RSC Chemical Biology

REVIEW

View Article Online
View Journal | View Issue

Check for updates

Cite this: *RSC Chem. Biol.*, 2024, 5, 401

Received 27th October 2023, Accepted 7th February 2024

DOI: 10.1039/d3cb00208j

rsc.li/rsc-chembio

Leveraging machine learning models for peptide–protein interaction prediction

Song Yin, †[a] Xuenan Mi †[b] and Diwakar Shukla *[abc]

Peptides play a pivotal role in a wide range of biological activities through participating in up to 40% protein–protein interactions in cellular processes. They also demonstrate remarkable specificity and efficacy, making them promising candidates for drug development. However, predicting peptide–protein complexes by traditional computational approaches, such as docking and molecular dynamics simulations, still remains a challenge due to high computational cost, flexible nature of peptides, and limited structural information of peptide–protein complexes. In recent years, the surge of available biological data has given rise to the development of an increasing number of machine learning models for predicting peptide–protein interactions. These models offer efficient solutions to address the challenges associated with traditional computational approaches. Furthermore, they offer enhanced accuracy, robustness, and interpretability in their predictive outcomes. This review presents a comprehensive overview of machine learning and deep learning models that have emerged in recent years for the prediction of peptide–protein interactions.

Introduction

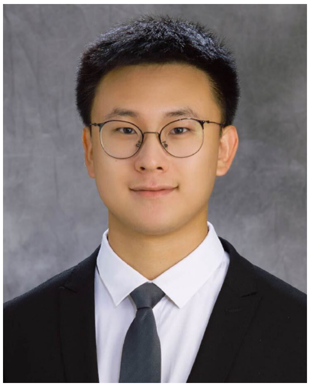
Peptides consist of short chains of amino acids connected by peptide bonds, typically comprising 2 to 50 amino acids. One of the most critical functions of peptides is their mediation of 15–40% of protein–protein interactions (PPIs).[1] PPIs play essential roles in various biological processes within living organisms, including DNA replication, DNA transcription, catalyzing metabolic reactions and regulating cellular signals.[2] Peptides have become promising drug candidates due to their ability to modulate PPIs. Over the past century, The Food and Drug Administration (FDA) has approved more than 80 peptide drugs,[3] with insulin being the pioneering therapeutic peptide used extensively in diabetes treatment. Compared with the small molecules, peptide drugs demonstrate high specificity and efficacy.[4] Additionally, compared with other classes of drug

[a] *Department of Chemical and Biomolecular Engineering, University of Illinois Urbana-Champaign, Urbana 61801, Illinois, USA. E-mail: diwakar@illinois.edu*

[b] *Center for Biophysics and Quantitative Biology, University of Illinois Urbana-Champaign, Urbana, IL, 61801, USA*

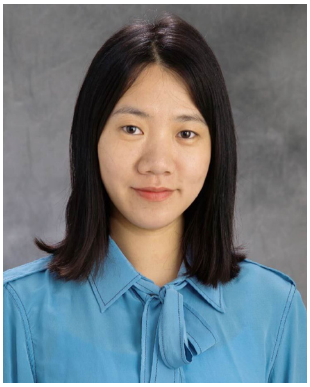
[c] *Department of Bioengineering, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA*

† These authors contributed equally to this work.



Song Yin

Song Yin is currently a PhD student in Prof. Diwakar Shukla's research group in the Department of Chemical and Biomolecular Engineering at University of Illinois at Urbana-Champaign. Song Yin obtained his BS and MS in Chemical Engineering from Tianjin University (TJU). His research work is focused on using machine learning and molecular dynamics simulations to investigate complex chemical and biological processes, including peptide drug discovery, peptide-protein interactions, etc.



Xuenan Mi

Xuenan Mi is a PhD candidate in Center for Biophysics and Quantitative Biology at University of Illinois at Urbana-Champaign under Prof. Diwakar Shukla. Her PhD research is focused on employing machine learning models and molecular dynamics simulations on peptide drug discovery and design. Specifically, she is focused on peptide substrate selectivity by leveraging large language model and active learning.

candidates, peptides have more flexible backbones, enabling their better membrane permeability.[4]

The rational design of peptide drugs is challenging and costly, due to the lack of stability and the big pool of potential target candidates. Therefore, computational methodologies that have proven effective in small molecule drug design have been adapted for modelling peptide–protein interactions (PepPIs). These computational techniques include docking, molecular dynamics (MD) simulations, and machine learning (ML) and deep learning (DL) models. Docking approaches enable exploration of peptide binding positions and poses in atomistic details, facilitating the prediction of binding affinities.[5–9] However, peptides are inherently flexible and they can interact with proteins in various conformations. These conformations often change during the binding process.[10] MD simulation is another approach to model the peptide–protein interaction. The peptide–protein binding and unbinding process can be studied thermodynamically and kinetically through MD simulations.[10–18] But sampling the complex energy landscapes associated with peptide–protein interactions typically requires intensive computational resources and time. The accuracy of both docking and MD simulations relies on the knowledge of protein structures, but the limited availability of peptide–protein complex structures has restricted the utility of these two approaches.

In recent years, ML and DL models have been widely used in the field of computer-aided drug design. These models offer an alternative way to address the inherent challenges associated with docking and MD simulations in modeling PepPIs. Due to the large amount of available biological data, many ML/DL models are routinely employed to obtain sequence–function relationship, achieving comparable predictive performance to structure-based models. This is because sequence data contain evolutionary, structural and functional information across protein space. Furthermore, compared with docking and MD simulations, ML/DL models exhibit greater efficiency and generalizability. Trained ML/DL models are capable of predicting PepPIs in a single pass, but it is hard to do large-scale docking and MD simulations due to their resource-intensive and time-consuming nature. Moreover, with the development of interpretable models, DL models are no longer regarded as black boxes; they can provide valuable insights into residue-level contributions to peptide–protein binding predictions.

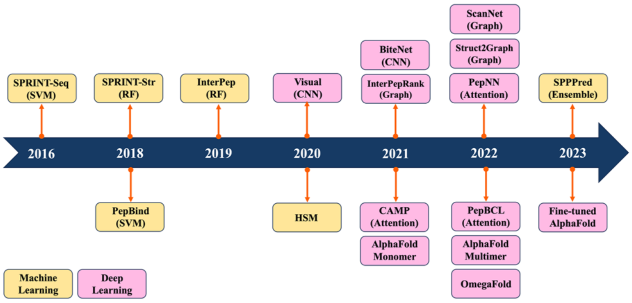


Fig. 1 Timeline of machine learning and deep learning methods for PepPI prediction.

