TECHNISCHE UNIVERSITÄT MÜNCHEN
Fachgebiet für Bioinformatik

Prediction of residue contacts and interaction sites in transmembrane proteins using deep learning

Jianfeng Sun

Vollständiger Abdruck der von der TUM School of Life Sciences der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Bernhard Küster

Prüfer der Dissertation:
1. Prof. Dr. Dmitrij Frischmann
2. Prof. Dr. Burkhard Rost

Die Dissertation wurde am 12.11.2020 bei der Technischen Universität München eingereicht und durch die TUM School of Life Sciences am 25.01.2021 angenommen.

Abstract

Proteins play a pivotal role in a variety of biological processes. Their structures are stabilized by tons of invisible physical contacts between residues and most of their functions are performed by interacting to other proteins, ligands, or other macromolecules. Atomic-level three-dimensional (3D) protein structures allow a detailed analysis of protein functions and interaction mechanisms. Notwithstanding a large quantity of protein structures have still not been determined experimentally due to the costly and laborious nature inherent to experimental techniques. Computational prediction of residue contacts and interaction sites can cope with the intractable tasks. Transmembrane (TM) proteins across plasma membranes mediate a wide range of signaling activities between intracellular and extracellular environments, which makes them indispensable to cellular activities. Besides, TM protein targets have been found to be pharmaceutically instrumental for drug development. Currently, most computational methods are confined to globular proteins. In this thesis, we have sought to establish a comprehensive pipeline for constructing accurate contact maps and interaction site potentials of TM proteins by computationally detecting residue contacts at an intra-protein level and interaction sites at an inter-protein level, respectively.

One of the longest-standing challenges in structural biology is the accurate prediction of residue contacts that provide spatial distance constraints used subsequently for 3D modeling of protein structures. The centerpiece of accurate contact prediction lies in advanced algorithms and well-curated informative features. Undoubtedly, evolutionary coupling analysis (ECA) methods have been successfully applied to tackling the conundrum and capturing covariant residues that are highly-relevant to spatial proximities between them. Over the past decade, deep learning heralding the next generation of intelligent algorithms has achieved unrivaled successes across a broad spectrum of biological applications compared to traditional techniques. Very recently deep residual neural networks (ResNets) have enabled considerable progress in predicting secondary structures, residue contacts, and 3D protein structures. We have built on this emerging technique and the coevolutionary features to develop two novel deep-learning-based systems, DeepHelicon and DeepTMInter, for sequence-based prediction of residue contacts and interaction sites in TM proteins, respectively. Both systems have undergone systematic supervised-learning processes followed by performance refinement on the currently largest datasets of TM proteins at the <22% and <25% sequence identity levels, respectively. Using a dataset of 44 TM proteins, DeepHelicon outperformed two state-of-the-art techniques, DeepMetaPSICOV and Membrane2, in terms of the mean precision/recall/f1-score/MCC values of 77.84%/27.52%/38.82%/44.43%, 87.42%/12.70%/21.48%/31.99%, and 91.33%/6.58%/12.06%/23.60% in predicting the top $L/2$, $L/5$, and $L/10$ inter-helical contacts, respectively. These results have so far been the best records based on these TM proteins. We also found that DeepHelicon is well-suited for predicting contacts in those accommodating abundant helices. Our second method, DeepTMInter, presented a substantial improvement for interaction site prediction on a rigorously redundancy-reduced test dataset with the AUC/AUCPR values of 0.689/0.598 compared to 0.589/0.493 of a previously best performing method, MBpred.

Furthermore, we used DeepTMInter to systematically investigate the interaction network connectivity of human transmembrane proteome and found that the percentage of per-protein interaction sites is directly proportional to the number of human interaction partners. Our findings also show that among all functional families of human TM protein, the ion channels were identified to accommodate the largest number of interaction sites per protein. The resulting data are helpful for both academia and industry in aiding the follow-up analysis related to human transmembrane proteins, such as drug development.

Zusammenfassung

Proteine spielen eine zentrale Rolle in einer Vielzahl von biologischen Prozessen. Ihre Strukturen werden durch Tonnen unsichtbarer physikalischer Kontakte zwischen Rückständen stabilisiert und die meisten ihrer Funktionen werden durch Interaktion mit anderen Proteinen, Liganden oder anderen Makromolekülen durchgeführt. Dreidimensionale (3D) Proteinstrukturen auf Atomebene ermöglichen eine detaillierte Analyse von Proteinfunktionen und Interaktionsmechanismen. Ungeachtet einer großen Menge an Proteinstrukturen wurden aufgrund der kostspieligen und mühsamen Natur, die experimentellen Techniken innewohnt, noch nicht experimentell bestimmt. Die Berechnungsvorhersage von Rückstandskontakten und Interaktionsstellen kann die unlösbaren Aufgaben bewältigen. Transmembran-Proteine (TM) über Plasmamembranen hinweg vermitteln eine breite Palette von Signalaktivitäten zwischen intrazellulärer und extrazellulärer Umgebung, was sie für zelluläre Aktivitäten unentbehrlich macht. Außerdem, TM Protein-Ziele wurden gefunden, um pharmazeutisch instrumental für die Entwicklung von Medikamenten. Derzeit beschränken sich die meisten Rechenmethoden auf Kugelproteine. In dieser These haben wir versucht, eine umfassende Pipeline für die Erstellung genauer Kontaktkarten und Interaktionsstandortpotenziale von TM-Proteinen zu etablieren, indem wir rechenweise Rückstandskontakte auf intraproteinebener Ebene und Interaktionsstandorte auf Interproteinebene erkennen.

Eine der ältesten Herausforderungen in der Strukturbiologie ist die genaue Vorhersage von Rückstandskontakten, die räumliche Entfernungseinschränkungen liefern, die anschließend für die 3D-Modellierung von Proteinstrukturen verwendet werden. Das Herzstück der genauen Kontaktvorhersage liegt in fortschrittlichen Algorithmen und gut kuratierten informativen Funktionen. Zweifellos wurden methoden der evolutionären Kopplungsanalyse (ECA) erfolgreich eingesetzt, um das Problem zu bekämpfen und kovariante Rückstände zu erfassen, die für räumliche Probleme zwischen ihnen hochrelevant sind. In den letzten zehn Jahren hat Deep Learning, das die nächste Generation intelligenter Algorithmen ankündigt, im Vergleich zu herkömmlichen Techniken unübertroffene Erfolge in einem breiten Spektrum biologischer Anwendungen erzielt. In jüngster Zeit haben tiefe verbleibende neuronale Netzwerke (ResNets) erhebliche Fortschritte bei der Vorhersage von Sekundärstrukturen, Rückstandskontakten und 3D-Proteinstrukturen ermöglicht. Wir haben auf dieser aufkommenden Technik und den koevolutionären Merkmalen aufgebaut, um zwei neuartige Deep-Learning-basierte Systeme zu entwickeln, DeepHelicon und DeepTMInter, für die sequenzbasierte Vorhersage von Rückstandskontakten bzw. Interaktionsstellen in TM-Proteinen. Beide Systeme wurden systematisch überwachten Lernprozessen unterzogen, gefolgt von einer Leistungsverfeinerung der derzeit größten Datensätze von TM-Proteinen auf den <22% bzw. <25%-Sequenzidentitätsniveaus. Mit einem Datensatz von 44 TM-Proteinen übertraf DeepHelicon zwei hochmoderne Techniken, DeepMetaPSICOV und Membrane2, in Bezug auf die durchschnittlichen Genauigkeits-/Rückruf-/F1-Score/MCC-Werte von 77.84%/27.52%/38.82%/44.43%, 87.42%/12.70%/21.48%/31.99%, und 91.33%/6.58%/12.06%/23.60% bei der Vorhersage der oberen L/2, L/5 und L/10 interhelical Kontakte. Diese Ergebnisse waren bisher die besten Aufzeichnungen, die auf diesen TM-Proteinen basieren. Wir fanden auch heraus, dass DeepHelicon gut geeignet ist, kontakte bei den Kontakten zu prognostizieren, die reichlich Helices aufnehmen. Unsere zweite Methode, DeepTMInter, präsentierte eine wesentliche Verbesserung für die Vorhersage von Interaktions-Standorten auf einem streng redundanzreduzierten Testdatensatz mit den AUC/AUCPR-Werten von 0.689/0.598 im Vergleich zu 0.589/0.493 einer zuvor leistungsstärksten Methode, MBpred.

Darüber hinaus haben wir DeepTMInter verwendet, um die Interaktionsnetzwerkkonnektivität von humanem Transmembranproteom systematisch zu untersuchen, und festgestellt, dass der Prozentsatz der Pro-Protein-Interaktionsstellen direkt proportional zur Anzahl der menschlichen Interaktionspartner ist. Unsere Ergebnisse zeigen auch, dass unter allen funktionellen Familien des menschlichen TM-Proteins die Ionenkanäle identifiziert wurden, um die größte Anzahl von Interaktionsstellen pro Protein aufzunehmen. Die daraus resultierenden Daten sind sowohl für die Wissenschaft als auch für die Industrie hilfreich, wenn es um die Folgeanalyse im Zusammenhang mit menschlichen Transmembranproteinen wie der Arzneimittelentwicklung geht.

Acknowledgements

Over a span of the past three years for my Ph.D. studies, my research scope has apparently expanded and the concomitant abilities in science research have largely been enhanced. I spent an eventful period of my life on learning, working, and living in Germany, and I gained valuable life and research experience. I am very impressed with the generous help from many people here in Germany and from those outside Germany during my Ph.D. studies.

First and foremost, I would like to extend my deepest gratitude to my supervisor, Prof. Dr. Dmitrij Frishman, for giving me the opportunity to do Ph.D. studies and to work closely with him in Germany, and for providing invaluable guidance throughout this research. I would like to particularly thank his tailor-made teaching way to the ability of students so that I can fully stretch myself to do this highly interdisciplinary research work. He taught me how to present research work clearly and scientifically. I have greatly and deeply benefited from his encyclopedic knowledge of both bioinformatics and biology. I am more than appreciative of his encouragement that always stimulated me to further improvement. It was a great honor and privilege to work and study under his guidance.

I am very grateful to Martina Rüttger and Léonie Corry for their generous help in solving the obstacles in my daily life in Germany. I would like to extend my special thanks to Martina Rüttger. We always had interesting conversations where she gave me her advice and introduced her interesting stories and life experiences in Germany and America.

I am very thankful that Drazen Jalsovec is consistently patient to solve my technical problems on using multiple servers.

I would like to express my sincere thanks to Jinlong Ru and Bo Zeng for academic discussions. I would like to thank Peter Hönigschmid and Stephan Breimann for sharing opinions about bioinformatics. I am deeply grateful to Marina Parr for the interchanges of ideas and many of her gifts from Russia. My grateful thanks go to all other colleagues in the laboratory for working together: Hongen Xu, and Fei Qi, Hengyuan Liu, Wei-Yun Tsai, Jan Zaucha, Michael Kiening, Anja Mösch, and Evans Kataka. I would like to say thanks to all who gave me support to my Ph.D. studies directly or indirectly.

I am very grateful to Prof. Dr. Bernhard Küster who kindly agreed to be the chairman of examining committee for my doctoral dissertation, and to Prof. Dr.

Burkhard Rost who kindly agreed to be the second examiner of my doctoral dissertation.

Last but not least, I would like to express my deepest gratitude to my family members for their continuous love and support. I would like to sincerely thank Prof. Dr. Jianxin Wang, my master supervisor, who first brought me into contact with the academic area, fostered my academic interest, and enlightened me as to the nature of my first research work.

Publication list

The following two publications are part of this thesis.

1. Jianfeng Sun and Dmitrij Frishman. DeepHelicon: Accurate prediction of inter-helical residue contacts in transmembrane proteins by residual neural networks. *Journal of Structural Biology*, 212(1):107574, 2020.
2. Jianfeng Sun and Dmitrij Frishman. Improved sequence-based prediction of interaction sites in α-helical transmembrane proteins by deep learning. *Computational and Structural Biotechnology Journal*, 19:1512-1530, 2021.

Contents

List of Figures

List of Tables

List of Abbreviations

3D	three-dimensional
AP-MS	affinity purification/mass spectrometry
AUC	area under the receiver operating characteristic curve
AUCPR	precision-recall curve
BN	batch normalization
CASP	critical assessment of protein structure prediction
CNN	convolutional neural network
cryo-EM	ryogenic electron microscopy
Cyto	cytoplasmic
DCA	direct coupling analysis
DI	direct information
DL	deep learning
EC	evolutionary coupling
ECA	evolutionary coupling analysis
Extra	extracellular
FP	false positive
FN	false negative
GNB	Gaussian naive Bayes
GPCR	G-protein-coupled receptor
JSC	Jaccard similarity coefficient
LGIC	ligand-gated ion channel
MCC	Matthews correlation coefficient
MI	mutual information
ML	machine learning
MLP	multi-layer perceptron
MSA	multiple sequence alignment
NMR	nuclear magnetic resonance
NI	number of interacting amino acid residues
NIP	number of interaction partner
NNI	number of non-interacting amino acid residues
OPM	orientations of proteins in membranes
PPI	protein-protein interaction

ReLU	rectified linear unit
ResNet	residual neural network
ROC	receiver operating characteristic
RU	residual unit
SG	stacked generalization
SS	sequence separation
TM	transmembrane protein
TN	true negative
TP	true positive
VGIC	voltage-gated ion channel
Y2H	yeast two-hybrid

Chapter 1

Introduction

1.1 Membrane proteins

1.1.1 Protein topology

Most of membrane proteins are characterized by membrane-spanning α-helices (Lee, 2011; Miyazawa, Fujiyoshi, and Unwin, 2003) roughly orientated perpendicularly to the membrane plane (Heijne, 2006; Fuchs, Kirschner, and Frishman, 2009), *e.g.*, one α-helix in the transmembrane region of glycophorin (Lemmon et al., 1992). The helix-bundle membrane proteins can either cross the lipid bilayer or be interrupted within the membrane (Heijne, 2006), which are therefore classified into transmembrane proteins and peripheral membrane proteins (Pollard et al., 2016). It has been established that the former ones which are predominant in size (Zaucha et al., 2020) are exposed to the intracellular (*i.e.*, cytoplasmic) and extracellular surfaces of the membranes. Thus, the topology of transmembrane proteins encompasses the three kinds of regions (see an example shown in Fig. 1.1a): transmembrane regions (Adamian and Liang, 2001), cytoplasmic regions, and extracellular regions, which are all vital to life activities.

1.1.2 Membrane permeability mediated by transmembrane proteins

Transmembrane proteins play a crucial role in bridging the gap between the intracellular and extracellular environment, allowing entrances of chemical substrates and ions into cytoplasm of cells and organelles through plasma membranes, binding of ligands to intra- or extra-cellular domains for specialized biochemical reactions, and communication between cells (Hopf et al., 2012). Transmembrane proteins are perceived as natural barriers to allow membranes to be impermeable to external ions and macromolecules (Phillips et al., 2009). The impermeability of membranes ensures compartmentalization as necessity for cellular activities

(Zaucha et al., 2020). Transmembrane proteins that regulate the membrane impermeability can be divided into the three classes: pumps, carriers, and channels according to a broad spectrum of characteristics (see *chapter 14* in Pollard et al., 2016), *e.g.*, pump needs energy input while the other two do not need it.

1.1.3 Human transmembrane protein families

Transmembrane proteins in the human transmembrane proteome are widely found to be targeted by chemical compounds or small molecules that are pharmacologically and immunologically therapeutic to human diseases (Armstrong et al., 2020; Alexander et al., 2019; Sokolina et al., 2017), *e.g.*, schizophrenia (Moreno, Sealfon, and González-Maeso, 2009) and Parkinson disease (Gan-día et al., 2013). According to the Guide to PHARMACOLOGY (GtoPdb available at https://www.guidetopharmacology.org/) (Armstrong et al., 2020; Alexander et al., 2019), an expert-curated database of ligand-activity-target relationships, human transmembrane proteins are categorized into 8 major classes, namely, G-protein-coupled receptors, catalytic receptors, ligand-gated ion channels, voltage-gated ion channels, other ion channels, transporter, enzyme, and other protein targets. On the other hand, Almén's work (Almén et al., 2009) classified human transmembrane proteins into three major functional groups: receptors (63 sub-groups), transporters (89 sub-groups), and enzymes (7 sub-groups). Transporters construct intricate networks of carriers, pumps, and translocators (Saier Jr et al., 2016) to carry substrates or other molecules across the membrane by exploiting electrochemical gradients (Pollard et al., 2016). Enzymes are responsible for catalyzing biochemical reactions (Omelchenko et al., 2010). G-protein-coupled receptors, the largest family of transmembrane receptors, are crucial for mediating signal transduction pathways (Sokolina et al., 2017). Transmembrane protein ion channels perform biological functions, e.g., regulating electrical potential, by allowing ions to diffuse across cell membranes (Pollard et al., 2016).

1.1.4 Experimentally determined structures

It has long been clear that the full understanding of biological mechanisms of proteins relies on their known 3-dimensional (3D) structures at an atomic level (Baker and Sali, 2001). To date, most of 3D structures of proteins in structural biology are experimentally determined by three key methods, namely, X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and cryogenic electron microscopy (cryo-EM) (Wlodawer, Li, and Dauter, 2017). Since its invention, the X-ray crystallography method has been the most prevailing method to determine structures (Wang and Wang, 2017), *e.g.*, with a total of 141,566 entries (see

Fig. 1.1b) available in PDB in 2019. The first structure, myoglobin, was determined by using the X-ray technique in 1958 (Wlodawer, Li, and Dauter, 2017; Kendrew et al., 1958). Thanks to the high sensitivity to the nuances of local structural changes, NMR spectroscopy has emerged as a powerful tool for determination of proteins that bind to ligands, which is extremely useful for drug discovery (Geraets, Pothula, and Schröder, 2020). The downside of both methods is the low-precision determination of large proteins (Schmidt and Urlaub, 2017). Cryo-EM has become increasingly instrumental in determining macromolecular complexes (Hendrickson, 2016; Shoemaker and Ando, 2018), *e.g.*, membrane protein assemblies (Cheng, 2018). One typical case of large protein determination is that the transmembrane spike glycoprotein of SARS-CoV-2 was determined by cryo-EM (Walls et al., 2020). Despite how experimentally determined structures grow, the number of transmembrane protein structures is still small. Around 20-30% gene-encoding proteins in genomes are membrane proteins (Wallin and Heijne, 1998; Sharpe, Stevens, and Munro, 2010) whose structures only account for 2%-3% of the all experimentally-determined structures available in PDB (Xia et al., 2018).

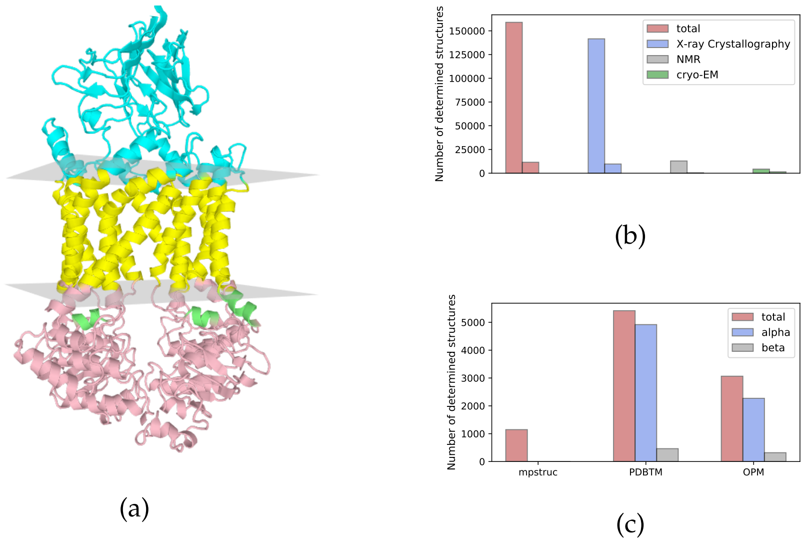


FIGURE 1.1: Experimentally determined structures. (a) shows an example of the 3D structure of pseudomonas aeruginosa PAO1 (PDB code: 5x5y) from Escherichia coli BL21(DE3). Portions in yellow represent transmembrane regions and portions in cyan and pink represent either intra- or extra-cellular regions. Membranes are bounded by grey planes. (b) shows the number of experimentally determined structures deposited in PDB by the end of 2019 by using X-ray crystallography, NMR spectroscopy, and cryo-EM techniques, respectively. Each technique includes two bars, the left one for the cumulative number and the right one for the annual number of determined structures. (c) shows the number of experimentally determined structures of α-helical, β-barrel, and all transmembrane proteins, by the end of 2019, curated in the *mpstruc* (α-helical and β-barrel data not shown), PDBTM, and OPM databases, respectively.

1.1.5 Databases of membrane protein structures

Membrane protein structures together with all others are comprehensively deposited in the protein data bank (PDB, `https://www.rcsb.org/`) (Goodsell et al., 2020). Detailed annotations of membrane protein structures are elucidated by the three widely-used databases (Shimizu et al., 2018, see Fig. 1.1c): *mpstruc* (`https://blanco.biomol.uci.edu/mpstruc/`) (White, 2009), PDBTM (Kozma, Simon, and Tusnady, 2012) (`http://pdbtm.enzim.hu/`), and orientations of proteins in membranes (OPM, `https://opm.phar.umich.edu/`) (Lomize et al., 2012). *mpstruc* is a well-curated database for annotations of membrane proteins in PDB but do not provide membrane protein topologies. PDBTM is a repository of only transmembrane protein structures by using the TMDET algorithm (Tusnády, Dosztányi, and Simon, 2004) to geometrically localize membrane planes and to distinguish transmembrane proteins from non-transmembrane proteins. OPM is an archive of membrane proteins whose orientation to the membrane is determined by minimizing a transfer energy function from water to the lipid bilayer. In addition, other subject-specific databases for membrane proteins also become available, *e.g.*, the Membranome (Lomize, Hage, and Pogozheva, 2018) (`https://membranome.org/`) database for single-spanning transmembrane proteins that are the most functionally diverse group of membrane proteins (Lomize et al., 2017).

1.2 Prediction of transmembrane protein structures and residue contacts

1.2.1 Retrieval tools for generating multiple sequence alignments

Multiple sequence alignments (MSAs) lay the foundation for structural biology and immensely impact on the accurate prediction of residue contacts. The quality of protein MSAs relies on retrieval tools and sequence databases. Two popularly used tools for retrieving homologous sequences are HHblits (Remmert et al., 2012) and JackHmmer (Johnson, Eddy, and Portugaly, 2010). Protein sequences databases against which HHblits retrieves the homologous sequences are often curated and regularly released by their own team. It also provides the way to customize a sequence database by users. In order to obtain reliable results, one often uses the alignments large enough for tackling structural biology problems (Söding, 2017). Recently, large alignments generated from metagenomics databases have been used for accurate modeling of a large number of protein families (Ovchinnikov et al., 2017).

1.2.2 Computational modeling of proteins

Although available protein sequence databases are growing exponentially (Finn et al., 2016), biological functions of proteins of unknown structures remain elusive in that only a far small number of proteins have been experimentally determined. Yet, despite the fast-paced growth of the number of protein sequences, efforts of accurate protein structure determination are still hampered by existing experimental technologies *e.g.*, time-consuming and costly nature through X-ray crystallography and NMR spectroscopy (Ding et al., 2013). To circumvent the problems, insight is therefore being given into devising computational algorithms to predict 3D protein structures from amino acid sequences (Dill and MacCallum, 2012). Spatial proximities at an inter-residue level are depicted by protein contact maps through which protein 3D models can be built by structure prediction programs, such as CONFOLD2 (Adhikari and Cheng, 2018) and Rosetta (Ovchinnikov et al., 2017), and by de novo modeling as exemplified in the three studies (Hopf et al., 2012; Hopf et al., 2017; Sjodt et al., 2018). Recently, the inter-helical residue contacts can also facilitate the structural modeling of TM proteins (Yang et al., 2016). Unsatisfactory prediction performance of inter-helical residue contacts in TM proteins has been the main impediment to the development of transmembrane protein modeling. Therefore, a direct motivation here is to predict inter-helical residue contacts using advanced computational approaches. In addition, distance-based prediction has recently started to show a powerful ability in protein modeling (Xu, 2019).

1.2.3 Dominant role of deep learning in contact prediction

Recent years have witnessed a prosperous development of machine- and deep-learning techniques in the area of information technology, led by using machines trained and optimized to perform extremely complicated tasks (LeCun, Bengio, and Hinton, 2015), such as image recognition (Krizhevsky, Sutskever, and Hinton, 2012) and automatic speech recognition (Hinton et al., 2012) as well as other artificial intelligent applications (Silver et al., 2017; He and Deng, 2017). In contrast to traditional machine learning techniques, deep learning approaches automatically learn representations from input data, which vastly reduces tedious and repetitive engineering work for feature extraction (LeCun, Bengio, and Hinton, 2015). The number of bioinformatics applications developed by deep learning has drastically increased since 2012 (Min, Lee, and Yoon, 2017) that tallies with the time when the deep-learning-based ImageNet achieved ideal performance in the ILSVRC-2012 image competition (Krizhevsky, Sutskever, and Hinton, 2012). ImageNet has attracted much attention since its inception. Nowadays, deep-learning techniques are expanding to different kinds of biological analyses driven by a considerable amount of multi-omics (*e.g.*, genomics, proteomics, and metabolomics) data (Li, Wu, and Ngom, 2018; Wainberg et al., 2018; Eraslan et al., 2019). Concomitantly, these learning techniques have led to a rapid growth of precise protein modeling; other structure-related prediction problems including the contact and distance prediction are also ongoing. With the fast development of deep learning, precision of predicting residue contacts has been substantially boosted by a dozen of methods (Wang et al., 2017; Kandathil, Greener, and Jones, 2019a; Ding et al., 2018), providing a promising perspective for how to incorporate deep learning algorithms into sequence-based contact prediction. In structural bioinformatics the quality of residue contact prediction is assessed by the critical assessment of protein structure prediction (CASP) organization (Ezkurdia et al., 2009; Monastyrskyy et al., 2016). The most recent two CASP competitions are briefly introduced below.

a. CASP12

One of the most striking successes in CASP12 is that among the best performing groups, the average precision of 47% on the $L/5$ long-range contacts in CASP12 nearly doubles the average precision of 27% in CASP11 (Schaarschmidt et al., 2018). In some cases with abundant homologous sequences the precision can even reach up to 100%. The average precision increment from CASP11 to CASP12 is only 6% between their respective best performing groups. In CASP12 ultra-deep residual neural networks (He et al., 2016a) are first applied into residue contact prediction.

b. CASP13

With the large improvement of predicting residue contacts in CASP12, the performance of predictors was systematically examined in CASP13 where we saw a quantum leap in average precision of almost up to 70% from 47% of CASP12 (Shrestha et al., 2019). Another silent feature of CASP13 is that deep learning approaches are widely used and are dominant in top performing groups. All best performing models, such as RaptorX (Xu and Wang, 2019), TripletRes (Li et al., 2019a), and DeepMetaPSICOV (Kandathil, Greener, and Jones, 2019a), were developed by using deep residual neural networks that significantly led in the CASP13 competition (Kandathil, Greener, and Jones, 2019b).

Looking deeper, approximately all off-the-shelf deep neural network models for residue contact prediction are generated using coevolutionary information. Several studies have investigated this issue and have partially revealed the central role of coevolutionary information in the strikingly successful performance. EPSILON-CP (Stahl, Schneider, and Brock, 2017) and MemConP (Hönigschmid and Frishman, 2016) have sought to quantify the contribution of coevolutionary information by using Gini impurity (GI, also known as mean decrease of impurity) (Louppe et al., 2013). In both studies, coevolutionary information has the highest GI scores that emphasize the importance of the features. Obviously, the coevolutionary information have geared the accurate prediction towards fast growth. Owing to the residue correlations inferred by coevolutionary information, accurate deep-learning predictors integrated with the coevolutionary information as features has undergone an unprecedented development and improvement (Wang and Wang, 2017). The underlying principle is that coevolutionary information agrees to physical contacts (or distances) between residues (see Figure 1.1 in (Marks et al., 2011)). Some statistical inference techniques generate correlation results carrying the coevolutionary information (Feinauer et al., 2014). As such, the more accurate the statistical inference techniques are, the more accurate the correlations between residues are. The correlations essentially reflect the physical contact strengths between residues. We will discuss the methods or the techniques that are used to yield the coevolutionary information in detail in the next section.

1.2.4 Residue contacts identified by evolutionary coupling analysis

Given 21 types of symbols $q_t, t = 1, 2, \ldots, L; L = 21$ (20 amino acids and 1 gap), a common way of measuring the correlation between any alignment column pair (i, j) is the mutual information (MI) (Lapedes et al., 1999):

$$MI_{ij} = \sum_{k=l=1}^{L} f_{ij}\left(q_k, q_l\right) \frac{f_{ij}\left(q_k, q_l\right)}{f_i\left(q_k\right) f_i\left(q_l\right)} \tag{1.1}$$

The result obtained by eq. (1.1) fails to describe an accurate coevolutionary strength. For example, if R_1R_2 and R_2R_3 are two contacting residue pairs, R_1R_3 is also thought of as being in contact because of the transitive correlation between R_1R_2 and R_2R_3 (Weigt et al., 2009; Morcos et al., 2011). However, R_1R_3 may not directly be in physical contact but using MI R_1R_3 is considered as indirectly contacting by the confounding factor (*i.e.*, transitivity by which MI is inherently limited) (Marks et al., 2011). One the other hand, MI is seen as a local method as it is calculated by considering only a column pair at one time, which ignores the influences of other alignment columns on it (Stein, Marks, and Sander, 2015). Apart from MI, other two local examples are Pearson's correlation (Stein, Marks, and Sander, 2015) and CRoSS (Thattai, Burak, and Shraiman, 2007; Weigt et al., 2009). Direct coupling analysis (DCA) methods obtained statistical inferences for residue contacts from maximum entropy modeling (a global approximation technique), which disentangles directly coupled residues from indirectly coupled ones. This method uses $p_{ij}^{DI}\left(q_k, q_l\right)$ to estimate the coevolutionary coupling strengths between alignment columns i and j, which satisfies the following condition:

$$p_{ij}^{DI}\left(q_k, q_l\right) = \sum_{\{q_t, t \neq i, j\}} P\left(q_1, q_2, \ldots, q_{Ls}\right) \tag{1.2}$$

where Ls is the length of the protein of interest. The right part in eq. (1.2) is a probability inferred from a global statistical model - maximum-entropy model. Direct information (DI) expounded by Morcos et al., 2011 (Morcos et al., 2011) substitutes $p_{ij}^{DI}(q_k, q_l)$ for $f_{ij}(q_k, q_l)$ in eq. (1.1), leading to

$$DI_{ij} = \sum_{k=l=1}^{L} p_{ij}^{DI}(q_k, q_l) \frac{p_{ij}^{DI}(q_k, q_l)}{f_i(q_k) f_i(q_l)} \tag{1.3}$$

DI_{ij} is able to eliminate the indirect coupling effects between any two columns of a MSA. Over a past decade, a collection of mathematical methods has been proposed to predict residue contacts using evolutionary coupling analysis (ECA) methods that have quantified proximities between residues and have extended to mutation detection (Figliuzzi et al., 2016; Hopf et al., 2017). ECA methods includes both the above DCA methods thereof and other closely related methods based on two clusters: *cluster i* - sparse inverse covariance estimation (SICE) solved by graphical LASSO methods (Friedman, Hastie, and Tibshirani, 2008; Loh and Wainwright, 2012) and *cluster ii* - pseudolikelihood maximization approaches (Ekeberg, Hartonen, and Aurell, 2014). We briefly review their representative methods as follows. For *cluster i*, PSICOV (Jones et al., 2012) could, to the best of our knowledge, emerge as the first method to refine residue contact detection by using graphical LASSO to tackle the SICE problem, followed by COUSCOus (Rawi et al., 2016); later, CoinDCA enhanced prediction performance by using a group graphical LASSO method at the expense of a high computational cost (Ma et al., 2015). plmDCA (Ekeberg et al., 2013) and Gremlin (Balakrishnan et al., 2011; Kamisetty, Ovchinnikov, and Baker, 2013) as well as CCMpred (Seemayer, Gruber, and Söding, 2014) are representative of *cluster ii*, showing more accurate prediction of residue contacts than those in *cluster i* (Ekeberg, Hartonen, and Aurell, 2014). Among the three methods, CCMpred vastly optimized the running time by taking advantage of graphics processing units (GPUs). In order to obviate phylogenetic and entropic bias (Lapedes et al., 1999), most of methods in *cluster i* and *cluster ii* obtained the contact likelihood S_{ij} between alignment i and j by adopting average-product correction (APC) (Dunn, Wahl, and Gloor, 2008), such that

$$S_{ij} = S_{ij}^{raw} - \frac{S_{i\cdot}^{raw} S_{\cdot j}^{raw}}{S_{\cdot\cdot}^{raw}} \tag{1.4}$$

where S_{ij}^{raw} represents the raw contact likelihood achieved directly by statistical inference in *cluster i* and *cluster ii*. $S_{i\cdot}^{raw}$is averaged over the values between column i (in the raw contact map) and all other columns. $S_{\cdot j}^{raw}$ is averaged over the values between column j and all other columns. $S_{\cdot\cdot}^{raw}$ is averaged over the values in the entire raw contact map. These above methods are, however, either too hard to be implemented in practice, or to be inefficient in dealing with large proteins. To allow a distinctly faster speed than the above methods, Gaussian DCA detected residue contacts using a multivariate Gaussian model (Baldassi et al., 2014). According to our test, it has so far been the method that has run at the fastest running speed when computational resources have been allocated to different ECA methods at the same level. Yet another fast running method is FreeContact (Kaján et al., 2014), an implementation of the EVfold method (Marks et al., 2011). More recently, pydca has provided a Python-based integrated platform for different ECA methods (Zerihun et al., 2020).

