chosen as representative. The final representative dataset contains 34 scales falling into the following 7 categories: conformation (16), polarity (5), energy (8), composition (1), accessible surface (1), shape (1), structure-activity (2) (Supplementary Table S5). Each physicochemical scale was rescaled to the range [0,1]. For comparison purposes we also used several other widely used amino acid physicochemical scales (Supplementary Fig. S6).

#### 2.5.3. Amino acid composition

Amino acid composition of each protein was represented by a vector of length 20 containing the relative frequency of each amino acid.

#### 2.5.4. MSA evolutionary profile

The evolutionary profile for each symbol Y of 21 symbols (20 amino acids and one gap symbol) at MSA column *i* was calculated as

$$EP_{Y,i} = \log_2 \frac{p_{Y,i}}{p_Y}$$

where  $p_{Y,i}$  is the relative frequency of the symbol Y in the MSA column *i* and  $p_Y$  is the relative frequency of Y in the whole MSA.

#### 2.5.5. Shannon entropy and evolutionary conservation

Shannon entropy for each MSA column *i* was computed as

$$E = - \sum_{i=1}^n p_{Y,i} \log_2 p_{Y,i}$$

where *n* is 21 (20 amino acids and one gap symbol) and  $p_{Y,i}$  is the relative frequency of each symbol Y at MSA column *i*. Lower values of Shannon entropy correspond to higher conservation. Entropy values were transformed in such a way that higher values correspond to a stronger evolutionary conservation *C*:

$$C = 1 - c \times E$$

where the constant *c* is  $\frac{1}{\log_2(20)}$ .

#### 2.5.6. Relative sequence position

Relative sequence position was computed by normalizing the actual position *i* by protein length *L*:  $\frac{i}{L}$ .

#### 2.5.7. Protein topology

For each amino acid position *i* we generated a boolean vector of length 3 containing a one-hot representation of three topological regions: cytoplasm, transmembrane helix, or extracellular region.

#### 2.5.8. Residue coevolution

The likelihood of two amino acid residues to be in contact can be measured by the evolutionary coupling (EC) values predicted by the evolutionary coupling analysis (ECA) methods [38]. In order to quantify the likelihood of a given residue to be involved in a contact, the evolutionary coupling ratio (ECR) has been proposed [22]:

$$ECR = \frac{EC_X}{EC/L}$$

where  $EC_X$  is the sum of all EC values involving the residue X at position *i* and *EC* is the sum of all EC values of all residues in protein of length *L*. In order to reduce the variance error, we employed four ECA tools to generate three types of EC values, namely: mutual information and EVfold (generated by FreeContact <ftp://roslab.org/free/>) and Gaussian DCA (<https://github.com/carlobaldassi/GaussDCA.jl>). The ECR feature is thus represented by a vector of length 3.

## 2.6. The deep learning approach

### 2.6.1. Sequence window size and feature vector dimension

The choice of the sequence window size is crucial for optimizing the speed of training and also because it determines the dimension of the feature vector. For each window centered around a certain sequence position we tested three different setups: *i*) a comparatively large window size of 9, *ii*) a comparatively small window size of 3, and *iii*) a combination of two different window sizes, 9 and 3, dependent on a particular group of features being used. Upon conducting extensive computational experiments (data not shown) we found the setup *iii* to deliver the most optimal results in terms of the number of epochs required for training. We finally chose the window size of 9 for three features - physicochemical scales, evolutionary profile, and residue coevolution - while for all other features we used the windows of length 3. This choice resulted in a feature vector of length 660 (Supplementary Table S7).

### 2.6.2. Residual neural network architecture

We developed a deep learning architecture based on a residual neural network (ResNet) for predicting interacting amino acid residues in transmembrane proteins (Fig. 2a). For each amino acid position, the 660-dimensional feature vector (Section 2.6.1) was reshaped into a  $26 \times 26$  matrix (i.e. with 676 elements, with 16 dimensions padded by 0), which was batch-normalized in order to speed up the training process [39]. The matrix was fed into an initial convolutional layer with 64 filters using stride 1, thus resulting in 64 separate output matrices with the same dimension  $26 \times 26$  as the input matrix. In line with previous work [18,33,40], the same number of filters (64) was used in all other convolutional layers within the proposed architecture. The  $l \times l$  values contained in the filter are the actual parameters to be learned during training. Since the computational cost of applying the convolution operation is directly proportional to the number of parameters, we used relatively small filters of size  $3 \times 3$  containing randomly generated normally distributed values.

At the core of our ResNet architecture are 27 residual units (also called residual blocks), all with the same structure (Fig. 2a). For comparison, the number of residual units in some of the recently published ResNet architectures were 9 [41], 18 [40], and 22 [34]. We found that increasing the number of residual units even further

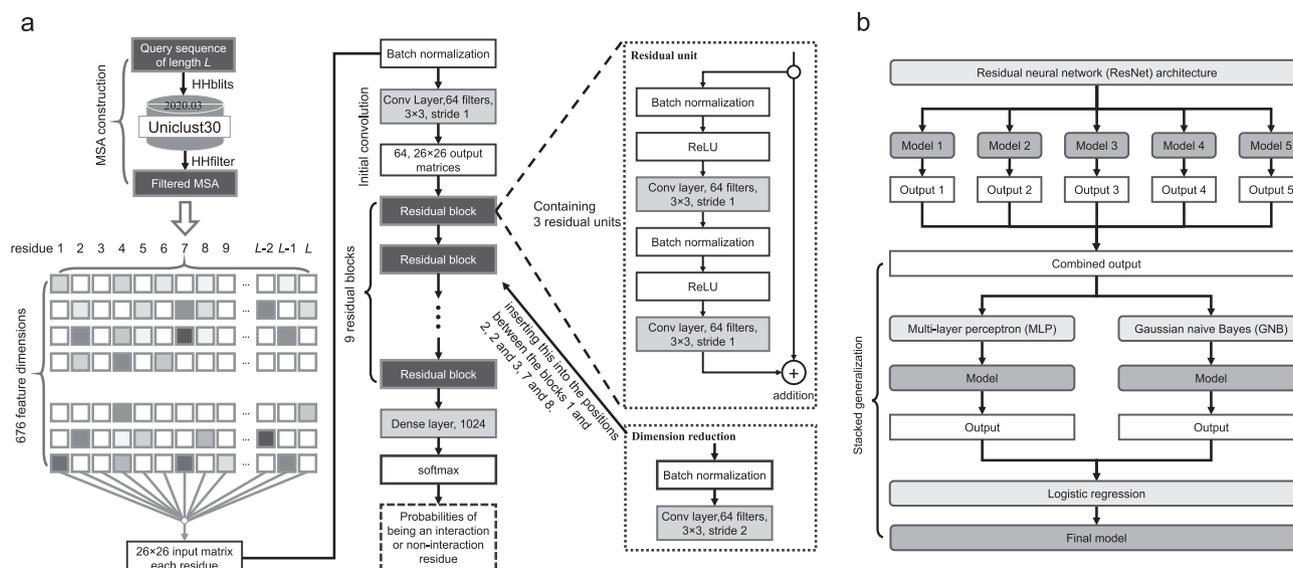
led to significantly higher requirements for computing power and GPU memory resources; a deep architecture with 27 residual units could be trained in an acceptable running time. Each residual unit consists of 2 repeated basic structures consisting of batch normalization [39], a rectified linear unit (ReLU) [42], and a convolutional layer. This design of the residual unit, already used in our previous work [17], allows to accelerate the optimization and to avoid overfitting of the ResNet architecture [43]. ReLU [42], which makes input data non-negative [44], can alleviate the vanishing gradient problem [44] and further improve the performance of ResNet [42]. In order to reduce data dimensionality and the computational cost, we plugged an additional block comprising batch normalization and a convolutional layer with stride 2 (Fig. 2a) evenly distributed across the architecture: in the beginning (between residual units 1 and 2), in the middle (between residual units 13 and 14), and at the end (between residual units 26 and 27). It has become a common practice to apply dimensionality reduction subsequent to a group of convolutional layers [45–48]. As a result, the original matrix dimension is progressively reduced from  $26 \times 26$  to  $13 \times 13$ , then to  $7 \times 7$ , and finally to  $4 \times 4$ . The last residual unit is connected to a dense layer with 1024 neurons, followed by a 2-way softmax function which outputs for each amino acid the probabilities  $([x_1, x_2])$  where  $0 \leq x_1, x_2 \leq 1$  of it being involved in an interaction (corresponding to the label [0, 1] if  $x_2 \geq x_1$ ) or not (corresponding to the label [1, 0] if  $x_1 \geq x_2$ ). We implemented the deep architecture of ResNet depicted in Fig. 2a by using Google's Tensorflow library (version 1.12.0) based on Python programming language.

### 2.6.3. Settings for training the ResNet architecture

Adam [49], a computationally efficient algorithm, was employed to train and optimize the ResNet architecture with the learning rate of 0.001 and the batch size of 100 training samples. The cross entropy objective function [50] was used to quantitatively measure the difference between the actual labels and predicted labels (see Section 2.6.2).

### 2.6.4. Cross validation of the ResNet architecture

We categorized protein chains in the TrainData dataset into 5 classes according to their length: <200 (104 chains), 200–400 (125 chains), 401–600 (51 chains), 601–800 (16 chains), and



**Fig. 2.** Overview of the supervised learning process of DeepTMInter. (a) Layout of the deep ResNet architecture to predict interacting amino acid residues in TM proteins. (b) Using stacked generalization to further enhance the prediction performance of the ResNet models. Rounded boxes in light grey: deep learning or machine learning methods; rounded boxes in dark grey: respective obtained models.

>800 (4 chains) (Supplementary Table S8). 5-fold stratified-shuffle cross validation [23] was employed to evenly allocate protein chains of different length classes for training and validation at each iteration (Supplementary Fig. S1). Protein chains of each class were then randomly selected for training and validation according to the ratio 4:1 at each of the 5 iterations. Specifically, at each iteration, 240 chains were used for training and 60 chains for validation.

### 2.6.5. Avoiding over-training

Over-training is detrimental to the performance of models on unseen validation data [51–53], even if they achieve ideal performance on training data [54] (Supplementary Fig. S2). In order to avoid over-training, the early stopping strategy [55] was adopted, which involves aborting the training when the performance on validation data begins to worsen [56]. At each round of cross-validation, the model was chosen at one of the training epochs over which the performance on validation data continued to show an optimal trend [54,55].

### 2.6.6. Stacked generalization

Stacked generalization [21], an approach for implementing model ensembles [57], was used to minimize the generalization errors of the models trained by the ResNet architecture [58] (Fig. 2b). The combined output, which was constructed by merging the output (by column) of the five models generated by a 5-round training on the full TrainData dataset (see Section 3.1.1), served as input for a multi-layer perceptron (MLP) [59] and a Gaussian Naive Bayes (GNB) classifier [60]. The output of the MLP and GNB models was then fitted by logistic regression. The resulting final model was used to report and evaluate the performance of the interaction site predictor reported in this paper. The MLP, GNB, and logistic regression models were implemented using the scikit-learn package (<https://scikit-learn.org>).

### 2.7. Evaluation criteria

The overall performance of DeepTMInter was evaluated based on two threshold-free [61,62] measures: the area under the ROC (Receiver Operating Characteristic) curve (AUC) and the area under the Precision-Recall curve (AUCPR) [63]. In addition, we also employed the following single-threshold performance measures [61]:

$$\text{Precision (P)} = \frac{TP}{TP + FP}$$

$$\text{Recall (R)} = \frac{TP}{TP + FN}$$

$$\text{Specificity (SP)} = \frac{TN}{TN + FP}$$

$$\text{F1 - score (F1)} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{Accuracy (ACC)} = \frac{TP + TN}{TP + TN + FP + FN}$$

Matthews Correlation Coefficient (MCC)

$$= \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

$$\text{Jaccard Similarity Coefficient (JSC)} = \frac{TP}{TP + FP + FN}$$

where *TP* (true positive), *FP* (false positive), *TN* (true negative) and *FN* (false negative) are the number of interacting residues predicted as interacting, the number of non-interacting residues predicted as interacting, the number of non-interacting residues predicted as non-interacting, and the number of interacting residues predicted as non-interacting, respectively. For each protein of length *L* in the three test datasets we evaluated predictions based on the top-ranked *L*/5 interaction sites (*L*-based performance evaluation is most commonly adopted in structural bioinformatics [18,29,33]).

### 2.8. Comparison with other methods

We compared the performance of DeepTMInter on three test datasets (see Section 2.1) with the MBPred algorithm previously developed in our group ([12]; <https://github.com/bojiigu/MBPred>). The standalone MBPred suite contains four individual predictors - MBPredTM, MBPredCyto, MBPredExtra, and MBPredAll - trained on transmembrane, cytoplasmic, and extracellular regions as well as full-length TM protein sequences, respectively. Additionally, we also compared DeepTMInter with MBPredCombined, which combines MBPredTM, MBPredCyto, and MBPredExtra predictions.

Another method we compared DeepTMInter with is DELPHI [10], which employs a hybrid convolutional and recurrent neural network architecture for predicting interaction sites in proteins of any type, both soluble and transmembrane.

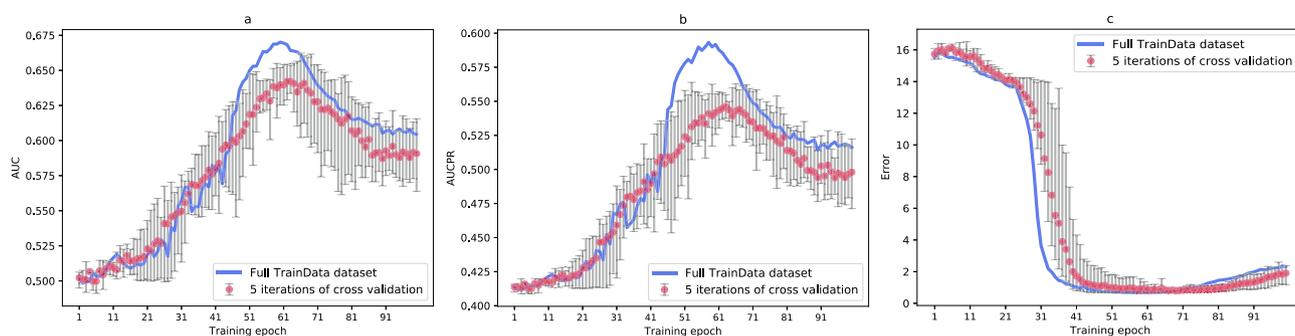
### 2.9. Human transmembrane proteins

We obtained 5178 human protein sequences with at least one annotated transmembrane region from the UniProtKB/Swiss-Prot database [8]. Interaction sites of the proteins were predicted by DeepTMInter. Topologies of the human transmembrane proteins were annotated according to UniProt. We finally retained for further analysis 5051 human transmembrane proteins with the MSA depth in the range between 20 and 10,000 after filtering (see Section 2.4 for alignment generation). Thus, shallow MSAs providing insufficient evolutionary information as well as excessively deep MSAs imposing excessively high CPU requirements were excluded. These proteins were classified into eight major functional classes - *G-protein-coupled receptor* (GPCR), *catalytic receptor*, *ligand-gated ion channel* (LGIC), *voltage-gated ion channel* (VGIC), *other ion channel*, *transporter*, *enzyme*, and *other protein target* - according to the expert-curated “Guide to PHARMACOLOGY” database (GtoPdb; <https://www.guidetopharmacology.org/>) [64,65].