Diwakar Shukla

Diwakar Shukla is an associate professor in the Department of Chemical and Biomolecular Engineering at the University of Illinois at Urbana-Champaign. He is affiliated with the Center for Biophysics and Quantitative Biology, Department of Plant Biology and Department of Bioengineering. His research work is focused on understanding complex biological processes using machine learning. He received his BTech and MTech degrees in Chemical Engineering at the Indian Institute of Technology (IIT) Bombay, India. He then joined Massachusetts Institute of Technology where he received his MS and PhD degrees in Chemical Engineering for his work on solution biochemistry. Before joining University of Illinois as a professor, he worked as a postdoc at Stanford University on developing distributed computing approaches understanding protein dynamics.

Previous reviews have mainly summarized ML/DL models for predicting PPIs.[19–24] They have traditionally categorized computational methods for predicting PPIs into two main classes: sequence-based and structure-based approaches. Sequence-based methods extract information only from sequence data, whereas structure-based methods rely on the information derived from peptide–protein complex structures. Recently, ML/DL models have increasingly integrated both sequence and structure information to enhance their predictive performance. In this review, we systematically summarize the progress made in predicting PepPIs. From ML perspective, we include support vector machine (SVM) and random forest (RF). ML models typically require manual feature extraction from sequence and structure datasets. But DL models, including convolutional neural network (CNN), graph convolutional network (GCN) and transformer, automatically extract multi-layer feature representations from data. To the best of our knowledge, this is the first review to summarize the ML/DL work for specifically predicting PepPIs. Fig. 1 shows the timeline illustrating the evolution of ML/DL methods in the context of PepPI predictions. Table 1 summarizes the details of ML/DL models discussed in this review.

Machine learning models for peptide–protein interaction prediction

Support vector machine (SVM)

SVM is a powerful ML algorithm commonly employed for classification tasks. The objective of SVM is to determine the optimal hyperplane that effectively separates data points belonging to different classes in the feature space. The

Table 1 Overview of machine learning models for PepPI prediction

Model name	Baseline model	Data type and datasets	Key ideas	Model performance
SPRINT-Seq[25]	SVM	Protein sequences from the BioLip[26] protein sequence	First ML model predicted PepPIs only based on sequence features	ACC: 0.66, AUC: 0.71, MCC: 0.33, SEN: 0.64, SP: 0.68
PepBind[27]	SVM	Protein sequences from BioLip[26]	Intrinsic disorder-based features were first introduced	AUC: 0.76, MCC: 0.33, SEN: 0.32, PRE: 0.45
SPRINT-Str[28]	RF	Protein–peptide complex sequences and structures from BioLip[26]	Used structural information and employed the RF classifier	ACC: 0.94, AUC: 0.78, MCC: 0.29, SEN: 0.24, SP: 0.98
InterPep[29]	RF	Protein–peptide complex structures from RCSB PDB[30]	Predicted what region of the protein structure the peptide is most likely to bind	ACC: 0.81, SEN: 0.51
SPPPred[31]	Ensemble: SVM, RF, KNN	Protein sequences from the BioLip database[26]	Ensemble learning model was applied for effectively handling imbalanced dataset	ACC: 0.95, AUC: 0.71, MCC: 0.23, F1: 0.31, SEN: 0.32, SP: 0.96
Hierarchical statistical mechanical (HSM)[32]	HSM	Peptide binding domain (PBD)-peptide structures from UniProt[33]	Introduced bespoke HSM model to predict the affinities of peptide binding domain (PBD)–peptide interactions	AUC: 0.97 (PBD: PDZ)
Visual[34]	CNN	Protein sequences from BioLip[26]	Protein sequence features were transformed into images and CNN was first applied to predict PepPIs	AUC: 0.73, MCC: 0.17, SEN: 0.67, SP: 0.69
$\text{BiteNet}_{\text{Pp}}$[35]	CNN	Protein–peptide complex structures from BioLip[26]	Utilized 3D CNN and protein structures directly to predict protein–peptide binding sites	AUC: 0.91, MCC: 0.49, PRE: 0.53
InterPepRank[36]	GCN	Protein–peptide complex structures from RCSB PDB[30]	Achieves high accuracy in predicting both binding sites and conformations for disordered peptides	AUC: 0.86
ScanNet[37]	Geometric DL Architecture	Protein–peptide complex structures from Dockground[38]	An end-to-end, interpretable geometric DL model that learns features directly from 3D structures	ACC: 0.88, AUC: 0.69, SEN: 0.50, PRE: 0.74
Struct2Graph[39]	GCN and attention	Protein–peptide complex structures from IntAct,[40] STRING,[41] and UniProt[33]	A GCN-based mutual attention classifier accurately predicting interactions between query proteins exclusively from 3D structural data	ACC: 0.99, AUC: 0.99, MCC: 0.98, F1: 0.99, SEN: 0.98, SP: 0.99, PRE: 0.99, NPV: 0.98
CAMP[42]	CNN and self-attention	Protein–peptide complex sequences from RCSB PDB[30] and DrugBank[43]	Took account of sequence information of both proteins and peptides, and identified binding residues of peptides	AUC: 0.87, AUPR: 0.64
PepNN[44]	Transformer	Protein–peptide complex sequences and structures from RCSB PDB[30]	Utilized a multi-head reciprocal attention layer to update the embeddings of both peptides and proteins; transfer learning was applied to solve the limited protein–peptide complex structure issue	AUC: 0.86, MCC: 0.41
PepBCL[45]	BERT-based contrastive learning framework	Protein sequences from the BioLip database[26]	An end-to-end predictive model; contrastive learning module was used to tackle the imbalanced data issue	AUC: 0.82, MCC: 0.39, SEN: 0.32, SP: 0.98, PRE: 0.54
AlphaFold monomer[46–48]	MSA based transformer	Protein sequences and structures from Uniclust30[49] and RCSB PDB[30]	Adding the peptide sequence *via* a polyglycine linker to the C-terminus of the receptor monomer sequence could mimic peptide docking as monomer folding	SR: 0.75 (within 1.5 Å RMSD) in Tsaban *et al.*[47] and SR: 0.33 (fraction of native contacts = 0.8 as cutoff) in Shanker and Sanner[48]
OmegaFold[48,50]	Protein language model	Protein sequences and structures from Uniref50,[51] RSCB PDB,[30] CASP,[52] and CAMEO[53]		SR: 0.20 (fraction of native contacts = 0.8 as cutoff) in Shanker and Sanner[48]
AlphaFold multimer[48,54]	MSA based transformer	Protein complex sequences and structures from RSCB PDB[30] and Benchmark 2[55]	Improved the accuracy of predicted multimeric interfaces between two or more proteins	SR: 0.53 (fraction of native contacts = 0.8 as cutoff) in Shanker and Sanner[48]
Fine-tuned AlphaFold[56]	MSA based transformer	Peptide–MHC complex structures from RSCB PDB[30]	Leveraging and fine-tuning AF2 with existing peptide–protein binding data could improve its PepPI predictions	AUC: 0.97 (class 1) and AUC: 0.93 (class 2)