1.2.5 DeepHelicon

We assumed that neighbors of contacting residues positioned in TMHs might provide structurally contextual information for the contacting residues. This assumption was implemented/made based on the following two considerations. First, DeepHelicon learned inter-helical residue contact patterns from complete contact maps while training only inter-helical residue contacts forms incomplete contact maps. Previous methods for TM inter-helical residue contact prediction are solely trained on residue contacts in TMHs (Fuchs, Kirschner, and Frishman, 2009; Hönigschmid and Frishman, 2016; Yang et al., 2016). Accordingly, structure- or prediction-derived annotations of TMHs should be involved in initially training these methods. However, it would be virtually impossible for further rounds of re-training deep learning models based on patches (*i.e.*, squares around residue pairs of interest) constructed from complete contact maps. Second, deep learning methods including residual neural networks are methods to learn representations from input data in an automatic manner; these learned representations containing structurally contextual information might supply more supporting information for an accurate prediction of inter-helical residue contacts. MemBrain2.0 (Yang and Shen, 2018) developed recently also achieved high predictive performance by learning residue contacts outside TMHs, although it was not inferred exactly to what degree the learning process and the high performance were linked. During training at stage 1, we compared the performance of two types of models trained using inter-helical residue pairs and using residue pairs in all sequences, respectively. We found no silent differences (*i.e.*, comparable performance) between them at stage 1, and therefore we directly ignored the way of using only inter-helical residue pairs in the follow-up training. DeepHelicon made use of such abundant information of neighbors of contacting residues in two ways, that is, different inter-helical evolutionary coupling values (Marks et al., 2011) at stage 1 and patch learning at stage 2 (Sun and Frishman, 2020). Exclusively evolutionary features also characterize DeepHelicon as a unique method differently from other TM residue contact predictors. Evidently, the plmConv (Golkov et al., 2016) and DeepCov (Jones and Kandathil, 2018) methods have successfully applied coupling matrices (Ekeberg et al., 2013) as input features to predict residue-residue contacts. These two works mainly considered globular proteins as opposed to TM proteins; therefore, it was for the first time that couplings matrices were used in predicting inter-helical residue contacts in TM proteins. Considering variance errors and/or calculation bias, previous methods are given to using a combination of coevolutionary coupling values produced by different ECA methods (Jones et al., 2015; Yang and Shen, 2018). DeepHelicon used as input coevolutionary coupling values of four methods: EVfold (Marks et al., 2011), plmDCA (Ekeberg et al., 2013), CCMpred (Seemayer, Gruber, and Söding, 2014), and Gaussian DCA (Baldassi et al., 2014). Sliding windows for residue pairs are also leveraged in DeepHelicon in order to capture the flanking sequential information of residue pairs.

1.3 Transmembrane protein interactions

1.3.1 Experimental techniques

Protein-protein interactions (PPIs) are experimentally identified by two wid-ely-used methods, namely, yeast two-hybrid (Y2H) assays and affinity purification/mass spectrometry (AP-MS) (Keilhauer, Hein, and Mann, 2015; Zhang et al., 2015; Morris et al., 2014; Xing et al., 2016). Both methods are suitable for high throughput experiments to map human protein interactome of PPIs on a genome-wide study (Figeys, 2008). For example, the most up-to-date database published in 2020, HuRI, comprises more than 50,000 binary PPIs detected by using the Y2H assay (Luck et al., 2020). This method is perceived as an affordable and inexpensive technique as it requires a low resource consumption (Liu et al., 2020a). It detects the physical contacts occurring in the DNA-binding domain (DBD) and the activation domain (AD) located in the transcription factor in living yeast cells (Luban and Goff, 1995; Causier and Davies, 2002; Jessulat et al., 2011). One shortcoming of the Y2H assay is however the high false-positive or false-negative rates for binary PPI pairs (Jessulat et al., 2011; Liu et al., 2020a). In contrast, AP-MS identifies interactions in a biological complex formed by a tag-fused bait protein (*e.g.*, antibody) and its interaction partners, followed by mass spectrometry for rendering potential interaction partners (Jessulat et al., 2011; Yugandhar, Gupta, and Yu, 2019). Nevertheless, weak transient interactions are often imperceptible to AP-MS in that protein complexes undergo intricately and dynamically conformational changes (Nooren and Thornton, 2003; Luck et al., 2020). Additionally, PPIs can also be probed by using fluorescence resonance energy transfer (Margineanu et al., 2016), protein microarrays (Popescu et al., 2007), X-ray crystallography and NMR spectroscopy (Shoemaker and Panchenko, 2007) and etc.

1.3.2 PPI Databases

Human interactome map. A complete map of human PPI interactome facilitates the clear understanding of genotype-phenotype relationships in humans (Rolland et al., 2014; Menche et al., 2015). The first-generation interactome maps for humans were made and started in 2005 (Rual et al., 2005). Later, a complementary database, HI-II-14, reported 13,944 interactions using the Y2H assay and much expanded the size of known PPIs in humans. Very recently, an aggregate of 64,006 binary PPIs is determined by the union of the HI-union and Lit-BM databases (Luck et al., 2020). The core of the HI-union database is a collection of 52,569 high-quality expert-curated PPIs from a systematic mapping of human ORFome v9.1 (Luck et al., 2020). The Lit-BM database contains 13,441 PPIs obtained by collating literature. In addition to human PPI databases, the three most commonly used PPI databases, covering different species, are BioGRID (Stark et al., 2006; Oughtred et al., 2019), IntAct (Orchard et al., 2014), and STRING (Szklarczyk et al., 2019), with each presenting a thorough curation for both/either genetic and/or protein interactions collected from publicly available sources. These resources are rich in PPIs, with BioGRID (version: 3.5.188) containing 1,858,173 and IntAct (version: 4.1.25) containing 1,063,382 interactions.

1.3.3 Computational prediction of PPIs

Physical PPIs have implication for understanding the roles of proteins and mechanisms of many biological processes, such as signal transduction and chemical substrate translocation. Thus, the physical PPI detection is of importance. Notwithstanding much effort of localizing physical interactions, experimental techniques are inherently limited by a large overlap between PPIs detected by different sorts of high throughput experiments (Jessulat et al., 2011). Still, PPI interactome maps remain incomplete. Computational techniques have therefore been exploited for PPI prediction to map interactions on a proteome-wide scale. Current efforts of PPI prediction are made primarily using two ways, namely, PPI site prediction and binary PPI prediction. The strength of either a binary PPI or a PPI site is inferred from a probability ranging from 0 to 1. Compared to binary PPI prediction, PPI site prediction involves relatively more complicated processes, including interaction site definition, interaction site extraction, and imbalanced nature of interaction and non-interaction sites and etc. The vast majority of the two types of methods predict interaction sites by extracting sequence-based features, including amino acid composition, relative position, amino acid representation, and evolutionary information based on MSAs of sequences and etc. Publicly available databases have amassed huge amounts of information that can be pulled for annotations of proteins to be involved in interactions in pairs. Developed by machine learning or deep learning approaches, prediction performance has been gradually improved by *i*) binary PPI methods, including Profppikernel (Hamp and Rost, 2015), ProfPPIdb (Tran, Hamp, and Rost, 2018), DPPI (Hashemifar et al., 2018), MCDPPI (You et al., 2014), Wei's work (Wei et al., 2017); *ii*) site-based PPI methods, including PSIVER (Murakami and Mizuguchi, 2010), Chen's work (Chen et al., 2012), Hamp's work (Hamp and Rost, 2012), DLPred (Zhang et al., 2019), and DELPHI (Li and Ilie, 2020). However, a rather small number of prediction methods, such as Bordner's work (Bordner, 2009), MBPred (Zeng, Hönigschmid, and Frishman, 2019), and MPLs-Pred (Lu et al., 2019), have so far been made available and specialized for transmembrane proteins. In addition, it would also be possible that DCA methods were successfully extended into inter-protein contact prediction to reveal PPI network connection (Gueudré et al., 2016; Uguzzoni et al., 2017; Szurmant and Weigt, 2018; Cong et al., 2019).

1.3.4 DeepTMInter

Transmembrane proteins play an indispensable role in cellular activities. However, only a few methods have become accessible to the prediction of interaction sites in transmembrane proteins because the vast majority of available methods have been trained based on solely globular proteins or a hybrid set of globular and transmembrane proteins. This could impair the performance of predicting interaction sites in transmembrane proteins in that such methods lack a systematical learning and a sufficient training on transmembrane proteins. Apparently, there is an urgent need for a substantial performance improvement of methods specialized for transmembrane proteins. Therefore, we developed DeepTMInter upon which we further made a comprehensive investigation of high-quality transmembrane protein sequences collected from the PDBTM database. The resulting sequences sharing only less than 25% sequence identity to each other comprise the largest dataset for transmembrane protein interaction site prediction. More than 50% transmembrane proteins in human proteome are pharmaceutically drug targets but lack a thorough analysis of the number of their interaction sites. We particularly explored how per-protein interaction sites in human transmembrane proteome are distributed over 8 human families.

Chapter 2

DeepHelicon: accurate prediction of inter-helical residue contacts in transmembrane proteins by residual neural networks

Accurate prediction of amino acid residue contacts is an important prerequisite for generating high-quality 3D models of transmembrane (TM) proteins. While a large number of compositional, evolutionary, and structural properties of proteins can be used to train contact prediction methods, recent research suggests that coevolution between residues provides the strongest indication of their spatial proximity. We have developed a deep learning approach, DeepHelicon, to predict inter-helical residue contacts in TM proteins by considering only coevolutionary features. DeepHelicon comprises a two-stage supervised learning process by residual neural networks for a gradual refinement of contact maps, followed by variance reduction by an ensemble of models. We present a benchmark study of 12 contact predictors and conclude that DeepHelicon together with the two other state-of-the-art methods DeepMetaPSICOV and Membrain2 outperforms the 10 remaining algorithms on all datasets and at all settings. On a set of 44 TM proteins with an average length of 388 residues DeepHelicon achieves the best performance among all benchmarked methods in predicting the top $L/5$ and $L/2$ inter-helical contacts, with the mean precision of 87.42% and 77.84%, respectively. On a set of 57 relatively small TM proteins with an average length of 298 residues DeepHelicon ranks second best after DeepMetaPSICOV. DeepHelicon produces the most accurate predictions for large proteins with more than 10 transmembrane helices. Coevolutionary features alone allow to predict inter-helical residue contacts with an accuracy sufficient for generating acceptable 3D models for up to 30% of proteins using a fully automated modeling method such as CONFOLD2.

2.1 Introduction

Approximately every third protein in a living cell crosses a biological membrane (Frishman and Mewes, 1997), and most of the transmembrane (TM) proteins adopt an α-helical bundle fold. Functional studies on membrane proteins rely heavily on their 3D structures, but only about 2%−3% of all experimentally determined 3D structures are actually TM proteins (Xia et al., 2018). Over the past decade the paucity of atomic structures is being increasingly compensated for by a remarkable progress in computing 3D models of TM proteins from sequence alone (Hopf et al., 2012; Hayat et al., 2015). At the core of this methodological advance are significantly improved approaches for predicting amino acid contacts based on the evolutionary coupling analysis (ECA), which can be additionally combined with machine learning methods for an even better performance. The former group of approaches (reviewed in (Stein, Marks, and Sander, 2015)) deduces residue contacts based on coevolutionary information contained in multiple sequence alignments (MSAs) and includes algorithms such as EVfold (Marks et al., 2011), plmDCA (Ekeberg et al., 2013), and CCMpred (Seemayer, Gruber, and Söding, 2014). The latter group of approaches, which includes algorithms such as MetaPSICOV (Jones et al., 2015), R2C (Yang et al., 2016), and PconsC3 (Michel et al., 2017), predicts residue contacts by a supervised learning process aimed at distinguishing contacting from non-contacting residue pairs based on a specific feature set. These new generation methods exhibit superior performance compared to the earlier techniques that did not use any coevolutionary (Wu and Zhang, 2008; Li, Fang, and Fang, 2011) or even MSA-based (Tegge et al., 2009) features. Adoption of deep learning methods has led to a further improvement in contact prediction accuracy, as documented by the most recent blind prediction experiment CASP13 (Shrestha et al., 2019). In particular, methods employing residual neural networks (ResNets) achieve state-of-the-art results for soluble proteins (Wang et al., 2017; Li et al., 2019b; Kandathil, Greener, and Jones, 2019a). Compared to other deep learning methods, ResNets allow to increase the depth of neural networks while maintaining low training cost and a relatively fast training speed (He et al., 2016a).

Specialized contact prediction algorithms for TM proteins utilize a broad spectrum of compositional, sequence-based, structural, and coevolutionary features to characterize residue pairs (Fuchs, Kirschner, and Frishman, 2009; Yang et al., 2013; Hönigschmid and Frishman, 2016; Yang and Shen, 2018). High dimensional feature space increases the computational complexity and may make the supervised learning process difficult to optimize (Stahl, Schneider, and Brock, 2017). Some of these standard features, such as amino acids composition, may in fact be of little value for contact prediction (Stahl, Schneider, and Brock, 2017), while coevolutionary features have been shown to be highly informative (Golkov et al., 2016; Wang et al., 2017; Jones and Kandathil, 2018). Moreover, satisfactory 3D models can be derived for α-helical TM proteins by considering only inter-helical contacts (Yang et al., 2013; Yang and Shen, 2018; Ovchinnikov et al., 2015).

Here, we present a novel computational method, DeepHelicon, which employs a two-stage ResNet combined with residual units (RU) (He et al., 2016b) for sequence-based prediction of inter-helical residue contacts in α-helical TM proteins. The first-stage ResNet generates coarse-grained contact maps, which are further refined by the second-stage ResNet. The prediction performance is further enhanced by using an ensemble of models, which allows to reduce the variance errors of individual prediction models. DeepHelicon is the first contact predictor for TM proteins trained exclusively on coevolutionary features and our experiments suggest that these features alone are sufficient to accurately infer inter-helical residue contacts in TM proteins.

2.2 Materials and methods

2.2.1 Dataset

We obtained from the PDBTM database (Kozma, Simon, and Tusnady, 2012) 5606 protein chains corresponding to α-helical TM proteins with a resolution better than 3.5 and with the number of TM helices ranging from 2 to 17. This initial dataset was made non-redundant at the 23% sequence identity level using an in-house implementation of the greedy algorithm described in (Curtis, 2003). Note that cd-hit (Huang et al., 2010), the most commonly used tool for reducing sequence redundancy, only works with similarity thresholds greater than 40%. Additionally, we required that no two protein chains share a significant structural similarity by imposing a TM-score (Xu and Zhang, 2010) threshold of 0.4. The resulting full non-redundant dataset (further referred to as FULL), containing 222 protein chains was used to train our final predictor. In order to evaluate the predictor we split the FULL dataset into two unequal parts: *i*) training dataset, containing 165 chains (TRAIN), and *ii*) independent test dataset (TEST), containing 57 chains. In addition, we also tested our method on a combination of two previously published datasets – 21 and 30 chains used to test TMhhcp (Wang et al., 2011) and MemConP (Hönigschmid and Frishman, 2016), respectively. Upon removal of the 7 protein chains contained in our TEST dataset, the final combined dataset, called PREVIOUS, contains 44 α-helical TM protein chains. Detailed information about the TRAIN, TEST, and PREVIOUS datasets can be found in Tables A.1, A.2, A.3, and Fig. 2.1 respectively.

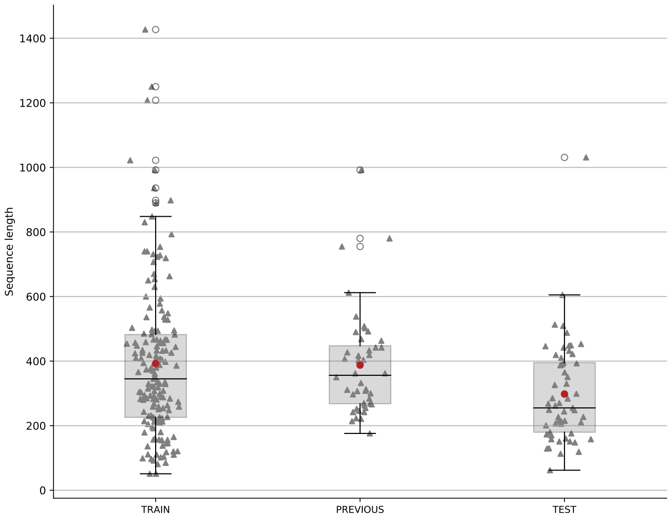


FIGURE 2.1: Sequence length distribution in the TRAIN, PREVIOUS, and TEST datasets (average values 392, 388, and 298, respectively). Each grey triangle represents a protein sequence in its respective dataset. Each red circle represents the average value of the sequence length in its respective dataset.

2.2.2 Transmembrane protein topology

Our method is designed to predict inter-helical contacts between residues located on different transmembrane helices. For comparing DeepHelicon with other methods, residues were labeled as being in transmembrane regions according to the PDBTM database. The publicly released DeepHelicon package relies on transmembrane helices predicted by the TMHMM2.0 algorithm (Krogh et al., 2001).

2.2.3 Definition of residue contacts

Following the previous work by us (Fuchs, Kirschner, and Frishman, 2009; Hönigschmid and Frishman, 2016) and others (Yang et al., 2013; Yang and Shen, 2018), we considered two residues to be in contact if the spatial distance between any pair of their heavy (non-hydrogen) atoms was below 5.5 and the sequence separation between the residues was no less than five positions.

2.2.4 Multiple sequence alignments

Multiple sequence alignments (MSA) were generated for all sequences in the TRAIN, TEST, and PREVIOUS datasets by running three iterations of HHblits searches

against the UniProt20 (version 2016) database. HHblits is an iterative protein sequence search tool to generate multiple sequence alignments (MSA) by profile hidden Markov models (Remmert et al., 2012). Uniprot20 (`https://github.com/soedinglab/hh-suite`) is a purpose-built da-tabase obtained by clustering of the UniProt database (Apweiler et al., 2004) at the 20-30% maximum pairwise sequence identity (Remmert et al., 2012). In order to generate as many multiple sequences as possible, we used the following recommended HHblits parameters (https://github.com/soe
dinglab/CCMpred/wiki/FAQ): maximum filter 100000, realign maximum hits 100000, maximum number of alignments in alignment list 100000, maximum number of lines in hit list 100000, and E-value cutoff 0.001. We also turned on the -all option used to obtain all sequences in the significantly similar UniProt20 clusters. As shown in Fig. 2.2, the average numbers of homologous sequences in the MSAs generated by the HHblits parameters described above are 32506, 32799, and 31317 for the TRAIN, PREVIOUS, and TEST datasets, respectively. We set an upper limit of 65000 on the number of sequences in the alignments in order to limit the CPU resources required for generating evolutionary features.

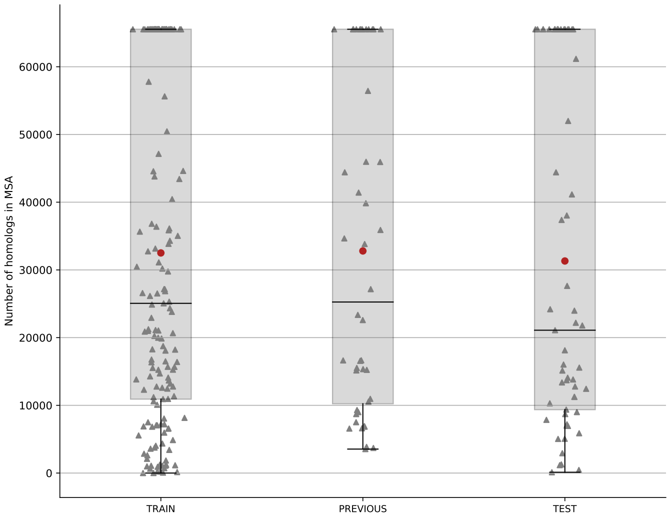


FIGURE 2.2: Number distribution of homologs in MSA in the TRAIN, PREVIOUS, and TEST datasets (average values 32506, 32799, and 31317, respectively). Each grey triangle represents a protein sequence in its respective dataset. Each red circle represents the average value of the number of homologs in MSA in its respective dataset.

2.2.5 Protein features

Recent studies by us (MemConP Hönigschmid and Frishman, 2016; MBpred Zeng, Hönigschmid, and Frishman, 2019) and others (EPSILON-CP, (Stahl, Schneider, and Brock, 2017)) indicate that many widely used sequence-derived features of proteins, such as amino acid composition, physico-chemical properties of amino acids, sequence separation between contacting residues, and evolutionary conservation, have very low importance in predicting both intra- and intermolecular interactions. By contrast, coevolutionary information ran-ked first in terms of feature importance in all three studies mentioned above. Given that excessively high-dimensional feature space creates the need for massive amounts of computing power and also the recent evidence that deep learning models for contact prediction remain well-trained after excluding unnecessary features (Stahl, Schneider, and Brock, 2017), we opted in this work to use only two most informative features: coupling matrix and coevolutionary score. These two features result in a feature space (741 dimensions, see below) comparable in size to the ones used in MetaPSICOV (Jones et al., 2015) (672-dimensional feature space in the first training stage and 731-dimensional feature space in the second training stage) and plmConv (Golkov et al., 2016) (441-dimensional feature space).

2.2.5.1 Evolutionary coupling values

Evolutionary coupling values reflect covariation between MSA columns i and j and can be inferred from a global maximal entropy model by the direct coupling analysis (Marks, Hopf, and Sander, 2012). It has been shown that combining evolutionary coupling methods based on different principles and trained on different datasets allows to achieve higher accuracy in contact prediction (Jones et al., 2015) and this strategy has been adopted by a number of recent prediction techniques (Michel et al., 2017; Stahl, Schneider, and Brock, 2017; Liu et al., 2018; Yang and Shen, 2018; Hanson et al., 2018). In this work we compute evolutionary coupling values by four established algorithms: *i*) EVfold (Marks et al., 2011), as implemented by the FreeContact software (Kaján et al., 2014), *ii*) plmDCA (Ekeberg et al., 2013), *iii*) CCMpred (Seemayer, Gruber, and Söding, 2014), and *iv*) Gaussian DCA (Baldassi et al., 2014).

Note that we are only interested in contacts between the amino acids located on different transmembrane helices and facing each other. Following our previous work (Hönigschmid and Frishman, 2016) for each pair of alignment positions (i, j) we actually consider the total of 25 evolutionary coupling values at positions $(i + x, j + y), (i + x, j - y), (i - x, j - y)$, and $(i - x, j + y)$ where $(x, y) \in \{(0, 0), (0, 1), (0, 3), (0, 4), (1, 0), (3, 0), (3, 4), (4, 0), (4, 3), (4, 4)\}$. These 25 values are computed by each of the four contact prediction methods mentioned above, resulting in a feature vector of length 100 (25 × 4). Furthermore, we applied a moving window of length 3 to each residue pair, so that the final size of the feature vector containing evolutionary coupling data is 300 (100 × 3).

2.2.5.2 Coupling matrix

In addition to computing evolutionary couplings, we follow the idea of Golkov et al. (Golkov et al., 2016). and utilize as features coupling matrices, which describe the co-constraint on the occurrence of 21 symbols (20 amino acids and one gap) in the alignment columns i and j (Hopf et al., 2017). Each element of these 21 × 21 matrices reflects the relative favorability of a specific pair of symbols. Each coupling matrix is actually represented by a vector of length 441 (21 × 21). For a protein sequence of length N, the total number of coupling matrices will be $\frac{N(N-1)}{2}$. We calculated the coupling matrices using the implementation of the pseudolikelihood maximization direct-coupling analysis available from `https://github.com/debbiemarkslab/plmc`.

2.2.6 Overview of the learning process

We developed a predictor for inter-helical residue contacts in TM proteins, which employs a two-stage learning process and an ensemble of models to gradually improve the prediction performance (Fig. 2.3). In the first stage, all residue contacts (*i.e.* the entire contact map) are predicted with a relatively low precision by a first-stage deep learning architecture described in section 2.2.6.1. The second-stage deep learning architecture involves the total of four iterations, as described in section 2.2.6.2. At the first three iterations complete contact maps are predicted with a progressively increasing accuracy due to recursive learning of discriminative features from refined contact maps of the previous iteration. At the forth iteration of the second stage, accurate predictions of inter-helical residue contacts are made. The final inter-helical residue contact predictions are obtained by an ensemble of models (see section 2.2.6.3). Both at the first stage and at all four iterations of the second stage we trained models on the TRAIN dataset and tested them (except for the iteration 4 of the second stage) on all residue contacts (contact maps) of the TRAIN dataset. To examine the performance of the inter-helical residue contact prediction, the models at the first stage and the models at all four iterations of the second stage as well as the ensemble of models were tested on the PREVIOUS and TEST datasets. Note that the final model was not trained at all; instead, it uses the average of the predictions generated by the iterations 2-4 of the second stage. The details on how our final model was built and trained are explained below.

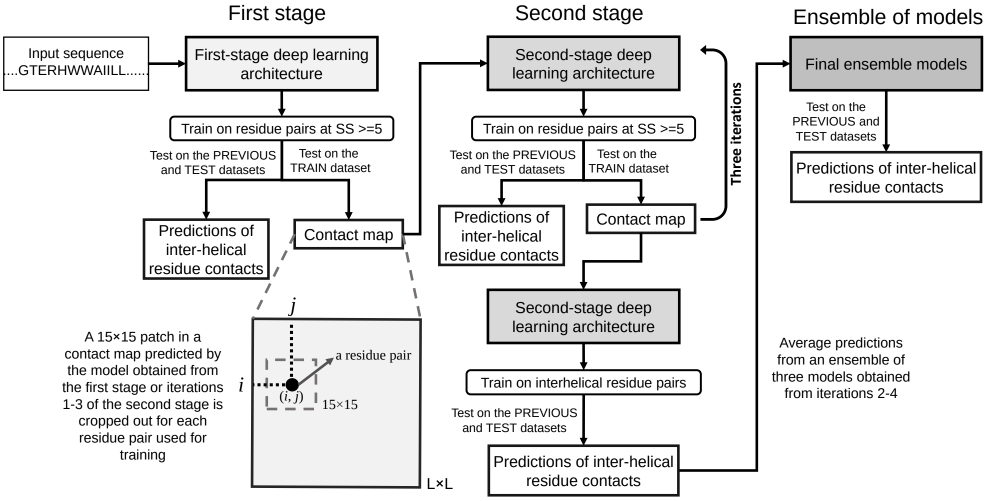

First stage
Second stage
Ensemble of models
Input sequence ...GTERHWWAIILL......
First-stage deep learning architecture
Train on residue pairs at SS >=5
Test on the PREVIOUS and TEST datasets
Test on the TRAIN dataset
Predictions of inter-helical residue contacts
Contact map
A 15×15 patch in a contact map predicted by the model obtained from the first stage or iterations 1-3 of the second stage is cropped out for each residue pair used for training
j
i
a residue pair
(i, j)
15×15
L×L
Second-stage deep learning architecture
Train on residue pairs at SS >=5
Test on the PREVIOUS and TEST datasets
Test on the TRAIN dataset
Predictions of inter-helical residue contacts
Contact map
Three iterations
Second-stage deep learning architecture
Train on interhelical residue pairs
Test on the PREVIOUS and TEST datasets
Predictions of inter-helical residue contacts
Final ensemble models
Test on the PREVIOUS and TEST datasets
Predictions of inter-helical residue contacts
Average predictions from an ensemble of three models obtained from iterations 2-4

FIGURE 2.3: Overview of the learning process for inter-helical residue contact prediction. SS: sequence separation.

2.2.6.1 The first stage of the learning process

At the first stage of the learning process we employed a residual neural network (ResNets) to predict residue contacts (i.e. contact map) for each TM protein with a comparatively low precision. Since the network relies on 2D convolutional layers, the 741-dimensional feature vector for each residue pair described above was converted to an almost square 26×28 matrix, ignoring a minor loss of data ($26 \times 28 = 728$ elements instead of 741 elements).

A new type of a residual unit (RU) described by (He et al., 2016b) was used, which facilitates optimization and allows to reduce overfitting (Fig. 2.4a). The RU consists of two identical groups of operations, each containing a 2D convolutional layer. Features are extracted by means of 16 filters with 3×3 elements each, which are moved both horizontally and vertically over the input data matrix by 1 position (*i.e.* stride=1). The convolution step thus results in 16 output matrices, each with the same dimension as the input matrix. Each convolutional layer is pre-activated by batch normalization (BN) (Ioffe and Szegedy, 2015) and a rectified linear unit (ReLU) (Nair and Hinton, 2010). BN is an effective way to speed up the training process and avoid overfitting by normalizing and rescaling its input data. ReLU sets all negative input values to zero, which results in a faster convergence of the training process and better performance (He et al., 2016a). As seen in Fig. 2.4a, the output of the second convolutional layer summed with the input data serves as the final output of the RU. This approach, referred to as residual mapping, allows to reduce learning error in deep network architectures (He et al., 2016a; He et al., 2016b).

The overall architecture of our ResNet is depicted in Fig. 2.4b. It contains the total of 30 convolutional layers and 1 dense layer and implements a hybrid approach, in which different kinds of deep models are combined to achieve better robustness and effectiveness of learning (see, *e.g.* (Hanson et al., 2018)). The core of the system is constituted by four blocks, each containing three RUs described above. Additional convolutional layers outside of the blocks are not bypassed by a residual connection. The first block is preceded by a batch-normalized convolutional layer employing filters with stride 1, while the other three blocks are preceded by batch-normalized convolutional layers employing filters with stride 2 and thus reducing the dimensionality of the input data (LeCun, Kavukcuoglu, and Farabet, 2010; Albawi, Mohammed, and Al-Zawi, 2017). Therefore, the input matrices of blocks 1, 2, 3, and 4 have the dimensions of 26×28, 13×14, 7×7, and 4×4, respectively. Such down-sampling reduces the computational load of the convolutional operation and makes the training process more robust with respect to minor variations in the input data. The last two convolutional layers constitute an additional 2-layer convolutional neural network (CNN). Subsequently, a dense (fully-connected) layer (He et al., 2016a) with 256 neurons transforms the result of the 2-layer CNN into two real values, whose magnitude reflects the likelihood of two amino acids residues to be in contact, which are then converted into the probabilities of two possible outcomes (contact/no contact) by a 2-way softmax activation function (He et al., 2016a).

2.2.6.2 The second stage of the learning process

At the second stage we implemented an iterative learning scheme (Heffernan et al., 2017) in which models are trained using the features extracted from contact maps. Parameter settings for the second-stage deep learning architecture were empirically determined. The ResNet shown in Fig. 2.4c uses the same RU as at the first stage (Fig. 2.4b), but the overall architecture is different. Each block contains 6 RUs and the number of neurons in the final dense layer is increased to 1024. We also found that increasing the number of filters in the convolutional layers to 64 leads to a better performance, which implies that the predictor is able to capture more informative features of contacting residue pairs.

The second-stage learning process was designed to take into account the structural context in which contacts occur. To this end we used as input for the first iteration 15×15 fragments of the contact map predicted at the first stage, centered around each residue pair *i* and *j*. For the subsequent iterations 2-4 the above operation was repeated, in that square patches centered around each residue pair were extracted from the contact map constructed at the preceding iterative training step (Fig. 2.4). Due to the application of the convolutional operation with the stride 2, at each iteration the dimension of the patches was reduced from the initial 15×15 to 8×8 and then to 4×4 (see Fig. 2.4c). Importantly, at iteration 4 we only extracted inter-helical residue pairs instead of the full contact map. Based on multiple in silico experiments we found that a further increase of the number of iterations at the second learning stage only led to a marginal gain in predictive performance.

2.2.6.3 Ensemble of models

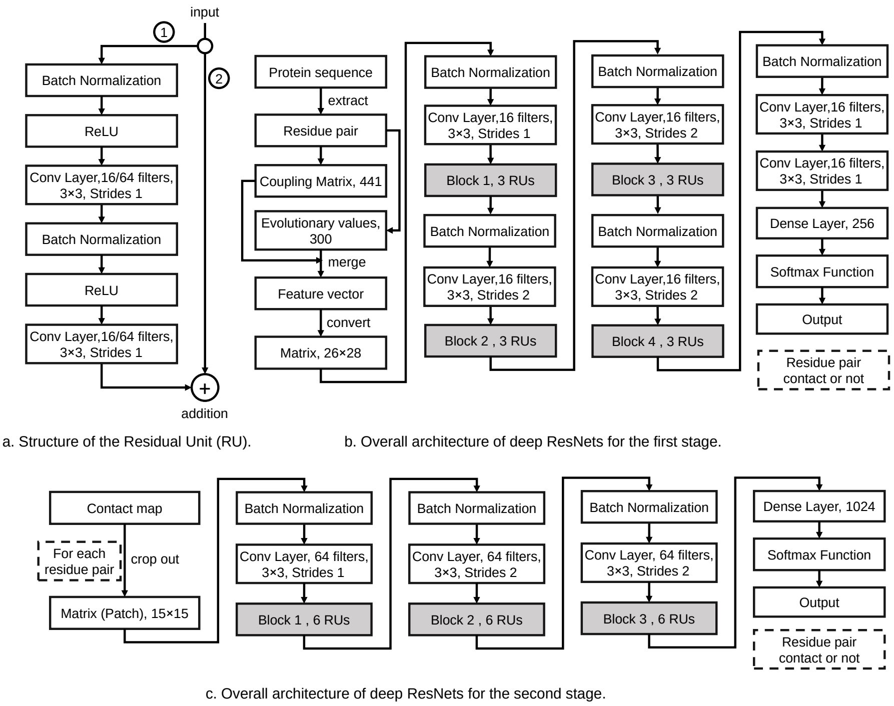
The final predictions of inter-helical contacts were obtained by averaging the output prediction values of models trained at the iterations 2-4 of stage 2 in order to reduce the variance error if each individual model (Naftaly, Intrator, and Horn, 1997; Keijzer and Babovic, 2000) (Fig. 2.3). Since the performance of the trained models is affected by a number of factors, such as parameter initialization and training times, they showed different predictive capacities on the training and testing datasets. For this reason, the prediction values of the better trained models at iterations 2-4 were given higher weights and thus contributed more strongly to the final ensemble prediction. Specifically, models obtained at iterations 3, 2, and 4 were given the weights 0.5, 0.4 and 0.1, respectively. Therefore, the prediction values p_e of the ensemble predictor were expressed as

$$p_e = \sum_{i=2}^{4} w_i \times p_i$$

where p_i represents the prediction values of the trained models at the iteration i of stage 2, and w_i represents the weights.

2.2.7 Training process

ResNets at both stages were trained using Adam, a computationally efficient variation of stochastic gradient-based descent method for problems with massive amounts of data and a large number of parameters (Kingma and Ba, 2014). The cross entropy objective function (Boer et al., 2005) was used for quantitatively measuring the difference between the actual labels ([0, 1] or [1, 0] for the presence or absence of a residue contact) and predicted labels, with predicted value learning rate and training batch size set to 0.001 and 100, respectively. The number of weight parameters to be learned was calculated as follows. In the first-stage deep learning architecture, the first convolutional layer takes as input 1



a. Structure of the Residual Unit (RU).

b. Overall architecture of deep ResNets for the first stage.

c. Overall architecture of deep ResNets for the second stage.