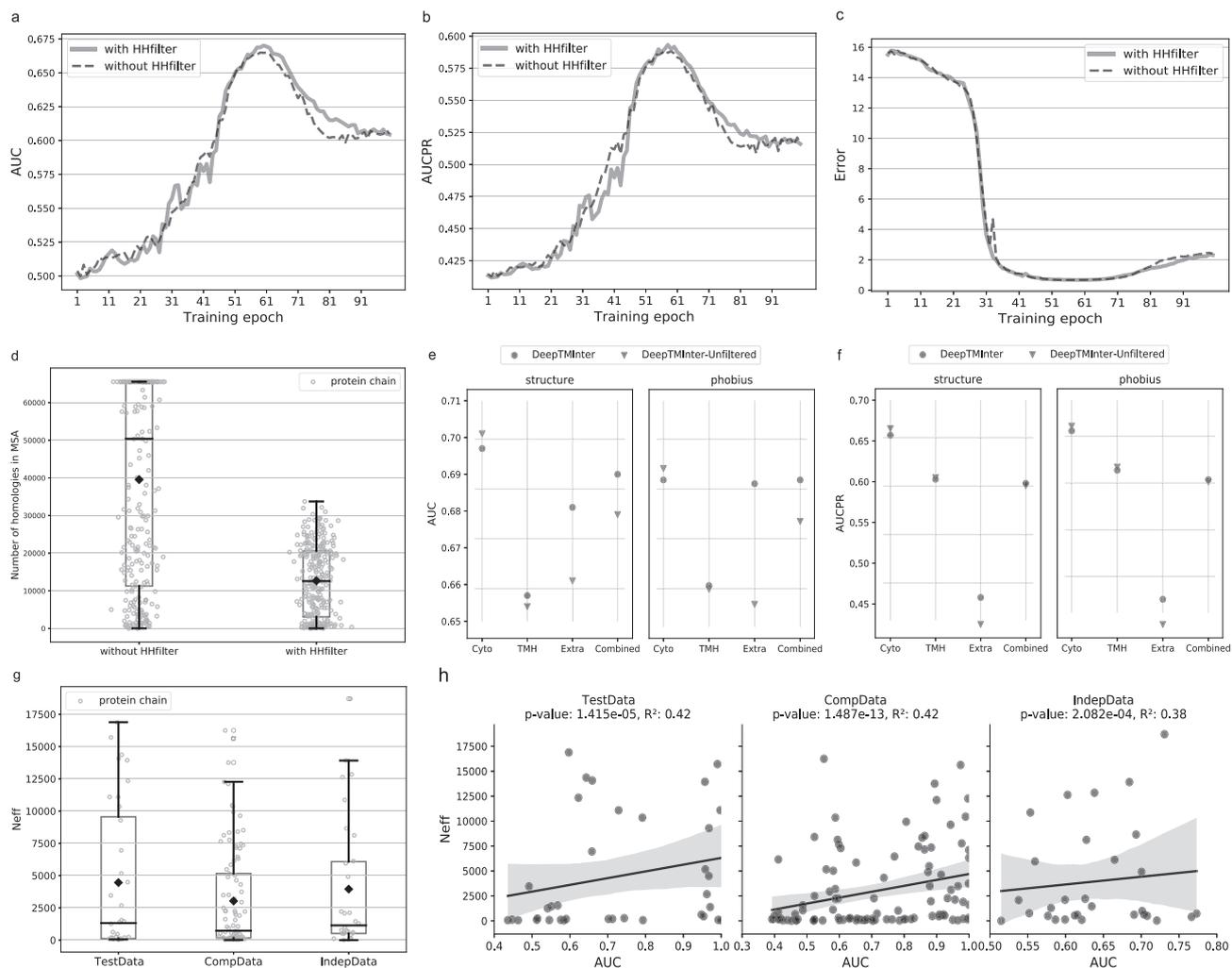
### 2.10. Protein-protein interaction databases

For human transmembrane proteins we obtained 76,584 unique pairs of interacting proteins from a high-quality expert-curated resource HuRI-Union [6], which represents the union of the HI-union and Lit-BM databases. The 64,006 binary proteins interactions (PPIs) in the HI-union database were systematically identified by the yeast two-hybrid (Y2H) assay, while the Lit-BM database comprises a collection of 13,441 high-confidence binary PPIs from literature.

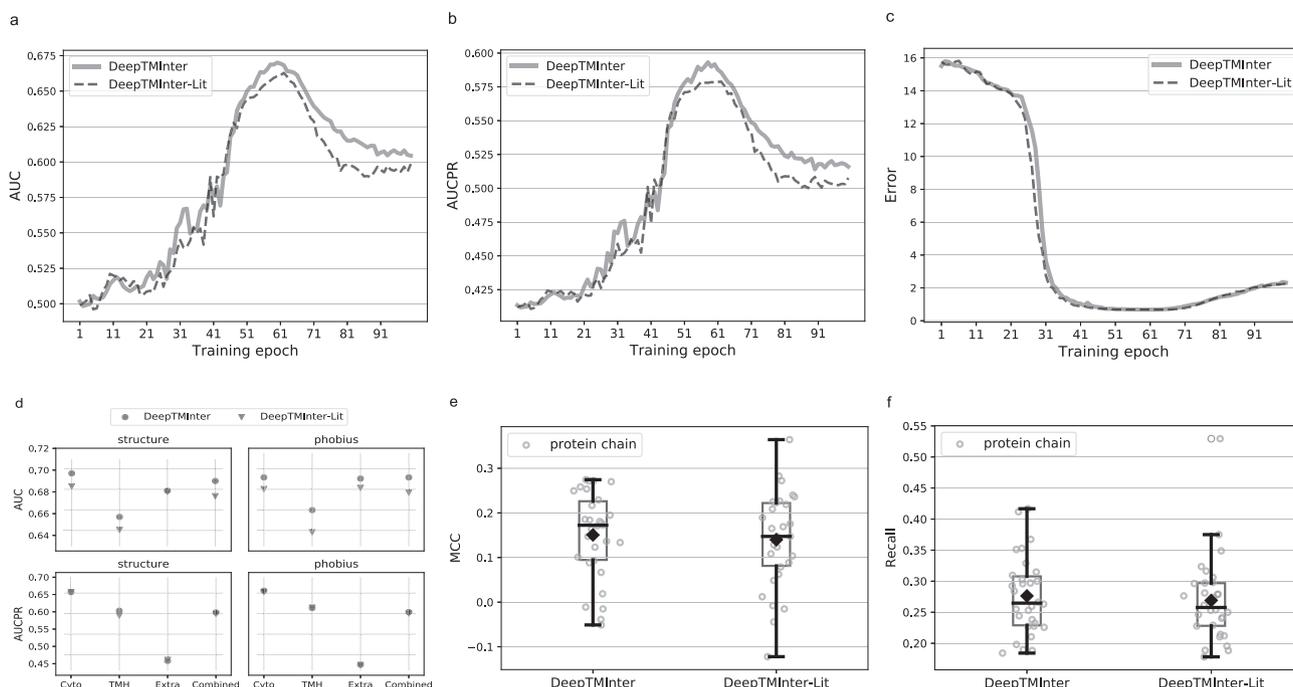
Interaction partners for the proteins in our three test datasets (TestData, CompData, and IndepData) were obtained by merging 1,858,173 binary interactions from the BioGRID database (version 3.5.188) [5] and 1,063,382 binary interactions from the IntAct database (version: 4.1.25) [66], respectively (Supplementary Figs. S3 and S4). A mapping between the PDB codes and UniProt IDs of proteins was obtained by PyPDB [67].



**Fig. 3.** Performance of our method on the IndepData dataset based on the 80% subsets and the full TrainData dataset. (a), (b), and (c) show AUC, AUCPR, and cross-entropy error values over 100 training epochs. The errors in (c) measure the difference between actual and predicted labels of interaction sites using the cross-entropy objective function (see Section 2.6.3). Blue lines and red dots represent the mean AUC, AUCPR, and error values produced by the models trained over 5 rounds on the full training set or the models trained on the 80% subsets in the course of 5-fold cross validation, respectively. For each red dot the upper and lower bounds correspond to the maximum and minimum values produced by 5 cross-validation models at each of the 100 epochs, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 4.** Performance comparison of DeepTMInter and DeepTMInter-Unfiltered on the IndepData dataset. (a), (b), and (c) show mean AUC, AUCPR, and cross-entropy error values with (solid line) and without (dashed line) HHfilter over 100 training epochs produced by the models trained over 5 rounds on the full training set. (d) presents the number of homologous sequences in MSAs generated with and without using HHfilter, with the mean values of 39,553 and 12,663, respectively. (e) and (f) show the AUC and AUCPR values produced by the final ensemble models (stacked generalization, see Section 2.6.6) on four types of structure-derived and Phobius-predicted regions (Cyto, TMH, Extra, and Combined) (Supplementary Tables S11). (g) shows the distribution of Neff values for the MSAs of proteins in the three test datasets. (h) shows the correlation between Neff and the prediction performance measured as AUC on the three test datasets. Each dot represents a protein.



**Fig. 5.** Performance comparison of DeepTMInter and DeepTMInter-Lit based on the IndepData dataset. (a), (b), and (c) show mean AUC, AUCPR, and cross-entropy error values over 100 training epochs produced by the DeepTMInter (solid line) and DeepTMInter-Lit (dashed line) models trained over 5 rounds on the full training set. (d) shows the AUC and AUCPR values produced by the final ensemble models (stacked generalization, see Section 2.6.6) on four types of structure-derived and Phobius-predicted regions (Cyto, TMH, Extra, and Combined). (e) and (f) display the distribution of MCC and recall values of protein chains.

### 3. Results

#### 3.1. Prediction performance of DeepTMInter

In addition to performing a 5-fold cross-validation procedure, we also conducted 5 rounds of training on the full TrainData dataset in order to eliminate the influence on the prediction performance of some random factors, such as the initialization parameters of the residual neural network (see Sections 2.6.4 and

2.6.6). Note that no protein chain in the TrainData dataset shares more than 25% sequence identity with any protein from the IndepData dataset (see Section 2.1), on which our method was assessed.

As seen in Fig. 3, the best performance on validation data is achieved in the vicinity of epoch 60 and the corresponding models were chosen for final assessment according to the early stopping strategy. Overall, the performance of models trained on the full TrainData dataset is significantly better in terms of mean AUC values, AUCPR values, and cross-entropy error function than that of

**Table 1**

Performance gauged by mean precision (P), recall (R), specificity (SP), accuracy (ACC), MCC, F1-score (F1), and AUC, and AUCPR (see Section 2.7) using structure-derived Cyto, TMH, Extra, and Combined regions on the TestData dataset.

Predictor	Region	L/5					L/10					AUC	AUCPR		
		P	R	SP	ACC	MCC	F1	P	R	SP	ACC			MCC	F1
MBPredCyto	Cyto	0.651	0.231	0.817	0.439	0.077	0.298	0.613	0.113	0.913	0.413	0.038	0.172	0.760	0.539
MBPredTM		0.593	0.215	0.806	0.422	0.008	0.276	0.540	0.109	0.907	0.397	0.001	0.157	0.612	0.488
MBPredExtra		0.645	0.219	0.811	0.440	0.068	0.295	0.566	0.093	0.905	0.403	0.001	0.153	0.581	0.444
MBPredAll		0.657	0.227	0.814	0.449	0.075	0.293	0.619	0.114	0.910	0.420	0.031	0.172	0.745	0.511
DELPHI		0.663	0.256	0.838	<b>0.485</b>	0.107	0.325	0.689	0.148	0.924	<b>0.460</b>	0.113	0.212	0.720	0.575
<b>DeepTMInter</b>		<b>0.743</b>	<b>0.322</b>	<b>0.871</b>	0.481	<b>0.202</b>	<b>0.376</b>	<b>0.733</b>	<b>0.197</b>	<b>0.944</b>	0.440	<b>0.158</b>	<b>0.250</b>	<b>0.807</b>	<b>0.744</b>
MBPredCyto	TMH	0.574	0.224	0.816	0.490	0.059	0.293	0.574	0.119	0.909	0.480	0.040	0.184	0.605	0.449
MBPredTM		0.620	0.248	0.837	0.509	0.106	0.324	0.665	0.141	0.926	0.500	0.105	0.219	0.753	0.507
MBPredExtra		0.572	0.210	0.814	0.489	0.051	0.284	0.592	0.120	0.911	0.484	0.051	0.189	0.581	0.410
MBPredAll		0.645	0.267	0.841	0.532	0.150	0.343	0.637	0.137	0.922	0.510	0.095	0.213	0.724	0.471
DELPHI		0.584	0.248	0.836	0.535	0.096	0.320	0.654	0.147	0.930	<b>0.535</b>	0.121	0.227	0.707	0.569
<b>DeepTMInter</b>		<b>0.734</b>	<b>0.350</b>	<b>0.890</b>	<b>0.554</b>	<b>0.240</b>	<b>0.429</b>	<b>0.775</b>	<b>0.205</b>	<b>0.958</b>	0.523	<b>0.200</b>	<b>0.303</b>	<b>0.820</b>	<b>0.721</b>
MBPredCyto	Extra	0.624	0.205	0.804	0.447	0.025	0.288	0.587	0.101	0.900	0.424	0.009	0.165	0.626	0.528
MBPredTM		0.572	0.184	0.797	0.418	0.001	0.258	0.516	0.085	0.891	0.408	0.001	0.141	0.594	0.453
MBPredExtra		0.660	0.228	0.809	0.463	0.072	0.312	0.658	0.112	0.910	0.437	0.061	0.184	0.685	0.497
MBPredAll		0.611	0.222	0.814	0.442	0.023	0.290	0.625	0.117	0.910	0.439	0.047	0.187	0.672	0.518
DELPHI		0.591	0.230	0.828	0.479	0.054	0.304	0.648	0.134	0.919	<b>0.470</b>	0.071	0.209	0.644	0.591
<b>DeepTMInter</b>		<b>0.763</b>	<b>0.325</b>	<b>0.867</b>	<b>0.504</b>	<b>0.194</b>	<b>0.405</b>	<b>0.778</b>	<b>0.181</b>	<b>0.938</b>	0.460	<b>0.161</b>	<b>0.265</b>	<b>0.738</b>	<b>0.685</b>
MBPredAll	Combined	0.654	0.245	0.829	0.493	0.111	0.329	0.646	0.120	0.916	0.458	0.065	0.190	0.721	0.497
MBPredCombined		0.665	0.242	0.827	0.492	0.118	0.331	0.662	0.119	0.914	0.457	0.075	0.191	0.732	0.517
DELPHI		0.616	0.251	0.839	0.513	0.094	0.331	0.653	0.137	0.926	<b>0.495</b>	0.094	0.213	0.695	0.572
<b>DeepTMInter</b>			<b>0.759</b>	<b>0.344</b>	<b>0.888</b>	<b>0.530</b>	<b>0.234</b>	<b>0.425</b>	<b>0.815</b>	<b>0.202</b>	<b>0.961</b>	0.488	<b>0.205</b>	<b>0.294</b>	<b>0.793</b>

models trained on TrainData subsets in the course of cross validation (Fig. 3a-c, Supplementary Tables S9 and S10). The ~20% increase in the number of protein chains between each cross-validation subset and the full training set and the concomitant increase in the number of interaction sites (from ~82,000 for each of cross validations to 102,685, see Supplementary Table S8) lead to a surge in prediction performance. Thus, we finally settled for the models trained based on the full TrainData dataset, which were further used to construct the final ensemble model referred to as DeepTMInter (see Section 2.6.6). The application of the stacked generalization to the final ensemble model results in an approximately 0.5%-3% increase compared to the separate models in terms of AUC performance using different regions (Supplementary Table S10).