Abbreviations: ACC: accuracy; AUC: area under the ROC curve; AUPR: area under the precision–recall curve; MCC: Matthews correlation coefficient; SEN: sensitivity; SP: specificity; PRE: precision; SR: success rate.

selection criteria for this optimal hyperplane aim to maximize the margins between the closest points of distinct classes, thereby minimizing misclassification rates.

SPRINT-Seq (Sequence-based prediction of Protein–peptide Residue-level INTeraction sites) is the first ML based prediction of peptide–protein binding sites only using sequence features.[25] Various types of information were extracted from protein sequences to create a feature dataset, including one-hot encoded protein sequences, evolutionary information,[57] predicted accessible surface area,[58] secondary structures,[58] and physiochemical properties.[59] These features were fed into a classification model, SVM, to predict the label for each residue

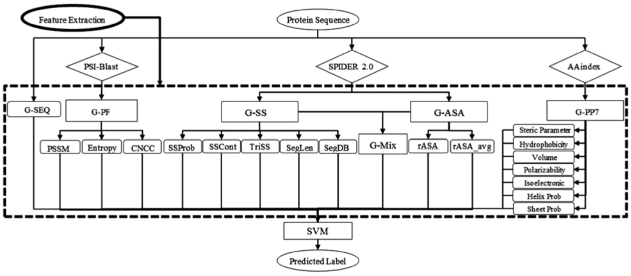


Fig. 2 The input features and architecture of SPRINT-Seq. G-SEQ: sequence feature; G-PF: sequence profile from the position specific scoring matrix (PSSM); G-SS: secondary structure-based features; G-ASA: accessible surface area-based features; G-PP7: physicochemical-based feature group. Adapted with permission from G. Taherzadeh, Y. Yang, T. Zhang, A. W.-C. Liew and Y. Zhou, *J. Comput. Chem.*, 2016, **37**, 1223–1229. Copyright 2024 John Wiley and Sons.

(Fig. 2). SPRINT-Seq yielded a Matthews correlation coefficient (MCC) of 0.326, a sensitivity of 0.64 and a specificity of 0.68 on an independent test set. The importance of each feature was also evaluated; the most crucial feature distinguishing binding from non-binding residues is the sequence evolution profile. This sequence-based technique's performance is comparable to or better than that of structure-based models (Peptimap,[60] Pepite,[61] PinUp,[62] VisGrid[63]) for peptide-binding site prediction.

To improve the accuracy of sequence-based prediction, Zhao *et al.* introduced intrinsic disorder as a feature within sequence representation.[27] Peptides that participate in peptide–protein interactions exhibit consistent attributes of short linear motifs, primarily found in the intrinsic disordered regions (IDRs). These attributes include short length, flexible structure and weak binding affinity.[64] In addition to the novel sequence representation, they designed a consensus-based method called PepBind.[27] This method combines the SVM classification model with the template-based methods S-SITE and TM-SITE.[65] The combination of these three individual predictors yielded better performance than all three individual methods and outperformed the first sequence-based method SPRINT-Seq.

Random forest (RF)

RF is another supervised ML algorithm for classification and regression, which combines multiple decision trees to create a "forest". During the training of a RF for classification, each tree contributes a vote. The forest subsequently selects the classification with the majority of votes as the predicted outcome. All decision trees comprising the RF are independent models. While individual decision trees may contain errors, the collective majority vote of the ensemble ensures more robust and accurate predictions, thereby enhancing the reliability of RF predicted results.

A RF model, SPRINT-Str[28] (Structure-based Prediction of Residue-level INTeraction), was developed to predict the putative peptide–protein binding residues and binding sites by combining both sequence-based and structure-based information. The sequence information in the input includes the position specific scoring matrix (PSSM) for all amino acids in the protein and entropy calculated based on the PSSM. Structural information includes accessible surface area (ASA) calculated by DSSP (define secondary structure of proteins),[66] secondary structure (SS) calculated by DSSP,[66] half-sphere exposure (HSE) representing the solvent exposure using residue contact numbers in upward and downward hemispheres along with the pseudo Cβ–Cα bond,[67] and flexibility calculated by iModeS[68] to describe the functional motions of proteins.[69] A RF classifier was further trained and tested to predict the binding residues. The density-based spatial clustering of applications with noise (DBSCAN) algorithm[70] was then applied to cluster spatially neighboring binding site residues. The largest cluster was selected as the predicted binding site with a corresponding reliability score. SPRINT-Str achieved robust performance in predicting binding residues with a MCC of 0.293 as well as an area under the receiver operating characteristic curve (ROC AUC) of 0.782. For instance, when testing the model's performance on peptide binding with the human tyrosine phosphatase protein PTPN4 PDZ domain (PDBID: 3NFK),[71] 15 out of 17 binding residues were correctly predicted, and the predicted binding sites were similar to the actual binding sites. SPRINT-Str is one of the representative ML models that pass structural features into the models and has achieved remarkable success in predicting PepPIs.