FIGURE 2.4: Prediction of inter-helical residue contacts in transmembrane proteins by deep learning. (a) shows the structure of the Residual Unit (RU) (He et al., 2016b). (b) and (c) show the overall architectures of the ResNets for the first stage and the second stage, respectively. See text for an explanation.

matrix and outputs 16 matrices, one for each 3×3 filter applied, which results in 144 (1×16×3×3) weight parameters. Each of the remaining 29 convolutional layers takes as input 16 matrices and outputs 16 matrices, which results in 2304 (16×16×3×3) weight parameters. The dense layer, which converts data from 256 input neurons to 2 output values for each residue pair, needs 512 (256×2) weight parameters. Thus, there are the total of 67472 (144+2304×29+512) weight parameters to be learned in the deep learning architecture at stage 1. Similarly, in the second-stage deep learning architecture the first convolutional layer takes as input 1 matrix and outputs 64 matrices, with the same-size filter as applied at stage 1, which results in 576 (1×64×3×3) weight parameters. Each of the subsequent 38 convolutional layers takes as input 64 matrices and outputs 64 matrices, leading to 1400832 (64×64×3×3×38) weight parameters in total. The dense layer involves 2048 (1024×2) weight parameters. As a result, the total of 1403456 weight parameters need to be learned at each of the 4 iterations of stage 2.

The first-stage deep learning architecture was trained on residue pairs with sequence separation (SS)≥5 (Fig. 2.3). At the second stage training was performed on residue pairs with SS≥5 at the first three iterations, while at the fourth iteration the deep learning architecture was trained on inter-helical residue pairs.

2.2.8 Assessment metrics

The prediction performance of our trained deep learning model (Bradley, 1997) was evaluated using the following measures:

$$precision = \frac{TP}{TP+FP}$$
$$recall = \frac{TP}{TP+FN}$$
$$F - score = \frac{(1+\beta^2)\times precision\times recall}{\beta^2\times precision+recall}$$
$$\text{Matthews Correlation Coefficient } (MCC) = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

where TP (true positive), FP (false positive), TN (true negative) and FN (false negative) are the number of contacting residue pairs predicted as contacting, the number of non-contacting residue pairs predicted as contacting, the number of non-contacting residue pairs predicted as non-contacting, and the number of contacting residue pairs predicted as non-contacting, respectively.

We evaluated the prediction performance of our method separately for inter-helical residue contacts. Correspondingly, precision was used to assess how many residue pairs are correctly predicted among the top L, $L/2$, and $L/5$ residue contact predictions, where L denote the cumulative length of concatenated transmembrane helices. Recall was calculated to quantify the percentage of correctly predicted residue contacts among all observed contacts in experimentally-determined structure. F-score is a weighted harmonic mean of precision and recall. A F_1-score is obtained by setting β to 1 in the F-score equation above, such that precision and recall are equally important. Given that our dataset is strongly imbalanced (approximately 1 to 100 ratio between the number of residue contacts and all possible residue pairs, see Table A.4), we additionally evaluate $F_{0.35}$ by setting β to 0.35 in the F-score equation in order to up-weight precision relative to recall. Note that MCC is not affected by the class imbalance problem.

2.2.9 Cross-validation of the predictor

We used the stratified-shuffle k-fold ($k = 5$) cross validation (Sharma et al., 2017) in order to ensure that proteins with different numbers of TM regions are distributed uniformly and randomly across all folds. To this end we split our full dataset (165 protein chains) into four TMH classes containing 64 (38.78%), 40 (24.24%), 27 (16.36%), and 34 (20.61%) proteins with 2-4, 5-7, 8-10, and over 10 transmembrane α-helices, respectively. As illustrated in Fig. 2.5, for each of the 5 folds the full dataset is subdivided into a training (132 chains) and a validation (33 chains) dataset. In each fold the training dataset (represented by grey rectangles in Fig. 2.5) consists of 51, 32, 22, and 27 protein chains obtained by randomly picking 80% of proteins from each TMH class described above. Similarly, the validation dataset in each fold (represented by purple rectangles in Fig. 2.5) contains 13, 8, 5, and 7 protein chains drawn from each TMH class. As a consequence, the training and validation datasets in each fold preserve approximately the same ratio between TMH classes as in the full dataset.

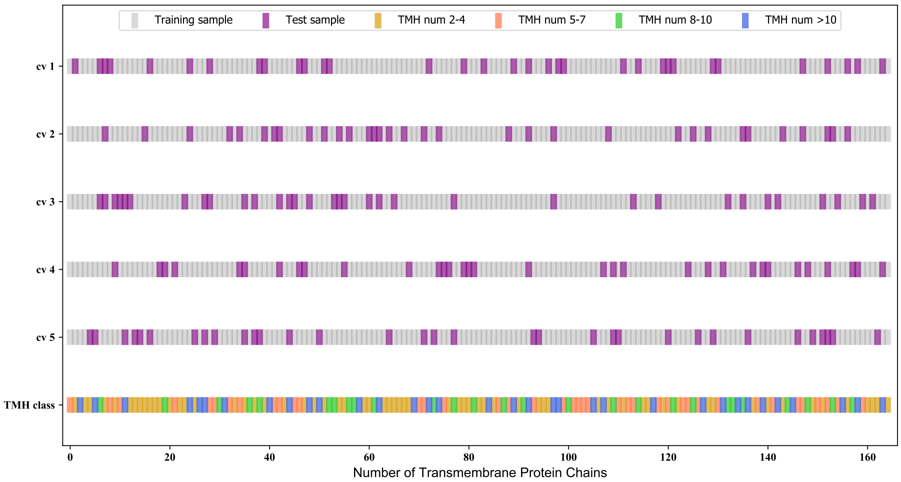


FIGURE 2.5: Visualization of 165 α-helical TM protein chains compartmentalized by stratified-shuffle 5-fold cross validation. Yellow, pink, green, or blue rectangles in the bottom plot (all 165 chains) represent protein chains with 2-4, 5-7, 8-10, or at least 11 TM helices, respectively. On all other plots grey rectangles (132 protein chains) or purple rectangles (33 protein chains) indicate whether these protein chains are chosen for training or validation in each fold, respectively.

2.2.10 Benchmarking predictor performance

We compared the performance of DeepHelicon with a collection of predictors employing ECA, machine learning (ML) or deep learning (DL) (Tables A.5 and A.6).

2.2.11 Protein structure prediction

We employed CONFOLD2 (Adhikari and Cheng, 2018) to conduct contact-driven 3D structure modelling of α-helical TM proteins. For each target protein of length N, CONFOLD2 returns top five spatial architectures assembled from the top N/x residue contacts and three-state secondary structure; the latter was predicted by SCRATCH1.0 (Magnan and Baldi, 2014). Results for each protein are reported based on the maximal TM-score and/or minimal Cα-RMSD (C-alpha atomic root mean square deviation) among the top five predicted structures compared to the native structure (`https://zhanglab.ccmb.med.umich.edu/TM-score/`).

2.3 Results and discussion

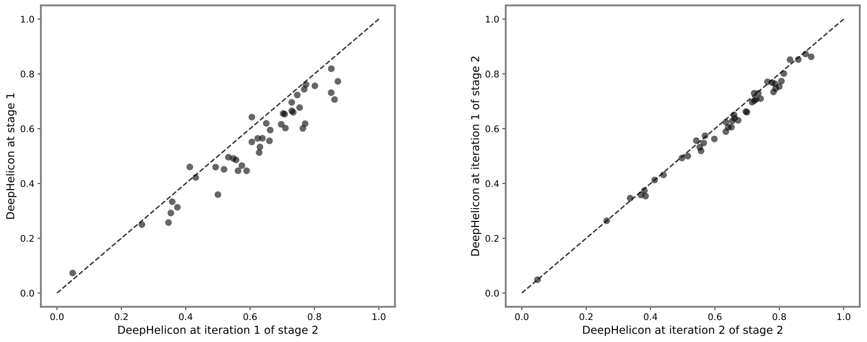
2.3.1 Performance assessment of DeepHelicon

As described in the section 2.2.6, we tested the performance of our models in predicting inter-helical contacts on the PREVIOUS and TEST datasets at the first stage (Table A.7) and at all four iterations of the second stage (Tables A.8 and A.9) (Figs. 2.6 and 2.7). The publicly released DeepHelicon package employs the ensemble of models, whose performance is summarized in Table A.10.

Stage 2 predictions (Tables A.8 and A.9) represent a significant improvement over stage 1 predictions (Fig. 2.6a, Table A.7 and Fig. 2.7a) in terms of precision. For example, for the top L predictions for the PREVIOUS dataset the precision increases from 55.81% at stage 1 to 61.35% at the first iteration of stage 2. These observations parallel the results obtained with the two-stage algorithms Membrain-contact 2.0 (Yang and Shen, 2018)(Yang and Shen, 2018) and MetaPSICOV (Jones et al., 2015), where deep refinement of the coarse-grained contact maps obtained in the first stage enhances the performance at the second stage. In addition, the gain in prediction performance by approximately 1% is due to using a larger number of filters in each layer of the second-stage architecture (Fig. 2.4).

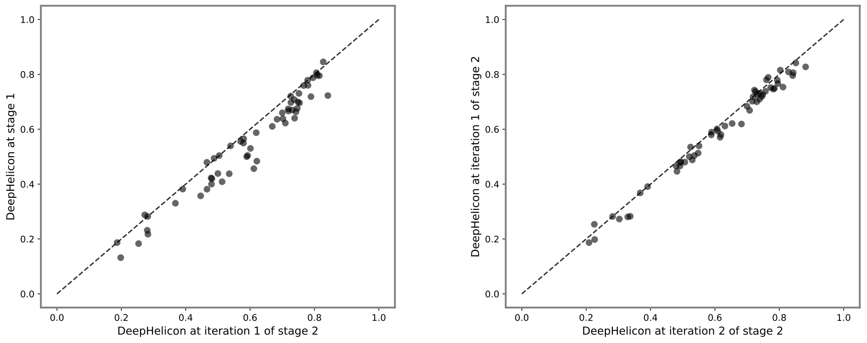
On the PREVIOUS dataset at stage 2 (Figs. 2.6b and 2.7b, and Table A.8), the precision at L raises sharply from 61.35% to 63.17% between the first and the second iteration, but then stagnates at the third and the fourth iteration (63.12% and 63.71%, respectively). At $L/10$ the precision actually drops off gradually over the four iterations (90.07%, 90.09%, 89.46%, 88.73%), while at $L/5$ there is no clear trend in the variation of precision (85.72%, 86.93%, 85.61%, 85.93%). A similar behavior was observed on the TEST dataset. These findings imply that increasing the number of iterations does not as such lead to further improvement of the prediction accuracy at the second stage. However, we found that combining the contact maps obtained at the last three iterations of stage 2 is beneficial for the ensemble models (Fig. 2.3, see section 2.2.6.3). For example, if only the first two iterations are used, the final results deteriorate by 1-2%. Based on a number of experiments we settled for the number of iterations of four.

Compared to the individual models at stage 2, the ensemble of models (Table A.10) achieves a slightly better prediction performance by most of evaluation metrics. In particular, on the PREVIOUS dataset at $L/10$ the precision increases from 90.09% (the best result among all iterations) to 91.33%, and on the TEST dataset at $L/2$ the precision increases from 75.14% to 76.16%.



(a) (b)

FIGURE 2.6: Precision of DeepHelicon for each protein in the PREVIOUS dataset. (a) Precision at stage 1 compared to iteration 1 of stage 2. (b) Precision at iteration 2 of stage 2 compared to iteration 1 of stage 2.



(a) (b)

FIGURE 2.7: Precision of DeepHelicon for each protein in the TEST dataset. (a) Precision at stage 1 compared to iteration 1 of stage 2. (b) Precision at iteration 2 of stage 2 compared to iteration 1 of stage 2.

2.3.2 Comparison of DeepHelicon with other contact prediction methods

We compared the inter-helical residue contact prediction performance of DeepHelicon with 12 publicly available methods listed in Table A.5 (Fig. 2.8). On the PREVIOUS dataset (Table 2.1) DeepHelicon outperforms all other predictors by all assessment metrics except for the performance for the top L contacts, which lags behind the DeepMetaPSICOV, one of the best predictors in CASP 13 (Kandathil, Greener, and Jones, 2019a). For instance, our method achieves the precision of 77.84%, 87.42%, and 91.33% at $L/2$, $L/5$, and $L/10$, respectively, compared to 77.60%, 86.24% and 88.28% for the next best method, DeepMetaPSICOV. DeepHelicon's recall values (27.52%, 12.70%, and 6.58%) at $L/2$, $L/5$, and $L/10$ are also higher than those of DeepMetaPSICOV (26.72%, 12.12%, and 6.03%). DeepHelicon is also superior to other methods in terms of the F_1, $F_{0.35}$, and MCC measures. For example, its F1 and MCC values are 49.47% and 50.41% at L compared to 44.51% and 45.05% of Membrain2.

On the TEST dataset (Table 2.2), DeepMetaPSICOV outperforms all other predictors (precision of 67.76%, 80.76%, 87.63%, and 90.39% at L, $L/2$, $L/5$, and $L/10$). DeepHelicon is second best at L and $L/2$ (62.13% and 76.16%), better than Membrain2 (58.75% and 74.46%), while at $L/5$ and $L/10$ Membrain2 achieves a higher precision (86.64% and 91.27%) than DeepHelicon (84.98% and 87.44%).

Overall, DeepHelicon, Membrain2, and DeepMetaPSICOV outperform all other predictors on both datasets and at all settings. The common property of these three predictors is that they exploit a combination of four, five, and three different ECA methods as input features for training, respectively. DeepMetaPSICOV and DeepHelicon also leverage evolutionary information - covariance matrices and coupling matrices. Both Membrain2 and DeepHelicon employ a multiple-stage training process to construct their models.

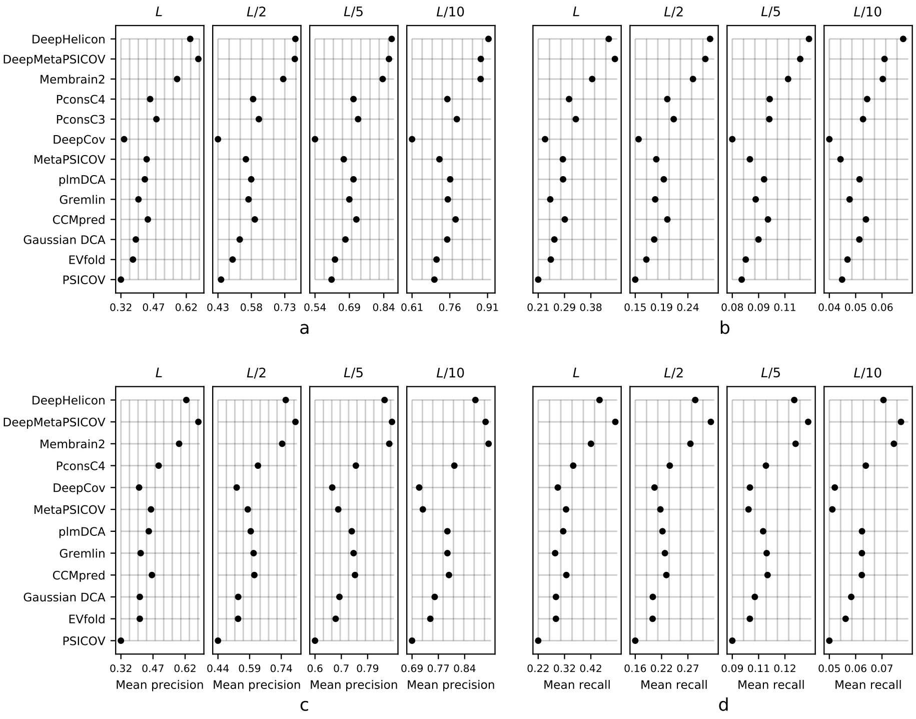


FIGURE 2.8: Prediction performance of DeepHelicon and other methods on inter-helical residue contacts. (a) and (b) show the mean precision and recall, respectively, on the PREVIOUS dataset, while (c) and (d) show the mean precision and recall on the TEST dataset, respectively.

Among the six ECA-based predictors, CCMpred achieves the best performance on both datasets. As seen in Fig. 2.8, Gremlin performs slightly better than plmDCA at $L/2$ and $L/5$ on the TEST dataset with precision (60.93% vs. 59.57% and 73.90% *vs.* 73.26%) while lagging marginally behind plmDCA at all thresholds on the PREVIOUS dataset. In terms of precision and recall, Gaussian DCA, EVfold, and PSICOV are ranked as the last three of all ECA-based predictors. Unexpectedly, we found that DeepCov's performance (precision 39.98%) was comparable to Gaussian DCA (precision 40.37%) for the top L inter-helical contact predictions on the TEST dataset. Interestingly, CCMpred exhibits an even better prediction performance for inter-helical residue contacts than some machine learning and deep learning techniques. For example, for the top $L/5$ inter-helical contact predictions it achieves the precision of 72.02% and 74.41% on the PREVIOUS and TEST datasets, respectively, while for MetaPSICOV the corresponding values are 66.51% and 68.43%. CCMpred outperforms PconsC4 in terms of precision (72.02% *vs.*70.75%) and MCC (25.64% *vs.*25.49%) on the top $L/5$ contact predictions.

TABLE 2.1: Prediction performance on the PREVIOUS dataset for inter-helical residue contacts.

Predictor	**Threshold**	**Precision**	**Recall**	**F_1**	**$F_{0.35}$**	**MCC**
PSICOV	*L*	31.90	21.31	24.75	29.84	24.10
EVfold	*L*	37.42	25.23	29.10	35.03	28.72
CCMpred	*L*	44.25	29.58	34.24	41.36	34.19
Gaussian DCA	*L*	38.74	26.34	30.21	36.27	29.91
plmDCA	*L*	42.87	29.05	33.29	40.10	33.21
Gremlin	*L*	39.95	25.05	30.19	37.17	29.87
DeepCov	*L*	33.36	23.44	26.24	31.30	25.80
MetaPSICOV	*L*	43.71	28.99	33.62	40.80	33.59
PconsC3	*L*	48.19	33.01	37.81	45.21	37.86
Pconsc4	*L*	45.30	30.91	35.28	42.41	35.33
Membrain2	*L*	57.69	38.11	44.51	53.91	45.05
DeepMetaPSICOV	*L*	67.42	45.18	52.35	63.13	53.28
DeepHelicon	*L*	63.69	43.26	49.47	59.59	50.41
PSICOV	*L/2*	44.48	14.87	21.67	35.75	24.25
EVfold	*L/2*	49.66	16.75	24.29	39.97	27.34
CCMpred	*L/2*	59.64	20.30	29.21	47.95	33.18
Gaussian DCA	*L/2*	52.86	18.07	25.99	42.56	29.32
plmDCA	*L/2*	58.02	19.69	28.38	46.64	32.21
Gremlin	*L/2*	56.81	18.24	27.05	45.37	30.85
DeepCov	*L/2*	43.08	15.43	21.58	34.86	24.04
MetaPSICOV	*L/2*	55.62	18.43	26.94	44.63	30.58
PconsC3	*L/2*	61.45	21.37	30.53	49.70	34.58
PconsC4	*L/2*	58.87	20.29	29.12	47.51	32.97
Membrain2	*L/2*	72.40	24.63	35.52	58.33	40.68
DeepMetaPSICOV	*L/2*	77.60	26.72	38.36	62.69	43.89
DeepHelicon	*L/2*	77.84	27.52	38.82	62.93	44.43
PSICOV	*L/5*	61.19	8.24	14.26	34.86	21.47
EVfold	*L/5*	62.68	8.51	14.67	35.77	22.07
CCMpred	*L/5*	72.02	9.98	17.07	41.16	25.64
Gaussian DCA	*L/5*	67.24	9.35	16.00	38.54	23.94
plmDCA	*L/5*	70.75	9.72	16.66	40.33	25.08
Gremlin	*L/5*	68.94	9.17	15.93	39.17	24.20
DeepCov	*L/5*	53.96	7.62	12.90	30.96	19.11
MetaPSICOV	*L/5*	66.51	8.78	15.27	37.66	23.23
PconsC3	*L/5*	72.75	10.07	17.39	42.01	26.07

PconsC4	*L/5*	70.75	10.09	17.14	40.81	25.49
Membrain2	*L/5*	83.50	11.31	19.61	47.82	29.79
DeepMetaPSICOV	*L/5*	86.24	12.12	20.69	49.80	31.15
DeepHelicon	*L/5*	87.42	12.70	21.48	50.74	31.99
PSICOV	*L/10*	69.84	4.78	8.83	27.12	17.52
EVfold	*L/10*	70.69	4.94	9.09	27.68	17.89
CCMpred	*L/10*	78.25	5.48	10.07	30.55	19.85
Gaussian DCA	*L/10*	74.94	5.29	9.71	29.41	19.06
plmDCA	*L/10*	76.10	5.29	9.73	29.60	19.22
Gremlin	*L/10*	75.17	5.00	9.30	28.95	18.74
DeepCov	*L/10*	60.96	4.40	7.98	23.93	15.45
MetaPSICOV	*L/10*	71.84	4.73	8.80	27.49	17.78
PconsC3	*L/10*	78.75	5.39	10.01	30.86	19.93
PconsC4	*L/10*	75.00	5.52	10.05	29.85	19.36
Membrain2	*L/10*	88.21	5.98	11.10	34.35	22.31
DeepMetaPSICOV	*L/10*	88.28	6.03	11.19	34.55	22.39
DeepHelicon	*L/10*	91.33	6.58	12.06	36.25	23.60

TABLE 2.2: Prediction performance on the TEST dataset for inter-helical residue contacts.

Predictor	**Threshold**	**Precision**		**F$_1$**	**F$_{0.35}$**	**MCC**
PSICOV	*L*	31.55	22.44	25.49	29.83	24.39
EVfold	*L*	40.37	29.18	32.89	38.25	32.16
CCMpred	*L*	46.04	33.14	37.41	43.58	36.93
Gaussian DCA	*L*	40.37	29.18	32.87	38.24	32.15
plmDCA	*L*	44.58	31.95	36.13	42.18	35.59
Gremlin	*L*	40.78	28.88	32.86	38.53	32.18
DeepCov	*L*	39.98	29.89	33.03	37.98	32.39
MetaPSICOV	*L*	45.62	33.03	37.19	43.22	36.70
PconsC4	*L*	49.16	35.78	40.28	46.65	39.87
Membrain2	*L*	58.75	42.50	47.89	55.67	48.01
DeepMetaPSICOV	*L*	67.76	51.76	56.70	64.60	57.19
DeepHelicon	*L*	62.13	45.75	51.01	58.95	51.26
PSICOV	*L/2*	44.04	16.00	22.83	36.17	24.88
EVfold	*L/2*	53.57	19.63	27.95	44.10	30.78
CCMpred	*L/2*	61.28	22.50	32.03	50.47	35.49
Gaussian DCA	*L/2*	53.64	19.67	27.99	44.15	30.83

plmDCA	*L/2*	59.57	21.69	30.95	48.97	34.31
Gremlin	*L/2*	60.93	22.20	31.67	50.09	35.15
DeepCov	*L/2*	52.91	20.05	28.16	43.81	30.90
MetaPSICOV	*L/2*	58.17	21.26	30.33	47.88	33.56
PconsC4	*L/2*	62.98	23.24	33.09	52.00	36.66
Membrain2	*L/2*	74.46	27.54	39.12	61.45	43.72
DeepMetaPSICOV	*L/2*	80.76	31.81	44.06	67.37	48.91
DeepHelicon	*L/2*	76.16	28.55	40.30	62.96	44.98
PSICOV	*L/5*	60.12	9.031	15.42	35.98	22.18
EVfold	*L/5*	67.51	10.17	17.34	40.42	25.06
CCMpred	*L/5*	74.41	11.32	19.27	44.72	27.86
Gaussian DCA	*L/5*	68.85	10.49	17.85	41.37	25.71
plmDCA	*L/5*	73.26	11.03	18.81	43.84	27.27
Gremlin	*L/5*	73.90	11.26	19.16	44.40	27.68
DeepCov	*L/5*	66.23	10.18	17.29	39.93	24.81
MetaPSICOV	*L/5*	68.43	10.08	17.31	40.78	25.20
PconsC4	*L/5*	74.81	11.24	19.23	44.92	27.91
Membrain2	*L/5*	86.64	13.14	22.37	52.02	32.61
DeepMetaPSICOV	*L/5*	87.63	13.95	23.54	53.53	33.71
DeepHelicon	*L/5*	84.98	13.05	22.17	51.24	32.12
PSICOV	*L/10*	69.08	5.28	9.68	28.69	18.24
EVfold	*L/10*	74.37	5.72	10.49	31.03	19.77
CCMpred	*L/10*	79.78	6.17	11.30	33.33	21.29
Gaussian DCA	*L/10*	75.65	5.88	10.76	31.69	20.20
plmDCA	*L/10*	79.32	6.17	11.30	33.23	21.23
Gremlin	*L/10*	79.34	6.17	11.30	33.22	21.22
DeepCov	*L/10*	69.90	5.43	9.96	29.57	18.81
MetaPSICOV	*L/10*	72.21	5.36	9.88	29.65	18.87
PconsC4	*L/10*	81.34	6.28	11.52	34.10	21.76
Membrain2	*L/10*	91.27	7.05	12.92	38.17	24.50
DeepMetaPSICOV	*L/10*	90.39	7.24	13.23	38.55	24.66
DeepHelicon	*L/10*	87.44	6.76	12.40	36.58	23.44

Overall, the results of our comparative evaluation are in line with the previous studies by us and others. CCMpred ranks above MetaPSICOV in terms of top L/5 contact predictions (Hönigschmid and Frishman, 2016) and achieves the best performance among all ECA-based predictors (Yang and Shen, 2018). CCMpred was reported to outperform Gremlin and plmDCA by Seemayer et al. for globular proteins (Seemayer, Gruber, and Söding, 2014), while plmDCA, in its turn, performs better than Gaussian DCA (Baldassi et al., 2014; Michel et al., 2017). Similar to our findings, all four predictors were previously reported to outperform PSICOV and EVfold (Kamisetty, Ovchinnikov, and Baker, 2013; Seemayer, Gruber, and Söding, 2014; Baldassi et al., 2014; Feinauer et al., 2014). In our work we have also confirmed the excellent performance of DeepMetaPSICOV in predicting residue contacts in transmembrane proteins (Kandathil, Greener, and Jones, 2019a). To the best of our knowledge, PconsC3, PconsC4, and DeepCov have not, so far, been systematically tested on transmembrane proteins, but on globular proteins PconsC3 and PconsC4 reportedly perform better than ECA-based predictors (Michel et al., 2017; Michel, Menéndez Hurtado, and Elofsson, 2019). In our tests, these two predictors also outperform the ECA-based predictors, but lag behind Membrain2, DeepMetaPSICOV, and DeepHelicon.

2.3.3 Dependence of mean precision on protein characteristics

We benchmarked the mean precision of DeepHelicon in predicting the top $L/2$ inter-helical residue contacts with respect to the log number of effective sequences in MSA *ln(Meff)* (Wang et al., 2017), the number of TM helices, and the total number of homologs in MSA. The *ln(Meff)* measure essentially reflects the amount of homologous information in an MSA and is calculated as the number of non-redundant protein sequences at a 70% sequence identity cutoff (Wang et al., 2017). For each of the three characteristics, its value range was split into five equal bins and the mean precision was calculated for each bin separately.

For the majority of the predictors the mean precision rises with the increase in the number of effective sequences both on the PREVIOUS (Fig. 2.9a) and, especially, on the TEST (Fig. A.1a) datasets. Overall, a similar trend is observed with respect to the total number of sequences in the MSA (Figs. 2.9b and A.1b), but it is less pronounced, which implies that the quality of the alignments is more important than their sheer size, as previously reported (Jones and Kandathil, 2018). DeepHelicon outperforms other predictors on proteins with a large number of TM helices (Figs. 2.9c and A.1c). For example, the mean precision for proteins with 12-15 TM helices is 75.59% compared to 73.68% for Membrain2 and 70.61% for DeepMetaPSICOV (PREVIOUS dataset), while for proteins with 10-12 TM helices it is 82.89% compared to 75.18% of Membrain2 and 82.04% of DeepMetaPSICOV (TEST dataset). Most of the predictors exhibit a good prediction performance on both datasets for the TM proteins with five to eight helices.

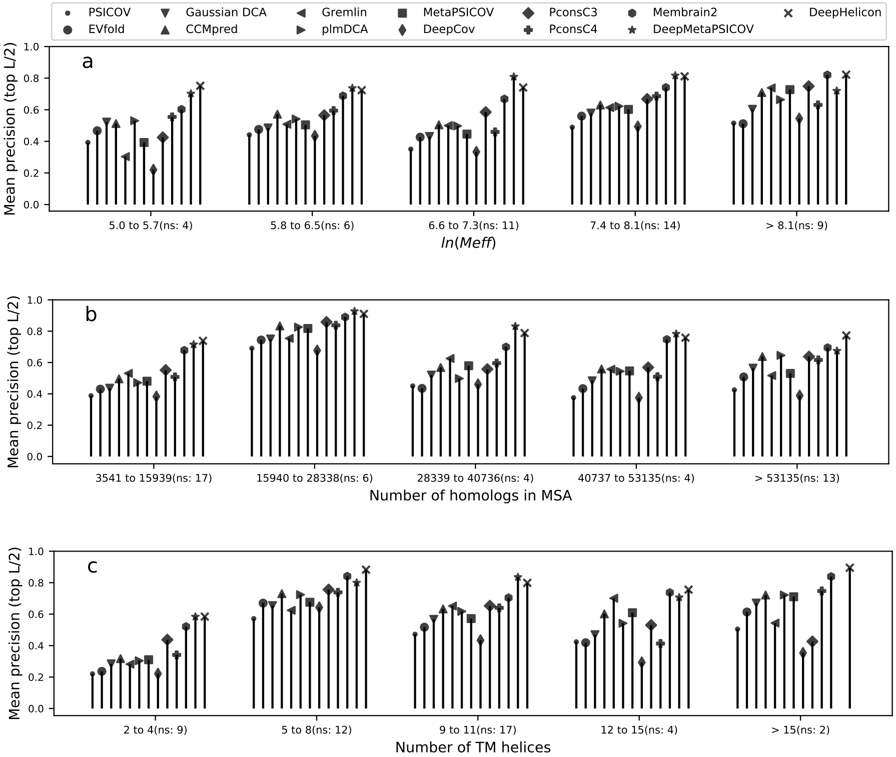
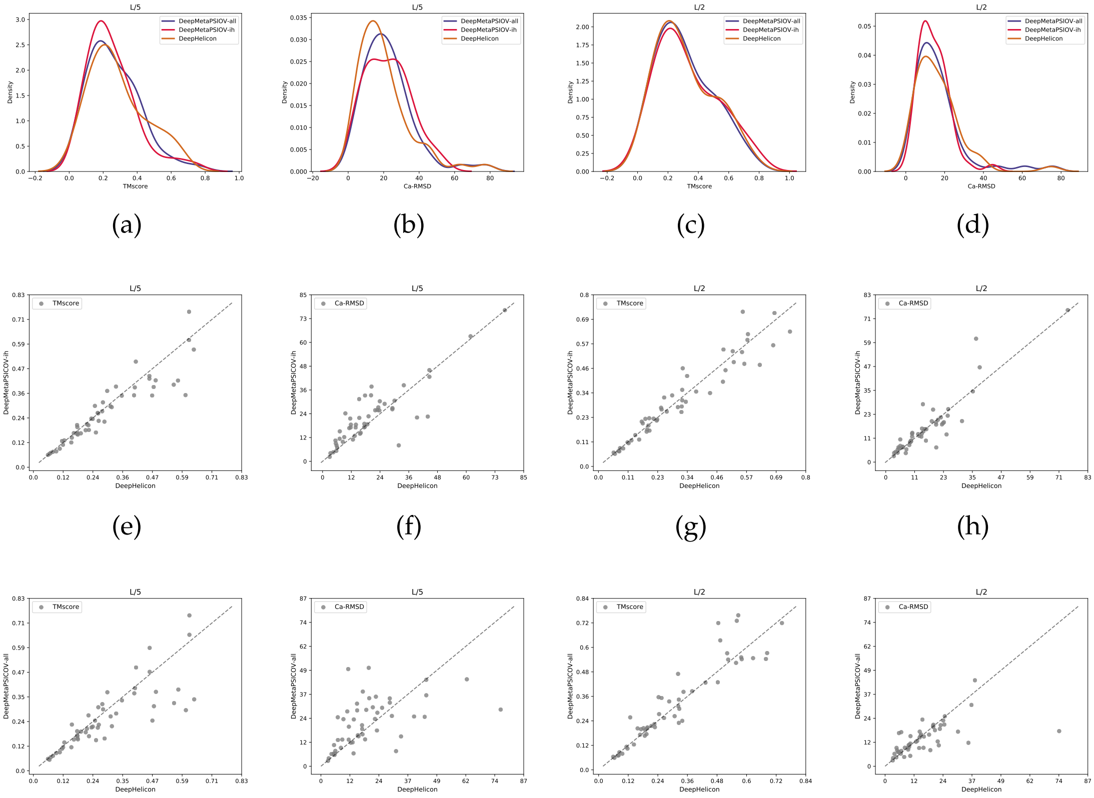


FIGURE 2.9: Mean precision of Top $L/2$ inter-helical contact predictions on the PREVIOUS dataset for different ranges of *ln(Meff)* (a), the number of all homologs in MSA (b), and the number of TM helices (c). ns: the number of sequences. Some of the large protein chains were omitted from this comparison because the webservers or standalone packages we tested did not return results for them (see section 2.2.10 and Table A.6).