### 3.2. Influence of MSA depth on prediction performance

MSA depth is a major factor determining the CPU and memory requirements for feature generation. We therefore evaluated the performance of our method on full MSAs (this model is referred to as DeepTMInter-Unfiltered) and on shallow MSAs filtered by HHfilter (see Section 2.4) (referred to as DeepTMInter). The performance of these two models is in general comparable (Fig. 4a-c), with DeepTMInter even overperforming DeepTMInter-Unfiltered in some cases using the four types of either structure-derived or Phobius-predicted regions (Cyto, TMH, Extra, and Combined) (Fig. 4e and f). For example, using the structure-derived Extra region the AUC and AUCPR values of DeepTMInter (0.688 and 0.458, respectively) are significantly higher than the values achieved by DeepTMInter-Unfiltered (0.656 and 0.425, respectively). Thus, significantly reducing alignment depth (Fig. 4d) allowed to speed up our method without sacrificing prediction performance. We examined how the prediction performance is influenced by the MSA quality based on the numbers of effective sequences (Neff) [68,69], which were computed using ConKit [70] with the 80% sequence-identity threshold [68] (Fig. 4g and h). According to the linear regression analysis there is overall a statistically significant correlation between the MSA quality and the prediction performance on all test datasets, as evidenced by low

p-values (e.g.,  $1.415e-05$  on TestData and  $2.084e-04$  on IndepData), although the R-square values are moderate (e.g., 0.42 on TestData and 0.38 on IndepData).

### 3.3. Selection of amino acid physicochemical scales

In order to investigate how the choice of amino acid physicochemical scales influences model performance, two groups of scales were prepared: one generated by the AAanalysis tool and the other one manually collected from literature (see Methods, Section 2.5.2). Our final model, DeepTMInter, was trained with the scales generated by the AAanalysis tool and all the other features. For comparison, the model trained with the scales collected from references and all the other features is further referred to as DeepTMInter-Lit.

Overall, DeepTMInter shows a better performance than DeepTMInter-Lit in terms of AUC values on all test datasets (Fig. 5). For instance, using Phobius-predicted combined regions on the IndepData dataset DeepTMInter achieves the AUC value of 0.690 (AUCPR = 0.599) compared to 0.676 (0.595) of DeepTMInter-Lit (Supplementary Tables S10 and 12). We assume that this gain in performance stems from the fact that the AAanalysis tool selects the representative scales of each kind and thus significantly reduces data redundancy, which is detrimental to learning algorithms [71,72].

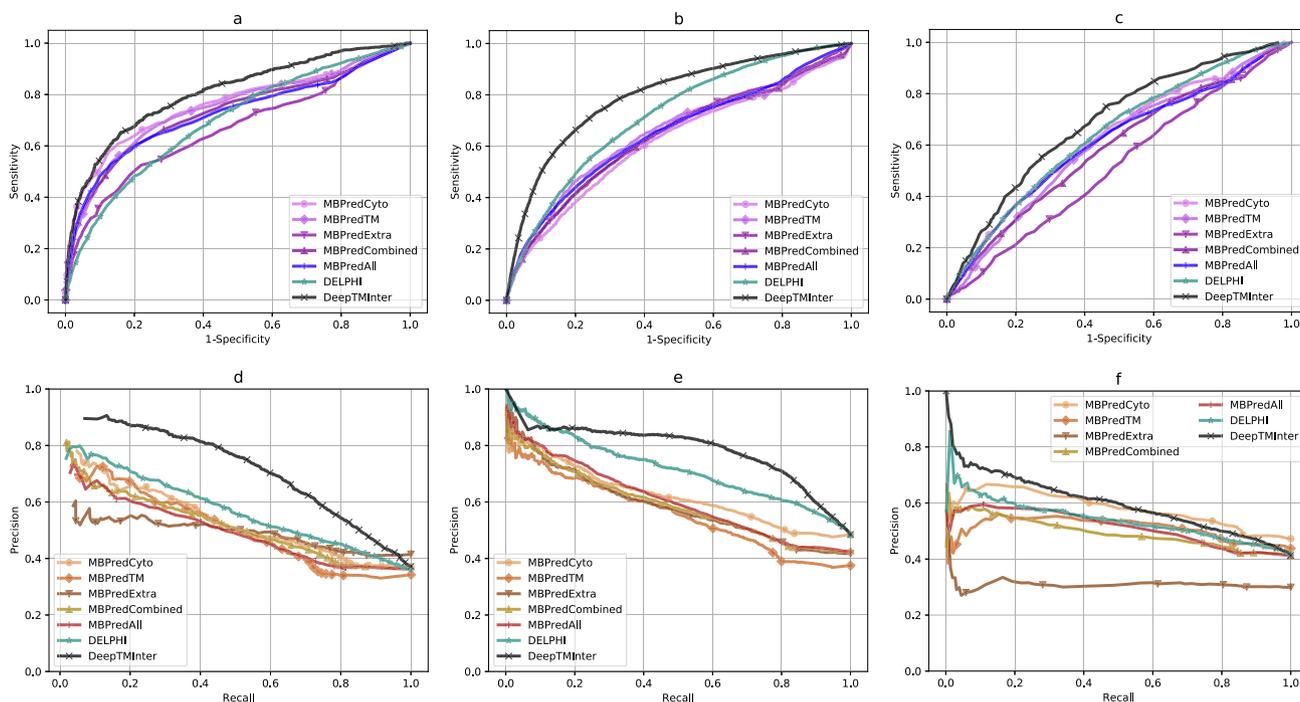
### 3.4. Performance comparison of DeepTMInter with other methods

We compared the prediction performance of DeepTMInter with the four underlying predictors (MBPredTM, MBPredCyto, MBPredExtra, and MBPredAll) in the MBPred suite, with the ensemble predictor MBPredCombined, as well as with DELPHI (see Section 2.8). Due to the adoption of the early stopping strategy to prevent over-training (see Section 2.6.5), our trained model achieved high prediction performance not only on the two previous test datasets (TestData and CompData) (Tables 1 and 2), but also on the independent test dataset (IndepData) (Figs. 6 and 7, Table 3, and Supplementary Tables S13-S14).

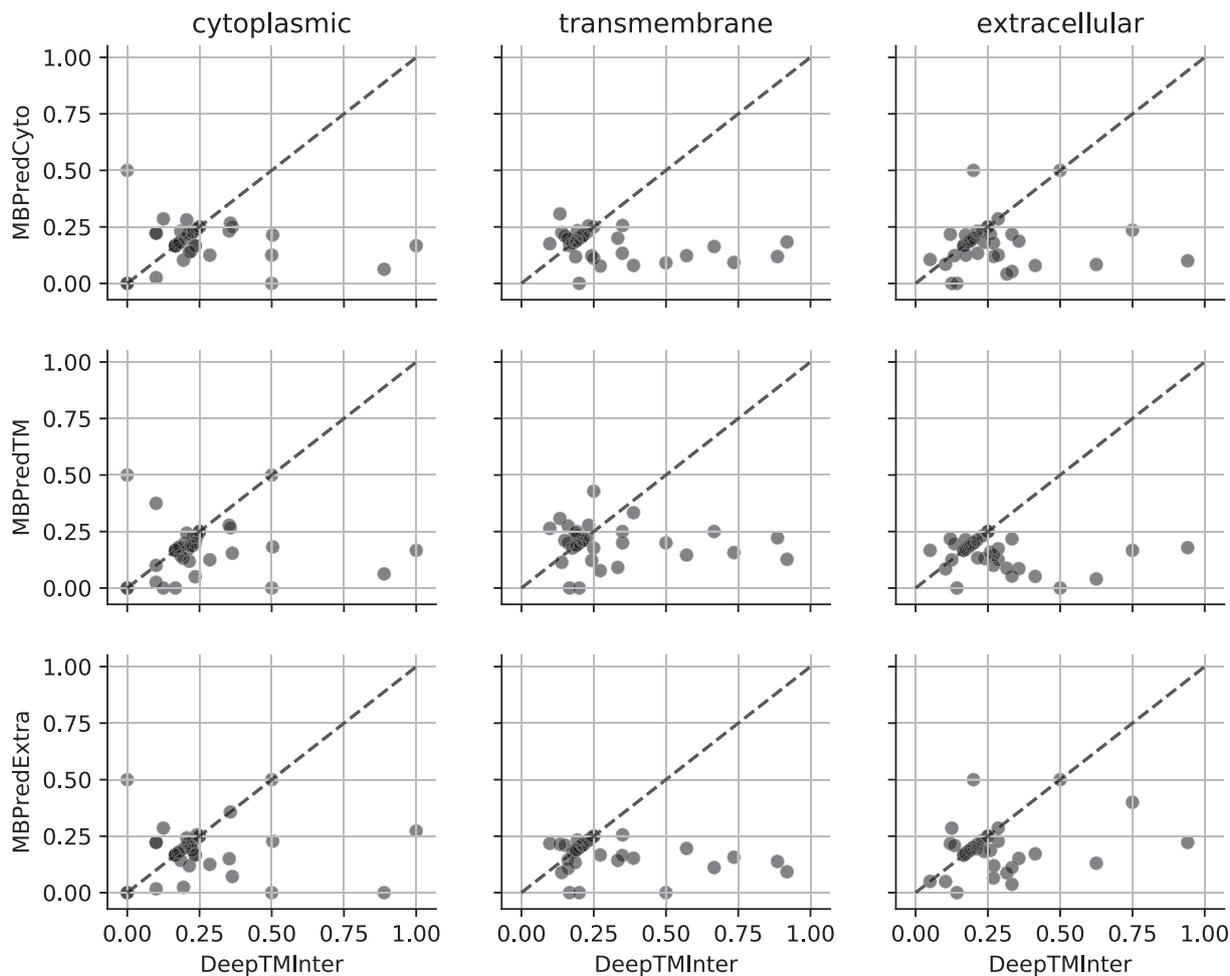
**Table 2**

Performance gauged by mean precision (P), recall (R), specificity (SP), accuracy (ACC), MCC, F1-score (F1), and AUC, and AUCPR (see Section 2.7) using structure-derived Cyto, TMH, Extra, and Combined regions on the CompData dataset.

Predictor	Region	L/5						L/10							
		P	R	SP	ACC	MCC	F1	P	R	SP	ACC	MCC	F1	AUC	AUCPR
MBPredCyto	Cyto	0.727	0.236	0.838	0.432	0.082	0.337	0.693	0.115	0.924	0.385	0.066	0.190	0.618	0.622
MBPredTM		0.674	0.215	0.809	0.410	0.020	0.309	0.616	0.102	0.907	0.371	0.008	0.168	0.571	0.552
MBPredExtra		0.681	0.216	0.821	0.411	0.030	0.308	0.663	0.112	0.914	0.379	0.048	0.181	0.585	0.570
MBPredAll		0.764	0.261	0.855	0.454	0.139	0.363	0.728	0.132	0.935	0.400	0.103	0.210	0.656	0.664
DELPHI		0.754	0.248	0.870	<b>0.463</b>	0.136	0.358	0.751	0.129	0.942	<b>0.414</b>	0.111	0.212	0.735	0.717
<b>DeepTMInter</b>		<b>0.804</b>	<b>0.289</b>	<b>0.900</b>	0.461	<b>0.174</b>	<b>0.387</b>	<b>0.764</b>	<b>0.156</b>	<b>0.957</b>	0.400	<b>0.129</b>	<b>0.229</b>	<b>0.807</b>	<b>0.779</b>
MBPredCyto	TMH	0.598	0.227	0.818	0.505	0.074	0.307	0.594	0.120	0.911	0.486	0.049	0.188	0.578	0.452
MBPredTM		0.663	0.269	0.850	0.531	0.147	0.354	0.715	0.149	0.931	0.507	0.133	0.232	0.650	0.558
MBPredExtra		0.574	0.209	0.812	0.495	0.047	0.289	0.570	0.108	0.906	0.481	0.030	0.173	0.576	0.444
MBPredAll		0.662	0.270	0.848	0.535	0.150	0.355	0.667	0.144	0.931	0.505	0.106	0.222	0.669	0.566
DELPHI		0.664	0.285	0.853	<b>0.553</b>	0.171	0.365	0.699	0.159	0.933	<b>0.528</b>	0.146	0.237	0.715	0.571
<b>DeepTMInter</b>		<b>0.719</b>	<b>0.319</b>	<b>0.889</b>	0.552	<b>0.211</b>	<b>0.403</b>	<b>0.757</b>	<b>0.181</b>	<b>0.955</b>	0.517	<b>0.174</b>	<b>0.267</b>	<b>0.803</b>	<b>0.716</b>
MBPredCyto	Extra	0.647	0.216	0.823	0.472	0.075	0.308	0.617	0.107	0.913	0.435	0.040	0.176	0.591	0.515
MBPredTM		0.589	0.194	0.801	0.450	0.016	0.277	0.545	0.094	0.899	0.422	-0.007	0.154	0.545	0.448
MBPredExtra		0.696	0.243	0.863	0.492	0.133	0.341	0.703	0.131	0.943	0.453	0.110	0.212	0.643	0.586
MBPredAll		0.681	0.244	0.854	0.496	0.124	0.339	0.701	0.135	0.937	0.463	0.110	0.216	0.635	0.581
DELPHI		0.664	0.244	0.861	0.499	0.110	0.338	0.700	0.135	0.936	<b>0.474</b>	0.110	0.216	0.706	0.626
<b>DeepTMInter</b>		<b>0.784</b>	<b>0.315</b>	<b>0.895</b>	<b>0.527</b>	<b>0.234</b>	<b>0.415</b>	<b>0.792</b>	<b>0.181</b>	<b>0.959</b>	<b>0.474</b>	<b>0.187</b>	<b>0.272</b>	<b>0.777</b>	<b>0.712</b>
MBPredAll	Combined	0.687	0.260	0.856	0.509	0.134	0.360	0.707	0.136	0.936	0.475	0.104	0.220	0.651	0.604
MBPredCombined		0.679	0.260	0.855	0.510	0.129	0.359	0.717	0.141	0.938	0.481	0.114	0.227	0.635	0.589
DELPHI		0.698	0.275	0.868	0.538	0.169	0.379	0.729	0.145	0.939	<b>0.505</b>	0.135	0.235	0.718	0.633
<b>DeepTMInter</b>		<b>0.773</b>	<b>0.320</b>	<b>0.903</b>	<b>0.544</b>	<b>0.229</b>	<b>0.426</b>	<b>0.799</b>	<b>0.174</b>	<b>0.960</b>	0.494	<b>0.177</b>	<b>0.270</b>	<b>0.796</b>	<b>0.738</b>



**Fig. 6.** Performance comparison of DeepTMInter with MBPred and DELPHI. (a), (b), and (c) show the ROC curves on the TestData, CompData, and IndepData datasets, respectively, while (d), (e), and (f) show the Precision-Recall curves on the TestData, CompData, and IndepData datasets, respectively.