The structures of proteins or peptide–protein complexes can also be directly used as input to ML models. The underlying premise of this approach is that, if a PepPI shares similarities with a certain interaction surface, that well-characterized surface can serve as a template for modeling other PepPIs. The InterPep model[29] constructs four steps to better represent this idea: mass structural alignment (MSA), feature extraction, RF classification, and clustering. A template modeling (TM) score larger than 0.5 was used to screen out candidate templates. Overall, InterPep accurately predicted 255 out of 502 (50.7%) binding sites for the top 1 prediction and correctly identified 348 out of 502 (69.3%) binding sites within the top 5 predictions, which demonstrates that it is a useful tool for the identification of peptide-binding sites.

Ensemble learning

In the pursuit of a more robust predictive model for protein–peptide binding sites, Shafiee *et al.* adopted an ensemble-based ML classifier named SPPPred.[31] Ensemble learning stands out as an effective strategy for handling imbalanced datasets, as it allows multiple models to collectively contribute to predictions, resulting in enhanced robustness, reduced variance, and improved generalization.[72]

In the SPPPred algorithm, the ensemble learning technique of bagging[73] was employed to predict peptide binding residues. The initial step in bagging involves generating various subsets of data through random sampling with replacement, a process known as bootstrapping. For each bootstrap dataset, distinct classification models are trained, including support vector machine (SVM), K-nearest neighbors (KNN), and random forest (RF). Subsequently, for each residue, the class with the majority of votes across these models is determined as the final predicted label. This ensemble method consistently demonstrates strong and comparable performance on independent test sets, with an F1 score of 0.31, an accuracy of 0.95, and an MCC of 0.23.

Other state-of-the-art (SOTA) models

There are some SOTA bespoke ML models that have achieved great success in the predictions of PepPIs, for example, hierarchical statistical mechanical modeling (HSM).[32] A dataset of 8 peptide-binding domain (PBD) families was applied to train and test the HSM model, including PDZ, SH2, SH3, WW, WH1, PTB, TK, and PTP, which cover 39% of human PBDs. The HSM model defines a pseudo-Hamiltonian, which is a machine-learned approximation of Hamiltonian that maps the system state to its energy.[74] The predicted PepPI probability is derived from the sum of pseudo-Hamiltonian corresponding to each PBD-peptide sequence pair. In total, 9 models were developed, including 8 separate HSM/ID models (ID means independent domain, one for each protein family) and a single unified HSM/D model covering all families (D means domains). The HSM model remarkably outperformed other ML models such as NetPhorest[75] and PepInt.[76] By computing the energies from pseudo-Hamiltonian, the HSM model can evaluate and rank the possibilities of different PepPI patterns, facilitating the verification of existing PepPI ensembles and the discovery of new possible PepPI ensembles. Furthermore, the HSM model provides detailed explanations of the peptide–protein binding mechanism, demonstrating a strong interpretability. Using peptide binding with the HCK-SH3 domain (PDBID: 2OI3)[77] as an example, the HSM model gave a detailed examination and explanation of the peptide-SH3 domain binding mechanism. The ''W114 tryptophan switch'' binding motif[78] was correctly recognized by the HSM model. Additionally, a conserved triplet of aromatic residues W114-Y132-Y87 was previously identified as contributing to peptide binding with the HCK-SH3 domain.[79,80] However, the HSM model also found that Y89 and Y127 had similar predicted energetic profiles as W114, suggesting a new possible W–Y–Y aromatic triplet. By mapping the predicted interaction energies to the complex structure, the HSM model successfully recognized the repulsive binding regions and attractive binding regions. The predicted attractive binding interface correctly aligns with the previously studied RT-loop and proline recognition pocket,[79,80] demonstrating the strong predictive and interpretative ability of the HSM model.

Deep learning models for peptide–protein interaction prediction

Convolutional neural network (CNN)

CNNs are a class of neural networks that have demonstrated great success in processing image data.[81] The design of CNNs was inspired by the biological visual system in humans. When humans see an image, each neuron in the brain processes information within its own receptive field and connects with other neurons in a way to cover the entire image. Similarly, each neuron in a CNN also only processes data in its receptive field. This approach allows CNNs to dissect simpler patterns initially and subsequently assemble them into more complex patterns. A typical CNN architecture consists of three layers: the convolutional layer, the pooling layer, and the fully connected layer. In the convolutional layer, a dot product is computed between two matrices – the first being a kernel with a set of learnable parameters, and the second representing a portion of the receptive field. The kernel slides across the entire image, generating a two-dimensional representation. The pooling layer replaces the output of the convolutional layer at each location by deriving a summary statistic of the nearby outputs. This serves to reduce the size of the feature maps, subsequently decreasing the training time. Finally, the fully connected layer connects the information extracted from the previous layers to the output layer and eventually classifies the input into a label. The biological data could be transformed into an image-like pattern; therefore CNNs could be applied to binding site identification.

Wardah *et al.* applied CNNs for identifying peptide-binding sites by introducing a CNN-based method named Visual.[34] In the Visual algorithm, features were extracted from protein sequences, like HSE,[67] secondary structures,[82] ASA,[82] local backbone angles,[82] PSSM[57] and physicochemical properties.[83] These features were stacked horizontally resulting in a feature vector with a length of 38. Visual employs a sliding window approach to capture the local context of each residue. For a given residue, the feature vectors of the three upstream and three downstream residues were combined into a matrix, resulting in a 2-dimensional array with a size of 7 × 38. An illustrative example of the input data in an image-like format is depicted in Fig. 3, showcasing the center residue serine (S) within a window size of 7. A 7 × 38 image is generated as the input of the CNN classifier. The Visual model comprises two sets of convolutional layers, followed by a pooling layer and a fully connected layer (Fig. 3). Visual was applied to identify the peptide binding sites of proteins and achieved a sensitivity of 0.67 and a ROC AUC of 0.73.