2.3.4 Contact-driven modeling of α-helical TM proteins by CONFOLD2

We employed CONFOLD2 to model 3D structures guided by the top-ranked $N/5$ and $N/2$ contacts predicted by DeepHelicon and DeepMetaPSICOV (Tables A.11 and A.12). Note that DeepHelicon is only trained to predict inter-helical contacts while for DeepMetaPSICOV we considered both inter-helical (Deep- MetaPSICOV-ih) as well as all non-local contacts (DeepMetaPSICOV-all) formed by residue pairs with a sequence separation at least 6. As seen in Fig. 2.10, CONFOLD2 models exhibit a comparable quality both in terms of TM-scores and Cα-RMSD values using predicted contacts generated by either of the methods. With regard to the 3D models guided by inter-helical contacts at $N/5$ we found that out of the 51 proteins in the TEST dataset DeepHelicon generates 14 models with TM-scores higher than 0.4 (27.45%), while DeepMetaPSICOV generates 8 models (15.69%) (Table A.12). At $N/2$ both methods lead to 15 models with TM-scores higher than 0.4 (29.41%). These results demonstrate that using a fully automated modeling method such as CONFOLD2 in conjunction with state-of-the-art contact prediction methods, acceptable models can be generated for almost 30% of α-helical transmembrane proteins, even when only inter-helical contacts are taken into account.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

FIGURE 2.10: Comparison of TM-scores and Cα-RMSD values obtained for individual 3D models guided by inter-helical residue contacts in the TEST dataset. Distribution of TM-scores and Cα-RMSD values at $L/5$ (a) and (b) and at $L/2$ (c) and (d), respectively. TM-scores and Cα-RMSD values of DeepHelicon for each protein compared to DeepMetaPSICOV-ih at $L/5$ (e) and (f) and at $L/2$ (g) and (h), respectively, and DeepMetaPSICOV-all at $L/5$ (i) and (j) and at $L/2$ (k) and (l), respectively.

We illustrate the CONFOLD2 results with the models of one small and one large protein. For the first example, we present CONFOLD2 models of succinate dehydrogenase (PDB code 2acz, chain C) from Escherichia coli, guided by top $N/5$ predicted inter-helical residue contacts. The backbone of this protein is composed of three α-helices in the transmembrane region and one α-helix in the extracellular region (Fig. 2.11a). The best structure assisted by the inter-helical contacts predicted by DeepHelicon (Fig. 2.11b) achieves the highest TM-score of 0.620 relative to the native structure, while contacts generated by DeepMetaPSICOV lead to a structure with a TM-score of 0.613 (Fig. 2.11c). The Cα-RMSD values for

DeepHelicon and DeepMetaPSICOV are 5.630 and 9.187, respectively. The second example is the CONFOLD2 models of the uracil transporter (PDB code 3qe7, chain A) from Escherichia coli, guided by top N/2 predicted inter-helical residue contacts. This protein possesses 14 helices (Fig. 2.11d). The best model guided by DeepHelicon predictions (Fig. 2.11e) achieves a TM-score of 0.500 and a Cα-RMSD value of 11.411 relative to the native structure while the DeepMetaPSICOV-guided model (Fig. 2.11f) has a TM-score of 0.377 and a Cα-RMSD value of 15.029.

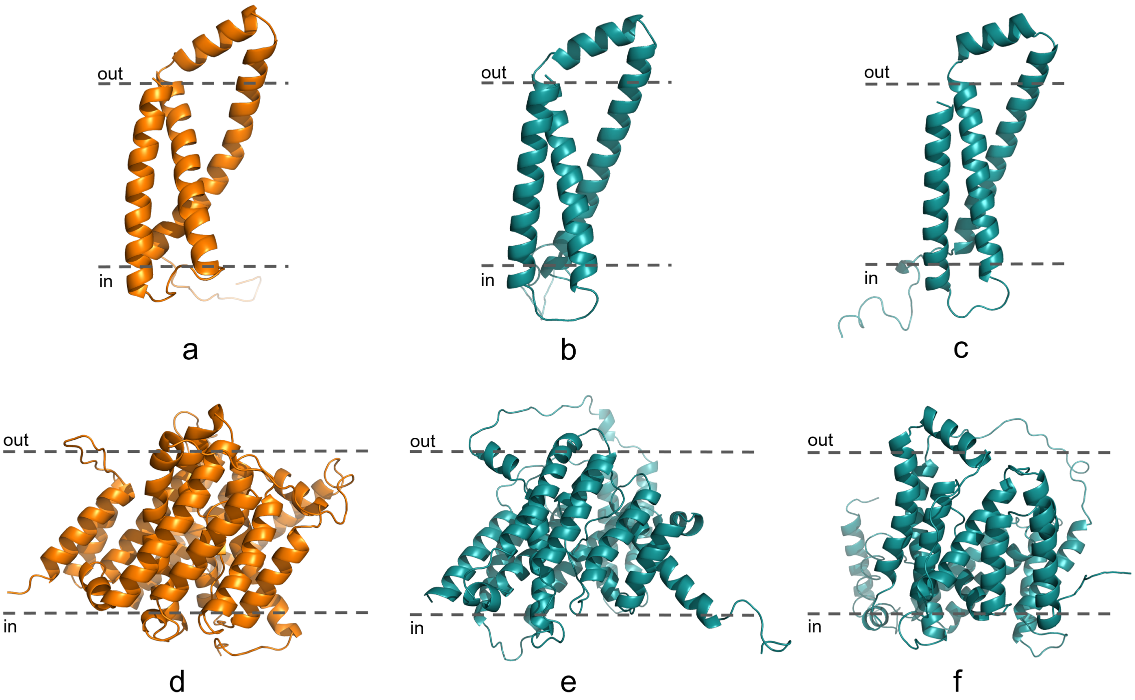


FIGURE 2.11: 3D modeling of succinate dehydrogenase (chain C) and uracil transporter (chain A) from *Escherichia coli*, guided by inter-helical residue contacts. (a), (b), and (c) correspond to the native structure, DeepHelicon-guided, and DeepMetaPSICOV-guided CONFOLD2 models of succinate dehydrogenase (chain C), respectively. (d), (e), and (f) correspond to the native structure, DeepHelicon-guided, and DeepMetaPSICOV-guided CONFOLD2 models for the uracil transporter (chain A), respectively.

2.4 Conclusion

The first specialized predictor for transmembrane proteins exploiting coevolving residues was developed by our group in 2007 (Fuchs et al., 2007). Subsequent iterations of the predictor, published in 2009 and 2016, employed neural networks (Fuchs, Kirschner, and Frishman, 2009) and a random forest model combined with direct coupling analysis (Hönigschmid and Frishman, 2016). In this work we have developed DeepHelicon, a next-generation deep-learning approach to predict inter-helical residue contacts in TM proteins. Our method has been trained on one of the currently largest datasets of membrane proteins, which is however still much smaller than the datasets containing thousands of globular and membrane proteins used to train general purpose predictors (Stahl, Schneider, and Brock, 2017; Xiong, Zeng, and Gong, 2017; Jones and Kandathil, 2018). Similar to other state-of-the-art methods, DeepHelicon relies on a two-stage deep learning architecture based on residual neural networks, which allow for very fast optimization. At the first stage two kinds of co-evolutionary features (coupling matrix and co-evolutionary features) are used to generate coarse-grained contact maps, which serve as input for the second stage. At the second stage we employ a novel iterative scheme, which leads to a progressive improvement of prediction performance due to recursive learning of contact maps from a previous iteration. Variance error is reduced by combining the decisions from multiple models. The contact prediction accuracy is sufficient to generate acceptable 3D models for up to 30% of proteins using a simple fully automated modeling method such as CONFOLD2. Moreover, we find that inter-helical contacts alone provide enough constraints for building 3D models of α-helical membrane proteins.

2.5 Software and data availability

The standalone DeepHelicon software is available at `https://github.com/2003100127/deephelicon`. It relies on the following external methods: HHBlits, CCMpred, Gaussian DCA, FreeContact, plmDCA, TMHMM2.0, and EVCouplings. DeepHelicon only takes as input a protein sequence in FASTA format. Residues located in the transmembrane regions are detected by the TMHMM2.0 algorithm. The output contains predicted inter-helical residue contacts. Training and testing data are available at `https://data.mendeley.com/datasets/k8tfvgftv3`. We are currently re-training DeepHelicon based on the most recent UniProt version, an updated version will be made available.

Chapter 3

DeepTMInter: improving prediction of interaction sites in transmembrane protein complexes using deep residual neural networks

Biophysical interactions between proteins are fundamental for a wide range of biological processes. Transmembrane (TM) proteins with known interaction sites are found to be pharmaceutically instrumental for drug discovery and therapy design. Due to pitfalls inherent to the laborious experimental determination of TM protein structures, the clear understanding of interaction details at an intermolecular level has been hampered for a few decades. Computational techniques are therefore required to allow large-scale functional annotations of TM protein interaction sites. Here, we present a novel deep-learning method, DeepTMInter, for sequence-based prediction of interaction sites in TM proteins by leveraging a collection of molecular physiochemical and evolutionary properties. Our method, trained using ultra-deep residual neural networks followed by stacked generalization for performance refinements, has enabled a substantial improvement for predicting interaction sites in cytoplasmic, transmembrane, and extracellular regions as well as full sequences. We showed that DeepTMInter outperformed our previously best performing method, MBpred, in terms of AUC/AUCPR values of 0.689/0.598 compared to 0.589/0.493 on a stringently redundancy-reduced independent dataset. Our systematical investigation of human transmembrane protein interactome first reveals that proteins of higher percentage of interaction sites are found to be significantly richer in interaction partners. In addition, the human ion channel group is identified by DeepTMInter as the largest functional family to accommodate 25.6% interaction sites per protein.

3.1 Introduction

Protein-protein interactions (PPIs) lay the foundations for manifold cellular activities (Kuzmanov and Emili, 2013; Zhang et al., 2019), such as signal transduction (Moore, Berger, and DeGrado, 2008) and immune response (Hubel et al., 2019; Li, Fang, and Fang, 2011). The accurate identification of interface patches and, in particular, specific interaction sites (e.g., drug-binding sites) has implication for drug discovery (Bai et al., 2016) and disease treatment (Yin and Flynn, 2016). Owing to difficulties of experimental structure determination, transmembrane proteins (TM), while accounting for 20-30% of gene-encoding proteins in living organisms (Fuchs, Kirschner, and Frishman, 2009; Sharpe, Stevens, and Munro, 2010), have a large number of interaction sites that still remain unknown and unidentified. It has been reported that transmembrane proteins are targeted by around 50% commercially released pharmaceutical drugs (Varga et al., 2016; Dobson, Reményi, and Tusnády, 2015). In particular, a clear understanding of interaction sites of human transmembrane proteome is key to catalyzing the development of various disease-associated drugs (Lin et al., 2019; Stone and Deber, 2017). On the other hand, experimentally PPI determined techniques such as yeast two-hybrid (Y2H) assays (Shoemaker and Panchenko, 2007), although supporting high throughput on a proteome-wide scale (Figeys, 2008), only provide binary protein interactome maps that are meanwhile intrinsically limited by high false-positive or false-negative rates (Jessulat et al., 2011; Liu et al., 2020a). Thus, computational techniques are urgently needed for identification of interaction sites in transmembrane proteins.

The current computational tools used for interaction site prediction can roughly be categorized into two groups *i*) predicting binary PPIs, such as Profppikernel (Hamp and Rost, 2015), ProfPPIdb (Tran, Hamp, and Rost, 2018), and DPPI (Hashemifar et al., 2018), and *ii*) predicting protein interaction sites, such as Hamp's work (Hamp and Rost, 2012), DLPred (Zhang et al., 2019), and DELPHI (Li and Ilie, 2020). The vast majority of the off-the-shelf techniques are trained on globular proteins while only two available methods, Bordner's work (Bordner, 2009) and MBpred (Zeng, Hönigschmid, and Frishman, 2019) are systematically trained on transmembrane proteins. Over the past decade, deep learning heralding the next generation of intelligent algorithms (LeCun, Bengio, and Hinton, 2015) has achieved unrivaled successes across a broad spectrum of biological applications (Li, Wu, and Ngom, 2018; Wainberg et al., 2018; Eraslan et al., 2019) compared to traditional techniques. More recently deep residual neural networks (ResNets) (He et al., 2016a) have enabled considerable progress in predicting secondary structures (Hanson et al., 2018) and residue contacts (Sun and Frishman, 2020;

Kandathil, Greener, and Jones, 2019b) as well 3D protein structures (Senior et al., 2020); nevertheless, rather few studies have indeed applied the ResNet technique to predicting protein interaction sites.

Here, we introduce a fully automated deep-learning tool, DeepTMInter, to predict interaction sites in transmembrane proteins. This method has been 5-fold cross-validated by stratified-shuffle methods (Liu et al., 2020b) using ultra-deep residual neural networks integrated with 27 residual units, followed by a thorough refinement of prediction performance using stacked generalization (Wolpert, 1992) for model ensemble and variance error reduction. DeepTMInter has been trained on the largest transmembrane protein training dataset consisting of 301 well-curated high-quality chains from 241 unique transmembrane protein assemblies. DeepTMInter has significantly outperformed MBpred, our previous-generation prediction tool, in the light of AUC/AUCPR values of 0.689/0.598 compared to 0.589/0.493 on the independent set of 30 chains whose redundancy has been strictly reduced to below a 25% sequence identity level to both themselves and training chains. We systematically investigated human TM protein interaction networks based on 76,584 high-quality PPIs in the HuRI-Union database (Li and Ilie, 2020). Our experiments unravel that a high percentage of per-protein interaction sites predicted by DeepTMInter corresponds to a high number of interaction partners. Furthermore, we found that alignment filtering allowed our method to run without accuracy loss at a very fast speed risen by around one order of magnitude.

3.2 Materials and method

3.2.1 Datasets of transmembrane proteins with known 3D structure

We obtained from the PDBTM database (version: July 2020) (Kozma, Simon, and Tusnady, 2012) a dataset of 3090 three-dimensional structures of α-helical TM proteins at better than 3.5Å resolution (Fig. 3.1a). Their biological oligomer structures were generated using the TMDET algorithm (Tusnády, Dosztányi, and Simon, 2004; Tusnády, Dosztányi, and Simon, 2005) 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 nonhydrogen atoms of less than 6Å(see section 3.2.2 for detailed information about interaction site definition). This procedure resulted in 10194 unique protein chains.

For comparison purposes we also used two additional datasets described in our previous work (Zeng, Hönigschmid, and Frishman, 2019). Briefly, the CompData dataset (101 TM protein chains, Table B.1) was derived by imposing a less than 30% sequence identity cutoff on a dataset of 267 TM protein chains benchmarked by Bordner (Bordner, 2009). The TestData dataset (Table B.2) 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 (Zeng, Hönigschmid, and Frishman, 2019). The structures of 81 and 35 chains in the CompData and TestData datasets were determined at better than 3.5Åresolution, respectively. Upon removing these 116 chains from the collection of 10194 chains described above, we were left with 10078 chains. Following the common practice in structural bioinformatics (Zou et al., 2020; Hanson et al., 2018; Heffernan et al., 2015), 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) (Tables B.3 and B.4).

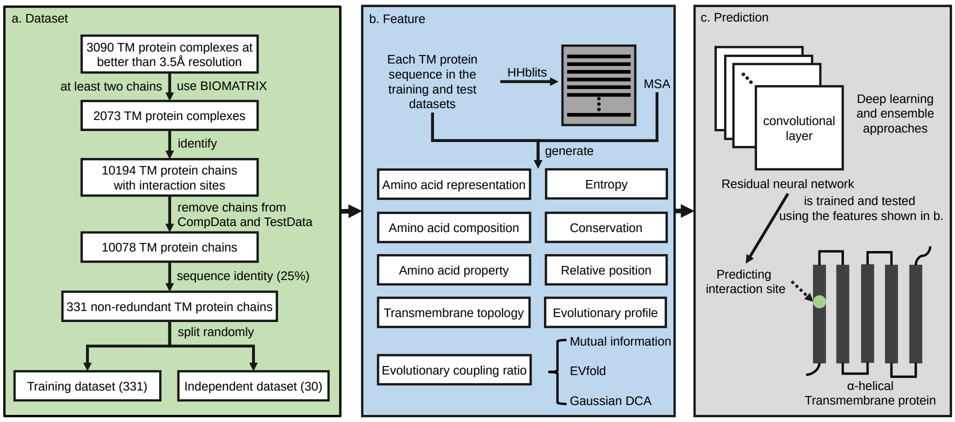

a. Dataset
3090 TM protein complexes at better than 3.5Å resolution
at least two chains
use BIOMATRIX
2073 TM protein complexes
identify
10194 TM protein chains with interaction sites
remove chains from CompData and TestData
10078 TM protein chains
sequence identity (25%)
331 non-redundant TM protein chains
split randomly
Training dataset (331)
Independent dataset (30)
b. Feature
Each TM protein sequence in the training and test datasets
HHblits
MSA
generate
Amino acid representation
Entropy
Amino acid composition
Conservation
Amino acid property
Relative position
Transmembrane topology
Evolutionary profile
Evolutionary coupling ratio
Mutual information
EVfold
Gaussian DCA
c. Prediction
convolutional layer
Deep learning and ensemble approaches
Residual neural network
is trained and tested using the features shown in b.
Predicting interaction site
α-helical Transmembrane protein

FIGURE 3.1: Flowchart of our method to predict interaction sites in TM proteins. (a), (b), and (c) schematically show the dataset generation, input feature, and prediction process, respectively.

3.2.2 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 (Zeng, Hönigschmid, and Frishman, 2019), 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 (Hamp and Rost, 2012), which is based on the distance between any two non-hydrogen atoms of less than 6Å.

3.2.3 Protein topology

Extracellular (Extra), transmembrane (TM), and cytoplasmic (Cyto) segments were structure-derived and predicted exactly as in our previous publication (Zeng, Hönigschmid, and Frishman, 2019). A combination (Combined) of the three segment types above was also used in benchmarking the performance of predictors. Note that different from determining protein topologies for DeepHelicon, DeepTMInter used structure-derived and Phobius-predicted topologies instead of those predicted by TMHMM (cf. section 2.2.2 of *Chapter* 2).

3.2.4 Multiple sequence alignments

The multiple sequence alignments (MSAs) were generated in the same way as described in section 2.2.4 of *Chapter* 2. The Uniclust30 database (`http://wwwuser.gwdg.de/~compbiol/uniclust/2020_03/`) was used. In order to keep the CPU and memory requirements for calculating features at a manageable level, HHfilter (Remmert et al., 2012) was applied to only keep sequences sharing <90% sequence identity, which resulted in a significant reduction of MSA depth.

3.2.5 Input features

For each amino acid and each MSA position we generated a series of sequence-based, physiochemical, and evolutionary characteristics (Fig. 3.1b), 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.

3.2.5.1 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.

3.2.5.2 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 (Kawashima et al., 2007) 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) (Table B.5). Each physicochemical scale was rescaled to the range [0,1]. For comparison purposes we also used several other widely used amino acid physicochemical scales (Table B.6).

3.2.5.3 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.

3.2.5.4 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.

3.2.5.5 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 $\frac{1}{\log_2(20)}$.

3.2.5.6 Relative sequence position

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

3.2.5.7 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.

3.2.5.8 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 (Stein, Marks, and Sander, 2015). In order to quantify the likelihood of a given residue to be involved in a contact, the evolutionary coupling ratio (ECR) has been proposed (Hopf et al., 2012):

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

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://rostlab.org/free/`) and Gaussian DCA (`https://github.com/carlobaldassi/`

`GaussDCA.jl`). The ECR feature is thus represented by a vector of length 3. Note that DeepHelicon utilizes four tools for generating coevolutionary features (see section 2.2.5.1).

3.2.6 The deep learning approach

3.2.6.1 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 (Table B.7).

3.2.6.2 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. 3.2). The architecture settings are similar to those in DeepHelicon (see explanations for batch normalization and ReLU described in section 2.2.6 of *Chapter* 2). For each amino acid position, the 678-dimensional feature vector (section 3.2.6.1) was reshaped into a 26×26 matrix (*i.e.* with 676 elements, with 16 dimensions padded by 0), which was batch-normalized in order to speed up the training process (Ioffe and Szegedy, 2015).

At the core of our ResNet architecture are 27 residual units (also called residual blocks), all with the same structure (Fig. 3.2). For comparison, the number of residual units in some of the recently published ResNet architectures were 9 (Wu et al., 2020), 18 (Jones and Kandathil, 2018), and 22 (Li et al., 2019b). This design of the residual unit, already used in our previous work (Sun and Frishman, 2020), allows to accelerate the optimization and to avoid overfitting of the ResNet architecture (He et al., 2016b). We plugged an additional block comprising batch normalization and a convolutional layer with stride 2 for reducing data dimensionality and the computational cost (Fig. 3.2). We implemented the deep architecture by using Google's Tensorflow library (version 1.12.0) based on Python programming language.

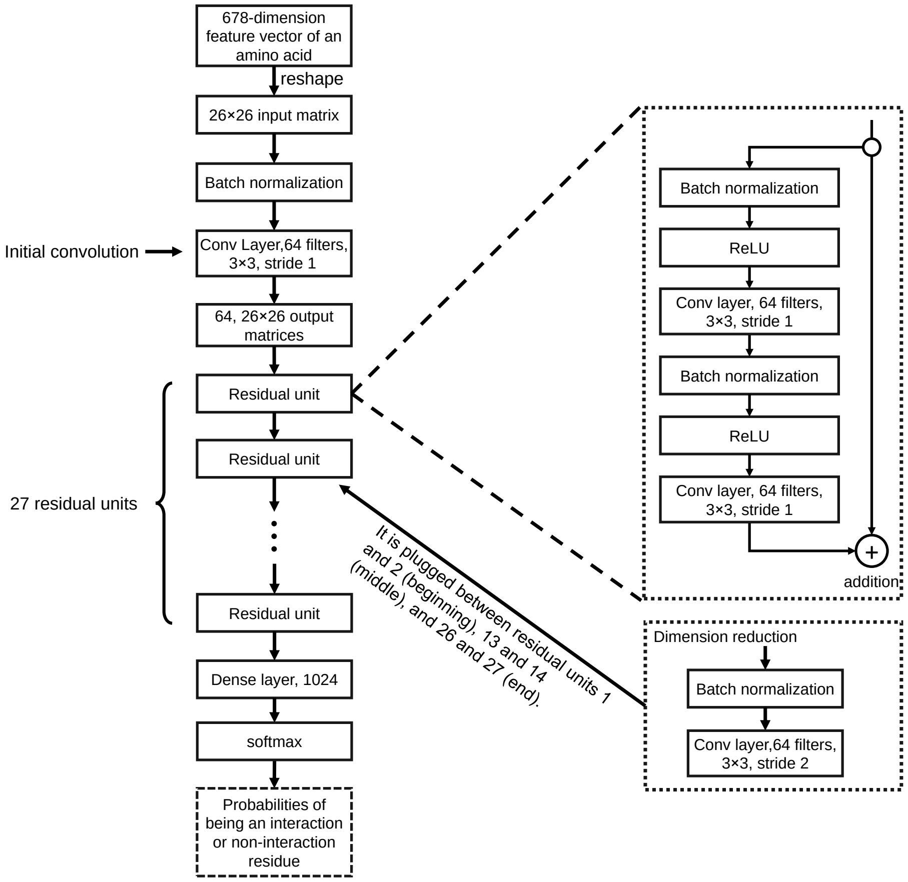

678-dimension feature vector of an amino acid
reshape
26×26 input matrix
Batch normalization
Initial convolution
Conv Layer,64 filters, 3×3, stride 1
64, 26×26 output matrices
27 residual units
Residual unit
Residual unit
Residual unit
Dense layer, 1024
softmax
Probabilities of being an interaction or non-interaction residue
Batch normalization
ReLU
Conv layer, 64 filters, 3×3, stride 1
Batch normalization
ReLU
Conv layer, 64 filters, 3×3, stride 1
+
addition
It is plugged between residual units 1 and 2 (beginning), 13 and 14 (middle), and 26 and 27 (end).
Dimension reduction
Batch normalization
Conv layer,64 filters, 3×3, stride 2

FIGURE 3.2: Layout of the deep ResNet architecture to predict interacting amino acid residues in TM proteins.

3.2.6.3 Settings for training the ResNet architecture

The training procedure is similar to that for training DeepHelicon described in section 2.2.7 of *Chapter* 2.

3.2.6.4 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 >800 (4 chains). The 5-fold stratified-shuffle cross validation (Liu et al., 2020b) method described in section 2.2.9 of *Chapter* 2 was employed to evenly allocate protein chains of different length classes for training and validation at each iteration (Fig. B.1).

3.2.6.5 Avoiding over-training

Over-training is detrimental to the performance of models on unseen validation data (Prechelt, 1998; Amari et al., 1997; Hawkins, 2004), even if they achieve ideal performance on training data (Tetko, Livingstone, and Luik, 1995) (Fig. B.2). In order to avoid over-training, the early stopping strategy (Tetko and Villa, 1997) was adopted, which involves aborting the training when the performance on validation data begins to worsen (Amari et al., 1996). 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 (Tetko, Livingstone, and Luik, 1995; Tetko and Villa, 1997).

3.2.6.6 Stacked generalization

Stacked generalization (Wolpert, 1992), an approach for implementing model ensembles (Anifowose, Labadin, and Abdulraheem, 2015), was used to minimize the generalization errors of the models trained by the ResNet architecture (He et al., 2013) (Fig. 3.3). 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 3.3.1), served as input for a multi-layer perceptron (MLP) (Gardner and Dorling, 1998) and a Gaussian Naive Bayes (GNB) classifier (Lou et al., 2014) (Fig. 3.3). 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`).

3.2.7 Evaluation criteria

The overall performance of DeepTMInter was evaluated based on two threshold-free (Saito and Rehmsmeier, 2015; Yuan et al., 2018) measures: the area under the ROC (Receiver Operating Characteristic) curve (AUC) and the area under the Precision-Recall curve (AUCPR) (Boyd, Eng, and Page, 2013). Most of single-threshold performance measures used here (Saito and Rehmsmeier, 2015) have been introduced in section 2.2.8 of *Chapter* 2, except:

$$\text{Jaccard similarity coefficient } (JSC) = \frac{TP}{TP+FP+FN}$$

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

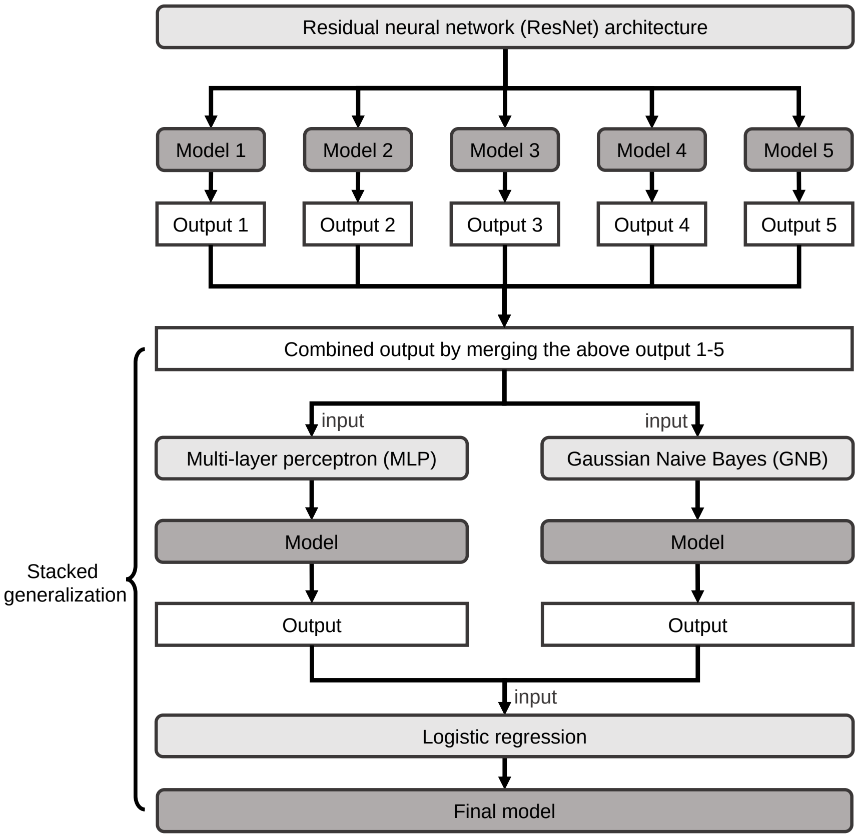


FIGURE 3.3: Using stacked generalization to further enhance the prediction performance of the ResNet models. Boxes in shallow grey: deep learning or machine learning methods; boxes in dark grey: respective obtained models.

residues predicted as non-interacting, and the number of interacting residues predicted as non-interacting, respectively. For each protein of length L we evaluated prediction based on the top-ranked $L/5$ interaction sites.

3.2.8 Comparison with MBPred

We compared the performance of DeepTMInter on three test datasets (see section 3.2.1) with the MBPred algorithm previously developed in our group ((Zeng, Hönigschmid, and Frishman, 2019); `https://github.com/bojigu/MBPred`). The standalone MBPred suite contains four individual predictors - MBPredTM, MBPred-Cyto, 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.

3.2.9 Human transmembrane proteins

We obtained 5,178 human protein sequences with at least one annotated transmembrane region from the UniProtKB/Swiss-Prot database (UniProt consortium, 2019). 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 5,051 human transmembrane proteins with the MSA depth in the range between 20 and 10,000 after filtering (see section 3.2.4 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/`) (Armstrong et al., 2020; Alexander et al., 2019).

3.2.10 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 (Luck et al., 2020), 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) (Oughtred et al., 2019) and 1,063,382 binary interactions from the IntAct database (version: 4.1.25) (Orchard et al., 2014), respectively. A mapping between the PDB codes and UniProt IDs of proteins was obtained by PyPDB (Gilpin, 2016).

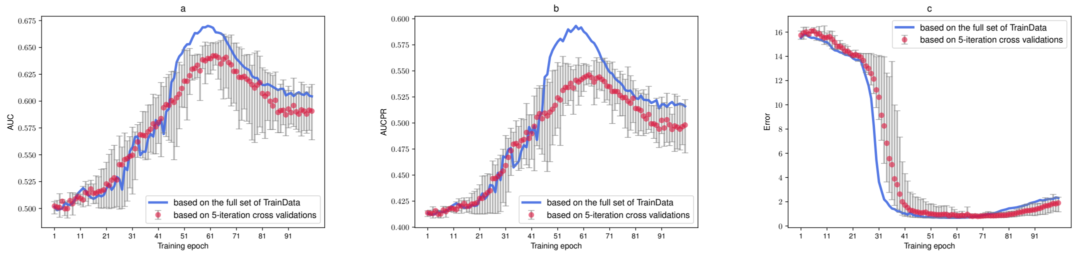
3.3 Results

3.3.1 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 3.2.6.4 and 3.2.6.6).

Note that no protein chain in the TrainData dataset shares more than 25% sequence identity with any protein from the IndepData dataset (see section 3.2.1), on which our method was assessed.

As seen in Fig. 3.4, 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 models trained on TrainData subsets in the course of cross validation (Figs. 3.4a-3.4c, Tables B.9 and B.10). 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 10,2685, see Table B.8) 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 3.2.6.6). The application of the stacked generalization to the final ensemble model results in an approximate 0.5%-3% increase in terms of AUC performance using different regions (Table B.10).

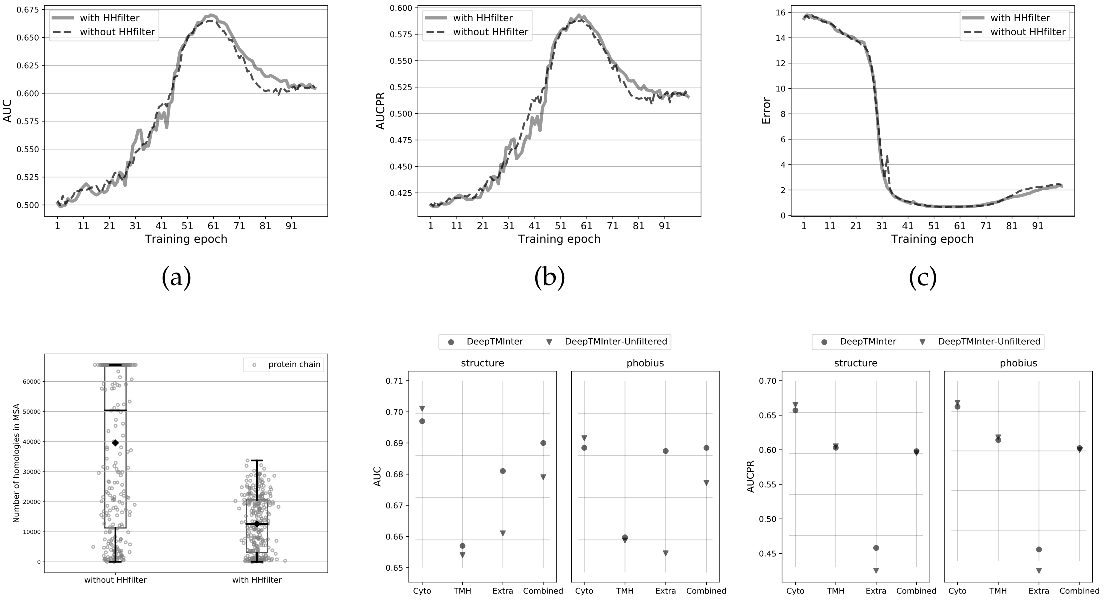


(a) (b) (c)

FIGURE 3.4: 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 3.2.6.3). 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.

3.3.2 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 3.2.4) (referred to as DeepTMInter). The performance of these two models is in general comparable (Figs. 3.5a-3.5c), 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) (Figs. 3.5e and 3.5f). 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) (Table B.11). Thus, significantly reducing alignment depth (Fig. 3.5d) allowed to speed up our method without sacrificing prediction performance.



(a) (b) (c)

(d) (e) (f)

FIGURE 3.5: 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).

3.3.3 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 3.2.5.2). 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. 3.6, Tables B.10 and B.12).