**Fig. 7.** Comparison of JSCs (Jaccard similarity coefficients) between DeepTMInter and the three specialized MBPred predictors (MBPredTM, MBPredCyto, and MBPredExtra) on the TestData dataset. Each dot corresponds to one protein chain.

**Table 3**

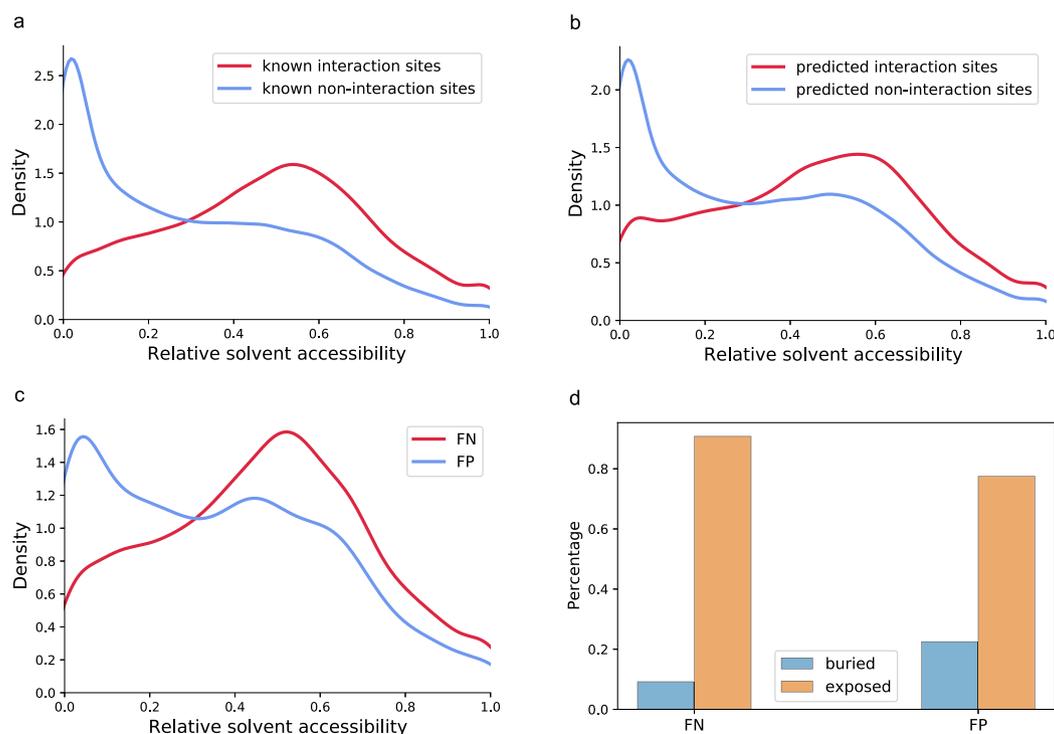
Performance gauged by mean precision (P), recall (R), specificity (SP), accuracy (ACC), MCC, F1-score (F1), and AUC, and AUCPR (see Section 2.7) using structure-derived Cyto, TMH, Extra, and Combined regions on the IndepData dataset.

Predictor	Region	L/5						L/10							
		P	R	SP	ACC	MCC	F1	P	R	SP	ACC	MCC	F1	AUC	AUCPR
MBPredCyto	Cyto	0.612	0.230	0.853	0.506	0.104	0.322	0.608	0.113	0.932	0.479	0.070	0.185	0.624	0.581
MBPredTM		0.538	0.206	0.824	0.476	0.027	0.285	0.569	0.108	0.919	0.470	0.040	0.176	0.568	0.522
MBPredExtra		0.524	0.205	0.819	0.471	0.019	0.278	0.554	0.111	0.919	0.467	0.037	0.174	0.591	0.540
MBPredAll		<b>0.659</b>	0.246	<b>0.868</b>	<b>0.519</b>	<b>0.153</b>	0.347	0.641	0.118	0.936	0.480	0.090	0.195	0.648	0.622
DELPHI		0.616	0.223	0.844	0.507	0.110	0.317	0.614	0.107	0.920	0.478	0.069	0.178	0.632	0.600
<b>DeepTMInter</b>		0.633	<b>0.269</b>	0.855	0.514	0.135	<b>0.352</b>	<b>0.687</b>	<b>0.147</b>	<b>0.937</b>	<b>0.493</b>	<b>0.129</b>	<b>0.228</b>	<b>0.689</b>	<b>0.657</b>
MBPredCyto	TMH	0.588	0.206	0.809	0.498	0.058	0.295	0.603	0.106	0.907	0.483	0.048	0.177	0.566	0.482
MBPredTM		0.635	0.242	<b>0.837</b>	0.515	0.090	0.335	0.652	0.124	<b>0.921</b>	0.493	0.073	0.204	0.603	0.513
MBPredExtra		0.594	0.211	0.815	0.500	0.063	0.299	0.622	0.113	0.908	0.488	0.061	0.185	0.548	0.476
MBPredAll		<b>0.649</b>	0.240	0.836	<b>0.524</b>	<b>0.125</b>	<b>0.336</b>	0.644	0.116	0.915	0.494	0.080	0.192	0.593	0.502
DELPHI		0.634	0.234	0.829	0.516	0.106	0.329	0.617	0.113	0.913	0.486	0.059	0.188	0.596	0.533
<b>DeepTMInter</b>		0.635	<b>0.246</b>	0.834	0.515	0.109	0.331	<b>0.676</b>	<b>0.131</b>	0.919	<b>0.499</b>	<b>0.107</b>	<b>0.209</b>	<b>0.661</b>	<b>0.603</b>
MBPredCyto	Extra	0.514	0.216	0.803	0.555	0.074	0.288	0.444	0.093	0.898	0.545	0.031	0.148	0.571	0.355
MBPredTM		0.518	0.238	0.807	0.556	0.085	0.301	0.471	0.112	0.909	0.551	0.055	0.170	0.581	0.386
MBPredExtra		0.463	0.204	0.804	0.539	0.029	0.265	0.465	0.106	0.902	0.543	0.042	0.163	0.519	0.312
MBPredAll		0.520	0.232	0.807	0.556	0.075	0.299	0.508	0.130	0.905	0.553	0.065	0.192	0.571	0.367
DELPHI		<b>0.560</b>	<b>0.269</b>	<b>0.822</b>	<b>0.576</b>	0.122	<b>0.342</b>	<b>0.543</b>	<b>0.131</b>	<b>0.915</b>	<b>0.563</b>	<b>0.105</b>	<b>0.202</b>	<b>0.703</b>	<b>0.476</b>
<b>DeepTMInter</b>		0.558	0.259	0.820	0.573	<b>0.133</b>	0.328	0.503	0.112	0.913	0.555	0.075	0.175	0.688	0.458
MBPredAll	Combined	0.608	0.249	0.840	0.571	0.139	0.344	0.597	0.122	0.918	0.548	0.084	0.200	0.610	0.514
MBPredCombined		0.573	0.240	0.834	0.557	0.104	0.325	0.573	0.118	0.916	0.542	0.068	0.190	0.589	0.493
DELPHI		0.578	0.238	0.831	0.559	0.109	0.325	0.603	0.126	0.919	0.548	0.091	0.202	0.642	0.541
<b>DeepTMInter</b>		<b>0.612</b>	<b>0.276</b>	<b>0.846</b>	<b>0.573</b>	<b>0.151</b>	<b>0.360</b>	<b>0.658</b>	<b>0.151</b>	<b>0.928</b>	<b>0.559</b>	<b>0.135</b>	<b>0.233</b>	<b>0.689</b>	<b>0.598</b>

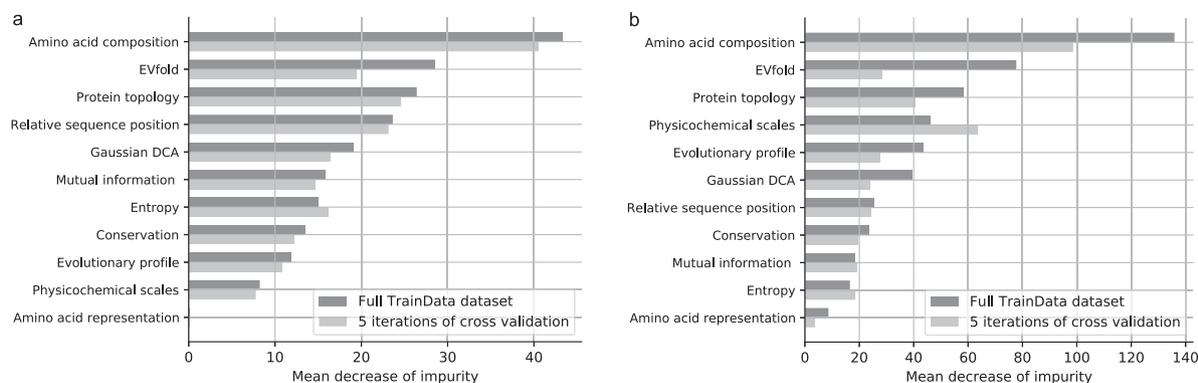
### 3.4.1. Performance comparison using threshold-free measures

On all test datasets the AUC and AUCPR performance of predictors was benchmarked using the Cyto, TMH, Extra and Combined regions either defined according to PDBTM or predicted by Phobius. For the three specialized predictors (MBPredCyto, MBPredTM, and MBPredExtra), we calculated the AUC and AUCPR values not only for their specialized regions but also for the regions on which

they were not trained (Table 3 and Supplementary Tables S13 and S14). Overall, DeepTMInter shows a significant improvement in terms of the AUC and AUCPR performance compared to the other two state-of-the-art methods. For example, on the IndepData dataset our method gives the AUC value of 0.661 (AUCPR = 0.603), distinctly higher than 0.603 (0.513) of MBPredTM and 0.596 (0.533) of DELPHI using structure-derived TMH regions. Note that DELPHI



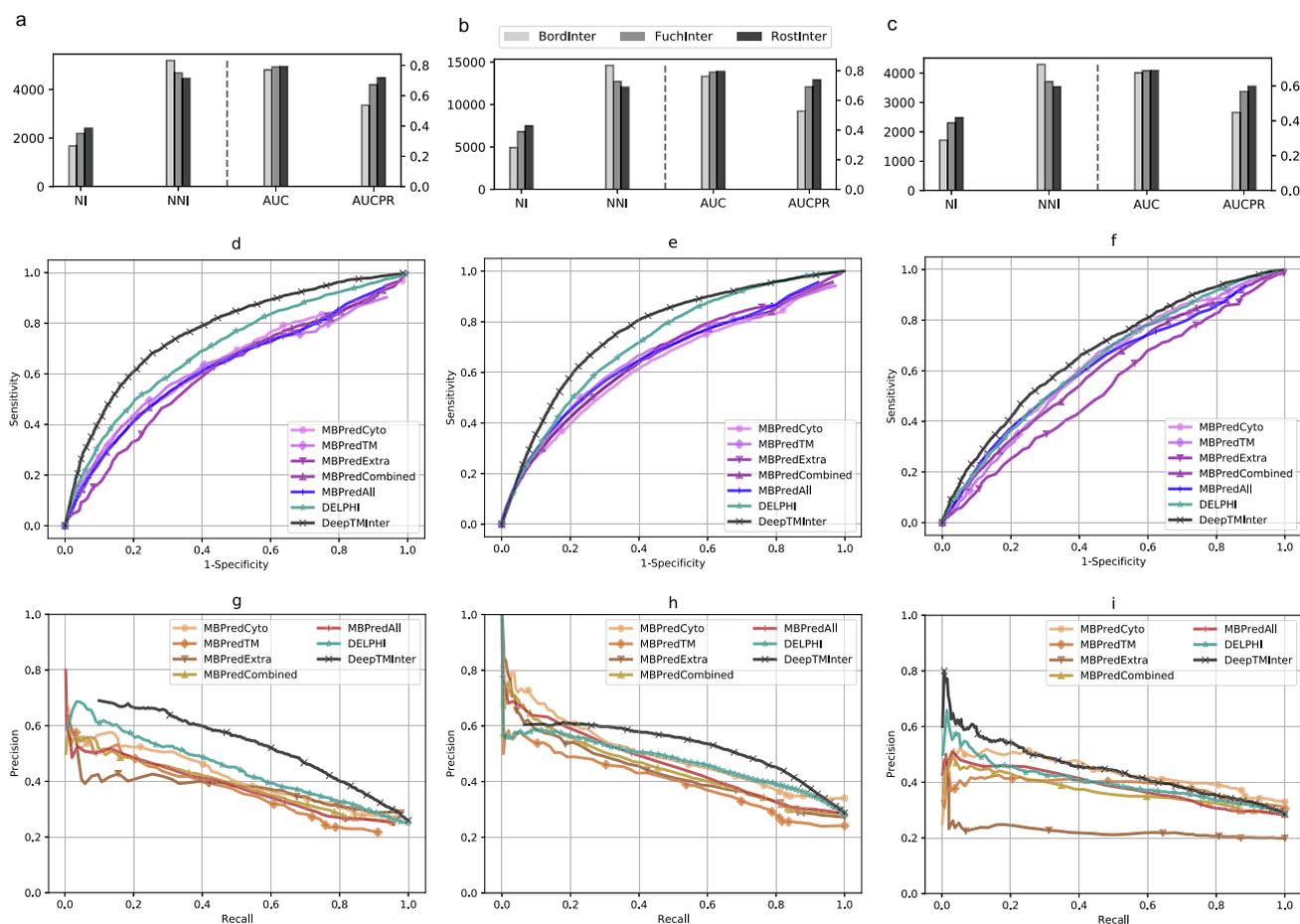
**Fig. 8.** Relative solvent accessibility of interaction and non-interaction sites. (a) and (b) show the RSA density of actual and predicted interaction and non-interaction sites, respectively. (c) shows the RSA density of FN and FP predictions. The densities in (a)–(c) are calculated by using the Gaussian kernel density estimate (Gaussian KDE) from the Seaborn library [78]. (d) shows the percentage of buried and exposed sites among the false predictions.