BiteNet$_{\mathrm{Pp}}$[35] is another CNN-based model that converts 3D protein structures to 4D tensor-based representations and feeds them into a 3D CNN to learn the probability of PepPIs and predict the peptide binding sites/domain. The 4D tensor has the first three dimensions corresponding to the x, y, and z dimensions, and the fourth dimension corresponding to 11 channels including atomic densities of 11 different atom types such as aromatic carbon, sulfur, amide nitrogen, carbonyl oxygen, and so forth. These four-dimensional tensor-based representations are then fed into 10 three-dimensional convolutional layers to obtain the probability score of ''hot spots'', which are determined as the geometric centers of each segmented peptide–protein interface. This model outperforms SOTA methods with a ROC AUC of 0.91 and a MCC of 0.49. The model showed promising power for the prediction of peptide–protein binding sites, but the model's performance is limited by the input protein orientation and sensitivity to the

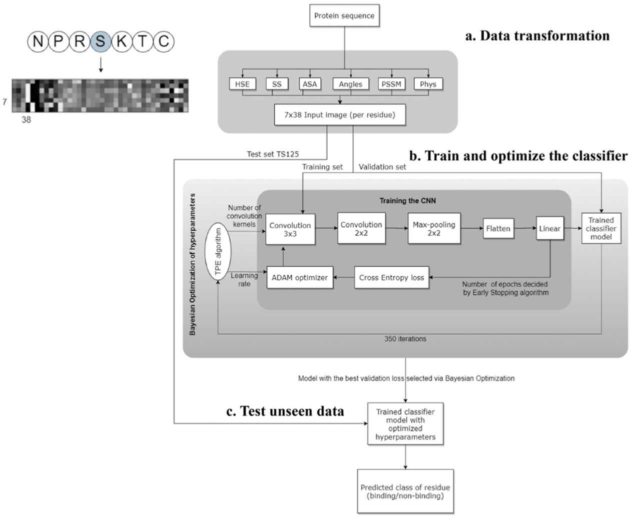

N P R S K T C
7
38
Protein sequence
a. Data transformation
HSE
SS
ASA
Angles
PSSM
Phys
7x38 Input image (per residue)
Test set TS125
b. Train and optimize the classifier
Training set
Validation set
Bayesian Optimization of hyperparameters
Training the CNN
Number of convolution kernels
TPE algorithm
Learning rate
Convolution 3x3
Convolution 2x2
Max-pooling 2x2
Flatten
Linear
Trained classifier model
ADAM optimizer
Cross Entropy loss
Number of epochs decided by Early Stopping algorithm
350 iterations
Model with the best validation loss selected via Bayesian Optimization
c. Test unseen data
Trained classifier model with optimized hyperparameters
Predicted class of residue (binding/non-binding)

Fig. 3 The workflow of the Visual model. (a) Transforming the protein sequence into a 7 × 38 input image (per residue). In the order from left to right of the image: 3 pixels represent half sphere exposure (HSE),[67] 3 pixels represent the predicted probabilities of different secondary structures, 1 pixel represents the accessible surface area (ASA) value, 4 pixels represent the local backbone angles, 20 pixels represent the position specific scoring matrix (PSSM), and 7 pixels represent the physicochemical properties of the amino acids. (b) Training and optimizing hyperparameters of the CNN. (c) Testing the optimized CNN on unseen test data to predict the label of each residue (binding/non-binding). Adapted with permission from W. Wardah, A. Dehzangi, G. Taherzadeh, M. A. Rashid, M. Khan, T. Tsunoda and A. Sharma, *J. Theor. Biol.*, 2020, **496**, 110278. Copyright 2024 Elsevier.

protein conformations. Therefore, BiteNet$_{\mathrm{Pp}}$ could be improved by using representations that could handle the protein rotation invariance.

Graph convolutional network (GCN)

Graph based models have been widely used to illustrate the PPIs and PepPIs based on the peptide/protein structures.[36,37,39,84–88] Graph embedding[89] includes nodes (vertices) representing different entities and edges (links) representing the relationships between them. For proteins, graphs typically assign amino acids and related information as nodes, with the distances and connections between amino acids represented as edges. This approach allows for the direct observation of information from protein 3D structures without involving hand-crafted features.[24,90] GCNs[91,92] are a type of neural network that can be used to learn graph embeddings. Similar to CNNs, GCNs take graph embeddings as input and progressively transform them through a series of localized convolutional and pooling layers where each layer updates all vertex features. The updated embeddings are passed through a classification layer to obtain the final classification results.[89,91] GCNs have been successfully applied to protein binding site prediction, with models such as PipGCN[84] and EGCN[85] achieving great success. More recently, a number of GCN-based models have also been applied for PepPI prediction.

InterPepRank[36] is a representative GCN that has been developed to predict the PepPIs. In this model, billions of decoys (computational protein folding structure) were generated by the PIPER[93] docking tool as the training and testing set, respectively. The peptide–protein complexes were then represented as graphs with one-hot encoded nodes illustrating individual residues, PSSM,[94] and self-entropy,[94] and one-hot encoded edges denoting the residue interactions. Both node and edge features were then passed through edge convolution layers with the output from each layer concatenated and fed into a global pooling layer and two dense layers to predict the LRMSD (ligand root-mean-square deviation) of decoys. InterPepRank achieved a median ROC AUC of 0.86, outperforming other benchmarking methods such as PIPER,[93] pyDock3,[95] and Zrank.[96] For example, in the case of a fragment from the center of troponin I (peptide) binding with the C-terminal domain of Akazara scallop troponin C (receptor),[97] the peptide was proved to be disordered when unbound and become an ordered α-helical structure upon binding,[98] following the induced-fit binding mechanism. Predicting the peptide binding conformation and binding sites for systems with induced-fit mechanisms is extremely challenging. The top 100 decoys predicted by both InterPepRank and Zrank showed that both methods can find the true binding site of the peptide. However, InterPepRank achieved an accuracy of 96% in predicting the peptide as an α-helical structure, while Zrank only achieved an accuracy of less than 50%, where half of the peptide decoys' secondary structures were predicted as either random coils or β-sheets. Therefore, InterPepRank is a powerful tool for predicting both binding sites and conformations, even in cases where the peptide is disordered when unbound. This is a significant advantage over other benchmarked energy-based docking methods, which may struggle with disordered structures that are more energetically favorable in unbound states or easier to fit into false positive binding sites.