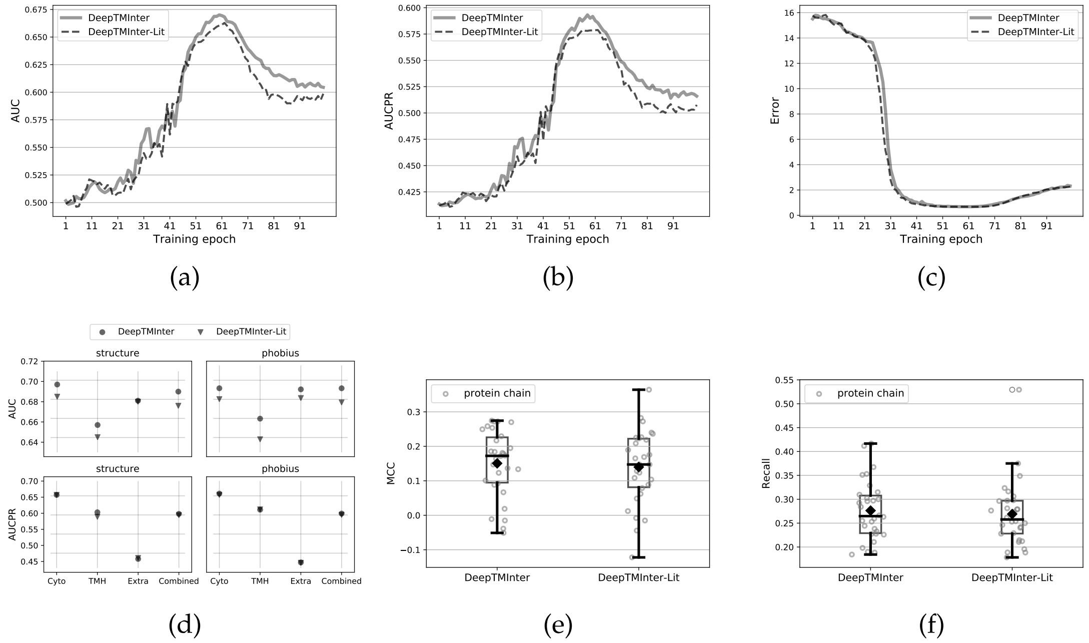
For instance, using Phobius-predicted combined regions on the IndepData dataset DeepTMInter achieves the AUC value of 0.690 (AUCPR=0.599) compared to 0.676 (0.595) of DeepTMInter-Lit (Tables B.10 and B.12). We assume that this gain in performance stems from the fact that the AAanalysis tool selects the representative scales of each kind and thus significantly reduces data redundancy, which is detrimental to learning algorithms (Mandal and Mukhopadhyay, 2013; Chormunge and Jena, 2018).

3.3.4 Performance comparison of DeepTMInter with the MBPred suite

We compared the prediction performance of DeepTMInter with the four underlying predictors (MBPredTM, MBPredCyto, MBPredExtra, and MBPredAll) in the MBPred suite and with the ensemble predictor MBPredCombined (see section 3.2.8). Due to the adoption of the early stopping strategy to prevent over-training (see section 3.2.6.5), our trained model achieved high prediction performance not only on the two previous test datasets (TestData and CompData), but also on the independent test dataset (IndepData) (Figs. 3.7-3.8 and Tables B.13-B.16).

3.3.4.1 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 (Tables B.13-B.14). Overall, DeepTMInter shows a significant improvement in terms of the AUC and AUCPR performance. For example, on the IndepData dataset our method gives the AUC value of 0.661



(a) (b) (c)

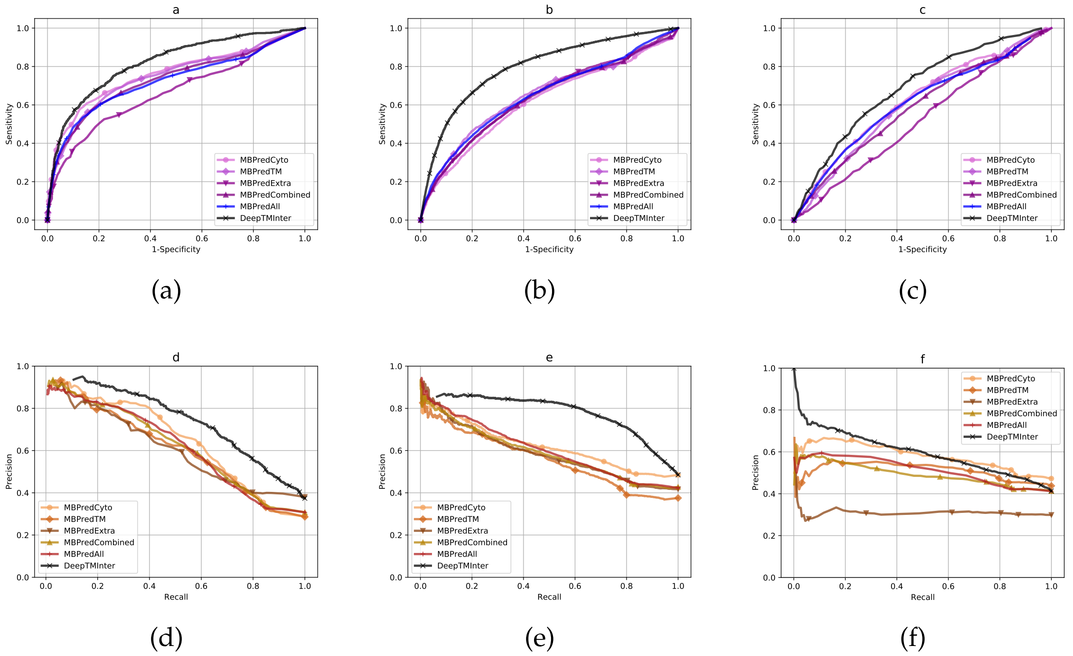
(d) (e) (f)

FIGURE 3.6: 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 3.2.6.6) 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.

(AUCPR=0.603), distinctly higher than 0.603 (0.513) of MBPredTM using structure-derived TMH regions. 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 (Zeng, Hönigschmid, and Frishman, 2019). Indeed, we 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. 3.7 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 achieves the highest AUC (0.793, 0.796, and 0.689) (Figs. 3.7a-3.7c) and AUCPR values (0.718, 0.738, and 0.598) (Figs. 3.7d-3.7f). 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.



(a) (b) (c)

(d) (e) (f)

FIGURE 3.7: Performance comparison between MBPred and DeepTMInter. (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.

3.3.4.2 Performance comparison using single-threshold measures

For proteins in all test datasets mean precision, recall, F1-score, MCC, and HL were calculated using entire combined structure-derived and Phobius-predicted regions (Tables B.15-B.16). Based on these performance measures DeepTMInter is also way ahead of the MBPred suite. 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 3.2.7) was used to assess the similarity between a set of actual labels and a set of predicted labels for sites in TM proteins (Tan, Steinbach, and Kumar, 2016; Fosso et al., 2018). 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. 3.8, 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. 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.

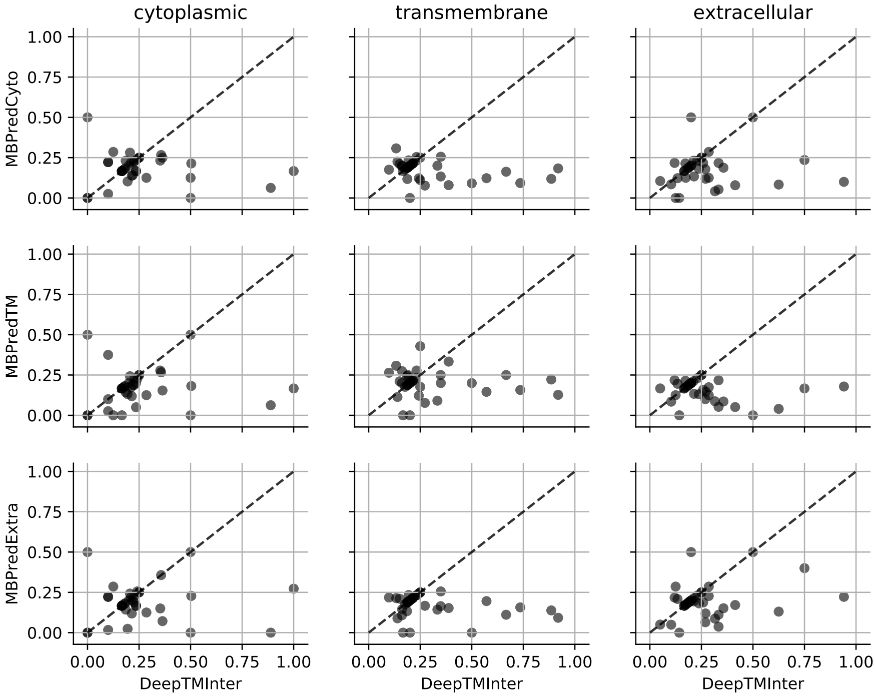
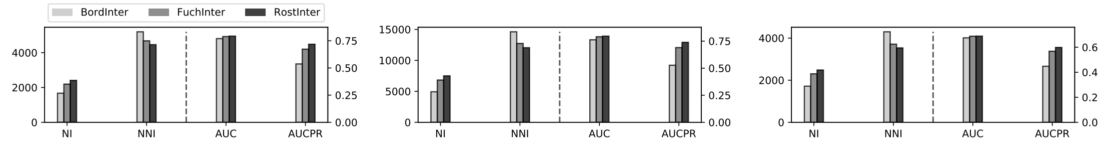


FIGURE 3.8: Comparison of JSCs (Jaccard similarity coefficients) between DeepTMInter and the three specialized MBPred predictors (MBPredTM, MBPredCyto, and MBPredExtra) on the TestData dataset. Each dot corresponds to one protein chain.

3.3.5 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 BordInter (Bordner, 2009), FuchInter (Fuchs, Kirschner, and Frishman, 2009), and RostInter (Hamp and Rost, 2012) residue contact definitions (Fig. 3.9 and Table B.17). 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. 3.9 and Table B.18). As expected, the number of residue contacts increases progressively with the spatial distance cutoff according to the BordInter (4Å), FuchInter (5.5Å), and RostInter (6Å) 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).



(a) (b) (c)

FIGURE 3.9: 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.

3.3.6 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. 3.10), disregarding alignment columns with more than 50% of gaps. In line with Bordner (Bordner, 2009) and our own previous work (Zeng, Hönigschmid, and Frishman, 2019), interaction sites 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 (*p*-value 2.39e-24, *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 (Lynch and Koshland, 1991; Riek et al., 1995).

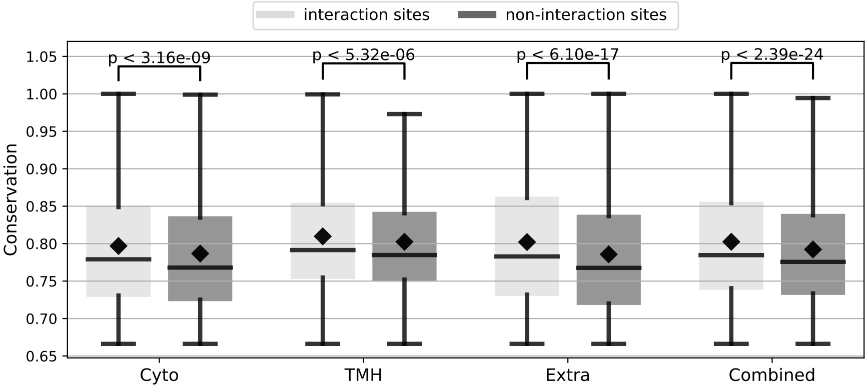
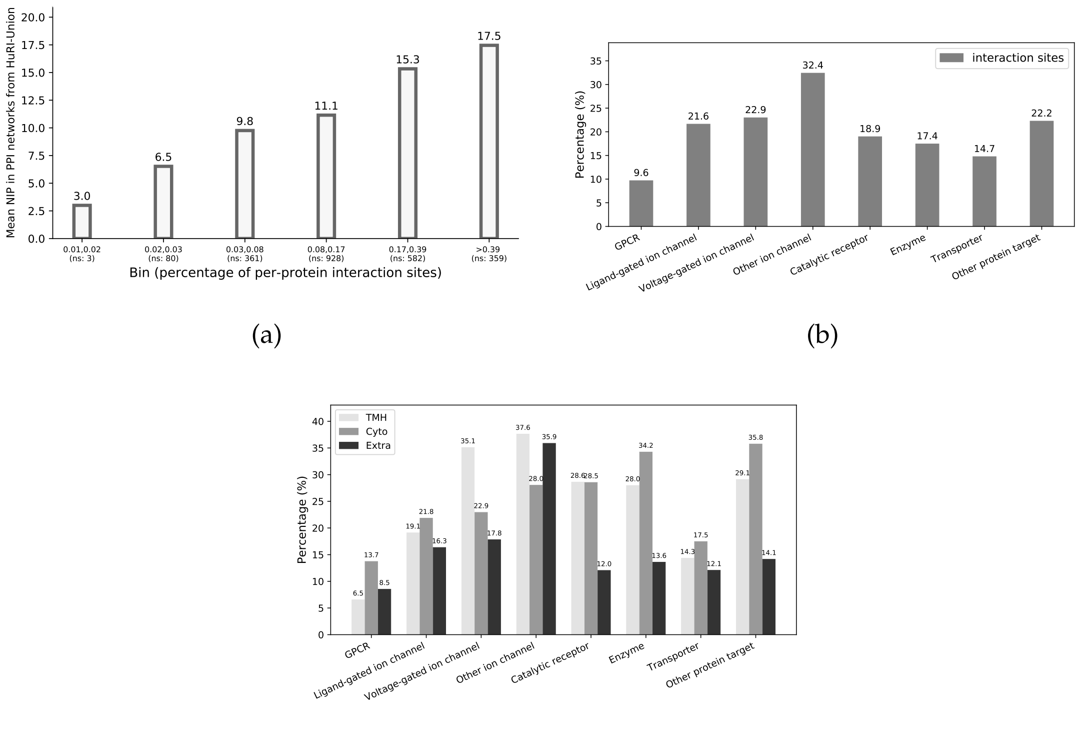


FIGURE 3.10: Evolutionary conservation of interaction sites across all the three test datasets (TestData, CompData, and IndepData) in the Cyto, TMH, Extra, and Combined (the full protein sequence) regions. Statistical significance of the difference between interaction and non-interaction sites is inferred by p-values obtained by t-Test.

3.3.7 Family-specific analysis of interaction sites and network connectivity in human transmembrane proteins

We investigated the relationship between the percentages of per-protein interaction sites predicted by DeepTMInter and the number of interaction partners on the human transmembrane PPI networks constructed using the HuRI-union database (see section 3.2.10). Using bins created by logarithm values, the number of interaction sites is directly proportional to the number of interaction partners across all human transmembrane proteins (Fig. 3.11a). For proteins in the three test datasets, the dependence of the interaction partners on the percentage of interaction sites is shown in Figs. B.3 and B.4. Additionally, in order to understand the relationship between the biological activities of proteins and their interaction patterns, we analyzed the average percentages of per-protein interaction sites in the eight major membrane protein families (see section 3.2.10) curated by the GtoPdb database (Figs. 3.11b and 3.11c, and Table B.19). Overall, ion channels (LGIC, VGIC, and other ion channel) account for the most abundant interaction sites per protein (Fig. 3.11b) with the highest percentages of 21.6%, 22.9%, and 32.4%, respectively. Interestingly, *Other ion channel* and *Other protein target* are the major families of the first two largest percentages of per-protein interaction sites. By ranking the percentages of 8 sub-families in *Other ion channel* (Fig. B.5), we found that orai channels (Liu et al., 2019), connexins, and pannexins (Molica et al., 2018) possess more than 60% and 40% of per-protein interaction sites, respectively, strongly contributing on to the high percentages.



(a)

(b)

(c)

FIGURE 3.11: Percentage of per-protein interaction sites in human transmembrane proteins. (a) shows the dependence of interaction partners (HuRI-union database, (Luck et al., 2020)) on percentages of per-protein interaction sites in all human transmembrane proteins. The number of interaction sites was equally divided into 6 bins according to the range of logarithm values. Logarithm binning is used to reveal the deeper significant trend and distribution behind data (Milojević, 2010; Wang et al., 2017). The mean number of interaction partners (NIPs) of human transmembrane proteins at each bin was evaluated. The percentage of per-protein interaction sites increases in ascending order of bin number. ns: number of sequences. (b) shows the average percentages of per-protein interaction sites in the full sequences with respect to eight major functional families. (c) shows the average percentages of per-protein interaction sites in the TMH, Cyto, and Extra regions with respect to eight major functional families.

Similarly, among 20 sub-families in *Other protein target* the largest three percentages of per-protein interaction sites (>40%) were displayed in other pattern recognition receptors, sigma receptors and abscisic acid receptor complex, respectively (Fig. B.9). Proteins in the three sub-families play an important role in signaling pathways (Ishikawa and Barber, 2008; Santiago et al., 2009; Aydar et al., 2002). The existence of these sub-families above essentially contributes to the high average percentage of per-protein interaction sites. *Other ion channels* and *other protein target* classes consist of proteins of the lowest average length (Fig. B.13). If proteins from these two classes and those from other classes have similar numbers of interaction sites, the proteins from the two classes are more likely to achieve higher percentages of interaction sites than those from other classes. *Other ion channel* reaches the maximum percentages of per-protein interaction sites in the TMH and the Extra regions (Fig. 3.11c and Figs. B.6-B.8), whereas *Other protein target* has the largest concentration of per-protein interaction site in the Cyto region (Figs. 3.11c and B.10-B.12). The enzyme family is highlighted by a high percentage of 34.2% on the Cyto region, which declares a functionally active role in intracellular activities.

3.3.8 Case studies

3.3.8.1 Human cardiac voltage-gated sodium channel

We present a case study for the assessment of the DeepTMInter prediction performance in the major interaction domains of human cardiac voltage-gated sodium channel – $Na_v1.5$ (encoded by gene SCN5A) (Grant, 2009). $Na_v1.5$ is crucial in mediating upstroke of the action potential (Schroeter et al., 2010; Rook et al., 2012). By pulling 19 interaction partner information from BioGRID, we found that the tail of the $Na_v1.5$ C-terminal is functionally important in Na^+ gating inactivation (Cormier et al., 2002). Four motifs, PY (sites: 1974-1976) (Luo et al., 2017), extended PY (sites: 1974-1980) (Rougier et al., 2005), SXV (sites: 2014-2016) (Gee et al., 1998), and IQ (sites: 1901-1927) (Chagot and Chazin, 2011), localized in the tail of the $Na_v1.5$ C-terminal, frequently interact with other proteins (see Table available at `https://data.mendeley.com/datasets/2t8kgwzp35`). Our results show that the interaction sites predicted by DeepTMInter give a better agreement to some of the experimentally established interfaces. For example, 5 out of 7 interaction sites were precisely detected in the extended PY motif. In addition, we also found that interactions between proteins can occur in the same interfaces, *e.g.*, three proteins encoded by genes NEDD4, NEDD4L, and WWP2 were discovered to interact with $Na_v1.5$ in the PY motif.

3.3.8.2 Comparison of predicted interfaces of testing proteins

Three example models from the TestData, CompData, and IndepData datasets are displayed in Figs. B.14-B.16, respectively.

3.4 Conclusions

We have developed a new deep learning approach, DeepTMInter, for sequence-based prediction of interaction sites in transmembrane proteins. DeepTMInter was first trained using deep residual neural networks integrated with 27 residual units, followed by further error reduction using a stacked generalization of ensemble of machine learning methods. We showed that by evaluating performance on an independent dataset <25% sequence-identical to the training dataset, DeepTMInter achieved the state-of-the-art performance with the overall highest AUC value of 0.689 and AUCPR value of 0.598 significantly better than MBPred previously established by our group. DeepTMInter revealed that among eight major functional families, the ion channel family of human transmembrane proteins were identified as the largest group to accommodate 25.6% interaction sites per protein. Our prediction strongly agreed to the experimentally validated interfaces of some functionally important motifs/domains in the human cardiac Na$_v$1.5 channel. Furthermore, analysis of interaction network connectivity based on the HuRI database discovered around 17 interaction partners per human transmembrane protein. Our findings also suggested that the number of interaction partners directly proportionally changed with the percentage of interaction sites in human transmembrane proteins.

3.5 Software availability

A repository of the standalone package of DeepTMInter was built at `https://github.com/2003100127/deeptminter`.

Chapter 4

Concluding remarks

The research presented in this thesis aims to decipher the intricate connection networks of residue-residue contacts and protein-protein interaction (PPI) sites in transmembrane proteins. The main contribution of this thesis is the integration of predicted contact maps at an intra-protein level and predicted interaction site potentials at an inter-protein level. The integration facilitates the construction of the whole interaction systems for transmembrane proteins to illuminate their biological roles in cellular activities. Deep residual neural networks have been shown to produce highly reliable and intelligible models for different biological applications. Aided by these recent advances, we have developed two novel methods for accurate prediction of residue contacts and interaction sites.

Our first deep-learning tool, termed DeepHelicon, described in *Chapter* 2, has been trained on 17,029,854 residue pairs, with each characterized by a 728-length feature vector. DeepHelicon has been shown to be powerful and resilient to those randomly-allocated and low sequence-identity proteins. We conclude that *i*) DeepHelicon is the most accurate method for large transmembrane proteins rich in helices and *ii*) using CONFOLD2 around 30% satisfactory transmembrane protein models can be guided by residue contacts predicted by DeepHelicon.

Our second deep-learning system, termed DeepTMInter, described in *Chapter* 3, has been designed to predict interaction sites in transmembrane proteins. By carrying out a thorough analysis of a set of 167 testing proteins, DeepTMInter has been confirmed to be more accurate than the previously best performing method, MBpred. Following up on the progress in prediction performance, we subsequently analyzed the network connectivity and the interaction-site occurrences of all human transmembrane proteins. We conclude that *i*) the percentage of interaction sites per human transmembrane protein is highly responsive to the number of its interaction partners and *ii*) human ion channels examined by DeepTMInter have the largest percentage of per-protein interaction sites.

Currently, one grand challenge that researchers are confronted with is the urgent need of new intelligent algorithms to cope with the rapidly growing volume of protein data. Apparently, residue contact prediction involves a huge amount of data that are extremely biased towards negative data samples. With more transmembrane proteins available, the emerging deep-learning techniques offer ample opportunities in more accurate prediction of residue contacts and interaction sites. In the future, since transmembrane proteins are disease-associated targets, it will be particularly interesting to discern disease-specific patterns in its PPI networks. We believe that the accurate results predicted by DeepHelicon and DeepTMInter can largely promote the understanding of transmembrane protein functions and their molecular mechanisms, which are important for the follow-up analysis in both academia and industry, such as drug development and disease therapy.

Appendix A

Tables

TABLE A.1: 165 α-helical TM protein chains in the TRAIN dataset.

PDB Codes							
1aig L	1h7c A	1jb0 A	1kqf C	1p49 A	1pw4 A	2a06 P	2ahy A
2axt A	2bhw A	2c3e A	2cfp A	2e74 B	2e74 A	2f93 B	2jkv A
2pri A	2vl1 A	2yev B	3a3y A	3abk B	3abk C	3abv D	3abv C
3aoa A	3aou A	3aqp A	3ayf A	3b44 A	3d31 C	3det A	3dh4 A
3egw C	3g6b A	3h90 A	3jyc A	3m71 A	3nym A	3qnq A	3tx3 A
3waj A	3wo6 A	3wvf A	3zcc A	4a01 A	4a2n B	4a82 A	4a97 A
4ain A	4al0 A	4aps A	4avm A	4aw6 B	4bbj A	4bem A	4bpd A
4cad C	4d1a A	4dji A	4dw0 A	4ev6 A	4ezc A	4f4c A	4fz0 A
4g7v S	4gd3 A	4gx0 A	4hkr A	4ikp A	4iu8 A	4jkv A	4jta B
4k1c A	4khz F	4m64 B	4mnd A	4phz A	4quv A	4rdq A	4u1w A
4wd7 A	4wis A	4ymk A	4zr1 A	5a1s A	5a43 A	5a44 A	5a63 B
5a63 C	5aex A	5aji A	5aww Y	5aym A	5azb A	5bw8 D	5bw8 C
5bzb A	5c65 A	5c6p A	5cgc A	5ckr A	5ctg A	5d0y A	5d3m D
5d91 A	5da0 A	5dir A	5djq A	5djq C	5doq B	5duo A	5ec5 A
5edl A	5egi A	5eiy A	5eke A	5er7 B	5ezm A	5fgn A	5gko A
5h1q A	5hwx A	5iji A	5irx A	5iwk A	5iws A	5jwy A	5khn B
5l8r G	5lki A	5lwy A	5m87 A	5mrw A	5n6h A	5nv9 A	5oge A
5ogl A	5sv0 B	5t4d A	5t77 A	5tcx A	5tj6 A	5u73 A	5ul0 A
5v7p A	5v8k A	5w3s A	5x5y F	5y78 A	5yi2 B	5z96 A	5zdh A
6b3j R	6bat A	6bcj B	6bhu A	6bml A	6bw5 A	6c96 A	6eti A
6ezn E	6ezn B	6ezn H	6ezn F	6ezn C			

TABLE A.2: 57 α-helical TM protein chains in the TEST dataset.

PDB Codes							
1jb0 L	2a06 C	2a65 A	2abm A	2acz C	2acz D	2axt B	2axt Z
2bs2 C	2zuq A	3abk A	3b4r A	3mp7 A	3o7p A	3tui A	3ux4 A
3wdo A	4a4m A	4bw5 A	4dnt A	4dxw A	4f35 B	4fc4 A	4he8 D
4he8 F	4j05 A	4kpp A	4mes A	4oqy A	4p79 A	4pgr A	4phz B
4phz K	4q2e A	4qtn A	4rp8 A	4ryi A	4tqu M	4xks A	4yms D
5a8e A	5b57 A	5c6n A	5doq A	5guf A	5guw B	5jki A	5kbw A
5l26 A	5o0t A	5x5y G	5xjj A	5xu1 M	6awf C	6awf D	6bar A
6cb2 A							

TABLE A.3: 44 α-helical TM protein chains in the PREVIOUS dataset.

PDB Codes							
1xqf A	2cfq A	2jln A	2nq2 A	2r6g F	2r6g G	2rh1 A	2w2e A
2wsc 2	2wsw A	2xq2 A	2yev A	2yvx A	2z73 A	2zxe A	2zy9 A
3b9w A	3c02 A	3ddl A	3eam A	3gd8 A	3gia A	3hd6 A	3k3f A
3kly A	3m7l A	3m73 A	3qe7 A	3rko L	3rvy A	3t9n A	3tij A
3ukm A	3usi A	3v5u A	4czb B	4hyg A	4ikw A	4m5b A	4q2g B
4r0c B	4twd A	4u1x C	4wd8 B				

TABLE A.4: Summary of residue contacts in the TRAIN (165 TM chains), PREVIOUS (44 TM chains), and TEST (57 TM chains) datasets.

Dataset	**Number of contacts**	**Number of non-contacts**	**Total**	**Contact *vs.* non-contacts** (%)
TRAIN	167216	16862638	17029854	0.99
PREVIOUS	45033	3757418	3802451	1.20
TEST	42493	3123828	3166321	1.36

TABLE A.5: Residue contact predictors benchmarked in this work.

Name	Typea	URL	Methodb	Datasetc
PSICOV	Local	http://bioinfadmin.cs.ucl.ac.uk/downloads/PSICOV/	ECA	P and T
EVfold	Local	http://packages.debian.org/freecontact	ECA	P and T
CCMpred	Local	https://github.com/soedinglab/CCMpred	ECA	P and T
Gaussian DCA	Local	https://github.com/carlobaldassi/GaussDCA.jl	ECA	P and T
plmDCA	Local	https://github.com/debbiemarkslab/plmc	ECA	P and T
Gremlin	Web	http://gremlin.bakerlab.org/	ECA	P and T
MetaPSICOV	Local	http://bioinfadmin.cs.ucl.ac.uk/downloads/MetaPSICOV/	ML	P and T
PconsC3	Web	http://pconsc3.bioinfo.se/	ML	P
DeepCov	Local	https://github.com/psipred/DeepCov	DL	P and T
Pconsc4	Local	https://github.com/ElofssonLab/PconsC4	DL	P and T
Membrain2	Web	http://www.csbio.sjtu.edu.cn/bioinf/MemBrain/	DL	P and T
DeepMetaPSICOV	Web	http://bioinf.cs.ucl.ac.uk/psipred/	DL	P and T

a) Type of installation. Local – downloadable and locally installable software package, Web – Web server.
b) Underlying algorithm. ECA – evolutionary coupling analysis, ML – machine learning, DL – deep learning.
c) Datasets used for benchmarking. P – PREVIOUS, T – TEST (see Section 2.2.1).

TABLE A.6: Information about no prediction results of predictors.

Dataset	Predictor	# of missing predictions	Missing TM chains	Reason
PREVIOUS	PconsC3	6	2yev A; 2zxe A; 3m73 A; 4czb B; 4q2g B; 4u1x C	Large proteins
	PconsC4	1	2zxe A	Large protein
	Membrain2	6	3m73 A; 4u1x C; 3tij A; 3ukm A; 3usi A; 4q2g B	Large proteins or no TM helices detected by their webserver
	DeepMetaPSICOV	7	2wsw A; 2xq2 A; 2yev A; 2zxe A; 3rko L; 3usi A; 4u1x C	Proteins with >500 amino acids limited by their webserver
TEST	PconsC4	1	2zxe A	Large protein
	DeepCov	1	4dnt A	Large protein
	Gremlin	2	4dnt A; 4phz K	Large protein and not enough sequences in MSA
	Membrain2	12	4a4m A; 2a65 A; 2acz C; 6awf C; 2axt B; 4dnt A; 2bs2 C; 4he8 D; 3o7p A; 4oqy A; 4xks A; 4pgr A	Large proteins or no TM helices detected by their webserver
	DeepMetaPSICOV	5	2a65 A; 3abk A; 4dnt A; 4dxw A; 4he8 F	Proteins with >500 amino acids limited by their webserver

TABLE A.7: Prediction performance at stage 1 on the PREVIOUS and TEST datasets for inter-helical residue contacts.

Metrics	**Threshold**	**Stage 1**	
		PREVIOUS	TEST
Precision	*L*	55.81	56.75
	L/2	69.03	69.93
	L/5	79.62	80.01
	L/10	83.91	84.14
Recall	*L*	38.51	41.61
	L/2	24.44	26.10
	L/5	11.19	12.24
	L/10	5.86	6.52
F1	*L*	43.43	46.48
	L/2	34.30	36.90
	L/5	19.08	20.81
	L/10	10.79	11.94
Fb	*L*	52.22	53.81
	L/2	55.71	57.76
	L/5	45.78	48.17
	L/10	32.86	35.18
MCC	*L*	44.10	46.49
	L/2	39.16	41.06
	L/5	28.66	30.12
	L/10	21.35	22.52

TABLE A.8: Prediction performance at 4 iterations of stage 2 on the PREVIOUS dataset for inter-helical residue contacts.

Metrics	**Threshold**	**Stage 2**			
		Iteration 1	Iteration 2	Iteration 3	Iteration 4
Precision	*L*	61.35	63.17	63.12	63.71
	L/2	75.13	76.23	77.20	77.18
	L/5	85.72	86.93	85.61	85.93
	L/10	90.07	90.09	89.46	88.73
Recall	*L*	41.76	42.88	42.87	43.30
	L/2	26.44	27.01	27.30	27.35
	L/5	12.32	12.45	12.48	12.53
	L/10	6.46	6.50	6.47	6.42
F1	*L*	47.68	49.05	49.04	49.51

	L/2	37.31	38.02	38.50	38.54
	L/5	20.91	21.15	21.11	21.17
	L/10	11.86	11.91	11.85	11.75
Fb	*L*	57.41	59.10	59.06	59.61
	L/2	60.65	61.62	62.41	62.41
	L/5	49.64	50.31	49.79	49.93
	L/10	35.71	35.76	35.56	35.27
MCC	*L*	48.52	49.97	49.95	50.45
	L/2	42.71	43.49	44.06	44.08
	L/5	31.23	31.64	31.37	31.48
	L/10	23.24	23.28	23.14	22.95

TABLE A.9: Prediction performance at 4 iterations of stage 2 on the TEST dataset for inter-helical residue contacts.

Metrics	**Threshold**	**Stage 2**			
		Iteration 1	Iteration 2	Iteration 3	Iteration 4
Precision	*L*	59.81	61.79	61.18	61.48
	L/2	74.17	75.14	74.57	74.95
	L/5	83.88	83.92	83.48	84.45
	L/10	87.61	86.41	86.74	87.27
Recall	*L*	43.93	45.52	45.07	45.11
	L/2	27.71	28.10	28.05	28.21
	L/5	12.82	12.88	12.78	12.95
	L/10	6.74	6.73	6.71	6.66
F1	*L*	49.05	50.74	50.25	50.42
	L/2	39.16	39.71	39.53	39.76
	L/5	21.80	21.88	21.73	22.02
	L/10	12.37	12.34	12.30	12.24
Fb	*L*	56.74	58.64	58.06	58.33
	L/2	61.28	62.10	61.67	62.00
	L/5	50.49	50.59	50.30	50.94
	L/10	36.59	36.32	36.30	36.32
MCC	*L*	49.20	50.98	50.46	50.62
	L/2	43.68	44.31	44.06	44.31
	L/5	31.62	31.70	31.50	31.91
	L/10	23.44	23.24	23.25	23.27

TABLE A.10: Performance of the final ensemble predictor on the PREVIOUS and TEST datasets for inter-helical residue contacts.

Metrics	**Threshold**	**Ensemble**	
		PREVIOUS	TEST
Precision	*L*	63.69	62.13
	L/2	77.84	76.16
	L/5	87.42	84.98
	L/10	91.33	87.44
Recall	*L*	43.26	45.75
	L/2	27.52	28.55
	L/5	12.70	13.05
	L/10	6.58	6.76
F1	*L*	49.47	51.01
	L/2	38.82	40.30
	L/5	21.48	22.17
	L/10	12.06	12.40
Fb	*L*	59.59	58.95
	L/2	62.93	62.96
	L/5	50.74	51.24
	L/10	36.25	36.58
MCC	*L*	50.41	51.26
	L/2	44.43	44.98
	L/5	31.99	32.12
	L/10	23.60	23.44

TABLE A.11: Performance of 3D modelling for proteins in the PREVIOUS dataset.