**Fig. 9.** Feature importance measured by mean decrease of impurity (MDI). The mean (a) and the max (b) MDI values of 11 unique features are achieved over the full training and the 5-iteration cross-validation datasets, respectively.

achieves the best AUC (0.703) and AUCPR (0.476) performance using the Extra region. Interestingly, on the TestData and CompData datasets, DeepTMInter also significantly outperforms all other methods in the Extra regions (Tables 1 and 2). To investigate the weaker performance of our method in this region on the IndepData dataset, we derived from protein structures the number of interaction and non-interaction sites (Supplementary Table S18) and found that unlike the even ratios of interaction vs. non-interaction sites in the Cyto (1199 vs. 1338) and TMH (766 vs.

984) regions, sites in the Extra region are clearly biased towards the non-interaction class with the ratio of 516 vs. 1210. We therefore speculate that this imbalance complicates distinguishing the interaction sites from the overrepresented non-interaction sites. By contrast, the method DELPHI, trained on non-TM proteins, still performs well in this non-TM region [10]. MBPredCyto, MBPredTM, and MBPredExtra have been reported to perform best in predicting interacting amino acid residues located in their respective regions (Cyto, TMH, and Extra) on the TestData dataset [12]. Indeed, we



**Fig. 10.** Statistics and performance comparison using the BordInter, FuchInter, and RostInter residue contact definitions on the TestData (a), CompData (b), and IndepData (c) datasets. Left side: NI - number of interacting amino acid residues; NNI - number of non-interacting amino acid residues. Right side: AUC and AUCPR. (d), (e), and (f) show the ROC curves on the TestData, CompData, and IndepData datasets, respectively based on the BordInter interaction definition, while (g), (h), and (i) show the Precision-Recall curves on the TestData, CompData, and IndepData datasets, respectively. Similar to Fig. 6, the ROC and Precision-Recall curves of three separate predictors (MBPredCyto, MBPredTM, and MBPredExtra) are evaluated using the predictions for the respective regions they were trained on.

found that this is the case both on the CompData and IndepData datasets. For example, on the CompData dataset among all specialized predictors in the MBPred suite MBPredCyto, MBPredTM, and MBPredExtra yield the highest AUC values of 0.618, 0.650, and 0.643 (AUCPR = 0.622, 0.558, and 0.586).

Fig. 6 shows the ROC and Precision-Recall curves of predictors using their specialized structure-derived regions on all test datasets. DeepTMInter is clearly superior to the MBPred suite and DELPHI and achieves the highest AUC (0.793, 0.796, and 0.689) (Fig. 6a-c) and AUCPR values (0.718, 0.738, and 0.598) (Fig. 6d-f). Overall, the MBPred suite is mostly comparable to DELPHI in terms of the AUC performance across the three test datasets on average, but is worse than DELPHI in terms of the AUCPR performance (Fig. 6). On all test datasets MBPred predictors exhibit comparable performance in predicting interaction sites located in the regions they are specifically trained on. For example, on the IndepData dataset MBPredCyto and MBPredTM produce similar ROC curves corresponding to the AUC values of 0.624 and 0.603, respectively.

### 3.4.2. Performance comparison using single-threshold measures

For proteins in all test datasets mean precision, recall, F1-score, and MCC were calculated using entire combined structure-derived regions (Tables 1–3). Based on these performance measures DeepTMInter is also way ahead of the MBPred suite and DELPHI. For example, on the CompData dataset DeepTMInter achieved the highest precision 0.783, recall 0.324, F1-score 0.432, and MCC 0.239 values. In addition, JSC (Jaccard similarity coefficient, see Section 2.7) was used to assess the similarity between a set of actual labels and a set of predicted labels for sites in TM proteins [73,74]. A high JSC is indicative of high performance of a predictor.

For the three structure-derived regions (Cyto, TMH, and Extra) on the TestData dataset we compared JSCs of DeepTMInter to those of the MBPredTM, MBPredCyto, and MBPredExtra, respectively, for each individual protein. As seen in Fig. 7, DeepTMInter vastly outperforms the three underlying MBPred predictors on all 36 proteins and in all the three structure-derived regions (Cyto, TMH, and Extra), with mean JSCs 0.258, 0.298, and 0.274 (averaged over JSCs of all proteins in that dataset) compared to 0.182, 0.199, and 0.194 of MBPredCyto, MBPredTM, and MBPredExtra, as well as 0.205, 0.197, and 0.185 of DELPHI. Mean JSC values indicate that the set of predicted labels corresponding to protein sites produced by DeepTMInter shows a stronger agreement to the experimentally determined sites (the set of their actual labels) than those produced by MBPred. Additionally, we also evaluated the JSC performance on the IndepData dataset, e.g., with DeepTMInter achieving 0.220, 0.204, and 0.221 compared to DELPHI achieving 0.196, 0.208, and 0.198 on the Cyto, TMH, and Combined regions, respectively, and furthermore compared to MBPredCyto, MBPredTM, and MBPredExtra achieving 0.199, 0.207, and 0.162 using their specialized regions, respectively. In the Extra region DELPHI achieves a significantly better JSC of 0.226 compared to 0.179 of DeepTMInter.

### 3.4.3. Exposure states of falsely classified amino acid positions

To get an understanding of how the falsely predicted interaction sites are correlated with interface properties, we calculated the occurrence of misclassified amino acid positions in exposed and buried regions of structures (Fig. 8). For each amino acid, the relative solvent accessibility (RSA) [75] was calculated based on known 3D structures using the DSSP software [76]. The RSA threshold for

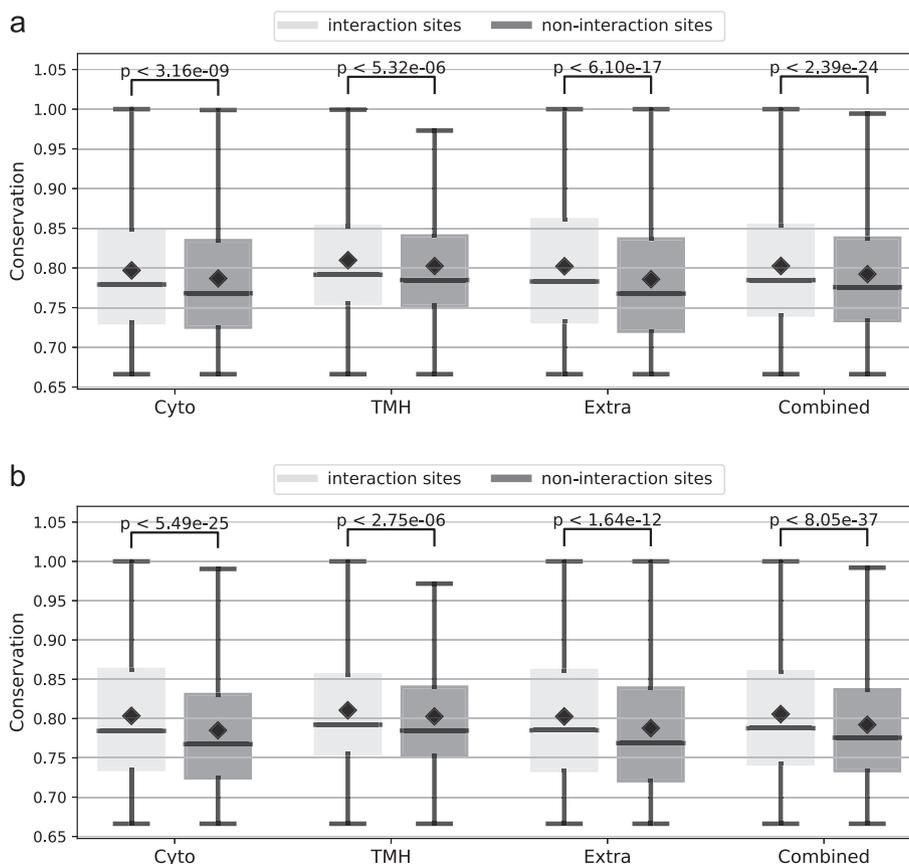


Fig. 11. Evolutionary conservation of interaction and non-interaction sites derived from structures (a) and predicted by DeepTMInter (b) across all three test datasets (TestData, CompData, and IndepData) in the Cyto, TMH, Extra, and Combined (the full protein sequence) regions. Statistical significance of the difference between interaction and non-interaction sites is inferred by p-values obtained by *t*-Test.

classifying residues as exposed or buried was set to 0.1, as suggested in [77]. We found that in terms of the RSA values the interaction sites predicted by DeepTMInter closely follow the distribution of values for the interaction sites derived from 3D structures (Fig. 8a and b). Both actual and predicted interaction sites are overwhelmingly located in the solvent (or lipid) accessible regions of proteins. While the false negative (FN) predictions (assigning established interaction sites as non-interacting) are apparent errors, the false positive (FP) predictions might point to unknown interaction sites that have yet to be experimentally confirmed. Overall, falsely predicted positions, both FN and FP, tend to reside in exposed areas of structure (Fig. 8c and d).

### 3.4.4. Feature importance analysis

We performed an importance analysis of topological, physiochemical, and evolutionary features on our full training and 5-iteration cross-validation datasets (Fig. 9). Since we are not aware of any measures that are able to deduce feature importance from deep-learning model training phases [79], we followed the common practice [12,79,80] to quantify the feature importance based on mean decrease of impurity (MDI) inferred from XGBoost [79,81]. The MDI values provide support for recursively splitting the feature set in the XGBoost method to classify sites as interact-

ing or non-interacting. In each split the features that best improve the classification performance are identified. For visualization purposes we evaluated the mean (Fig. 9a) and the maximal (Fig. 9b) MDI values of the different feature categories within sliding windows (see reference [79]). Amino acid composition makes the strongest contribution to the interaction site classification among all features. The EVfold coevolution feature remains very powerful for distinguishing between interaction and non-interaction sites (the EVfold feature has also been ranked best in our previous analysis for MBPred [12]) while the Gaussian DCA co-evolution makes a moderate contribution. This finding is in line with previous reports that residues involved in binding sites are collectively under functional constraints and are therefore subject to co-evolution [82]. Features representing amino acid types and reflecting their physiochemical properties only have run-of-the-mill importance (Fig. 9).

### 3.5. Performance evaluation using different residue contact definitions

To evaluate how DeepTMInter performance in predicting interaction sites depends on the choice of a particular residue contact definition, the AUC and AUCPR values were calculated on TestData, CompData, and IndepData datasets and compared using the

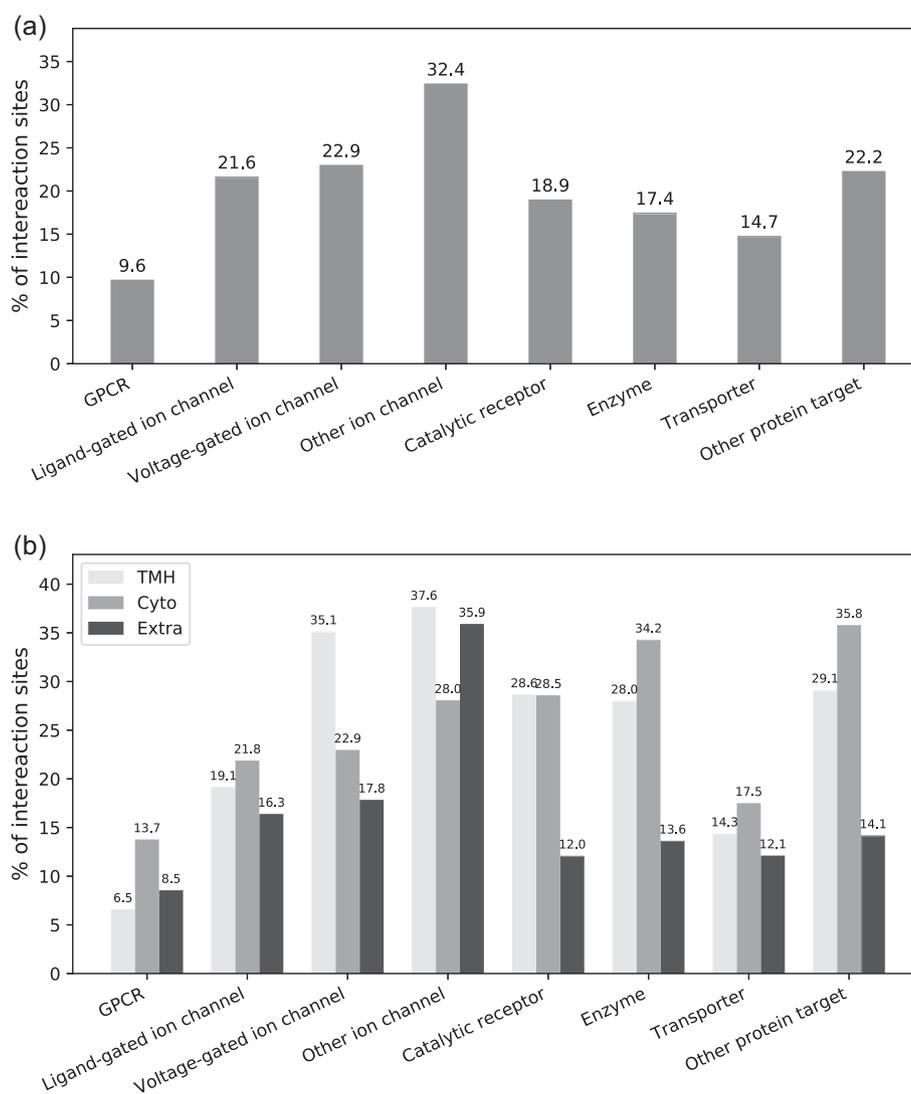


Fig. 12. Percentage of amino acids predicted to be involved in interactions in the complete protein sequences (a) and in the TMH, Cyto, and Extra regions (b).