Struct2Graph[39] is a novel multi-layer mutual graph attention convolutional network for structure-based predictions of PPIs (Fig. 4). Coarse-grained graph embeddings were generated by two GCNs with weight sharing for both components of the protein complexes. These embeddings were then passed through a mutual attention network to extract the relevant features for both proteins and concatenated into a single embedding vector. Attention weights and context vectors were calculated from the GCN-transformed hidden embeddings. Residues with large learned attention weights are more important and more likely to contribute towards interaction. The context vectors were concatenated and further passed into a feed-forward network (FFN) and a final Softmax layer to get the probability for PPI. Struct2Graph outperformed the feature-based ML models and other SOTA sequence-based DL models, achieving an accuracy of 98.89% on a positive/negative sample balanced dataset and an accuracy of 99.42% on a positive/negative sample unbalanced dataset (positive : negative = 1 : 10). Residue-level interpretation was conducted to identify the residues' contribution to PepPIs. For example, *Staphylococcus aureus* phenol soluble modulins (PSMs) peptide PSMα_1[99] competes with the high mobility group box-1 protein (HMGB1) to bind with toll-like receptor-4 (TLR4),[100] thus inhibiting HMGB1-mediated phosphorylation of NF-κB.[101] For the

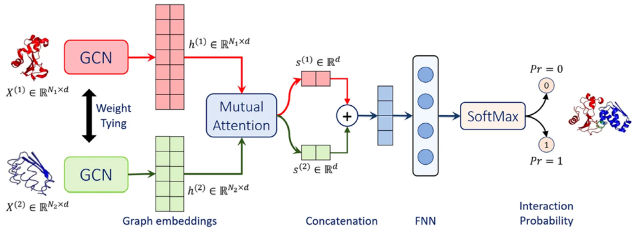

GCN
$h^{(1)} \in \mathbb{R}^{N_1 \times d}$
$X^{(1)} \in \mathbb{R}^{N_1 \times d}$
Weight
Tying
GCN
$h^{(2)} \in \mathbb{R}^{N_2 \times d}$
$X^{(2)} \in \mathbb{R}^{N_2 \times d}$
Mutual
Attention
$s^{(1)} \in \mathbb{R}^{d}$
$s^{(2)} \in \mathbb{R}^{d}$
+
SoftMax
$Pr = 0$
0
1
$Pr = 1$
Graph embeddings
Concatenation
FNN
Interaction
Probability

Fig. 4 Struct2Graph model architecture. Struct2Graph model loads graph embeddings of both components into two weight sharing graph convolutional networks (GCNs) separately. GCNs' outputs are integrated into a mutual attention network to predict the probability of PPI and the interaction sites. Adapted with permission from M. Baranwal, A. Magner, J. Saldinger, E. S. Turali-Emre, P. Elvati, S. Kozarekar, J. S. VanEpps, N. A. Kotov, A. Violi and A. O. Hero, *BMC Bioinf.*, 2022, **23**, 370. This article is licensed under a Creative Commons Attribution 4.0 International License, permitting unrestricted reproduction and adaptation provided proper crediting to author and source. Copyright 2024 Springer Nature.

PSMα_1-TLR4 complex, Struct2Graph demonstrated an impressive accuracy of 92%, and the predicted binding residues aligned with the previously identified TLR4 active binding sites. Notably, peptide residues 2Gly and 10Val were accurately predicted as the peptide binding residues. Furthermore, Struct2Graphs predictions corroborated the previously studied competitive binding mechanism, indicating that both PSMα_1 peptide and HMGB1 bind to the same area of TLR4.

Interpretable DL graph models have also been employed for the PepPI predictions. Recently, an end-to-end geometric DL architecture known as ScanNet (Spatio-chemical arrangement of neighbors neural NETwork)[37] was developed that integrated multi-scale spatio-chemical arrangement information of atoms and amino acids, along with multiple sequence alignment (MSA) for detecting protein–protein binding sites (PPBS). The model took the protein sequence, tertiary structure, and optionally position-weight matrix from MSA of evolutionarily related proteins as input. It first extracted all the atomic neighborhood embeddings, which were then passed through several filters to learn the atomic scale representations. To further reduce the dimensions, atom-wise representations were pooled at the amino acid scale, mixed with extracted amino acid information, and fed into trainable filters to yield amino acid scale representations (Fig. 5(a)). With these representations containing multi-scale spatio-chemical information, ScanNet was trained for the prediction of PPBS on 20k proteins with annotated binding sites. When compared with the traditional ML method XGBoost with handcrafted features, and designed pipeline based on structural homology, ScanNet achieved the highest accuracy of 87.7%. While the structural homology baseline performed almost the same as ScanNet, the accuracy dropped quickly when meeting with the unseen fold during the test because of its strong dependence on the homology that was previously developed. Therefore, it is crucial to understand what ScanNet has actually learned. Specifically, does the network only memorize the training data, or does it really understand the underlying protein–protein binding principles? Detailed visualization and interpretation were explored to illustrate the learned atom-wise representations and amino acid-wise representations. The network has learned different atomic patterns, such as the N–H–O hydrogen bond (Fig. 5(b)), the SH or NH2 side-chain hydrogen donor surrounded by oxygen atoms (Fig. 5(c)), a carbon in the vicinity of a methyl group and an aromatic ring (Fig. 5(d)), and so on. The detected pattern with solvent-exposed residues frequently appearing in the protein–protein interface (Fig. 5(e)), such as arginine (R), was positively correlated with the output probability of PPBS. However, that with the buried hydrophobic amino acids (Fig. 5(f)), such as phenylalanine (F), was negatively correlated with the output probability of PPBS. Interestingly, the pattern with the exposed hydrophobic amino acid surrounded by charged amino acids, which is the hotspot O-ring[102] architecture in protein interfaces, was positively correlated with the output probability (Fig. 5(g)). 2D t-distributed stochastic neighbor embedding (t-SNE) projections further verified that the model has already learned various amino acid-level structural features. 2D t-SNE projections on secondary structures (Fig. 5(h)) clearly illustrated that the model has learned the secondary structural information of the training complexes. With the multi-level knowledge of protein structures, ScanNet captures the underlying chemical principles of protein–protein binding. This SOTA interpretable DL model aids in a deeper understanding of PepPIs and PPIs.