PDB ID	TM-score						Ca-RMSD					
	L/5			*L*/2			*L*/5			*L*/2		
	DHC	DMP-ih	DMP-all	DHC	DMP-ih	DMP-all	DHC	DMP-ih	DMP-all	DHC	DMP-ih	DMP-all
1xqfA	0.207	0.227	0.222	0.300	0.228	0.239	19.935	19.643	22.501	17.369	21.504	17.732
2cfqA	0.440	0.270	0.277	0.537	0.606	0.558	25.317	27.843	28.797	14.276	12.994	14.335
2jlnA	0.233	0.180	0.259	0.216	0.197	0.359	45.554	51.067	26.193	36.053	45.055	16.338
2nq2A	0.172	0.155	0.157	0.165	0.167	0.167	20.971	26.154	25.373	21.445	21.361	22.438
2r6gF	0.126	0.122	0.117	0.137	0.110	0.131	-	-	-	-	-	-
2r6gG	0.182	0.158	0.131	0.212	0.156	0.173	31.611	37.428	48.44	19.464	36.783	32.111
2rh1A	0.122	0.113	0.113	0.127	0.116	0.120	47.396	52.446	50.399	46.118	50.925	49.105
2w2eA	0.202	0.189	0.221	0.226	0.209	0.204	17.181	23.708	16.557	16.66	20.579	16.586
2wsc2	0.134	0.136	0.139	0.134	0.134	0.140	52.39	56.942	23.282	48.837	53.968	25.24
2yvxA	0.136	0.129	0.109	0.146	0.135	0.150	-	-	-	-	-	-
2z73A	0.249	0.220	0.271	0.291	0.312	0.327	19.119	28.399	25.408	20.606	17.577	17.644
2zy9A	0.125	0.139	0.099	0.135	0.116	0.120	-	-	-	-	-	-
3b9wA	0.233	0.231	0.177	0.254	0.274	0.266	32.432	36.849	39.854	28.477	20.849	30.278
3c02A	0.233	0.231	0.250	0.254	0.239	0.298	19.156	22.071	17.123	15.401	18.655	11.402
3ddlA	0.390	0.216	0.329	0.387	0.443	0.436	9.523	22.063	13.19	9.754	9.043	9.28
3eamA	0.179	0.187	0.126	0.182	0.185	0.205	-	-	-	-	-	-
3gd8A	0.173	0.180	0.195	0.200	0.179	0.180	21.982	20.435	17.183	19.34	21.913	17.739
3giaA	0.269	0.358	0.269	0.417	0.539	0.484	31.26	16.959	35.143	25.282	12.201	17.051

3hd6A	0.154	0.164	0.177	0.177	0.172	0.165	43.158	31.83	31.406	41.231	22.676	34.737
3k3fA	0.271	0.205	0.250	0.358	0.397	0.459	22.513	31.714	35.64	20.205	15.571	14.065
3klyA	0.154	0.169	0.165	0.184	0.180	0.181	24.297	21.669	21.143	23.332	23.601	22.31
3m73A	0.543	0.368	0.202	0.718	0.784	0.742	11.55	25.59	69.535	7.53	5.504	5.639
3m7lA	0.283	0.191	0.182	0.353	0.318	0.301	14.08	28.178	23.186	12.563	13.246	12.651
3qe7A	0.248	0.177	0.215	0.500	0.377	0.268	34.863	29.133	27.604	11.411	15.029	24.619
3rvyA	-	-	-	-	-	-	-	-	-	-	-	-
3t9nA	0.128	0.160	0.160	0.168	0.141	0.185	78.015	77.013	39.835	75.779	75.783	27.707
3tijA	0.209	0.195	0.189	0.285	0.270	0.272	27.801	38.21	28.933	21.629	21.361	17.839
3ukmA	0.199	0.193	0.158	0.199	0.190	0.187	24.511	32.601	35.137	24.585	27.233	23.139
3v5uA	0.233	0.307	0.244	0.357	0.424	0.381	17.249	12.561	25.495	12.161	10.246	11.823
4czbB	0.329	0.297	0.264	0.503	0.662	0.614	26.103	23.798	25.623	12.803	7.676	7.404
4hygA	0.188	0.193	0.187	0.194	0.199	0.208	29.576	20.791	29.314	28.738	19.887	19.894
4ikwA	0.334	0.262	0.225	0.465	0.532	0.462	31.255	28.139	31.715	18.338	15.221	17.85
4m5bA	0.486	0.395	0.371	0.642	0.498	0.539	16.811	14.031	15.997	7.635	11.12	11.887
4q2gB	0.152	0.152	0.161	0.176	0.169	0.160	24.963	30.196	26.23	21.191	21.609	20.623
4r0cB	0.140	0.127	0.167	0.161	0.145	0.172	38.891	43.523	33.312	28.157	30.823	26.984
4twdA	0.135	0.148	0.153	0.144	0.126	0.162	-	-	-	-	-	-
4wd8B	0.149	0.154	0.161	0.162	0.162	0.190	34.733	36.584	30.36	36.751	32.8	25.731

* DMP-ih: DeepMetaPSICOV for inter-helical contacts; DMP-all: DeepMetaPSICOV for all contacts; DHC: DeepHelicon; ‘-’ no output returned.

TABLE A.12: Performance of 3D modelling for proteins in the TEST dataset.

PDB ID	TM-score						Ca-RMSD					
	L/5			*L/2*			*L/5*			*L/2*		
	DHC	**DMP-ih**	**DMP-all**	**DHC**	**DMP-ih**	**DMP-all**	**DHC**	**DMP-ih**	**DMP-all**	**DHC**	**DMP-ih**	**DMP-all**
1jb0L	0.277	0.312	0.295	0.286	0.333	0.335	13.637	13.050	12.42	13.802	12.959	13.149
2a06C	0.161	0.164	0.150	0.194	0.192	0.175	29.611	27.234	33.008	23.908	19.785	21.312
2abmA	0.462	0.439	0.590	0.566	0.481	0.757	14.016	17.308	6.686	10.832	14.082	5.386
2aczC	0.62	0.612	0.653	0.578	0.593	0.551	5.63	9.187	11.0	5.196	7.297	8.27
2aczD	0.401	0.346	0.372	0.379	0.351	0.385	6.191	6.982	7.954	6.778	7.423	6.519
2axtB	0.175	0.193	0.157	0.182	0.223	0.204	-	-	-	-	-	-
2axtZ	0.620	0.748	0.746	0.560	0.725	0.730	3.032	2.281	2.801	3.295	2.946	2.994
2bs2C	0.408	0.508	0.496	0.580	0.620	0.542	15.88	14.126	16.082	13.304	13.882	12.703
2zuqA	0.174	0.203	0.169	0.172	0.202	0.199	16.163	14.598	15.558	16.69	16.348	15.403
3b4rA	0.219	0.202	0.266	0.243	0.275	0.356	30.493	31.040	26.116	23.1	18.715	10.855
3mp7A	0.069	0.069	0.065	0.078	0.072	0.070	12.867	11.139	12.229	10.16	10.010	10.987
3o7pA	0.064	0.064	0.053	0.059	0.065	0.064	4.542	4.623	6.37	8.225	4.462	4.809
3tuiA	0.576	0.417	0.390	0.678	0.568	0.543	9.664	24.544	24.433	5.962	11.238	17.218
3ux4A	0.208	0.178	0.185	0.226	0.215	0.224	5.948	8.072	6.63	4.875	4.642	4.644
3wdoA	0.258	0.259	0.305	0.255	0.323	0.352	45.266	46.697	44.883	38.625	47.083	44.634
4a4mA	0.404	0.384	0.397	0.493	0.451	0.634	22.295	26.141	30.607	21.44	21.000	18.72
4bw5A	0.114	0.126	0.106	0.117	0.109	0.104	18.981	18.941	13.946	15.49	15.303	16.489
4f35B	0.058	0.059	0.057	0.064	0.059	0.059	5.824	5.335	5.86	4.046	5.031	6.435

4fc4A	0.329	0.388	0.275	0.485	0.548	0.719	17.664	17.415	21.295	14.659	12.638	9.453
4he8D	0.352	0.344	0.338	0.433	0.344	0.427	45.277	43.275	36.846	10.923	14.505	15.628
4j05A	0.151	0.119	0.113	0.143	0.147	0.124	15.505	31.846	32.498	16.046	16.423	24.228
4kppA	0.245	0.295	0.241	0.327	0.460	0.299	29.574	26.649	35.238	24.992	13.721	23.764
4mesA	0.260	0.223	0.221	0.265	0.257	0.254	11.925	17.363	13.79	10.314	10.608	10.034
4oqyA	0.153	0.142	0.222	0.128	0.122	0.257	77.251	77.379	29.432	74.829	75.483	18.239
4p79A	0.189	0.163	0.188	0.198	0.168	0.208	20.743	38.259	35.324	18.925	20.299	16.405
4pgrA	0.256	0.261	0.207	0.323	0.281	0.312	14.128	18.765	24.291	10.853	12.672	11.755
4phzB	0.231	0.234	0.208	0.197	0.226	0.203	44.581	22.908	25.648	25.471	23.049	21.652
4phzK	0.174	0.16	0.190	0.168	0.210	0.172	23.232	26.249	24.55	23.545	19.336	19.396
4q2eA	0.175	0.166	0.168	0.187	0.163	0.167	24.257	30.745	27.784	31.345	20.370	17.957
4qtnA	0.078	0.074	0.072	0.081	0.071	0.071	3.346	4.155	4.113	3.46	4.530	4.253
4rp8A	0.123	0.129	0.136	0.158	0.126	0.203	62.71	64.018	45.147	37.138	61.295	31.819
4ryiA	0.307	0.293	0.261	0.325	0.359	0.344	23.618	27.617	36.111	20.76	7.259	21.525
4tquM	0.091	0.076	0.085	0.096	0.084	0.079	5.464	10.668	6.517	4.927	6.200	9.528
4xksA	0.487	0.418	0.379	0.743	0.632	0.719	11.315	17.000	28.452	5.97	8.045	7.277
4ymsD	0.639	0.566	0.343	0.683	0.719	0.572	7.239	11.547	25.428	6.956	7.623	17.698
5a8eA	0.213	0.207	0.202	0.214	0.224	0.215	19.476	23.045	29.272	20.678	19.799	20.379
5b57A	0.105	0.089	0.089	0.084	0.086	0.086	17.72	19.189	19.355	20.708	18.621	21.029
5c6nA	0.473	0.345	0.241	0.521	0.540	0.571	11.717	21.894	50.461	9.924	9.806	8.646
5doqA	0.606	0.347	0.291	0.626	0.476	0.547	18.002	33.709	38.768	15.271	28.745	15.774
5gufA	0.236	0.232	0.212	0.245	0.267	0.273	17.867	17.617	16.578	16.522	10.647	9.578

5guwB	0.249	0.167	0.147	0.308	0.278	0.263	20.542	33.795	51.096	14.888	16.647	17.971
5jkiA	0.479	0.386	0.310	0.525	0.489	0.540	40.024	22.425	25.818	22.62	22.183	12.868
5kbwA	0.559	0.397	0.325	0.557	0.536	0.524	7.299	15.343	13.633	5.624	6.739	7.416
5l26A	0.283	0.219	0.155	0.325	0.310	0.230	15.566	22.227	29.001	14.851	14.234	14.924
5o0tA	-	-	-	-	-	-	-	-	-	-	-	-
5x5yG	0.312	0.289	0.214	0.339	0.303	0.240	34.426	38.934	15.47	35.823	35.121	12.167
5xjjA	0.177	0.157	0.151	0.189	0.173	0.195	26.301	29.413	30.35	25.697	26.305	25.962
5xu1M	0.118	0.108	0.113	0.113	0.113	0.113	8.588	9.977	9.567	6.977	7.330	7.946
6awfC	0.462	0.426	0.475	0.483	0.397	0.429	9.177	12.396	13.849	9.673	8.465	9.533
6awfD	0.294	0.367	0.377	0.344	0.424	0.382	32.282	8.163	7.836	8.496	6.331	9.608
6barA	0.221	0.179	0.166	0.228	0.220	0.209	12.008	18.774	20.528	10.43	10.328	11.077
6cb2A	0.274	0.269	0.320	0.322	0.255	0.470	24.066	26.043	18.572	19.461	26.068	8.342

* DMP-ih: DeepMetaPSICOV for inter-helical contacts; DMP-all: DeepMetaPSICOV for all contacts; DHC: DeepHelicon; ‘-’ no output returned.

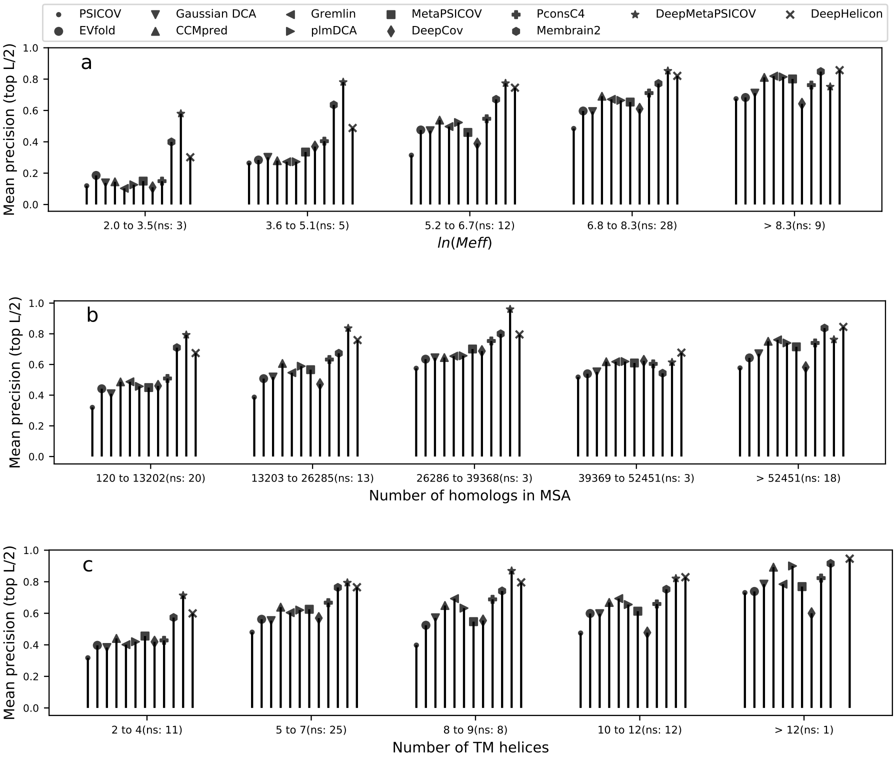


FIGURE A.1: Mean precision of top $L/2$ inter-helical contact predictions on the TEST dataset with respect to *ln(Meff)* (a), number of all homologs in MSA (b), and number of TM helices (c). ns: the number of sequences.

Appendix B

Tables

TABLE B.1: 101 α-helical TM protein chains in CompData dataset used in MBpred.

PDB Codes							
1fft A	1fft B	1fft C	1h2s A	1jb0 A	1jb0 F	1jb0 I	1jb0 K
1kf6 C	1kf6 D	1kqf B	1kqf C	1lgh A	1lgh B	1lnq A	1m56 B
1nek C	1nek D	1nkz A	1ots A	1q16 C	1q90 A	1q90 B	1q90 G
1rh5 A	1rh5 B	1rzh L	1rzh M	1s5l B	1s5l C	1s5l D	1s5l E
1s5l I	1s5l J	1s5l K	1s5l L	1s5l M	1s5l T	1s5l X	1s5l Z
1v54 I	1v54 J	1v54 K	1v54 L	1v54 M	1vf5 B	1vf5 D	1vf5 F
1xl4 A	1xme A	1xme B	1yew A	1yew B	1yew C	1zcd A	2bhw A
2fyu E	2fyu G	2fyu K	2h88 C	2h88 D	2hyd A	2ih3 C	2iub A
2nq2 A	2nwl A	2o01 G	2o01 H	2o01 I	2o01 J	2o01 L	2oar A
2r6g F	2rdd B	2vl0 A	2vv5 A	2yvx A	3cx5 C	3cx5 D	3cx5 H
3eam A	1jb0 M	1jb0 X	1m56 C	1m56 D	1q90 N	1q90 R	1s5l F
1s5l H	1v54 D	1v54 G	1vf5 G	1vf5 H	2bl2 A	2bs2 C	2j8s A
2j58 A	2q67 A	2qts A	3cx5 I	3d31 C			

TABLE B.2: 36 α-helical TM protein chains in TEST dataset used in MBpred.

PDB Codes							
3jcu H	3jcu I	3jcu T	3jcu W	3jcu X	4y28 K	5azd A	5b0w A
5b1a L	5b1a M	5b57 A	5b5e A	5b5e M	5b5e T	5b5e Z	5bn2 A
5djq N	5eg1 A	5eiy A	5fl7 K	5hv9 A	5i32 A	5jje B	5jnq A
5mkk A	5mrw C	5mrw D	5ul7 A	5x3x q	5x5y G	5kaf Y	5l22 B
5bqg A	5c2t D	5b1a J	5b1a K				

TABLE B.3: 301 α-helical TM protein chains in the TrainData dataset.

PDB Codes							
1aig L	1bcc E	1be3 G	1dxr H	1ezv G	1fx8 A	1h2s C	1kf6 O
1kf6 P	1kpl B	1occ K	1occ L	1zcd C	2axt E	2j58 B	2nq2 B
1kqf E	1kqf F	1lgh D	1m57 B	1nek G	1nek H	1occ D	1occ I
1ocr J	1orq C	1q90 E	1qle C	1sqq K	1vf5 Q	1w5c D	1xl6 A
2b6p A	2bl2 B	2bs2 F	2e74 A	2e74 B	2fyn B	2h8a A	2hyd B
2nrf A	2nuu A	2onk C	2q72 A	2qpd A	2uui A	2vpw C	2vr0 F
2w1p A	2wpd J	3hd7 B	3jcu w	3oax A	3odu A	3rko N	3tui A
2wsw A	2xq2 A	2yev A	2yev B	2zy9 B	3b9y A	3egw C	3hd7 A
3lut B	3m6e A	3m73 A	3mk7 A	3mk7 B	3mk7 C	3mp7 A	3mp7 B
3puv F	3q7k C	3qbg A	3qe7 A	3qnq A	3rko J	3rko L	3rko M
3ukm A	3vou A	3w5a A	3wgu D	4a01 A	4bem J	4bpd A	4c9q B
4cz8 A	4dnt A	4huq T	4i0u B	4o93 B	4phz B	4tnw A	4u2p B
4dxw B	4ev6 E	4gd3 A	4gd3 Q	4gx2 B	4hg6 A	4hkr A	4huq S
4iff A	4ire B	4jkv A	4lep A	4mnd A	4n7w A	4o6m B	4o7g B
4pi0 A	4pl0 B	4qnd A	4qtn C	4r0c A	4rdq A	4rp9 A	4ryi A
4u4w A	4uuj C	5dwy B	5ec5 A	5jnq B	5l2a A	5oy0 K	5sv0 A
4wis A	4xig M	4xig N	4xu4 A	4ymk A	4yzi A	4z90 A	5a1s A
5a63 C	5a63 D	5aww Y	5bz3 A	5ctg B	5doq A	5doq B	5dqq A
5eik A	5eiy B	5f1c B	5f8u A	5fvn A	5h1q A	5iji A	5j4i A
5lil B	5lwe B	5m94 A	5mrw A	5mrw B	5nik A	5nkq A	5oy0 A
5tj6 A	5uen A	5zji L	5zx5 A	6cnm A	6coy A	6f34 A	6fv8 A
5ul7 B	5v2c b	5v2c c	5v8k A	5vre A	5w3s A	5x3x m	5x5y F
5xan B	5xnl 4	5xu1 M	5y78 A	5ys3 A	5z96 A	5zdh A	5zji H
6btm A	6btm C	6btm F	6bvg A	6bwj B	6c5w A	6c96 A	6cjt A
6d0j B	6djz B	6e3y R	6ezn A	6ezn B	6ezn E	6ezn F	6ezn G
6g2j J	6g2j N	6hu9 g	6hum L	6idf B	6idf E	6k7g C	6k7l A
6g2j Y	6g2j Z	6g2j a	6g2j b	6g2j g	6g2j h	6g2j i	6g2j j
6g2j k	6g2j l	6h2f H	6h5a B	6hbu A	6hcy A	6hd8 B	6hu9 J
6hum P	6hwh A	6hwh L	6hwh N	6hwh b	6i1z B	6i8w A	6idf A
6irs A	6itc B	6iu3 A	6iv3 A	6j5i 8	6j5i b	6j5t C	6j8g B
6kkr A	6kls C	6qti A	6qum N	6rx4 B	6sem D	6v4j A	6vja C
6lum D	6lum G	6lyp A	6m17 A	6m18 B	6m96 B	6mgv A	6mit C
6mit G	6mrt A	6nf4 A	6np0 A	6oht A	6oly B	6p25 A	6p2j A
6peq F	6pl6 A	6pl6 B	6pqp A	6pw5 B	6qp6 A	6qq6 A	6qsk E
6r7x A	6rd4 3	6rd4 9	6rfq 6	6rfq X	6rfq g	6rfq i	6rfq j
6sp2 A	6su4 A	6t15 e	6t9o A	6u9w A	6uqf A	6v00 A	6v1q A

6vtk A	6vwk a	6wbf A	6wc9 A	6wej A	6wiv B	6wqz A	6ww7 A
7bvc B	6pe4 A	6pe4 E	6ww7 C	6ww7 F			

TABLE B.4: 30 α-helical TM protein chains in the IndepData dataset.

PDB Codes							
6e3y E	6rfq S	6t0b m	5eke C	6wdn E	4tsy A	6rfq J	5guf A
6uiw A	6hwh W	6ww7 E	6g2j f	3vr8 D	5ndc B	6li9 A	6ezn H
4rfs S	4pi2 C	6csm A	3rko A	6o7t h	6j5j f	6cxh C	4kjs A
6btm D	3udc A	3pux G	6kls B	6rd4 6	6ezn C		

TABLE B.5: Amino acid physicochemical scales collected from literature.

Scale name	**Type**	**Length**	**Source**
positive	discrete	3	Barnes and Gray, 2003
negative		2	
charged		4	
polar		3	
aliphatic		2	
aromatic		2	
hydrophobic		4	
small		3	
active	continuous	1	
weight		1	Edition, 2018
pI		1	
pka		1	
pkb		1	
hydrophobicity		1	Argos et. al., 1982
hydration		1	
free energy of transfer		1	
volume		1	Grantham, 1974
polarity		1	
hydrophilicity		1	Hopp and Woods, 1981
total	-	**34**	-

TABLE B.6: 34 representative amino acid physicochemical scales generated by the AAanalysis tool.

Scale id	Category	Sub-category
JANJ780102	Accessible Surface	Buried
JOND920101	Composition	General AA Composition
CHAM830101	Conformation	Coil
QIAN880123	Conformation	Extended (Beta-Sheet C-terminal)
KANM800102	Conformation	Extended (Beta-Sheet)
KANM800101	Conformation	Helix
QIAN880108	Conformation	Helix
FINA910103	Conformation	Helix (C-terminal inside)
RICJ880113	Conformation	Helix (C-terminal inside)
AURR980118	Conformation	Helix (C-terminal outside)
RICJ880116	Conformation	Helix (C-terminal outside)
QIAN880112	Conformation	Helix (C-terminal)
PALJ810108	Conformation	Helix (N-terminal inside)
AURR980105	Conformation	Helix (N-terminal N-cap
RICJ880103	Conformation	Helix (N-terminal N-cap)
RICJ880106	Conformation	Helix (N-terminal)
CHOP780214	Conformation	Turn (C-terminal)
CHOP780215	Conformation	Turn (C-terminal)
COSI940101	Energy	Electron-ion Interaction Potential
MUNV940105	Energy	Free Energy (Extended)
VASM830102	Energy	Free Energy (Extended)
OOBM850104	Energy	Non-bonded Energy per Atom
MIYS990104	Energy	Partition Energies
CHAM820101	Energy	Polarizability
RADA880102	Energy	Transfer Free Energy (TFE)
SIMZ760101	Energy	Transfer Free Energy (TFE)
WILM950104	Polarity	Hydrophobicity
MEEJ800101	Polarity	Hydrophobicity
JOND750102	Polarity	pK-C
ENGD860101	Polarity	Polarity (Hydrophilicity)
PONP800102	Polarity	Surrounding Hydrophobicity
KARS160108	Shape	Graph-model based
KRIW710101	Structure-Activity	Side Chain Interaction
KRIW790102	Structure-Activity	Side Chain Interaction

TABLE B.7: Summary of input features.

Feature	Length of vector	Window size	Final feature number
Amino acid representation	20	3	60
Amino acid property	34	9	306
Amino acid composition	20	3	60
Evolutionary profile	21	9	189
Entropy	1	3	3
Conservation	1	3	3
Relative position	1	3	3
Transmembrane topology	3	3	9
Evolutionary coupling ratio (ECR)			27
mutual information	1	9	
EVfold	1	9	
Gaussian DCA	1	9	
Total	-	-	**660**

TABLE B.8: Number of non-interacting and interacting amino acid residues in the full set and the 5 cross validation sets of the TrainData dataset used for training.

Dataset	Number of protein chains	Number of non-interacting amino acid residues	Number of interacting amino acid residues	Number of all amino acid residues
The full set	301	74104	28581	102685
No.1 cv set	240	58745	23346	82091
No.2 cv set	240	59121	22560	81681
No.3 cv set	240	58295	22934	81229
No.4 cv set	240	58638	22813	81451
No.5 cv set	240	58356	23316	81672

Note: cv: cross validation. The 5 cross validation sets are also available alongside the TrainData list on Mendeley at `https://xxx`.

TABLE B.9: AUC and AUCPR values of models obtained from 5 cross validations on validation data.

Model	Criterion	Structure				Phobius			
		Cyto	TMH	Extra	Combined	Cyto	TMH	Extra	Combined
CV 1	AUC	0.667	0.652	0.670	0.662	0.667	0.657	0.672	0.662
CV 2		0.635	0.607	0.657	0.640	0.631	0.619	0.662	0.640
CV 3		0.642	0.606	0.658	0.642	0.654	0.596	0.653	0.643
CV 4		0.661	0.640	0.648	0.656	0.667	0.633	0.654	0.656
CV 5		0.642	0.614	0.650	0.643	0.643	0.631	0.641	0.643
CV 1	AUCPR	0.620	0.594	0.451	0.563	0.612	0.613	0.442	0.563
CV 2		0.586	0.572	0.420	0.543	0.575	0.593	0.418	0.544
CV 3		0.606	0.552	0.444	0.553	0.607	0.568	0.427	0.553
CV 4		0.603	0.591	0.417	0.551	0.598	0.601	0.414	0.551
CV 5		0.591	0.549	0.424	0.540	0.587	0.585	0.409	0.540

Note: cross validation.

TABLE B.10: AUC and AUCPR values of models trained on the full TrainData set on validation data.

Model	Criterion	Structure				Phobius			
		Cyto	TMH	Extra	Combined	Cyto	TMH	Extra	Combined
Round 1	AUC	0.672	0.649	0.684	0.678	0.684	0.642	0.681	0.678
Round 2		0.683	0.647	0.680	0.681	0.689	0.651	0.667	0.681
Round 3		0.681	0.632	0.684	0.675	0.688	0.639	0.675	0.675
Round 4		0.676	0.643	0.668	0.673	0.683	0.640	0.664	0.673
Round 5		0.676	0.636	0.682	0.672	0.686	0.636	0.679	0.673
SG (Ensemble)		**0.689**	**0.661**	**0.688**	**0.689**	**0.697**	**0.657**	**0.681**	**0.690**
Round 1	AUCPR	0.650	0.594	0.458	0.594	0.659	0.603	0.446	0.594
Round 2		0.652	0.591	0.458	0.593	0.656	0.607	0.442	0.593
Round 3		0.663	0.578	0.476	0.598	0.663	0.608	0.451	0.599
Round 4		0.655	0.582	0.436	0.587	0.656	0.596	0.422	0.587
Round 5		0.654	0.589	0.438	0.588	0.660	0.606	0.428	0.589
SG (Ensemble)		**0.657**	**0.603**	**0.458**	**0.598**	**0.661**	**0.611**	**0.447**	**0.599**

Note: SG: stacked generalization.

TABLE B.11: AUC and AUCPR values of DeepTMInter-Unfiltered.

Model	Criterion	Structure				Phobius			
		Cyto	TMH	Extra	Combined	Cyto	TMH	Extra	Combined
SG (Ensemble)	AUC	0.692	0.660	0.656	0.678	0.701	0.654	0.661	0.679
SG (Ensemble)	AUCPR	0.665	0.605	0.425	0.595	0.667	0.615	0.415	0.596

TABLE B.12: AUC and AUCPR values of DeepTMInter-Lit.

Predictor	**Criterion**	**Structure**				**Phobius**			
		Cyto	**TMH**	**Extra**	**Combined**	**Cyto**	**TMH**	**Extra**	**Combined**
SG (Ensemble)	AUC	0.679	0.642	0.680	0.676	0.685	0.645	0.680	0.676
SG (Ensemble)	AUCAP	0.657	0.589	0.461	0.594	0.657	0.612	0.446	0.595

Note that DeepTMInter-Lit was trained by combining the amino acid physiochemical scales randomly collected from references and other features.

TABLE B.13: AUC values of predictors on the TestData, CompData, and IndepData datasets.

Predictor	**Dataset**	**Structure**				**Phobius**			
		Cyto	**TMH**	**Extra**	**Combined**	**Cyto**	**TMH**	**Extra**	**Combined**
MBPredCyto	TestData	0.760	0.605	0.626	-	0.688	0.625	0.629	-
MBPredTM		0.612	0.753	0.594	-	0.594	0.714	0.597	-
MBPredExtra		0.581	0.581	0.685	-	0.628	0.614	0.644	-
MBPredAll		0.745	0.724	0.672	0.721	0.709	0.720	0.674	0.704
MBPredCombined		-	-	-	0.732	-	-	-	0.682
DeepTMInter		**0.807**	**0.820**	**0.738**	**0.793**	**0.810**	**0.827**	**0.721**	**0.794**
MBPredCyto	CompData	0.618	0.578	0.591	-	0.615	0.614	0.590	-
MBPredTM		0.571	0.650	0.545	-	0.569	0.669	0.545	-
MBPredExtra		0.585	0.576	0.643	-	0.581	0.611	0.640	-
MBPredAll		0.656	0.669	0.635	0.651	0.651	0.673	0.641	0.654
MBPredCombined		-	-	-	0.635	-	-	-	0.640
DeepTMInter		**0.807**	**0.803**	**0.777**	**0.796**	**0.818**	**0.803**	**0.774**	**0.799**
MBPredCyto	IndepData	0.624	0.566	0.571	-	0.616	0.584	0.593	-
MBPredTM		0.568	0.603	0.581	-	0.558	0.605	0.597	-
MBPredExtra		0.591	0.548	0.519	-	0.582	0.565	0.529	-
MBPredAll		0.648	0.593	0.571	0.610	0.642	0.605	0.579	0.611
MBPredCombined		-	-	-	0.589	-	-	-	0.585
DeepTMInter		**0.689**	**0.661**	**0.688**	**0.689**	**0.697**	**0.657**	**0.681**	**0.690**

TABLE B.14: AUCPR values of predictors on the TestData, CompData, and IndepData datasets.

Predictor	**Dataset**	**Structure**				**Phobius**			
		Cyto	**TMH**	**Extra**	**Combined**	**Cyto**	**TMH**	**Extra**	**Combined**
MBPredCyto	TestData	0.539	0.449	0.528	-	0.492	0.510	0.546	-
MBPredTM		0.488	0.507	0.453	-	0.445	0.541	0.465	-
MBPredExtra		0.444	0.410	0.497	-	0.422	0.441	0.510	-
MBPredAll		0.511	0.471	0.518	0.497	0.470	0.513	0.527	0.500
MBPredCombined		-	-	-	0.517	-	-	-	0.515
DeepTMInter		**0.744**	**0.721**	**0.685**	**0.718**	**0.732**	**0.750**	**0.663**	**0.720**
MBPredCyto	CompData	0.622	0.452	0.515	-	0.603	0.497	0.520	-
MBPredTM		0.552	0.558	0.448	-	0.526	0.579	0.453	-
MBPredExtra		0.570	0.444	0.586	-	0.547	0.494	0.586	-
MBPredAll		0.664	0.566	0.581	0.604	0.648	0.586	0.591	0.606
MBPredCombined		-	-	-	0.589	-	-	-	0.587
DeepTMInter		**0.779**	**0.716**	**0.712**	**0.738**	**0.779**	**0.730**	**0.709**	**0.741**
MBPredCyto	IndepData	0.581	0.482	0.355	-	0.567	0.539	0.346	-
MBPredTM		0.522	0.513	0.386	-	0.507	0.547	0.368	-
MBPredExtra		0.540	0.476	0.312	-	0.531	0.512	0.306	-
MBPredAll		0.622	0.502	0.367	0.514	0.612	0.545	0.357	0.514
MBPredCombined		-	-	-	0.493	-	-	-	0.489
DeepTMInter		**0.657**	**0.603**	**0.458**	**0.598**	**0.661**	**0.611**	**0.447**	**0.599**

TABLE B.15: Performance gauged by mean precision, recall, F1-score, and MCC at $L/5$ (L represents protein length) threshold using structure-derived Combined regions on the TestData, CompData, and IndepData datasets.

Predictor	**Dataset**	**precision**	**recall**	**F-score**	**MCC**
MBPredAll	TestData	0.654	0.245	0.329	0.111
MBPredCombined		0.665	0.242	0.331	0.118
DeepTMInter		0.759	0.344	0.425	0.234
MBPredAll	CompData	0.679	0.260	0.359	0.129
MBPredCombined		0.687	0.260	0.360	0.134
DeepTMInter		0.773	0.320	0.426	0.229
MBPredAll	IndepData	0.608	0.249	0.344	0.139
MBPredCombined		0.573	0.240	0.325	0.104
DeepTMInter		0.612	0.276	0.360	0.151

TABLE B.16: Performance gauged by mean precision, recall, F1-score, and MCC at $L/5$ (L represents protein length) threshold using Phobius-predicted Combined regions on the TestData, CompData, and IndepData datasets.

Predictor	**Dataset**	**precision**	**recall**	**F-score**	**MCC**
MBPredAll	TestData	0.655	0.247	0.329	0.111
MBPredCombined		0.657	0.238	0.324	0.097
DeepTMInter		0.770	0.349	0.428	0.239
MBPredAll	CompData	0.679	0.260	0.358	0.128
MBPredCombined		0.685	0.258	0.358	0.128
DeepTMInter		0.783	0.324	0.432	0.239
MBPredAll	IndepData	0.607	0.248	0.343	0.135
MBPredCombined		0.560	0.232	0.316	0.085
DeepTMInter		0.617	0.277	0.360	0.151

TABLE B.17: AUC and AUCPR values on the TestData, CompData, and IndepData datasets based on the choices of interaction site definitions.

Criterion	**Dataset**	**Definition**	**Structure-derived**	**Phobius-predicted**
AUC	TestData	BordInter	0.770	0.773
		FuchInter	0.789	0.791
		RostInter	0.793	0.794
	CompData	BordInter	0.762	0.765
		FuchInter	0.790	0.792
		RostInter	0.796	0.799
	IndepData	BordInter	0.675	0.676
		FuchInter	0.688	0.689
		RostInter	0.689	0.690
AUCPR	TestData	BordInter	0.537	0.540
		FuchInter	0.673	0.676
		RostInter	0.718	0.720
	CompData	BordInter	0.527	0.528
		FuchInter	0.690	0.692
		RostInter	0.738	0.741
	IndepData	BordInter	0.448	0.449
		FuchInter	0.568	0.568
		RostInter	0.598	0.599

TABLE B.18: Number of non-interacting and interacting amino acid residues in the TrainData, TestData, CompData, and IndepData datasets based on the choices of interaction site definitions.