BordInter [11], FuchsInter [83], and RostInter [31] residue contact definitions (Fig. 10a-c and Supplementary Table S15). Note that the sites in the full protein sequences (the Combined region) obtained by DeepTMInter were involved in the calculation of the two criteria above. The numbers of interacting (NI) and non-interacting (NNI) amino acid residues were derived from experimental 3D structures (Fig. 10a-c and Supplementary Table S16). As expected, the number of residue contacts increases progressively with the spatial distance cutoff according to the BordInter (4 Å), FuchInter (5.5 Å), and RostInter (6 Å) definitions. The RostInter definition leads to the highest AUC and AUCPR values on the three test datasets. For example, on the CompData dataset the AUC (0.762, 0.790, and 0.796) and AUCPR values (0.527, 0.690, and 0.738) were obtained using the BordInter, FuchInter, and

RostInter definitions, respectively. A higher distance threshold (RostInter) also results in more residues labeled as interacting, thus partially alleviating the imbalance between the two residue classes (interacting and non-interacting), while the low distance threshold BordInter impairs the performance by increasing the imbalance as discussed in [12] (see also Supplementary Table S16). Using the most stringent contact definition (BordInter) DeepTMInter remains the most accurate method on structure-derived Combined regions (Fig. 10d-i). For example, it achieves an AUC value of 0.675 (AUCPR = 0.448) compared to 0.639 (0.402) of DELPHI on the IndepData dataset. Based on the Extra region and the IndepData dataset, DELPHI has the highest AUC value of 0.700 (AUCPR = 0.338) compared to that of 0.668 (AUCPR = 0.303) for DeepTMInter. Note that the distance threshold utilized for training DELPHI - the sum

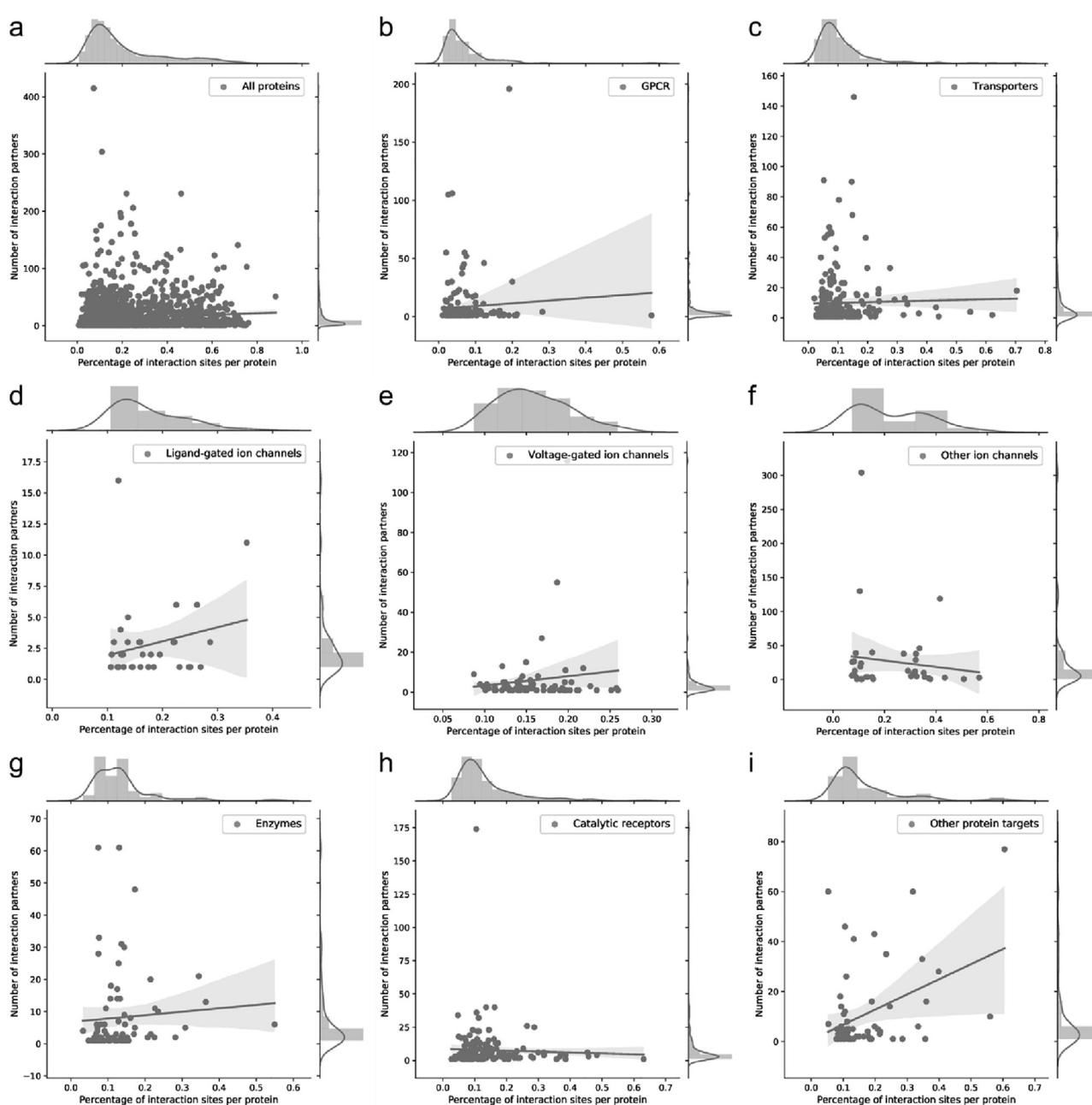


Fig. 13. Number of interaction partners versus the percentage of amino acids involved in interactions in all human transmembrane proteins (a) and in separate functional families (b)-(i).

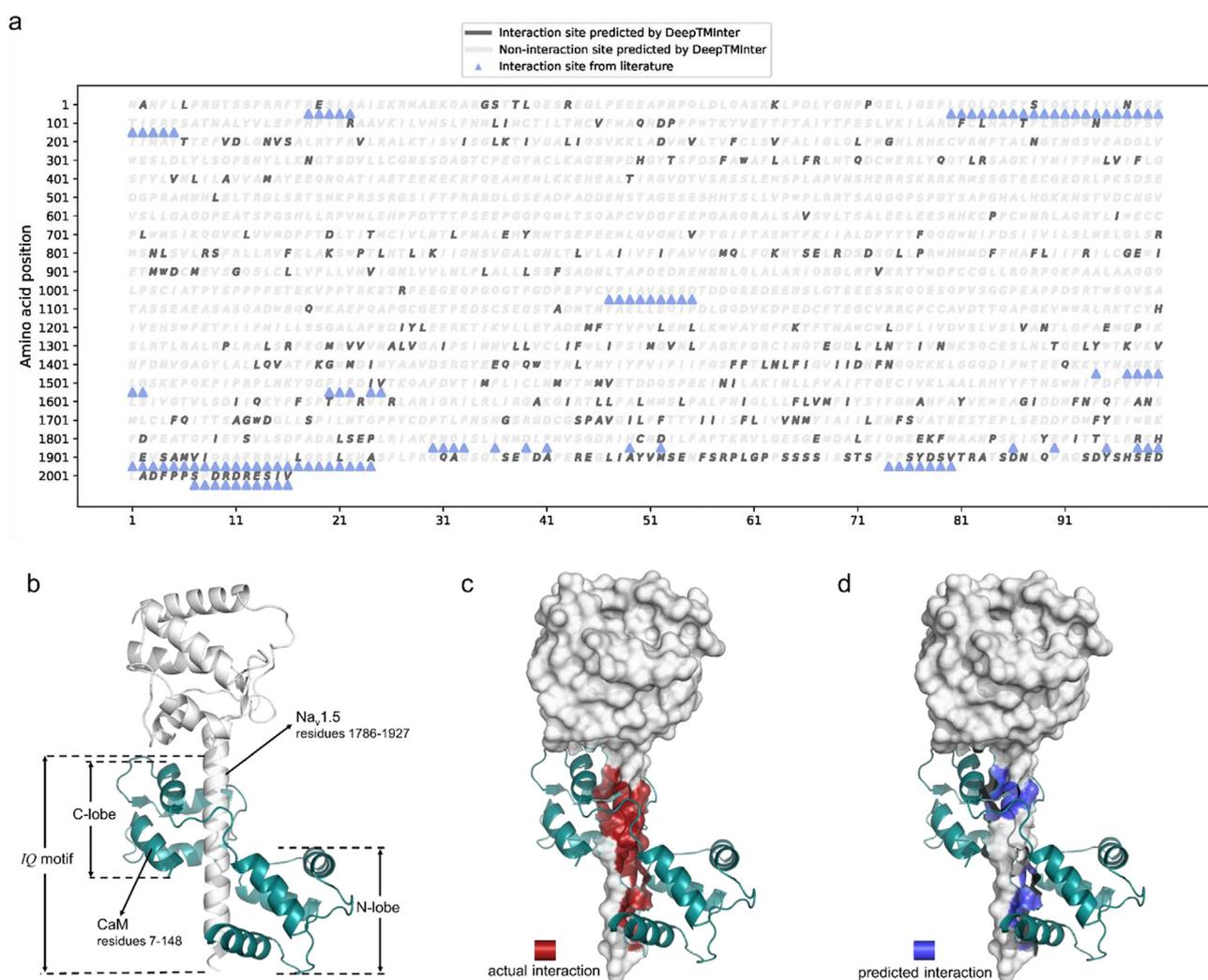
of the Van der Waal's radii of two atoms +0.5 Å [10] – is in most cases smaller than the BordInter, FuchInter, and RostInter distance thresholds.

### 3.6. Evolutionary conservation of interaction sites

We compared the conservation scores (ranging from 0 to 1) of interaction and non-interaction sites in the Cyto, TMH, Extra, and Combined regions (Fig. 11), disregarding alignment columns with more than 50% of gaps. In line with [11] and our own previous work [12], interaction sites both derived from structures (Fig. 11a) and predicted by DeepTMInter (Fig. 11b) are significantly more evolutionarily conserved than non-interaction sites in all four regions (Cyto, TMH, Extra, and Combined). The Combined region (i.e., the full sequence) displays the most statistically significant difference between interaction and non-interaction sites both derived from structures (p-value 2.39e–24, *t-Test*) and predicted by DeepTMInter (p-value 8.05e–37, *t-Test*). Interaction sites in the transmembrane domains, while still more conserved compared to the positions not involved in interactions, exhibit a lower p-value of 5.32e–06 due to the degenerate amino acid composition and hence stronger overall conservation of hydrophobic, lipid-immersed sequence segments [84,85].

### 3.7. Family-specific analysis of interaction sites and network connectivity in human transmembrane proteins

We investigated the relationship between the biological activities of proteins and their interaction patterns by analyzing the average percentage of interaction sites in the eight major membrane protein classes (see Section 2.9) curated by the GtoPdb database (Fig. 12a and b, and Supplementary Tables S19–S26 and Fig. S14). Overall, ligand-gated, voltage-gated, and other ion channels appear to have the largest percentage of their amino acid sequence involved in interactions (21.6%, 22.9%, and 32.4%, respectively) (Fig. 12a). Within the latter class (Supplementary Figs. S5 and S13) Orai channels [86] as well as connexins and pannexins [87] stand out as having the highest proportion of interaction sites. Similarly, among the 20 sub-families within the “other protein target” class, *other pattern recognition receptors*, *sigma receptors* and *abscisic acid receptor complex* all have more than 40% of their sequence predicted to mediate interactions (Supplementary Fig. S9). Proteins in these three sub-families play an important role in signaling pathways [88–90]. Among the eight functional groups, the “other ion channel” class harbors the highest percentages (37.6% and 35.9%) of interaction sites in the TMH and the Extra regions (Fig. 12b and Supplementary Figs. S6–S8), respectively,

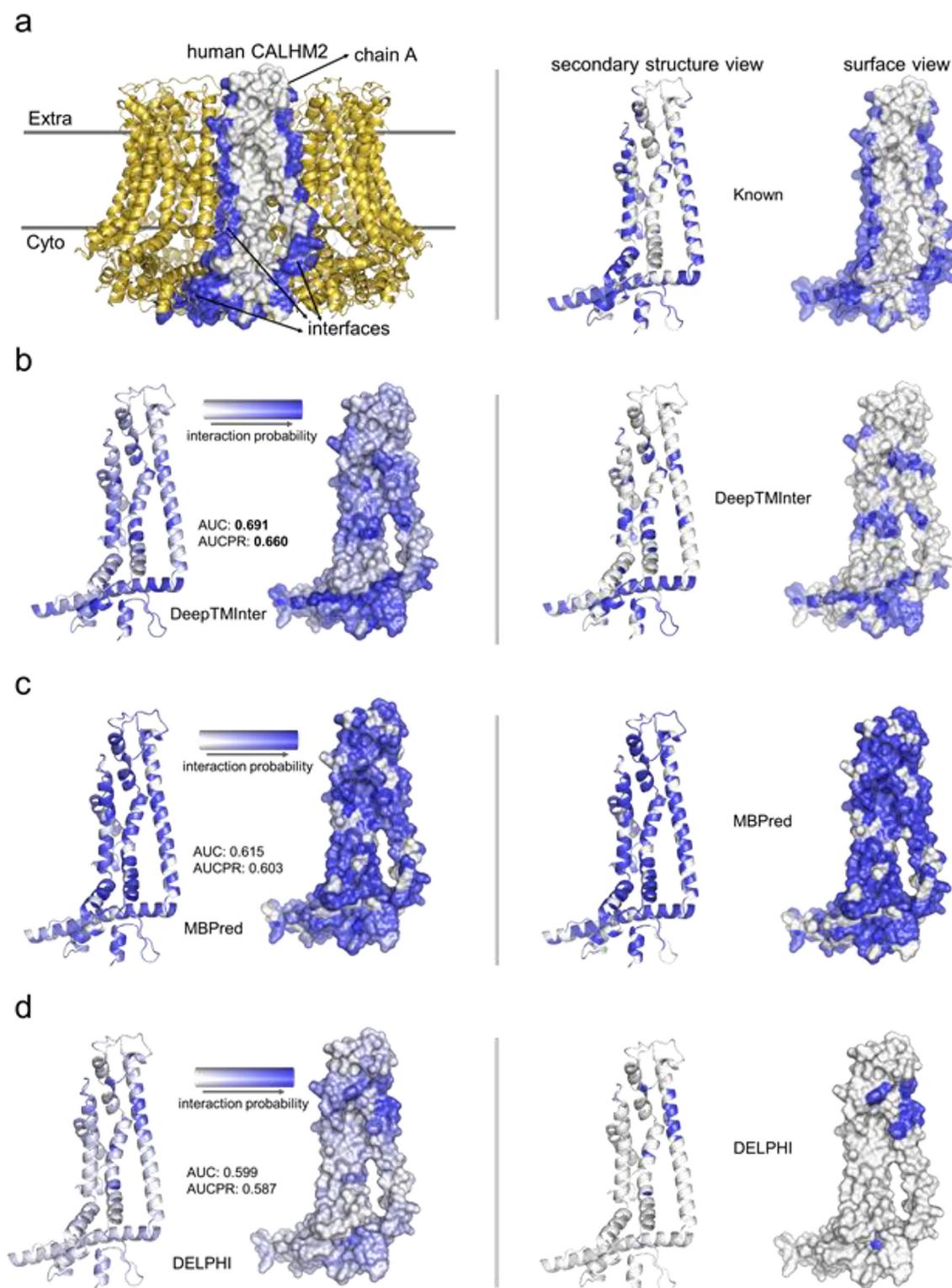


**Fig. 14.** Overview of interaction sites in Na<sub>v</sub>1.5. (a) Comparison of known interaction sites reported in the literature with DeepTMInter predictions. (b) The Nav1.5 fragment colored in white (PDB code: 4jq0 chain D) interacts with the C- and N-lobes of calmodulin (PDB code: 4jq0 chain A) colored in cyan through its IQ motif. The known and predicted interaction surfaces of Nav1.5 are colored in red (c) and blue (d), respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

while in the “Enzyme” and “other protein target” classes interaction sites reside in the Cyto region (Fig. 12b and Supplementary Figs. S10–S12).