Attention based models

Recurrent neural networks (RNNs) and long short-term memory (LSTM) are the most common models for language modeling and machine translation.[103] But both RNNs and LSTM suffer from the issue of handling long range dependencies; in other words they become ineffective when there is a significant gap between relevant information and the point where it is needed. The attention mechanism was introduced to address this limitation, which enables the modeling of dependencies without being constrained by their distance in input or output sequences.[104] The attention mechanism is one of the most important developments in natural language processing. Vaswani *et al.* introduced a new form of attention, called self-attention, which relates different positions of a single sequence to obtain a representation of the sequence.[103] A new architectural class, Transformer, was conceived, primarily based on the self-attention mechanism.[104] Transformer consists of multiple encoders and decoders with self-attention layers. The self-attention layer allows the transformer model to process all input words at once and model the relationship between all words in a sentence. The Transformer architecture led to the development of a new language model, called bidirectional encoder representations from transformers (BERT).[105] BERT is designed to pre-train deep bidirectional representations from unlabeled text. It utilizes a ''masked language model'' (MLM) objective, where some tokens from the input are randomly masked, and the model is trained to predict the masked word based on its context from both directions. Numerous deep learning architectures have emerged, either directly employing self-attention mechanisms or drawing inspiration from the

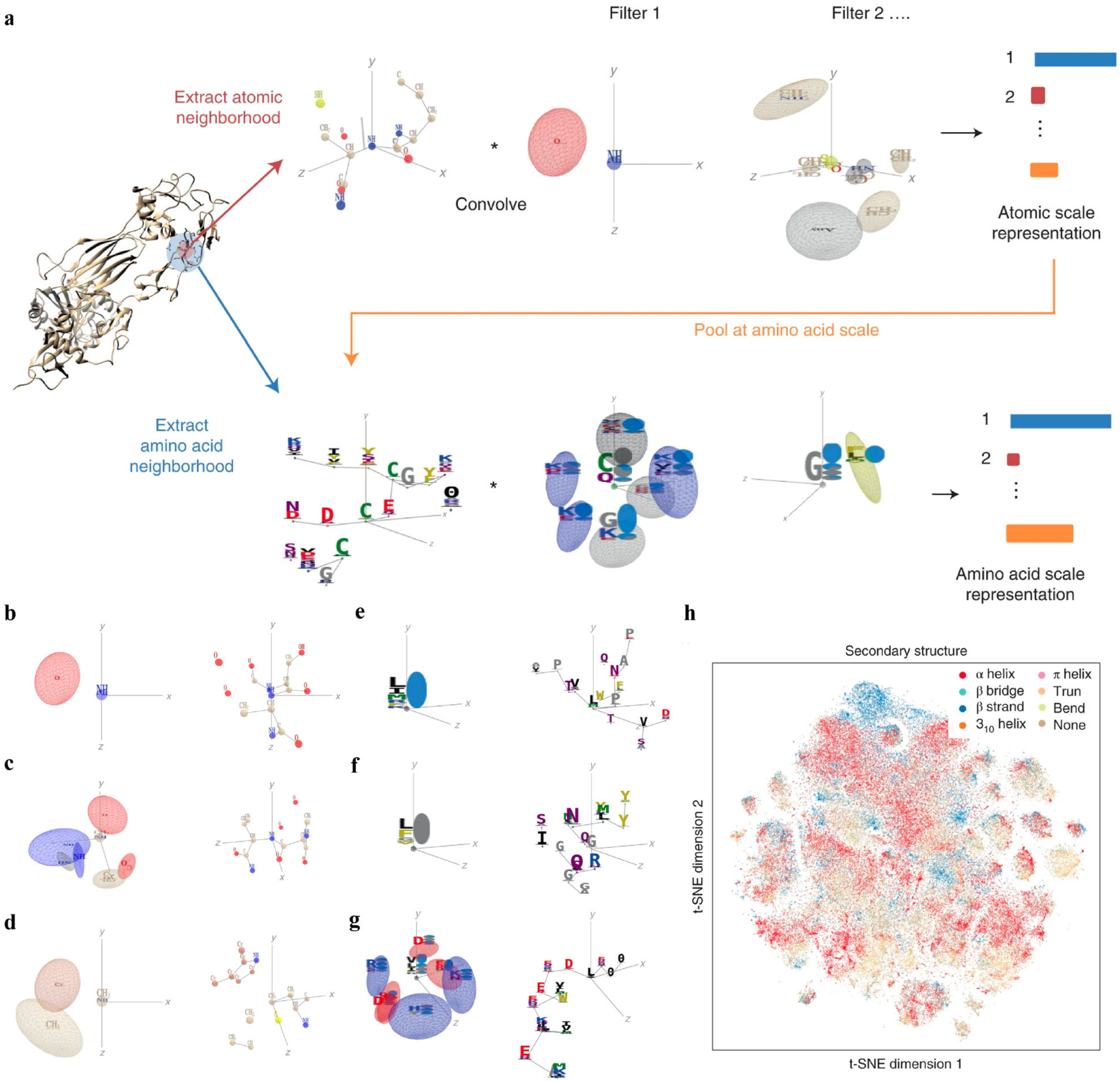


Fig. 5 (a) Overview of the ScanNet model architecture. Point cloud including neighboring atoms' information was first extracted for each atom from the protein structure. Point cloud was then passed through linear filters to detect specific atom interaction patterns, yielding an atomic-scale representation. This representation was pooled at the amino acid scale, concatenated with the extracted neighboring amino acid attributes from the protein structure, and then subject to a similar procedure as before to identify amino acid neighborhood and representations. (b)–(f) Each panel shows one learned atom-level spatio-chemical pattern on the left and the corresponding top-activating neighborhood on the right. (b) N–H–O hydrogen bond, (c) two oxygen atoms and three NH groups in a specific arrangement, and (d) a carbon in the vicinity of a methyl group and an aromatic ring. (e)–(g) Each panel shows one learned amino acid-level spatio-chemical pattern on the left and one corresponding top-activating neighborhood on the right. (e) Solvent-exposed residues, positively correlated with the output probability (r = 0.31) and (f) buried hydrophobic amino acids, negatively correlated with the output probability (r = −0.32). (g) The hotspot O-ring architecture, an exposed hydrophobic amino acid surrounded by exposed, charged amino acids, positively correlated with the output probability (r = 0.29). (h) Two-dimensional projection on the secondary structure of the learned amino acid scale representation using t-SNE. Reproduced with permission from J. Tubiana, D. Schneidman-Duhovny and H. J. Wolfson, *bioRxiv*, 2021. This article is licensed under a CC BY 4.0 International License, permitting unrestricted reproduction and adaptation provided proper crediting to author and source. Copyright 2024 Cold Spring Harbor Laboratory.

Transformer architecture. These advancements have also been applied forward in predicting PepPIs.