Dataset	**Definition**	**Number of non-interacting amino acid residues**	**Number of interacting amino acid residues**	**Number of all amino acid residues**
TrainData	BordInter	83978	18707	102685
	FuchInter	76739	25946	
	RostInter	74104	28581	
TestData	BordInter	5199	1671	6870
	FuchInter	4679	2191	
	RostInter	4461	2409	
CompData	BordInter	14613	4930	19543
	FuchInter	12723	6820	
	RostInter	12047	7496	
IndepData	BordInter	4298	1715	6013
	FuchInter	3711	2302	
	RostInter	3532	2481	

TABLE B.19: Auxiliary reference to sub-families of other protein target and other ion channel.

	Sub-family	**Detail**
Other protein target	Bcl-2	B-cell lymphoma 2 (Bcl-2) protein family
	BTN and BTN-like	Butyrophilin and butyrophilin-like proteins
	CD molecules	CD molecules
	CLRs	C-type lectin-like receptors (CLRs)
	AARC	Abscisic acid receptor complex
	Immunoglobulin-like	Immunoglobulin like domain containing proteins
	Fc epsilon receptors	Fc epsilon receptors
	Immunoglobulin C1-set	Immunoglobulin C1-set domain-containing proteins
	Other immune checkpoint	Other immune checkpoint proteins
	Reticulons and associated	Reticulons and associated proteins
	Leucine-rich	Leucine-rich repeat proteins
	Immunoglobulin C2-set	Immunoglobulin C2-set domain-containing proteins
	Neuropilins and Plexins	Neuropilins and Plexins
	Adiponectin receptors	Adiponectin receptors
	Sigma receptors	Sigma receptors
	SAB Ig like lectins	Sialic acid binding Ig like lectins
	Other PR receptors	Other pattern recognition receptors
	Tumour-associated antigens	Tumour-associated antigens
	Mitochondrial-associated	Mitochondrial-associated proteins
	Notch receptors	Notch receptors
Other ion channel	CaCC	Calcium activated chloride channel
	Aquaporins	Aquaporins
	ClC family	ClC family
	Orai channels	Orai channels
	Connexins and Pannexins	Connexins and Pannexins
	CFTR	CFTR
	NaVI2.1	Sodium leak channel, non-selective

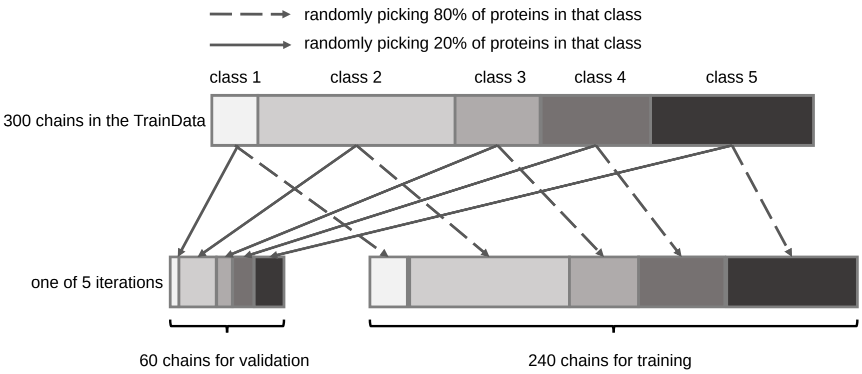

randomly picking 80% of proteins in that class
randomly picking 20% of proteins in that class
class 1
class 2
class 3
class 4
class 5
300 chains in the TrainData
one of 5 iterations
60 chains for validation
240 chains for training

FIGURE B.1: Sketch of 5-fold stratified-shuffle cross validation for allocating protein chains of different length classes.

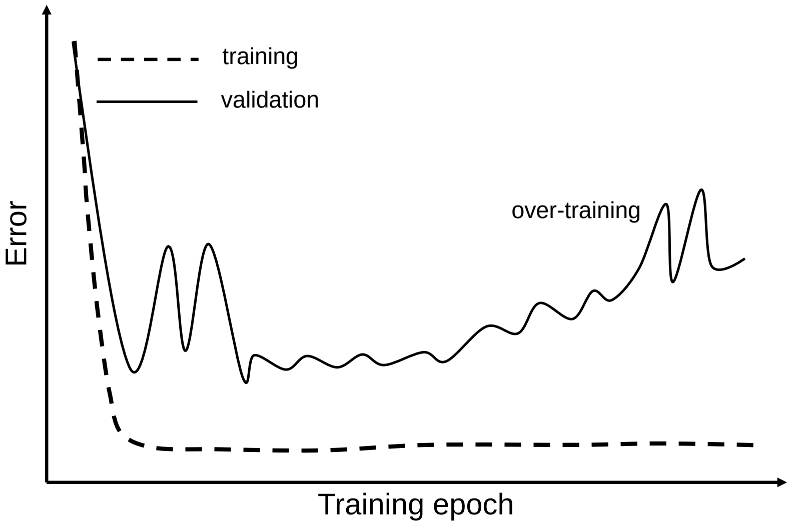


FIGURE B.2: Schematic illustration of the over-training issue occurring during training process. The performance of validation data largely fluctuates as training epoch increases. The model continues to well fit training data but gives bad performance on validation data with the increase of training epochs, leading to over-training (Amari et al., 1996 and Tetko and Villa, 1997). The error shown on the y-axis refers to a measure of differences between actual labels and predicted labels in terms of binary or multiclass classification problems. For example, in deep learning the errors on training or validation data are often measured using the cross entropy objective function (see section 3.2.6.3).

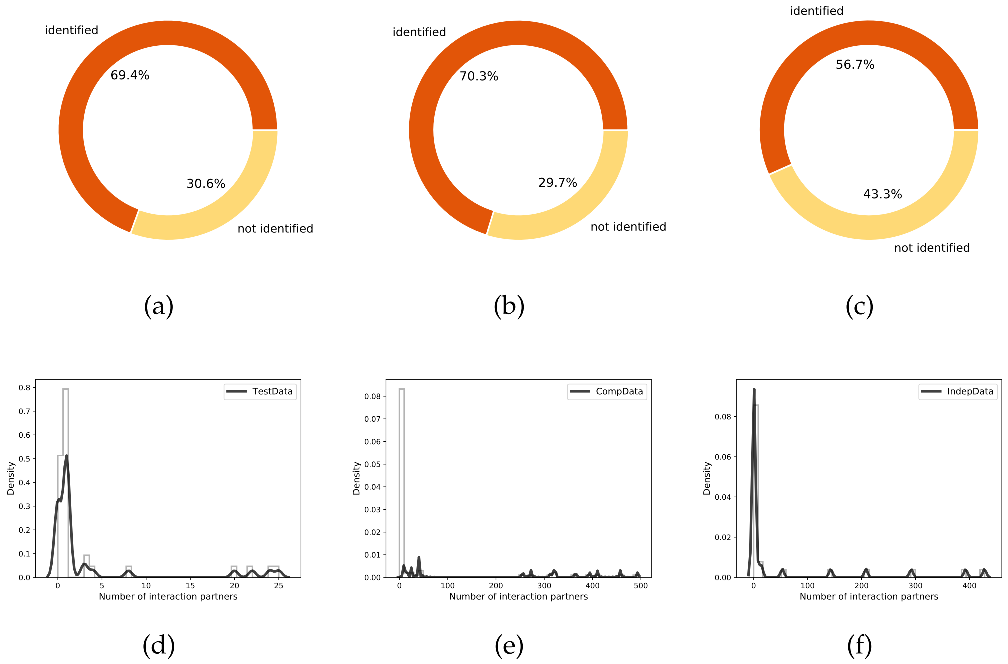


FIGURE B.3: Summary of interaction partners extracted from the BioGRID and IntAct databases. Altogether, out of all chains in the TestData, CompData, and IndepData test datasets, respectively, 25 (69.4%), 72 (71.3%), and 16 (53.3%) were identified with at least one interaction partner documented in the BioGRID and IntAct databases. (d), (e), and (f) show the distribution of interaction partners on the TestData, CompData, and IndepData datasets, respectively. Interaction partners of most of these protein chains identified are densely distributed at a small range approximately from 1 to 10, with only a few interaction partners (<25) on the TestData dataset (h) and with a rather wide range of numbers of interaction partners on the CompData (g) and IndepData (i) datasets.

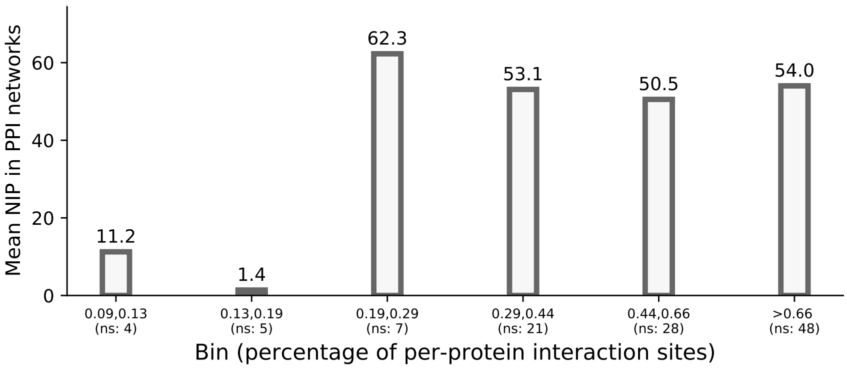


FIGURE B.4: Dependence of interaction partners (constructed by the BioGRID and IntAct databases) on the interaction sites of 167 testing proteins in the three test datasets (TestData, CompData, and IndepData). The number of interaction sites was equally divided into 6 bins according to the range of logarithm values and the mean number of interaction partners (NIPs) of human transmembrane proteins at each bin was evaluated. The number of interaction sites increases in ascending order of bin number. The graph was plotted by using the testing protein chains with interaction partners (at least one) that were found from the union of the BioGRID and IntAct databases (see section 3.2.10).

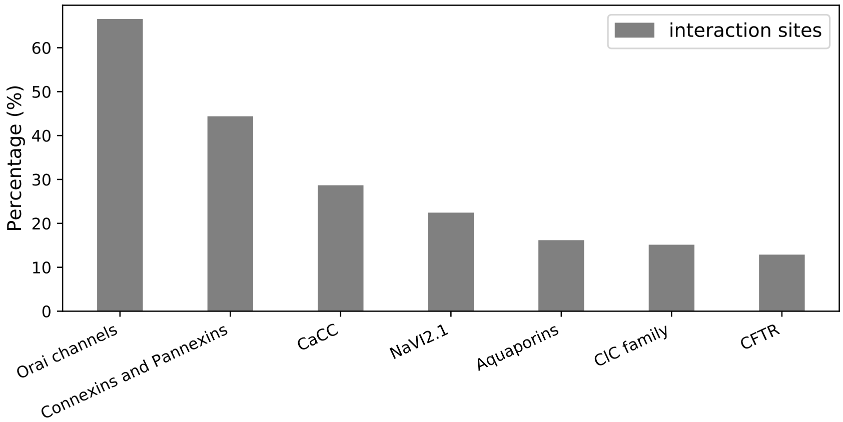


FIGURE B.5: Average percentage of per-protein interaction sites in the full sequences in the seven sub-families (see Table B.19 for reference) of the *Other ion channel* family.

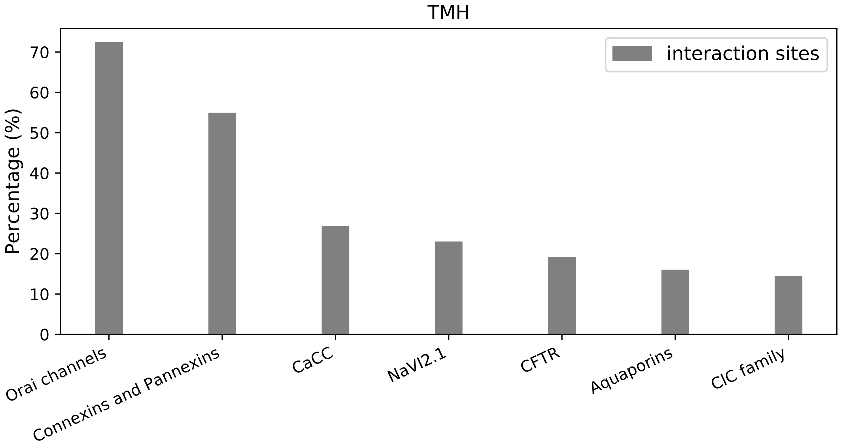


FIGURE B.6: Average percentage of per-protein interaction sites in the TMH regions in the seven sub-families (see Table B.19 for reference) of the *Other ion channel* family.

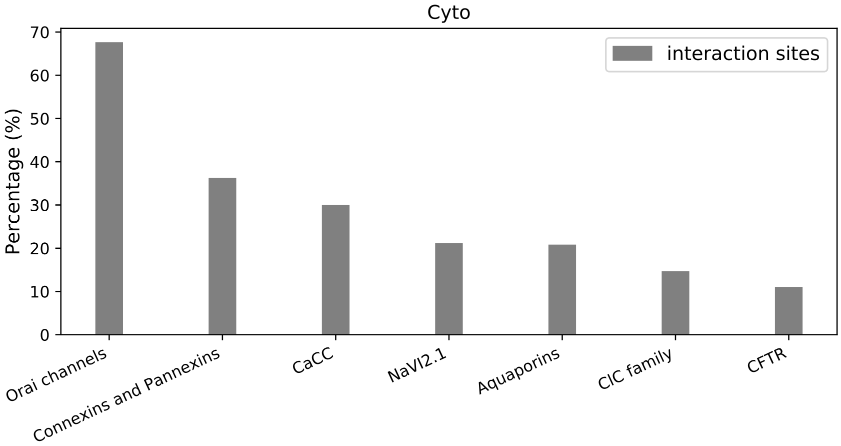


FIGURE B.7: Average percentage of per-protein interaction sites in the Cyto regions in the seven sub-families (see Table B.19 for reference) of the *Other ion channel* family.

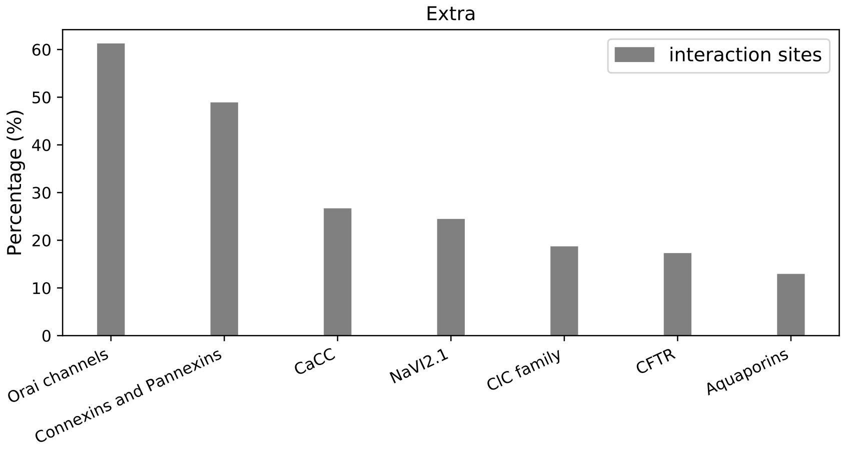


FIGURE B.8: Average percentage of per-protein interaction sites in the Extra regions in the seven sub-families (see Table B.19 for reference) of the *Other ion channel* family.

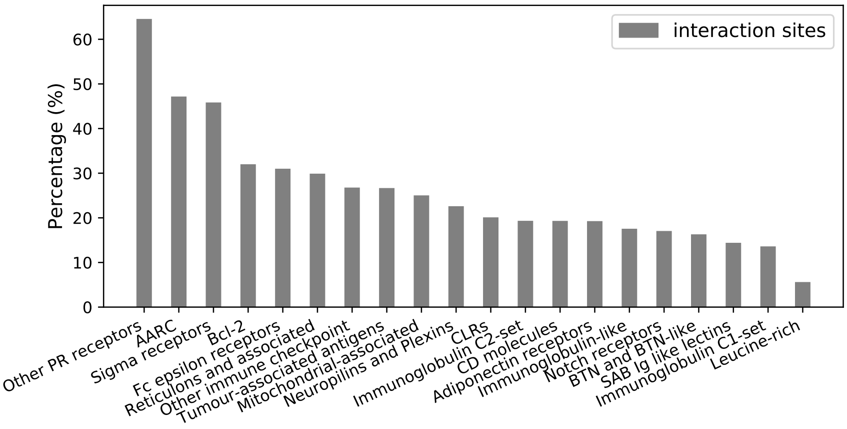


FIGURE B.9: Average percentage of per-protein interaction sites in the full sequences in the twenty sub-families (see Table B.19 for reference) of the *Other protein target* family.

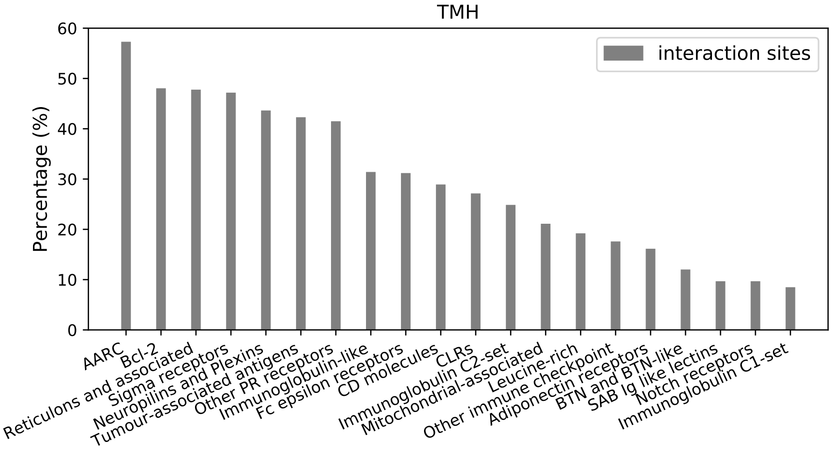


FIGURE B.10: Average percentage of per-protein interaction sites in the TMH regions in the twenty sub-families (see Table B.19 for reference) of the *Other protein target* family.

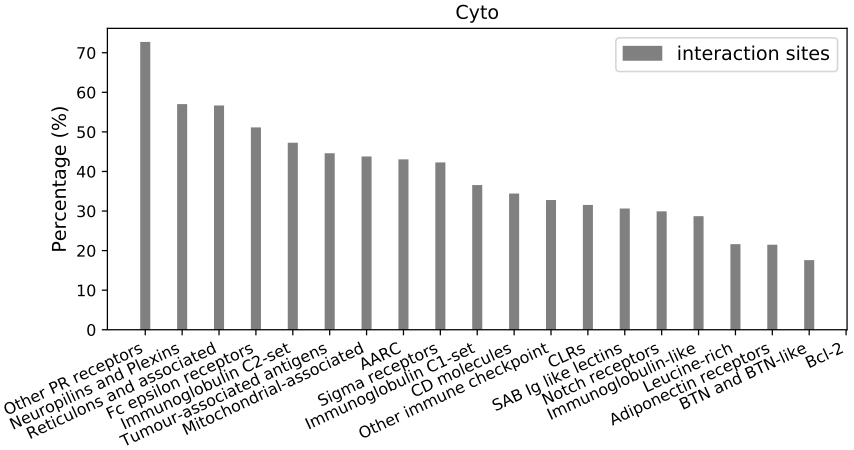


FIGURE B.11: Average percentage of per-protein interaction sites in the Cyto regions in the twenty sub-families (see Table B.19 for reference) of the *Other protein target* family.

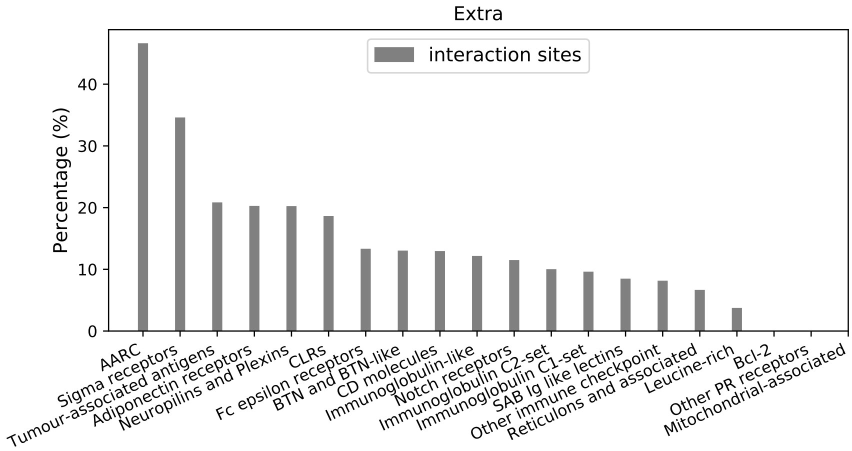


FIGURE B.12: Average percentage of per-protein interaction sites in the Extra regions in the twenty sub-families (see Table B.19 for reference) of the *Other protein target* family.

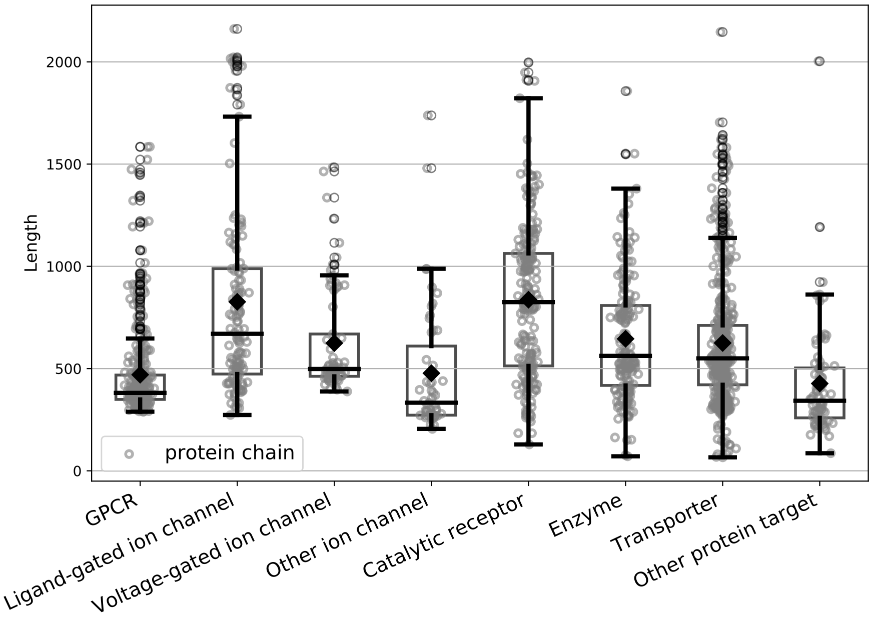
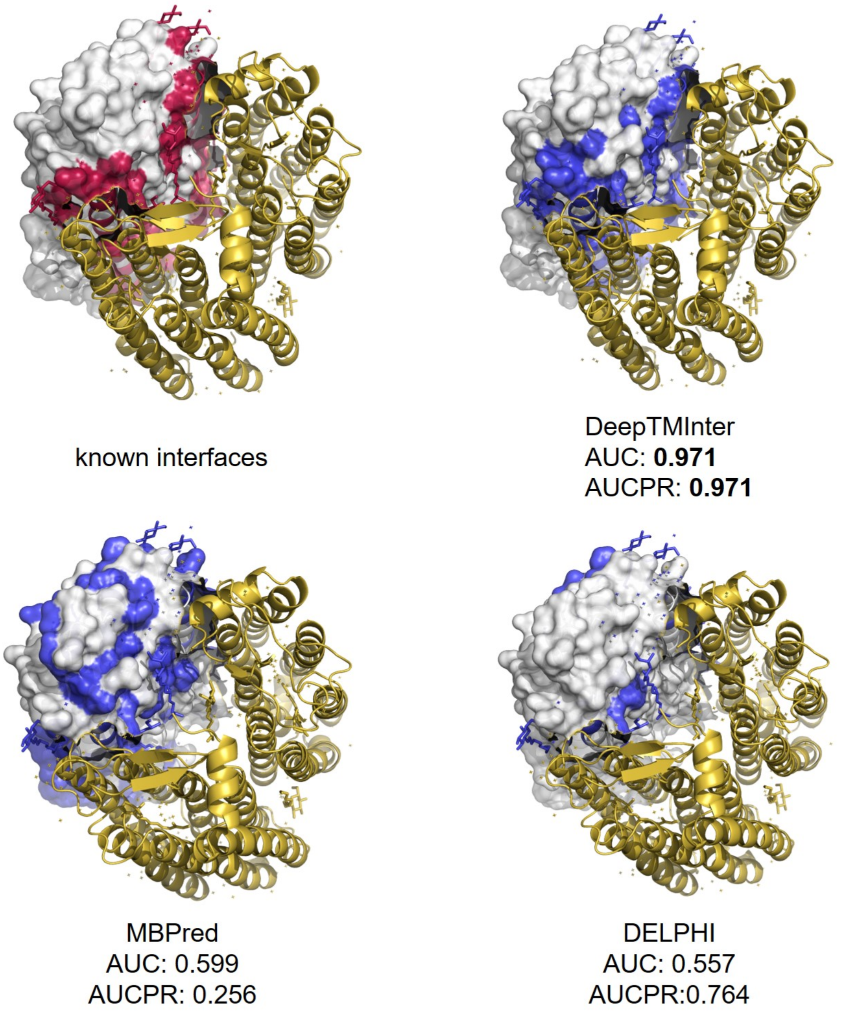


FIGURE B.13: Protein lengths in eight protein families. The black horizontal line in each box represents the average length of proteins in that family.



known interfaces

DeepTMInter
AUC: **0.971**
AUCPR: **0.971**

MBPred
AUC: 0.599
AUCPR: 0.256

DELPHI
AUC: 0.557
AUCPR:0.764

FIGURE B.14: Example of predicted interaction interfaces of a protein (PDB code: 5b0w chain A, shown in 'surface' view) in the Test-Data dataset using DeepTMInter, MBPred, and DFLPHI. Known and predicted interaction interfaces are colored in red and blue, respectively.

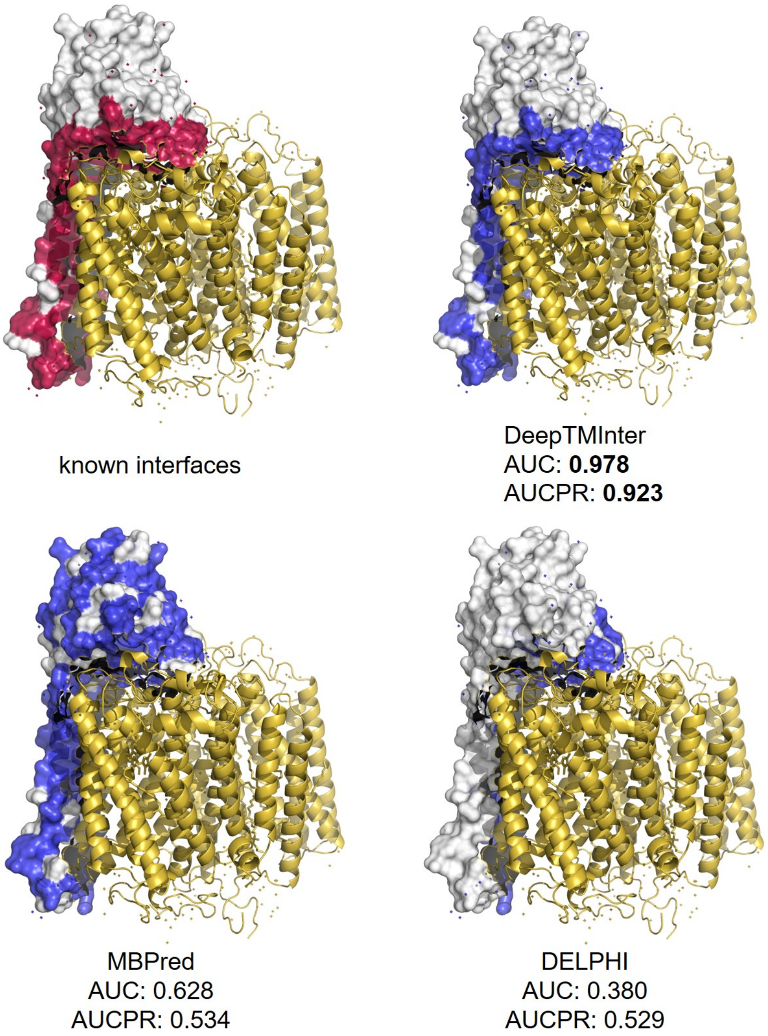


FIGURE B.15: Example of predicted interaction interfaces of a protein (PDB code: 1m56 chain A, shown in 'surface' view) in the CompData dataset using DeepTMInter, MBPred, and DFLPHI. Known and predicted interaction interfaces are colored in red and blue, respectively.

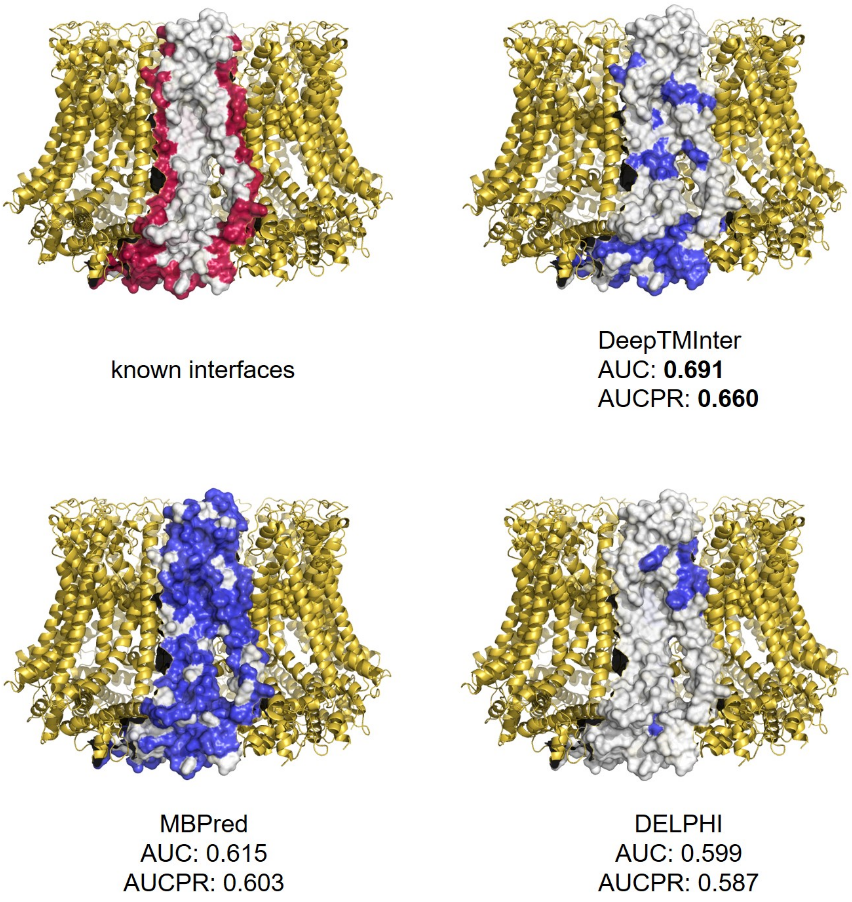


FIGURE B.16: Example of predicted interaction interfaces of a protein (PDB code: 6uiw chain A, shown in 'surface' view) in the IndepData dataset using DeepTMInter, MBPred, and DFLPHI. Known and predicted interaction interfaces are colored in red and blue, respectively.

Bibliography

Adamian, Larisa and Jie Liang (2001). "Helix-helix packing and interfacial pairwise interactions of residues in membrane proteins". In: *J Mol Biol* 311.4, pp. 891–907.

Adhikari, Badri and Jianlin Cheng (2018). "CONFOLD2: improved contact-driven ab initio protein structure modeling". In: *BMC Bioinformatics* 19.1, p. 22.

Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi (2017). "Understanding of a convolutional neural network". In: *2017 International Conference on Engineering and Technology (ICET)*. IEEE, pp. 1–6.

Alexander, Stephen PH et al. (2019). "The concise guide to pharmacology 2019/20: Ion channels". In: *Br. J. Pharmacol.* 176, S142–S228.

Almén, Markus Sällman et al. (2009). "Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin". In: *BMC Biol.* 7.1, pp. 1–14.

Amari, Shun-ichi et al. (1996). "Statistical theory of overtraining-Is cross-validation asymptotically effective?" In: *Advances in neural information processing systems*, pp. 176–182.

Amari, Shun-ichi et al. (1997). "Asymptotic statistical theory of overtraining and cross-validation". In: *IEEE Trans Neural Networ* 8.5, pp. 985–996.

Anifowose, Fatai, Jane Labadin, and Abdulazeez Abdulraheem (2015). "Improving the prediction of petroleum reservoir characterization with a stacked generalization ensemble model of support vector machines". In: *Appl Soft Comput* 26, pp. 483–496.

Apweiler, Rolf et al. (2004). "UniProt: the universal protein knowledgebase". In: *Nucleic Acids Res* 32.suppl_1, pp. D115–D119.

Armstrong, Jane F et al. (2020). "The IUPHAR/BPS Guide to PHARMACOLOGY in 2020: extending immunopharmacology content and introducing the IUPHAR/MMV Guide to MALARIA PHARMACOLOGY". In: *Nucleic Acids Res* 48.D1, pp. D1006–D1021.

Aydar, Ebru et al. (2002). "The sigma receptor as a ligand-regulated auxiliary potassium channel subunit". In: *Neuron* 34.3, pp. 399–410.

Bai, Fang et al. (2016). "Elucidating the druggable interface of protein- protein interactions using fragment docking and coevolutionary analysis". In: *Proc Natl Acad Sci* 113.50, E8051–E8058.

Baker, David and Andrej Sali (2001). "Protein structure prediction and structural genomics". In: *Science* 294.5540, pp. 93–96.

Balakrishnan, Sivaraman et al. (2011). "Learning generative models for protein fold families". In: *Proteins* 79.4, pp. 1061–1078.

Baldassi, Carlo et al. (2014). "Fast and accurate multivariate Gaussian modeling of protein families: predicting residue contacts and protein-interaction partners". In: *PloS One* 9.3, e92721.