In most of the families the percentage of the amino acids involved in interactions lies within the range of 10%–25%, while

the “catalytic receptors” and “other ion channel” classes are characterized by a wider range of 10%–50% (Fig. 13). On average human transmembrane protein interact with 10.5 protein partners according to the HuRI-Union database (see Section 2.10). We found no clear dependence (low  $R^2$  values) of the number of interaction



**Fig. 15.** Interaction interfaces of the human CALHM2 chain A. (a) Amino acid positions involved in protein interactions derived from the experimental 3D structure according to the RostInter definition (see *Methods*) are shown in blue. (b), (c), and (d) Interaction probabilities (left) and binary predictions of interacting (blue) vs. non-interacting residues (right) generated by DeepTMInter, MBPred, and DELPHI, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

partners on the number of amino acid positions predicted to be involved in interactions both across all human transmembrane proteins (Fig. 13a) and in each of the eight major functional families (Fig. 13b–i). Such dependence might be obscured by the fact that the same binding site often mediates interactions with a large number of partners [91].

### 3.8. Case studies

#### 3.8.1. Human cardiac voltage-gated sodium channel

The human cardiac voltage-gated sodium channel Nav1.5, encoded by the gene SCN5A, is expressed in the cardiomyocyte sarcolemma to carry an inward depolarizing current during phase 0 of the cardiac action potential [92]. According to the BioGRID database Nav1.5 has 22 interaction partners, and we identified additional 21 interaction partners based on literature analysis (Supplementary Table S27) [93–99]. Only a small number of interaction sites have been described in these studies, of which most are situated in the C-terminal portion of Nav1.5 (positions 1773–2016, indicated by blue triangles in Fig. 14a) [100] and only a few appear in the N-terminal portion [101–103]. The tail of its C-terminal domain is involved in  $Na^+$  gating inactivation [104] and mutations in this area cause many diseases such as long QT and Brugada syndromes [105]. One ten-residue peptide (positions 2007–2016) [106] and four motifs - PY (1974–1976) [107], extended PY (1974–1980) [108], SXV (2014–2016) [106], and IQ (1901–1927) [105] - localized in the tail of the C-terminal portion, frequently interact with other proteins (Supplementary Table S27). DeepTMInter predicts 17.10% of amino acid sites in Nav1.5 to be involved in interactions (Supplementary Table S28) and these predicted interaction sites occur more frequently in the C-terminal part of the Nav1.5 sequence (Fig. 14a). DeepTMInter was able to capture some of the experimentally established interfaces, including 9 out of 10 residue positions mediating the interaction with syntrophin [106], 5 out of 7 residue positions in the extended PY motif, and 8 positions within the IQ motif involved in the interaction with calmodulin (Fig. 14b–d) [100,105,109]. Interactions between  $Na_v1.5$  and other proteins also occur via the same interfaces; e.g., three proteins encoded by the genes NEDD4, NEDD4L, and WWP2 interact with the PY motif of  $Na_v1.5$  (see Supplementary Table S27).

#### 3.8.2. Human calcium homeostasis modulator-2

The human calcium homeostasis modulator-2 (CALHM2), whose 3D structure (PDB code: 6uiw) was released in 2019, is a voltage-gated  $Ca^{2+}$ -inhibited non-selective channel controlling ATP release [110]. The entire biological assembly of the human CALHM2 channel (Fig. 15a) consists of 11 subunits arranged in a circular fashion. We used the CALHM2 chain A from the IndepData dataset to examine the performance of three interaction site prediction methods: DeepTMInter, MBPred, and DELPHI. Overall, predictions generated by DeepTMInter (Fig. 15b) correspond better to the experimentally confirmed interfaces than the MBPred and DELPHI results (Figs. 15c and 15d). DeepTMInter achieves the highest AUC value of 0.691 (AUCPR = 0.660) compared to 0.615 (0.603) of MBPred and 0.599 (0.587) of DELPHI. Additionally, we display two models in Supplementary Figs. S15 and S16 from the TestData and CompData datasets, respectively.

## 4. Conclusions

Here, we have introduced DeepTMInter, a sequence-based predictor of interaction sites in transmembrane proteins. Methodologically, we capitalized on the ResNet advances, supplemented by stacked generalization for further optimization. On an independent

dataset DeepTMInter achieved the AUC and AUCPR values of 0.698 and 0.598, respectively, significantly outperforming MBPred, previously developed by our group, as well as the most recent method, DELPHI. Among the eight major functional families of transmembrane proteins, the percentage of amino acids predicted to reside in interaction sites varies approximately between 10 and 30%, with ion channels exhibiting the largest proportion of sequence involved in interactions.

## CRediT authorship contribution statement

**Jianfeng Sun:** Investigation, Data curation, Conceptualization, Methodology, Formal analysis, Visualization, Software, Validation, Writing - original draft. **Dmitrij Frishman:** Supervision, Investigation, Data curation, Conceptualization, Methodology, Validation, Project administration, Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

We are grateful to Bo Zeng for his helpful advice on MBPred, Stephan Breimann for assistance in using the AAanalysis tool, and Peter Hönigschmid for constructive discussions. This work was supported by the China Scholarship Council (CSC) and the DFG grant FR1411/14-1.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.csbj.2021.03.005>.

## References

- [1] Cournia Z, Allen TW, Andricioaei I, Antony B, Baum D, Brannigan G, et al. Membrane protein structure, function, and dynamics: a perspective from experiments and theory. *J Membr Biol* 2015;248:611–40. <https://doi.org/10.1007/s00232-015-9802-0>.
- [2] Hedin LE, Illergård K, Elofsson A. An introduction to membrane proteins. *J Proteome Res* 2011;10:3324–31. <https://doi.org/10.1021/pr200145a>.
- [3] Babu M, Vlasblom J, Pu S, Guo X, Graham C, Bean BDM, et al. Interaction landscape of membrane-protein complexes in *Saccharomyces cerevisiae*. *Nature* 2012;489:585–9. <https://doi.org/10.1038/nature11354>.
- [4] Jones AM, Xuan Y, Xu M, Wang R-S, Ho C-H, Lalonde S, et al. Border control - a membrane-linked interactome of arabidopsis. *Science* 2014;344:711–6. <https://doi.org/10.1126/science.1251358>.
- [5] Oughtred R, Stark C, Breitkreutz B-J, Rust J, Boucher L, Chang C, et al. The BioGRID interaction database: 2019 update. *Nucleic Acids Res* 2018;47:D529–41. <https://doi.org/10.1093/nar/gky1079>.
- [6] Luck K, Kim D-K, Lambourne L, Spirohn K, Begg BE, Bian W, et al. A reference map of the human binary protein interactome. *Nature* 2020;580:402–8. <https://doi.org/10.1038/s41586-020-2188-x>.
- [7] Frishman D, Mewes HW. Protein structural classes in five complete genomes. *Nat Struct Biol* 1997;4:626–8. <https://doi.org/10.1038/nsh0897-626>.
- [8] Consortium TU. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res* 2019;47:D506–15. DOI:10.1093/nar/gky1049.
- [9] Zhang B, Li J, Quan L, Chen Y, Lü Q. Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network. *Neurocomputing* 2019;357:86–100. <https://doi.org/10.1016/j.neucom.2019.05.013>.
- [10] Li Y, Golding GB, Ilie L. DELPHI: accurate deep ensemble model for protein interaction sites prediction. *Bioinformatics* 2020. <https://doi.org/10.1093/bioinformatics/btaa750>.
- [11] Bordner AJ. Predicting protein-protein binding sites in membrane proteins. *BMC Bioinf* 2009;10:312. <https://doi.org/10.1186/1471-2105-10-312>.
- [12] Zeng B, Hönigschmid P, Frishman D. Residue co-evolution helps predict interaction sites in  $\alpha$ -helical membrane proteins. *J Struct Biol* 2019;206:156–69. <https://doi.org/10.1016/j.jsb.2019.02.009>.

- [13] Li Y, Wu F-X, Ngom A. A review on machine learning principles for multi-view biological data integration. *Brief Bioinform* 2016;19:325–40. <https://doi.org/10.1093/bib/bbw113>.
- [14] Min S, Lee B, Yoon S. Deep learning in bioinformatics. *Brief Bioinform* 2016;18:851–69. <https://doi.org/10.1093/bib/bbw068>.
- [15] He K, Zhang X, Ren S, Deep SJ. Deep residual learning for image recognition. *IEEE Conf. Comput. Vis. Pattern Recognit.* 2016;2016:770–8. <https://doi.org/10.1109/CVPR.2016.90>.
- [16] Hanson J, Paliwal K, Litfin T, Yang Y, Zhou Y. Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks. *Bioinformatics* 2018;35:2403–10. <https://doi.org/10.1093/bioinformatics/bty1006>.
- [17] Sun J, Frishman D. DeepHelicon: accurate prediction of inter-helical residue contacts in transmembrane proteins by residual neural networks. *J Struct Biol* 2020;212:.. <https://doi.org/10.1016/j.jsb.2020.107574>107574.
- [18] Kandathil SM, Greener JG, Jones DT. Prediction of interresidue contacts with DeepMetaPSICOV in CASP13. *Proteins Struct Funct Bioinforma* 2019;87:1092–9. <https://doi.org/10.1002/prot.25779>.
- [19] Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, et al. Improved protein structure prediction using potentials from deep learning. *Nature* 2020;577:706–10. <https://doi.org/10.1038/s41586-019-1923-7>.
- [20] Cao Y, Geddes TA, Yang JYH, Yang P. Ensemble deep learning in bioinformatics. *Nat Mach Intell* 2020;2:500–8. <https://doi.org/10.1038/s42256-020-0217-y>.
- [21] Wolpert DH. Stacked generalization. *Neural Networks* 1992;5:241–59. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1).
- [22] Hopf TA, Colwell LJ, Sheridan R, Rost B, Sander C, Marks DS. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell* 2012;149:1607–21. <https://doi.org/10.1016/j.cell.2012.04.012>.
- [23] Liu Y, Xu L, Li J, Yu J, Yu X. Attentional connectivity-based prediction of autism using heterogeneous rs-fMRI data from CC200 atlas. *Exp Neurobiol* 2020;29:27–37. <https://doi.org/10.5607/en.2020.29.1.27>.
- [24] Remmert M, Biegert A, Hauser A, Söding J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat Methods* 2012;9:173–5. <https://doi.org/10.1038/nmeth.1818>.
- [25] Kozma D, Simon I, Tusnády GE. PDBTM: Protein Data Bank of transmembrane proteins after 8 years. *Nucleic Acids Res* 2013;41:D524–9. <https://doi.org/10.1093/nar/gks1169>.
- [26] Tusnády GE, Dosztányi Z, Simon I. Transmembrane proteins in the Protein Data Bank: identification and classification. *Bioinformatics* 2004;20:2964–72. <https://doi.org/10.1093/bioinformatics/bth340>.
- [27] Tusnády GE, Dosztányi Z, Simon I. TMDET: web server for detecting transmembrane regions of proteins by using their 3D coordinates. *Bioinformatics* 2004;21:1276–7. <https://doi.org/10.1093/bioinformatics/bti121>.
- [28] Zou Q, Lin G, Jiang X, Liu X, Zeng X. Sequence clustering in bioinformatics: an empirical study. *Brief Bioinform* 2018;21:1–10. <https://doi.org/10.1093/bib/bby090>.
- [29] Hanson J, Paliwal K, Litfin T, Yang Y, Zhou Y. Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics* 2018;34:4039–45. <https://doi.org/10.1093/bioinformatics/bty481>.
- [30] Heffernan R, Paliwal K, Lyons J, Dehzangi A, Sharma A, Wang J, et al. Improving prediction of secondary structure, local backbone angles and solvent accessible surface area of proteins by iterative deep learning. *Sci Rep* 2015;5:11476. <https://doi.org/10.1038/srep11476>.
- [31] Hamp T, Rost B. Alternative protein-protein interfaces are frequent exceptions. *PLOS Comput Biol* 2012;8:e1002623.
- [32] Käll L, Krogh A, Sonnhammer ELL. Advantages of combined transmembrane topology and signal peptide prediction—the Phobius web server. *Nucleic Acids Res* 2007;35:W429–32. <https://doi.org/10.1093/nar/gkm256>.
- [33] Adhikari B. DEEPCON: protein contact prediction using dilated convolutional neural networks with dropout. *Bioinformatics* 2019;36:470–7. <https://doi.org/10.1093/bioinformatics/btz593>.
- [34] Li Y, Hu J, Zhang C, Yu D-J, Zhang Y. ResPRE: high-accuracy protein contact prediction by coupling precision matrix with deep residual neural networks. *Bioinformatics* 2019;35:4647–55. <https://doi.org/10.1093/bioinformatics/btz291>.
- [35] Fukuda H, Tomii K. DeepECA: an end-to-end learning framework for protein contact prediction from a multiple sequence alignment. *BMC Bioinform* 2020;21:10. <https://doi.org/10.1186/s12859-019-3190-x>.
- [36] Seemayer S, Gruber M, Söding J. CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics* 2014;30:3128–30. <https://doi.org/10.1093/bioinformatics/btu500>.
- [37] Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M. AAindex: amino acid index database, progress report 2008. *Nucleic Acids Res* 2007;36:D202–5. <https://doi.org/10.1093/nar/gkm998>.
- [38] Stein RR, Marks DS, Sander C. Inferring pairwise interactions from biological data using maximum-entropy probability models. *PLOS Comput Biol* 2015;11:e1004182.
- [39] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.* – Vol. 37, JMLR.org; 2015, p. 448–456.
- [40] Jones DT, Kandathil SM. High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. *Bioinformatics* 2018;34:3308–15. <https://doi.org/10.1093/bioinformatics/bty341>.
- [41] Wu Q, Peng Z, Anishchenko I, Cong Q, Baker D, Yang J. Protein contact prediction using metagenome sequence data and residual neural networks. *Bioinformatics* 2019;36:41–8. <https://doi.org/10.1093/bioinformatics/btz477>.
- [42] Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. In *ICML Work. Deep Learn. Audio, Speech Lang. Process.*, vol. 30, 2013, p. 3.
- [43] He K, Zhang X, Ren S, Sun J. Identity Mappings in Deep Residual Networks BT - *Computer Vision – ECCV 2016*. In: Leibe B, Matas J, Sebe N, Welling M, editors., Cham: Springer International Publishing; 2016, p. 630–45.
- [44] Baldassi C, Malatesta EM, Zecchina R. Properties of the geometry of solutions and capacity of multilayer neural networks with rectified linear unit activations. *Phys Rev Lett* 2019;123:.. <https://doi.org/10.1103/PhysRevLett.123.170602>170602.
- [45] Lin M, Chen Q, Yan S. Network In Network. *CoRR* 2014;abs/1312.4.
- [46] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* 2015;abs/1409.1.
- [47] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM* 2017;60:84–90. <https://doi.org/10.1145/3065386>.
- [48] Huang G, Liu Z, Van Der ML, Weinberger KQ. Densely connected convolutional networks. In: *2017 IEEE Conf Comput Vis Pattern Recognit.* p. 2261–9. <https://doi.org/10.1109/CVPR.2017.243>.
- [49] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization 2015.
- [50] de Boer P-T, Kroese DP, Mannor S, Rubinstein RY. A tutorial on the cross-entropy method. *Ann Oper Res* 2005;134:19–67. <https://doi.org/10.1007/s10479-005-5724-z>.
- [51] Prechelt L. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks* 1998;11:761–7. [https://doi.org/10.1016/S0893-6080\(98\)00010-0](https://doi.org/10.1016/S0893-6080(98)00010-0).
- [52] Amari S, Murata N, Müller K-, Finke M, Yang HH. Asymptotic statistical theory of overtraining and cross-validation. *IEEE Trans Neural Networks* 1997;8:985–96. <https://doi.org/10.1109/72.623200>.
- [53] Hawkins DM. The problem of overfitting. *J Chem Inf Comput Sci* 2004;44:1–12. <https://doi.org/10.1021/ci0342472>.
- [54] Tetko IV, Livingstone DJ, Luik AI. Neural network studies. 1. Comparison of overfitting and overtraining. *J Chem Inf Comput Sci* 1995;35:826–33. <https://doi.org/10.1021/ci00027a006>.
- [55] Tetko IV, Villa AEP. An enhancement of generalization ability in cascade correlation algorithm by avoidance of overfitting/overtraining problem. *Neural Process Lett* 1997;6:43–50. <https://doi.org/10.1023/A:1009610808553>.
- [56] Amari S, Murata N, Müller K-R, Finke M, Yang HH. Statistical Theory of Overtraining - Is Cross-Validation Asymptotically Effective? In: Touretzky DS, Mozer MC, Hasselmo ME, editors. *Adv. Neural Inf. Process. Syst.* 8, MIT Press; 1996, p. 176–82.
- [57] Anifwose F, Labadin J, Abdurraheem A. Improving the prediction of petroleum reservoir characterization with a stacked generalization ensemble model of support vector machines. *Appl Soft Comput* 2015;26:483–96. <https://doi.org/10.1016/j.asoc.2014.10.017>.
- [58] He L, Yang Z, Zhao Z, Lin H, Li Y. Extracting drug-drug interaction from the biomedical literature using a stacked generalization-based approach. *PLoS One* 2013;8:e65814.
- [59] Gardner MW, Dorling SR. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmos Environ* 1998;32:2627–36. [https://doi.org/10.1016/S1352-2310\(97\)00447-0](https://doi.org/10.1016/S1352-2310(97)00447-0).
- [60] Lou W, Wang X, Chen F, Chen Y, Jiang B, Zhang H. Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian Naïve Bayes. *PLoS One* 2014;9:e86703.
- [61] Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One* 2015;10:e0118432.
- [62] Yuan Y, Zhou QM, Li B, Cai H, Chow EJ, Armstrong GT. A threshold-free summary index of prediction accuracy for censored time to event data. *Stat Med* 2018;37:1671–81. <https://doi.org/10.1002/sim.7606>.
- [63] Boyd K, Eng KH, Page CD. Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals BT - *Machine Learning and Knowledge Discovery in Databases*. In: Blockeel H, Kersting K, Nijssen S, Železný F, editors., Berlin, Heidelberg: Springer Berlin Heidelberg; 2013, p. 451–66.
- [64] Armstrong JF, Faccenda E, Harding SD, Pawson AJ, Southan C, Sharman JL, et al. The IUPHAR/BPS Guide to PHARMACOLOGY in 2020: extending immunopharmacology content and introducing the IUPHAR/MMV Guide to MALARIA PHARMACOLOGY. *Nucleic Acids Res* 2020;48:D1006–21. <https://doi.org/10.1093/nar/gkz951>.
- [65] Alexander SPH, Mathie A, Peters JA, Veale EL, Striessnig J, Kelly E, et al. The concise guide to pharmacology 2019/20: ion channels. *Br J Pharmacol* 2019;176:S142–228. <https://doi.org/10.1111/bph.14749>.
- [66] Orchard S, Ammari M, Aranda B, Breuza L, Briganti L, Broackes-Carter F, et al. The MIntAct project—IntAct as a common curation platform for 11 molecular

- interaction databases. *Nucleic Acids Res* 2013;42:D358–63. <https://doi.org/10.1093/nar/gkt1115>.
- [67] Gilpin W. PyPDB: a Python API for the protein data bank. *Bioinformatics* 2015;32:159–60. <https://doi.org/10.1093/bioinformatics/btv543>.
- [68] Zhang C, Zheng W, Mortuza SM, Li Y, Zhang Y. DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics* 2019;36:2105–12. <https://doi.org/10.1093/bioinformatics/btz863>.
- [69] Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. vol. 13. 2017. DOI:10.1371/journal.pcbi.1005324.
- [70] Simkovic F, Thomas JMH, Rigden DJ. ConKit: a python interface to contact predictions. *Bioinformatics* 2017;33:2209–11. <https://doi.org/10.1093/bioinformatics/btx148>.
- [71] Mandal M, Mukhopadhyay A. Unsupervised Non-redundant Feature Selection: A Graph-Theoretic Approach BT - Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA). In: Satapathy SC, Udgata SK, Biswal BN, editors., Berlin, Heidelberg: Springer Berlin Heidelberg; 2013, p. 373–80.
- [72] Chormunge S, Jena S. Correlation based feature selection with clustering for high dimensional data. *J Electr Syst Inf Technol* 2018;5:542–9. <https://doi.org/10.1016/j.jesit.2017.06.004>.
- [73] Tan P-N, Steinbach M, Kumar V. *Introduction to data mining. First Edition. USA: Addison-Wesley Longman Publishing Co., Inc.; 2005.*
- [74] Fosso B, Pesole G, Rosselló F, Valiente G. Unbiased taxonomic annotation of metagenomic samples. *J Comput Biol* 2018;25:348–60. <https://doi.org/10.1089/cmb.2017.0144>.
- [75] Savojardo C, Manfredi M, Martelli PL, Casadio R. Solvent accessibility of residues undergoing pathogenic variations in humans: from protein structures to protein sequences. *Front Mol Biosci* 2021;7:460. <https://doi.org/10.3389/fmolb.2020.626363>.
- [76] Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 1983;22:2577–637. <https://doi.org/10.1002/bip.360221211>.
- [77] Serafimova K, Mihaylov I, Vassilev D, Avdjieva I, Zielenkiewicz P, Kaczanowski S. Using Machine Learning in Accuracy Assessment of Knowledge-Based Energy and Frequency Base Likelihood in Protein Structures. In: Krzhizhanovskaya V V, Závodszy G, Lees MH, Dongarra JJ, Sloot PMA, Brissos S, et al., editors. *Comput. Sci. – ICCS 2020*, Cham: Springer International Publishing; 2020, p. 572–84.
- [78] Waskom M, the seaborn development team. *mwaskom/seaborn* 2020. 10.5281/zenodo.592845.
- [79] Stahl K, Schneider M, Brock O. EPSILON-CP: Using deep learning to combine information from multiple sources for protein contact prediction. *BMC Bioinf* 2017;18. <https://doi.org/10.1186/s12859-017-1713-x>.
- [80] Hönigschmid P, Frishman D. Accurate prediction of helix interactions and residue contacts in membrane proteins. *J Struct Biol* 2016;194:112–23. <https://doi.org/10.1016/j.jsb.2016.02.005>.
- [81] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, New York, NY, USA: Association for Computing Machinery; 2016, p. 785–794. DOI:10.1145/2939672.2939785.
- [82] Schelling M, Hopf TA, Rost B. Evolutionary couplings and sequence variation effect predict protein binding sites. *Proteins Struct Funct Bioinf* 2018;86:1064–74. <https://doi.org/10.1002/prot.25585>.
- [83] Fuchs A, Kirschner A, Frishman D. Prediction of helix–helix contacts and interacting helices in polytopic membrane proteins using neural networks. *Proteins Struct Funct Bioinf* 2009;74:857–71. <https://doi.org/10.1002/prot.22194>.
- [84] Lynch BA, Koshland DE. Disulfide cross-linking studies of the transmembrane regions of the aspartate sensory receptor of *Escherichia coli*. *Proc Natl Acad Sci* 1991;88:10402 LP – 10406. DOI:10.1073/pnas.88.23.10402.
- [85] Riek PR, Handschumacher MD, Sung S-S, Tan M, Glynias MJ, Schluchter MD, et al. Evolutionary conservation of both the hydrophilic and hydrophobic nature of transmembrane residues. *J Theor Biol* 1995;172:245–58. <https://doi.org/10.1006/jtbi.1995.0021>.
- [86] Liu X, Wu G, Yu Y, Chen X, Ji R, Lu J, et al. Molecular understanding of calcium permeation through the open Orai channel. *PLoS Biol* 2019;17:e3000096.
- [87] Molica F, Figueroa XF, Kwak BR, Isakson BE, Gibbins JM. Connexins and Pannexins in vascular function and disease. *Int J Mol Sci* 2018;19. <https://doi.org/10.3390/ijms19061663>.
- [88] Ishikawa H, Barber GN. STING is an endoplasmic reticulum adaptor that facilitates innate immune signalling. *Nature* 2008;455:674–8. <https://doi.org/10.1038/nature07317>.
- [89] Santiago J, Dupeux F, Round A, Antoni R, Park S-Y, Jamin M, et al. The abscisic acid receptor PYR1 in complex with abscisic acid. *Nature* 2009;462:665–8. <https://doi.org/10.1038/nature08591>.
- [90] Aydar E, Palmer CP, Klyachko VA, Jackson MB. The sigma receptor as a ligand-regulated auxiliary potassium channel subunit. *Neuron* 2002;34:399–410. [https://doi.org/10.1016/S0896-6273\(02\)00677-3](https://doi.org/10.1016/S0896-6273(02)00677-3).
- [91] Johnson ME, Hummer G. Interface-resolved network of protein-protein interactions. *PLoS Comput Biol* 2013;9:e1003065.
- [92] Veerman CC, Wilde AAM, Lodder EM. The cardiac sodium channel gene SCN5A and its gene product Nav1.5: role in physiology and pathophysiology. *Gene* 2015;573:177–87. <https://doi.org/10.1016/j.gene.2015.08.062>.
- [93] Iqbal SM, Lemmens-Gruber R. Phosphorylation of cardiac voltage-gated sodium channel: potential players with multiple dimensions. *Acta Physiol* 2019;225:. <https://doi.org/10.1111/apha.13210>e13210.
- [94] Remme CA. Cardiac Sodium Channel (Dys)Function and Inherited Arrhythmia Syndromes. In: Thomas D, Remme CA, editors. *Channelopathies Hear. Dis.*, Cham: Springer International Publishing; 2018, p. 9–45. DOI:10.1007/978-3-319-77812-9\_2.
- [95] Detta N, Frisso G, Salvatore F. The multi-faceted aspects of the complex cardiac Nav1.5 protein in membrane function and pathophysiology. *Biochim Biophys Acta - Proteins Proteomics* 2015;1854:1502–9. <https://doi.org/10.1016/j.bbapap.2015.07.009>.
- [96] Willis BC, Ponce-Balbuena D, Jalife J. Protein assemblies of sodium and inward rectifier potassium channels control cardiac excitability and arrhythmogenesis. *Am J Physiol Circ Physiol* 2015;308:H1463–73. <https://doi.org/10.1152/ajpheart.00176.2015>.
- [97] Shy D, Gillet L, Abriel H. Cardiac sodium channel Nav1.5 distribution in myocytes via interacting proteins: the multiple pool model. *Biochim Biophys Acta - Mol Cell Res* 2013;1833:886–94. <https://doi.org/10.1016/j.bbamcr.2012.10.026>.
- [98] Rook MB, Evers MM, Vos MA, Bierhuizen MFA. Biology of cardiac sodium channel Nav1.5 expression. *Cardiovasc Res* 2011;93:12–23. <https://doi.org/10.1093/cvr/cvr252>.
- [99] Abriel H. Cardiac sodium channel Nav1.5 and interacting proteins: Physiology and pathophysiology. *J Mol Cell Cardiol* 2010;48:2–11. <https://doi.org/10.1016/j.yjmcc.2009.08.025>.
- [100] Wang C, Chung BC, Yan H, Wang H-G, Lee S-Y, Pitt GS. Structural analyses of Ca<sup>2+</sup>/CaM interaction with Nav channel C-termini reveal mechanisms of calcium-dependent regulation. *Nat Commun* 2014;5:4896. <https://doi.org/10.1038/ncomms5896>.
- [101] Poon W-YL, Malik-Hall M, Wood JN, Okuse K. Identification of binding domains in the sodium channel Nav1.8 intracellular N-terminal region and annexin II light chain p11. *FEBS Lett* 2004;558:114–8. DOI:10.1016/S0014-5793(03)01512-6.
- [102] Matamoros M, Pérez-Hernández M, Guerrero-Serna G, Amorós I, Barana A, Núñez M, et al. Nav1.5 N-terminal domain binding to  $\alpha$ 1-syntrophin increases membrane density of human Kir2.1, Kir2.2 and Nav1.5 channels. *Cardiovasc Res* 2016;110:279–90. <https://doi.org/10.1093/cvr/cvw009>.
- [103] Wang Z, Vermij SH, Sottas V, Shestak A, Ross-Kaschitzka D, Zaklyazminskaya EV, et al. Calmodulin binds to the N-terminal domain of the cardiac sodium channel Nav1.5. *Channels* 2020;14:268–86. <https://doi.org/10.1080/19336950.2020.1805999>.
- [104] Cormier JW, Rivolta I, Tateyama M, Yang A-S, Kass RS. Secondary structure of the human cardiac Na<sup>+</sup> channel C terminus: evidence for a role of helical structures in modulation of channel inactivation. *J Biol Chem* 2002;277:9233–41. <https://doi.org/10.1074/jbc.M110204200>.
- [105] Chagot B, Chazin WJ. Solution NMR structure of apo-calmodulin in complex with the IQ motif of human cardiac sodium channel Nav1.5. *J Mol Biol* 2011;406:106–19. <https://doi.org/10.1016/j.jmb.2010.11.046>.
- [106] Gee SH, Madhavan R, Levinson SR, Caldwell JH, Sealock R, Froehner SC. Interaction of Muscle and Brain Sodium Channels with Multiple Members of the Syntrophin Family of Dystrophin-Associated Proteins. *J Neurosci* 1998;18:128 LP – 137. DOI:10.1523/JNEUROSCI.18-01-00128.1998.
- [107] Luo L, Ning F, Du Y, Song B, Yang D, Salvage SC, et al. Calcium-dependent Nedd4-2 upregulation mediates degradation of the cardiac sodium channel Nav1.5: implications for heart failure. *Acta Physiol* 2017;221:44–58. <https://doi.org/10.1111/apha.12872>.
- [108] Rougier J-S, van Bemmelen MX, Bruce MC, Jespersen T, Gavillet B, Apothéloz F, et al. Molecular determinants of voltage-gated sodium channel regulation by the Nedd4/Nedd4-like proteins. *Am J Physiol Physiol* 2005;288:C692–701. <https://doi.org/10.1152/ajpcell.00460.2004>.
- [109] Young KA, Caldwell JH. Modulation of skeletal and cardiac voltage-gated sodium channels by calmodulin. *J Physiol* 2005;565:349–70. <https://doi.org/10.1113/jphysiol.2004.081422>.
- [110] Choi W, Clemente N, Sun W, Du J, Lü W. The structures and gating mechanism of human calcium homeostasis modulator 2. *Nature* 2019;576:163–7. <https://doi.org/10.1038/s41586-019-1781-3>.