Boer, Pieter-Tjerk de et al. (2005). "A tutorial on the cross-entropy method". In: *Annals of Operations Research* 134.

Bordner, Andrew J (2009). "Predicting protein-protein binding sites in membrane proteins". In: *BMC Bioinformatics* 10.1, p. 312.

Boyd, Kendrick, Kevin H Eng, and C David Page (2013). "Area under the precision-recall curve: point estimates and confidence intervals". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer, pp. 451–466.

Bradley, Andrew P (1997). "The use of the area under the ROC curve in the evaluation of machine learning algorithms". In: *Pattern Recognit* 30.7, pp. 1145–1159.

Causier, Barry and Brendan Davies (2002). "Analysing protein-protein interactions with the yeast two-hybrid system". In: *Plant Mol Biol* 50.6, pp. 855–870.

Chagot, Benjamin and Walter J Chazin (2011). "Solution NMR structure of Apo-calmodulin in complex with the IQ motif of human cardiac sodium channel $\mathrm{Na}_v 1.\ 5$". In: *Journal of molecular biology* 406.1, pp. 106–119.

Chen, Ching-Tai et al. (2012). "Protein-protein interaction site predictions with three-dimensional probability distributions of interacting atoms on protein surfaces". In: *PloS One* 7.6, e37706.

Cheng, Yifan (2018). "Membrane protein structural biology in the era of single particle cryo-EM". In: *Curr Opin Struct Biol* 52, pp. 58–63.

Chormunge, Smita and Sudarson Jena (2018). "Correlation based feature selection with clustering for high dimensional data". In: *J Electrical Syst Inf Technol* 5.3, pp. 542–549.

Cong, Qian et al. (2019). "Protein interaction networks revealed by proteome coevolution". In: *Science* 365.6449, pp. 185–189.

Cormier, Joseph W et al. (2002). "Secondary Structure of the Human Cardiac Na^+ Channel C Terminus Evidence for a Role of Helical Structures in Modulation of Channel Inactivation". In: *J Biol Chem* 277.11, pp. 9233–9241.

Curtis, Sharon A (2003). "The classification of greedy algorithms". In: *Sci. Comput. Program.* 49.1-3, pp. 125–157.

Dill, Ken A and Justin L MacCallum (2012). "The protein-folding problem, 50 years on". In: *science* 338.6110, pp. 1042–1046.

Ding, Wang et al. (2013). "CNNcon: improved protein contact maps prediction using cascaded neural networks". In: *PloS One* 8.4, e61533.

Ding, Wenze et al. (2018). "DeepConPred2: an improved method for the prediction of protein residue contacts". In: *Comput Struct Biotechnol J* 16, pp. 503–510.

Dobson, László, István Reményi, and Gábor E Tusnády (2015). "The human transmembrane proteome". In: *Biol Direct* 10.1, p. 31.

Dunn, Stanley D, Lindi M Wahl, and Gregory B Gloor (2008). "Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction". In: *Bioinformatics* 24.3, pp. 333–340.

Ekeberg, Magnus, Tuomo Hartonen, and Erik Aurell (2014). "Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences". In: *J Comput Phys* 276, pp. 341–356.

Ekeberg, Magnus et al. (2013). "Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models". In: *Phys Rev E* 87.1, p. 012707.

Eraslan, Gökcen et al. (2019). "Deep learning: new computational modelling techniques for genomics". In: *Nat Rev Genet* 20.7, pp. 389–403.

Ezkurdia, Iakes et al. (2009). "Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8". In: *Proteins* 77.S9, pp. 196–209.

Feinauer, Christoph et al. (2014). "Improving contact prediction along three dimensions". In: *PLoS Comput Biol* 10.10, e1003847.

Figeys, Daniel (2008). "Mapping the human protein interactome". In: *Cell Res* 18.7, pp. 716–724.

Figliuzzi, Matteo et al. (2016). "Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-1". In: *Mol Biol Evol* 33.1, pp. 268–280.

Finn, Robert D et al. (2016). "The Pfam protein families database: towards a more sustainable future". In: *Nucleic Acids Res* 44.D1, pp. D279–D285.

Fosso, Bruno et al. (2018). "Unbiased taxonomic annotation of metagenomic samples". In: *J Comput Biol* 25.3, pp. 348–360.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2008). "Sparse inverse covariance estimation with the graphical lasso". In: *Biostatistics* 9.3, pp. 432–441.

Frishman, Dmitrij and H Werner Mewes (1997). "Protein structural classes in five complete genomes." In: *Nat Struct Biol* 4.8, pp. 626–628.

Fuchs, Angelika, Andreas Kirschner, and Dmitrij Frishman (2009). "Prediction of helix–helix contacts and interacting helices in polytopic membrane proteins using neural networks". In: *Proteins Struct Funct Bioinforma* 74.4, pp. 857–871.

Fuchs, Angelika et al. (2007). “Co-evolving residues in membrane proteins”. In: *Bioinformatics* 23.24, pp. 3312–3319.

Gan-día, Jorge et al. (2013). “The Parkinson’s disease-associated GPR 37 receptor-mediated cytotoxicity is controlled by its intracellular cysteine-rich domain”. In: *J Neurochem* 125.3, pp. 362–372.

Gee, Stephen H et al. (1998). “Interaction of muscle and brain sodium channels with multiple members of the syntrophin family of dystrophin-associated proteins”. In: *J Neurosci* 18.1, pp. 128–137.

Geraets, James A, Karunakar R Pothula, and Gunnar F Schröder (2020). “Integrating cryo-EM and NMR data”. In: *Curr Opin Struct Biol* 61, pp. 173–181.

Gilpin, William (2016). “PyPDB: a Python API for the protein data bank”. In: *Bioinformatics* 32.1, pp. 159–160.

Golkov, Vladimir et al. (2016). “Protein contact prediction from amino acid co-evolution using convolutional networks for graph-valued images”. In: *Advances in Neural Information Processing Systems*, pp. 4222–4230.

Goodsell, David S et al. (2020). “RCSB Protein Data Bank: Enabling biomedical research and drug discovery”. In: *Protein Sci* 29.1, pp. 52–65.

Grant, Augustus O (2009). “Cardiac ion channels”. In: *Circ-Arrhythmia Electrophysiol* 2.2, pp. 185–194.

Gueudré, Thomas et al. (2016). “Simultaneous identification of specifically interacting paralogs and interprotein contacts by direct coupling analysis”. In: *Proc Natl Acad Sci* 113.43, pp. 12186–12191.

Hamp, Tobias and Burkhard Rost (2012). “Alternative protein-protein interfaces are frequent exceptions”. In: *PLoS Comput Biol* 8.8, e1002623.

— (2015). “Evolutionary profiles improve protein–protein interaction prediction from sequence”. In: *Bioinformatics* 31.12, pp. 1945–1950.

Hanson, Jack et al. (2018). “Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks”. In: *Bioinformatics* 34.23, pp. 4039–4045.

Hashemifar, Somaye et al. (2018). “Predicting protein–protein interactions through sequence-based deep learning”. In: *Bioinformatics* 34.17, pp. i802–i810.

Hawkins, Douglas M (2004). “The problem of overfitting”. In: *J Chem Inf Comp Sci* 44.1, pp. 1–12.

Hayat, Sikander et al. (2015). “All-atom 3D structure prediction of transmembrane β-barrel proteins from sequences”. In: *Proc Natl Acad Sci* 112.17, pp. 5413–5418.

He, Kaiming et al. (2016a). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

— (2016b). “Identity mappings in deep residual networks”. In: *European conference on computer vision*. Springer, pp. 630–645.

He, Linna et al. (2013). “Extracting drug-drug interaction from the biomedical literature using a stacked generalization-based approach”. In: *PloS One* 8.6, e65814.

He, Xiaodong and Li Deng (2017). “Deep learning for image-to-text generation: A technical overview”. In: *IEEE Signal Process Mag* 34.6, pp. 109–116.

Heffernan, Rhys et al. (2015). “Improving prediction of secondary structure, local backbone angles and solvent accessible surface area of proteins by iterative deep learning”. In: *Sci Rep* 5.1, pp. 1–11.

Heffernan, Rhys et al. (2017). “Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility”. In: *Bioinformatics* 33.18, pp. 2842–2849.

Heijne, Gunnar von (2006). “Membrane-protein topology”. In: *Nat Rev Mol Cell Biol* 7.12, pp. 909–918.

Hendrickson, Wayne A (2016). “Atomic-level analysis of membrane-protein structure”. In: *Nat Struct Mol Biol* 23.6, pp. 464–467.

Hinton, Geoffrey et al. (2012). “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal Process Mag* 29.6, pp. 82–97.

Hönigschmid, Peter and Dmitrij Frishman (2016). “Accurate prediction of helix interactions and residue contacts in membrane proteins”. In: *J Struct Biol* 194.1, pp. 112–123.

Hopf, Thomas A et al. (2012). “Three-dimensional structures of membrane proteins from genomic sequencing”. In: *Cell* 149.7, pp. 1607–1621.

Hopf, Thomas A et al. (2017). “Mutation effects predicted from sequence co-variation”. In: *Nat Biotechnol* 35.2, pp. 128–135.

Huang, Ying et al. (2010). “CD-HIT Suite: a web server for clustering and comparing biological sequences”. In: *Bioinformatics* 26.5, pp. 680–682.

Hubel, Philipp et al. (2019). “A protein-interaction network of interferon-stimulated genes extends the innate immune system landscape”. In: *Nat Immunol* 20.4, pp. 493–502.

Ioffe, Sergey and Christian Szegedy (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167*.

Ishikawa, Hiroki and Glen N Barber (2008). “STING is an endoplasmic reticulum adaptor that facilitates innate immune signalling”. In: *Nature* 455.7213, pp. 674–678.

Jessulat, Matthew et al. (2011). "Recent advances in protein–protein interaction prediction: experimental and computational methods". In: *Expert Opin Drug Discov* 6.9, pp. 921–935.

Johnson, L Steven, Sean R Eddy, and Elon Portugaly (2010). "Hidden Markov model speed heuristic and iterative HMM search procedure". In: *BMC Bioinformatics* 11.1, p. 431.

Jones, David T and Shaun M Kandathil (2018). "High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features". In: *Bioinformatics* 34.19, pp. 3308–3315.

Jones, David T et al. (2012). "PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments". In: *Bioinformatics* 28.2, pp. 184–190.

Jones, David T et al. (2015). "MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins". In: *Bioinformatics* 31.7, pp. 999–1006.

Kaján, László et al. (2014). "FreeContact: fast and free software for protein contact prediction from residue co-evolution". In: *BMC Bioinformatics* 15.1, p. 85.

Kamisetty, Hetunandan, Sergey Ovchinnikov, and David Baker (2013). "Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era". In: *Proc Natl Acad Sci* 110.39, pp. 15674–15679.

Kandathil, Shaun M, Joe G Greener, and David T Jones (2019a). "Prediction of interresidue contacts with DeepMetaPSICOV in CASP13". In: *Proteins Struct Funct Bioinforma* 87.12, pp. 1092–1099.

— (2019b). "Recent developments in deep learning applied to protein structure prediction". In: *Proteins* 87.12, pp. 1179–1189.

Kawashima, Shuichi et al. (2007). "AAindex: amino acid index database, progress report 2008". In: *Nucleic Acids Res* 36.suppl_1, pp. D202–D205.

Keijzer, Maarten and Vladan Babovic (2000). "Genetic programming, ensemble methods and the bias/variance tradeoff–introductory investigations". In: *European Conference on Genetic Programming*. Springer, pp. 76–90.

Keilhauer, Eva C, Marco Y Hein, and Matthias Mann (2015). "Accurate protein complex retrieval by affinity enrichment mass spectrometry (AE-MS) rather than affinity purification mass spectrometry (AP-MS)". In: *Mol Cell Proteomics* 14.1, pp. 120–135.

Kendrew, John C et al. (1958). "A three-dimensional model of the myoglobin molecule obtained by x-ray analysis". In: *Nature* 181.4610, pp. 662–666.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kozma, Daniel, Istvan Simon, and Gabor E Tusnady (2012). “PDBTM: Protein Data Bank of transmembrane proteins after 8 years”. In: *Nucleic Acids Res* 41.D1, pp. D524–D529.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*, pp. 1097–1105.

Krogh, Anders et al. (2001). “Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes”. In: *J Mol Biol* 305.3, pp. 567–580.

Kuzmanov, Uros and Andrew Emili (2013). “Protein-protein interaction networks: probing disease mechanisms using model systems”. In: *Genome Med* 5.4, pp. 1–12.

Lapedes, Alan S et al. (1999). “Correlated mutations in models of protein sequences: phylogenetic and structural effects”. In: *Lecture Notes-Monograph Series*, pp. 236–256.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *nature* 521.7553, pp. 436–444.

LeCun, Yann, Koray Kavukcuoglu, and Clément Farabet (2010). “Convolutional networks and applications in vision”. In: *Proceedings of 2010 IEEE international symposium on circuits and systems*. IEEE, pp. 253–256.

Lee, Anthony G (2011). “Biological membranes: the importance of molecular detail”. In: *Trends Biochem Sci* 36.9, pp. 493–500.

Lemmon, Mark A et al. (1992). “Glycophorin A dimerization is driven by specific interactions between transmembrane alpha-helices.” In: *J Biol Chem* 267.11, pp. 7683–7689.

Li, Yang et al. (2019a). “Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13”. In: *Proteins* 87.12, pp. 1082–1091.

Li, Yang et al. (2019b). “ResPRE: high-accuracy protein contact prediction by coupling precision matrix with deep residual neural networks”. In: *Bioinformatics* 35.22, pp. 4647–4655.

Li, Yifeng, Fang-Xiang Wu, and Alioune Ngom (2018). “A review on machine learning principles for multi-view biological data integration”. In: *Brief Bioinform* 19.2, pp. 325–340.

Li, Yiwei and Lucian Ilie (2020). “DELPHI: accurate deep ensemble model for protein interaction sites prediction”. In: *bioRxiv*.

Li, Yunqi, Yaping Fang, and Jianwen Fang (2011). “Predicting residue–residue contacts using random forest models”. In: *Bioinformatics* 27.24, pp. 3379–3384.

Lin, Chun-Yu et al. (2019). “Membrane protein-regulated networks across human cancers”. In: *Nat Commun* 10.1, pp. 1–17.

Liu, Xiaofen et al. (2019). “Molecular understanding of calcium permeation through the open Orai channel”. In: *PLoS Biol* 17.4, e3000096.

Liu, Xiaonan et al. (2020a). “Combined proximity labeling and affinity purification-mass spectrometry workflow for mapping and visualizing protein interaction networks”. In: *Nat Protoc*, pp. 1–30.

Liu, Yang et al. (2018). “Enhancing evolutionary couplings with deep convolutional neural networks”. In: *Cell Syst* 6.1, pp. 65–74.

Liu, Yaya et al. (2020b). “Attentional Connectivity-based Prediction of Autism Using Heterogeneous rs-fMRI Data from CC200 Atlas”. In: *Exp Neurobiol* 29.1, p. 27.

Loh, Po-Ling and Martin J Wainwright (2012). “Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses”. In: *Advances in Neural Information Processing Systems*, pp. 2087–2095.

Lomize, Andrei L, Jacob M Hage, and Irina D Pogozheva (2018). “Membranome 2.0: database for proteome-wide profiling of bitopic proteins and their dimers”. In: *Bioinformatics* 34.6, pp. 1061–1062.

Lomize, Andrei L et al. (2017). “Membranome: a database for proteome-wide analysis of single-pass membrane proteins”. In: *Nucleic Acids Res* 45.D1, pp. D250–D255.

Lomize, Mikhail A et al. (2012). “OPM database and PPM web server: resources for positioning of proteins in membranes”. In: *Nucleic Acids Res* 40.D1, pp. D370–D376.

Lou, Wangchao et al. (2014). “Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naive Bayes”. In: *PloS One* 9.1, e86703.

Louppe, Gilles et al. (2013). “Understanding variable importances in forests of randomized trees”. In: *Advances in neural information processing systems*, pp. 431–439.

Lu, Chang et al. (2019). “MPLs-Pred: Predicting Membrane Protein-Ligand Binding Sites Using Hybrid Sequence-Based Features and Ligand-Specific Models”. In: *Int J Mol Sci* 20.13, p. 3120.

Luban, Jeremy and Stephen P Goff (1995). “The yeast two-hybrid system for studying protein—protein interactions”. In: *Curr Opin Biotechnol* 6.1, pp. 59–64.

Luck, Katja et al. (2020). “A reference map of the human binary protein interactome”. In: *Nature* 580.7803, pp. 402–408.

Luo, Ling et al. (2017). “Calcium-dependent Nedd4-2 upregulation mediates degradation of the cardiac sodium channel Na$_v$1.5: implications for heart failure”. In: *Acta Physiol* 221.1, pp. 44–58.

Lynch, Berkley A and DE Koshland (1991). “Disulfide cross-linking studies of the transmembrane regions of the aspartate sensory receptor of Escherichia coli”. In: *Proc Natl Acad Sci* 88.23, pp. 10402–10406.

Ma, Jianzhu et al. (2015). “Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning”. In: *Bioinformatics* 31.21, pp. 3506–3513.

Magnan, Christophe N and Pierre Baldi (2014). “SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity”. In: *Bioinformatics* 30.18, pp. 2592–2597.

Mandal, Monalisa and Anirban Mukhopadhyay (2013). “Unsupervised non-redundant feature selection: a graph-theoretic approach”. In: *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*. Springer, pp. 373–380.

Marks, Debora S, Thomas A Hopf, and Chris Sander (2012). “Protein structure prediction from sequence variation”. In: *Nat Biotechnol* 30.11, pp. 1072–1080.

Marks, Debora S et al. (2011). “Protein 3D structure computed from evolutionary sequence variation”. In: *PloS One* 6.12, e28766.

Menche, Jörg et al. (2015). “Uncovering disease-disease relationships through the incomplete interactome”. In: *Science* 347.6224.

Michel, Mirco, David Menéndez Hurtado, and Arne Elofsson (2019). “PconsC4: fast, accurate and hassle-free contact predictions”. In: *Bioinformatics* 35.15, pp. 2677–2679.

Michel, Mirco et al. (2017). “Predicting accurate contacts in thousands of Pfam domain families using PconsC3”. In: *Bioinformatics* 33.18, pp. 2859–2866.

Milojević, Staša (2010). “Power law distributions in information science: Making the case for logarithmic binning”. In: *J Am Soc Inf Sci Technol* 61.12, pp. 2417–2425.

Min, Seonwoo, Byunghan Lee, and Sungroh Yoon (2017). “Deep learning in bioinformatics”. In: *Brief Bioinform* 18.5, pp. 851–869.

Miyazawa, Atsuo, Yoshinori Fujiyoshi, and Nigel Unwin (2003). “Structure and gating mechanism of the acetylcholine receptor pore”. In: *Nature* 423.6943, pp. 949–955.

Molica, Filippo et al. (2018). “Connexins and pannexins in vascular function and disease”. In: *Int J Mol Sci* 19.6, p. 1663.

Monastyrskyy, Bohdan et al. (2016). "New encouraging developments in contact prediction: Assessment of the CASP 11 results". In: *Proteins* 84, pp. 131–144.

Moore, David T, Bryan W Berger, and William F DeGrado (2008). "Protein-protein interactions in the membrane: sequence, structural, and biological motifs". In: *Structure* 16.7, pp. 991–1001.

Morcos, Faruck et al. (2011). "Direct-coupling analysis of residue coevolution captures native contacts across many protein families". In: *Proc Natl Acad Sci* 108.49, E1293–E1301.

Moreno, José L, Stuart C Sealfon, and Javier González-Maeso (2009). "Group II metabotropic glutamate receptors and schizophrenia". In: *Cell Mol Life Sci* 66.23, p. 3777.

Morris, John H et al. (2014). "Affinity purification–mass spectrometry and network analysis to understand protein-protein interactions". In: *Nature protocols* 9.11, p. 2539.

Murakami, Yoichi and Kenji Mizuguchi (2010). "Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites". In: *Bioinformatics* 26.15, pp. 1841–1848.

Naftaly, Ury, Nathan Intrator, and David Horn (1997). "Optimal ensemble averaging of neural networks". In: *Netw Comput Neural Syst* 8.3, pp. 283–296.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 807–814.

Nooren, Irene MA and Janet M Thornton (2003). "Structural characterisation and functional significance of transient protein–protein interactions". In: *J Mol Biol* 325.5, pp. 991–1018.

Omelchenko, Marina V et al. (2010). "Non-homologous isofunctional enzymes: a systematic analysis of alternative solutions in enzyme evolution". In: *Biol Direct* 5.1, p. 31.

Orchard, Sandra et al. (2014). "The MIntAct project—IntAct as a common curation platform for 11 molecular interaction databases". In: *Nucleic Acids Res* 42.D1, pp. D358–D363.

Oughtred, Rose et al. (2019). "The BioGRID interaction database: 2019 update". In: *Nucleic Acids Res* 47.D1, pp. D529–D541.

Ovchinnikov, Sergey et al. (2015). "Large-scale determination of previously unsolved protein structures using evolutionary information". In: *Elife* 4, e09248.

Ovchinnikov, Sergey et al. (2017). "Protein structure determination using metagenome sequence data". In: *Science* 355.6322, pp. 294–298.

Phillips, Rob et al. (2009). "Emerging roles for lipids in shaping membrane-protein function". In: *Nature* 459.7245, pp. 379–385.

Pollard, Thomas D et al. (2016). *Cell Biology (Third Edition)*. Elsevier Health Sciences.

Popescu, Sorina C et al. (2007). "Differential binding of calmodulin-related proteins to their targets revealed through high-density Arabidopsis protein microarrays". In: *Proc Natl Acad Sci* 104.11, pp. 4730–4735.

Prechelt, Lutz (1998). "Automatic early stopping using cross validation: quantifying the criteria". In: *Neural Netw* 11.4, pp. 761–767.

Rawi, Reda et al. (2016). "COUSCOus: improved protein contact prediction using an empirical Bayes covariance estimator". In: *BMC Bioinformatics* 17.1, pp. 1–9.

Remmert, Michael et al. (2012). "HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment". In: *Nat Methods* 9.2, pp. 173–175.

Riek, Peter R et al. (1995). "Evolutionary conservation of both the hydrophilic and hydrophobic nature of transmembrane residues". In: *J Theor Biol* 172.3, pp. 245–258.

Rolland, Thomas et al. (2014). "A proteome-scale map of the human interactome network". In: *Cell* 159.5, pp. 1212–1226.

Rook, Martin B et al. (2012). "Biology of cardiac sodium channel Na$_v$1. 5 expression". In: *Cardiovascular research* 93.1, pp. 12–23.

Rougier, Jean-Sébastien et al. (2005). "Molecular determinants of voltage-gated sodium channel regulation by the Nedd4/Nedd4-like proteins". In: *Am J Physiol-Cell Physiol* 288.3, pp. C692–C701.

Rual, Jean-François et al. (2005). "Towards a proteome-scale map of the human protein–protein interaction network". In: *Nature* 437.7062, pp. 1173–1178.

Saier Jr, Milton H et al. (2016). "The transporter classification database (TCDB): recent advances". In: *Nucleic Acids Res* 44.D1, pp. D372–D379.

Saito, Takaya and Marc Rehmsmeier (2015). "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PloS One* 10.3, e0118432.

Santiago, Julia et al. (2009). "The abscisic acid receptor PYR1 in complex with abscisic acid". In: *Nature* 462.7273, pp. 665–668.

Schaarschmidt, Joerg et al. (2018). "Assessment of contact predictions in CASP12: co-evolution and deep learning coming of age". In: *Proteins* 86, pp. 51–66.

Schmidt, Carla and Henning Urlaub (2017). "Combining cryo-electron microscopy (cryo-EM) and cross-linking mass spectrometry (CX-MS) for structural elucidation of large protein assemblies". In: *Curr Opin Struct Biol* 46, pp. 157–168.

Schroeter, Annett et al. (2010). "Structure and function of splice variants of the cardiac voltage-gated sodium channel Nav1. 5". In: *J Mol Cell Cardiol* 49.1, pp. 16–24.

Seemayer, Stefan, Markus Gruber, and Johannes Söding (2014). "CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations". In: *Bioinformatics* 30.21, pp. 3128–3130.

Senior, Andrew W et al. (2020). "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792, pp. 706–710.

Sharma, Harshita et al. (2017). "Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology". In: *Comput Med Imaging Graph* 61, pp. 2–13.

Sharpe, Hayley J, Tim J Stevens, and Sean Munro (2010). "A comprehensive comparison of transmembrane domains reveals organelle-specific properties". In: *Cell* 142.1, pp. 158–169.

Shimizu, Kentaro et al. (2018). "Comparative analysis of membrane protein structure databases". In: *Biochim Biophys Acta* 1860.5, pp. 1077–1091.

Shoemaker, Benjamin A and Anna R Panchenko (2007). "Deciphering protein–protein interactions. Part I. Experimental techniques and databases". In: *PLoS Comput Biol* 3.3, e42.

Shoemaker, Susannah C and Nozomi Ando (2018). "X-rays in the cryo-electron microscopy era: Structural biology's dynamic future". In: *Biochemistry* 57.3, pp. 277–285.

Shrestha, Rojan et al. (2019). "Assessing the accuracy of contact predictions in CASP13". In: *Proteins Struct Funct Bioinforma* 87.12, pp. 1058–1068.

Silver, David et al. (2017). "Mastering the game of go without human knowledge". In: *Nature* 550.7676, pp. 354–359.

Sjodt, Megan et al. (2018). "Structure of the peptidoglycan polymerase RodA resolved by evolutionary coupling analysis". In: *Nature* 556.7699, pp. 118–121.

Söding, Johannes (2017). "Big-data approaches to protein structure prediction". In: *Science* 355.6322, pp. 248–249.

Sokolina, Kate et al. (2017). "Systematic protein–protein interaction mapping for clinically relevant human GPCR s". In: *Mol Syst Biol* 13.3, p. 918.

Stahl, Kolja, Michael Schneider, and Oliver Brock (2017). "EPSILON-CP: using deep learning to combine information from multiple sources for protein contact prediction". In: *BMC Bioinformatics* 18.1, p. 303.

Stark, Chris et al. (2006). "BioGRID: a general repository for interaction datasets". In: *Nucleic Acids Res* 34.suppl_1, pp. D535–D539.

Stein, Richard R, Debora S Marks, and Chris Sander (2015). "Inferring pairwise interactions from biological data using maximum-entropy probability models". In: *PLoS Comput Biol* 11.7, e1004182.

Stone, Tracy A and Charles M Deber (2017). "Therapeutic design of peptide modulators of protein-protein interactions in membranes". In: *Biochim Biophys Acta-Biomembr* 1859.4, pp. 577–585.

Sun, Jianfeng and Dmitrij Frishman (2020). "DeepHelicon: Accurate prediction of inter-helical residue contacts in transmembrane proteins by residual neural networks". In: *J Struct Biol* 212.1, p. 107574.

Szklarczyk, Damian et al. (2019). "STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets". In: *Nucleic Acids Res* 47.D1, pp. D607–D613.

Szurmant, Hendrik and Martin Weigt (2018). "Inter-residue, inter-protein and inter-family coevolution: bridging the scales". In: *Curr Opin Struct Biol* 50, pp. 26–32.

Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar (2016). *Introduction to data mining*. Pearson Education India.

Tegge, Allison N et al. (2009). "NNcon: improved protein contact map prediction using 2D-recursive neural networks". In: *Nucleic Acids Res* 37.suppl_2, W515–W518.

Tetko, Igor V, David J Livingstone, and Alexander I Luik (1995). "Neural network studies. 1. Comparison of overfitting and overtraining". In: *J Chem Inf Comput Sci* 35.5, pp. 826–833.

Tetko, Igor V and Alessandro EP Villa (1997). "An enhancement of generalization ability in cascade correlation algorithm by avoidance of overfitting/overtraining problem". In: *Neural Process Lett* 6.1-2, pp. 43–50.

Thattai, Mukund, Yoram Burak, and Boris I Shraiman (2007). "The origins of specificity in polyketide synthase protein interactions". In: *PLoS Comput Biol* 3.9, e186.

Tran, Linh, Tobias Hamp, and Burkhard Rost (2018). "ProfPPIdb: pairs of physical protein-protein interactions predicted for entire proteomes". In: *Plos One* 13.7, e0199988.

Tusnády, Gábor E., Zsuzsanna Dosztányi, and István Simon (2004). "Transmembrane proteins in the Protein Data Bank: identification and classification". In: *Bioinformatics* 20.17, pp. 2964–2972.

— (2005). "TMDET: web server for detecting transmembrane regions of proteins by using their 3D coordinates". In: *Bioinformatics* 21.7, pp. 1276–1277.

Uguzzoni, Guido et al. (2017). "Large-scale identification of coevolution signals across homo-oligomeric protein interfaces by direct coupling analysis". In: *Proc Natl Acad Sci* 114.13, E2662–E2671.

Varga, Julia et al. (2016). "TSTMP: target selection for structural genomics of human transmembrane proteins". In: *Nucleic Acids Res*, gkw939.

Wainberg, Michael et al. (2018). "Deep learning in biomedicine". In: *Nat Biotechnol* 36.9, pp. 829–838.

Wallin, Erik and Gunnar Von Heijne (1998). "Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms". In: *Protein Sci* 7.4, pp. 1029–1038.

Walls, Alexandra C et al. (2020). "Structure, function, and antigenicity of the SARS-CoV-2 spike glycoprotein". In: *Cell*.

Wang, Hong-Wei and Jia-Wei Wang (2017). "How cryo-electron microscopy and X-ray crystallography complement each other". In: *Protein Science* 26.1, pp. 32–39.

Wang, Sheng et al. (2017). "Accurate de novo prediction of protein contact map by ultra-deep learning model". In: *PLOS Comput Biol* 13.1, e1005324.

Wang, Xiao-Feng et al. (2011). "Predicting residue-residue contacts and helix-helix interactions in transmembrane proteins using an integrative feature-based random forest approach". In: *PloS One* 6.10, e26767.

Wei, Leyi et al. (2017). "Improved prediction of protein–protein interactions using novel negative samples, features, and an ensemble classifier". In: *Artif Intell Med* 83, pp. 67–74.

Weigt, Martin et al. (2009). "Identification of direct residue contacts in protein–protein interaction by message passing". In: *Proc Natl Acad Sci* 106.1, pp. 67–72.

White, Stephen H (2009). "Biophysical dissection of membrane proteins". In: *Nature* 459.7245, p. 344.

Wlodawer, Alexander, Mi Li, and Zbigniew Dauter (2017). "High-resolution cryo-EM maps and models: a crystallographer's perspective". In: *Structure* 25.10, pp. 1589–1597.

Wolpert, David H (1992). "Stacked generalization". In: *Neural Netw* 5.2, pp. 241–259.

Wu, Qi et al. (2020). "Protein contact prediction using metagenome sequence data and residual neural networks". In: *Bioinformatics* 36.1, pp. 41–48.

Wu, Sitao and Yang Zhang (2008). "A comprehensive assessment of sequence-based and template-based methods for protein contact prediction". In: *Bioinformatics* 24.7, pp. 924–931.

Xia, Yan et al. (2018). "Integrated structural biology for α-Helical membrane protein structure determination". In: *Structure* 26.4, pp. 657–666.

Xing, Shuping et al. (2016). "Techniques for the analysis of protein-protein interactions in vivo". In: *Plant Physiol* 171.2, pp. 727–758.

Xiong, Dapeng, Jianyang Zeng, and Haipeng Gong (2017). “A deep learning framework for improving long-range residue–residue contact prediction using a hierarchical strategy”. In: *Bioinformatics* 33.17, pp. 2675–2683.

Xu, Jinbo (2019). “Distance-based protein folding powered by deep learning”. In: *Proc. Natl. Acad. Sci.* 116.34, pp. 16856–16865.

Xu, Jinbo and Sheng Wang (2019). “Analysis of distance-based protein structure prediction by deep learning in CASP13”. In: *Proteins* 87.12, pp. 1069–1081.

Xu, Jinrui and Yang Zhang (2010). “How significant is a protein structure similarity with TM-score= 0.5?” In: *Bioinformatics* 26.7, pp. 889–895.

Yang, Jing and Hong-Bin Shen (2018). “MemBrain-contact 2.0: a new two-stage machine learning model for the prediction enhancement of transmembrane protein residue contacts in the full chain”. In: *Bioinformatics* 34.2, pp. 230–238.

Yang, Jing et al. (2013). “High-accuracy prediction of transmembrane inter-helix contacts and application to GPCR 3D structure modeling”. In: *Bioinformatics* 29.20, pp. 2579–2587.

Yang, Jing et al. (2016). “R_2C: improving ab initio residue contact map prediction using dynamic fusion strategy and Gaussian noise filter”. In: *Bioinformatics* 32.16, pp. 2435–2443.

Yin, Hang and Aaron D Flynn (2016). “Drugging membrane protein interactions”. In: *Annu Rev Biomed Eng* 18, pp. 51–76.

You, Zhu-Hong et al. (2014). “Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set”. In: *BMC Bioinformatics*. Vol. 15. S15. Springer, S9.

Yuan, Yan et al. (2018). “A threshold-free summary index of prediction accuracy for censored time to event data”. In: *Statistics in medicine* 37.10, pp. 1671–1681.

Yugandhar, Kumar, Shagun Gupta, and Haiyuan Yu (2019). “Inferring protein-protein interaction networks from mass spectrometry-based proteomic approaches: a mini-review”. In: *Computational and Structural Biotechnology Journal* 17, pp. 805–811.

Zaucha, Jan et al. (2020). “Mutations in transmembrane proteins: diseases, evolutionary insights, prediction and comparison with globular proteins”. In: *Brief Bioinform*.

Zeng, Bo, Peter Hönigschmid, and Dmitrij Frishman (2019). “Residue co-evolution helps predict interaction sites in α-helical membrane proteins”. In: *J Struct Biol* 206.2, pp. 156–169.

Zerihun, Mehari B et al. (2020). “Pydca v1.0: a comprehensive software for Direct Coupling Analysis of RNA and Protein Sequences”. In: *Bioinformatics* 36.7, pp. 2264–2265.

Zhang, Buzhong et al. (2019). “Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network”. In: *Neurocomputing* 357, pp. 86–100.

Zhang, Xiao-Fei et al. (2015). “Identifying binary protein-protein interactions from affinity purification mass spectrometry data”. In: *BMC Genomics* 16.1, pp. 1–14.

Zou, Quan et al. (2020). “Sequence clustering in bioinformatics: an empirical study”. In: *Brief Bioinform* 21.1, pp. 1–10.