Existing ML and DL models for predicting peptide–protein binding sites mainly focus on identifying binding residues on the protein surface. Sequence-based methods typically take protein sequences as inputs, assuming that a protein maintains fixed binding residues across different peptide binders. However, this assumption doesn't hold true for most cellular processes, as various peptides may interact with distinct protein residues to carry out diverse functions. Structure-based methods would require a target protein structure and a peptide sequence, thus limiting their applicability to proteins with available structural data. A novel DL framework for peptide–protein binding prediction was proposed, called CAMP,[42] to address the above limitations. CAMP takes account of information from the sequence of both peptides and target proteins, and also detects crucial binding residues of peptides for peptide drug discovery.

CAMP extracted data from difference sources, including RCSB PDB[30,106] and the known peptide drug–target pairs from DrugBank.[43,107–110] For each PDB complex, protein–ligand interaction predictor (PLIP) is employed to identify non-covalent interactions between the peptide and the protein, considering these interactions as positive samples for training. Additionally, PepBDB[111] aids in determining the binding residues of peptides involved in the specific protein–peptide complexes. Various features are extracted based on their primary sequences to construct comprehensive sequence profiles for peptides and proteins. These features include secondary structures, physicochemical properties, intrinsic disorder tendencies, and evolutionary information.[27,112–115] CAMP utilizes two multi-channel feature extractors to process peptide and protein features separately (Fig. 6). Each extractor contains a numerical channel for numerical features (PSSM and the intrinsic disorder tendency of each residue), along with multiple categorical channels for diverse categorical features (raw amino acid, secondary structure, polarity and hydropathy properties). Two CNN modules extract hidden contextual features from peptides and proteins. Self-attention layers are also employed to capture long-range dependencies between residues and assess the contribution of each residue to the final interaction. CAMP applies fully connected layers on all integrated features to predict the interaction between proteins and peptides. In addition to binary interaction prediction, CAMP can identify which residue of peptides interacts with target proteins by adding a sigmoid activation function to the output of the peptide CNN module. Compared with three baseline models (DeepDTA,[116] PIPR,[117] NRLMF[118]), CAMP demonstrates consistent better performance with an increase by up to 10% and 15% in terms of area under the curve (AUC) and area under the precision–recall curve (AUPR). To evaluate its ability to identify binding residues of peptides, the predicted label of each residue of the peptide is compared with the real label for four existing peptide binders. The results show that CAMP correctly predicts binding residues and thus provides reliable evidence for peptide drug design.

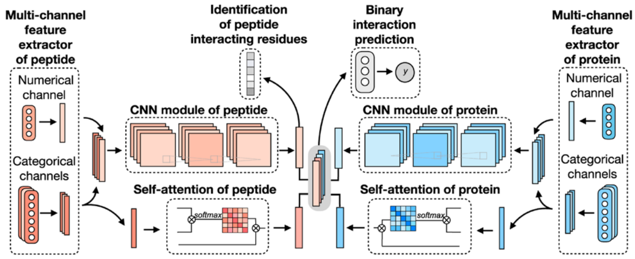

Multi-channel feature extractor of peptide
Numerical channel
Categorical channels
Identification of peptide interacting residues
CNN module of peptide
Self-attention of peptide
softmax
Binary interaction prediction
y
CNN module of protein
Self-attention of protein
softmax
Multi-channel feature extractor of protein
Numerical channel
Categorical channels

Fig. 6 The network architecture of CAMP. For each protein–peptide pair, the numerical and categorical features of peptide and protein sequences are extracted and fed into CNN modules. The outputs of the amino acid representations of the peptide and protein are also fed into the self-attention modules to learn the importance of individual residue to the final prediction. Then the outputs of CNN and self-attention modules are taken together as the input of three fully connected layers to predict the binding score for each peptide–protein pair. The output of CNN modules is also used for predicting the binding score for each residue from the peptide sequence. Adapted with permission from Y. Lei, S. Li, Z. Liu, F. Wan, T. Tian, S. Li, D. Zhao and J. Zeng, *Nat. Commun.*, 2021, **12**, 5465. This article is licensed under the Creative Commons CC BY license, permitting unrestricted reproduction and adaptation provided proper crediting to author and source. Copyright 2024 Springer Nature.

Instead of only applying the self-attention layer, Adbin *et al.* developed a transformer-based architecture known as PepNN, enabling both sequence-based (PepNN-Seq) and structure-based (PepNN-Struct) predictions of peptide binding sites.[44] PepNN takes representations of a protein and a peptide sequence as inputs and generates a confidence score for each residue, indicating the likelihood of being part of binding sites. PepNN-Struct learns a contextual representation of a protein structure through the use of graph attention layers (Fig. 7(a)). In contrast, PepNN-Seq only takes the protein and peptide sequence as inputs (Fig. 7(b)). In the PepNN algorithm, the encoding of the peptide sequence is independent from the protein encoding module, under the assumption that the peptide sequence carries all the necessary information regarding peptide–protein binding. However, in many scenarios, the peptide sequence is not sufficient to determine the bound conformation, as the same peptide can adopt different conformations when bound to different proteins.[119] Motivated by this, PepNN incorporates a multi-head reciprocal attention layer that simultaneously updates the embeddings of both the peptide and protein (Fig. 7(a)). This module attempts to learn the interactions between protein and peptide residues involved in binding.

Another challenge in predicting the protein–peptide binding sites is the limited availability of protein–peptide complex training data. Protein–protein complex information was added to the training set to overcome the limited data issue. Notably, not the entire protein–protein complex data were included, because the interactions between two proteins can be mediated by a linear segment in one protein that contributes to the majority of the interface energy. Pre-training of the model was conducted using a substantial dataset of large protein fragment–protein complexes (717 932).[120] Fine-tuning of the model then took place with a smaller set of peptide–protein complexes (2828), resulting in a considerable enhancement in predictive performance, particularly for the PepNN-Struct model (Fig. 7(c)). PepNN reliably predicts peptide binding sites on an independent test set and three benchmark datasets from the other studies.[27–29] PepNN-Struct surpassed most peptide binding site prediction approaches, achieving a higher AUC score. While PepNN generally exhibits lower MCC than the SOTA method AlphaFold-Multimer in most cases, its independence from multiple sequence alignments may render PepNN more suitable for modeling synthetic PepPIs.

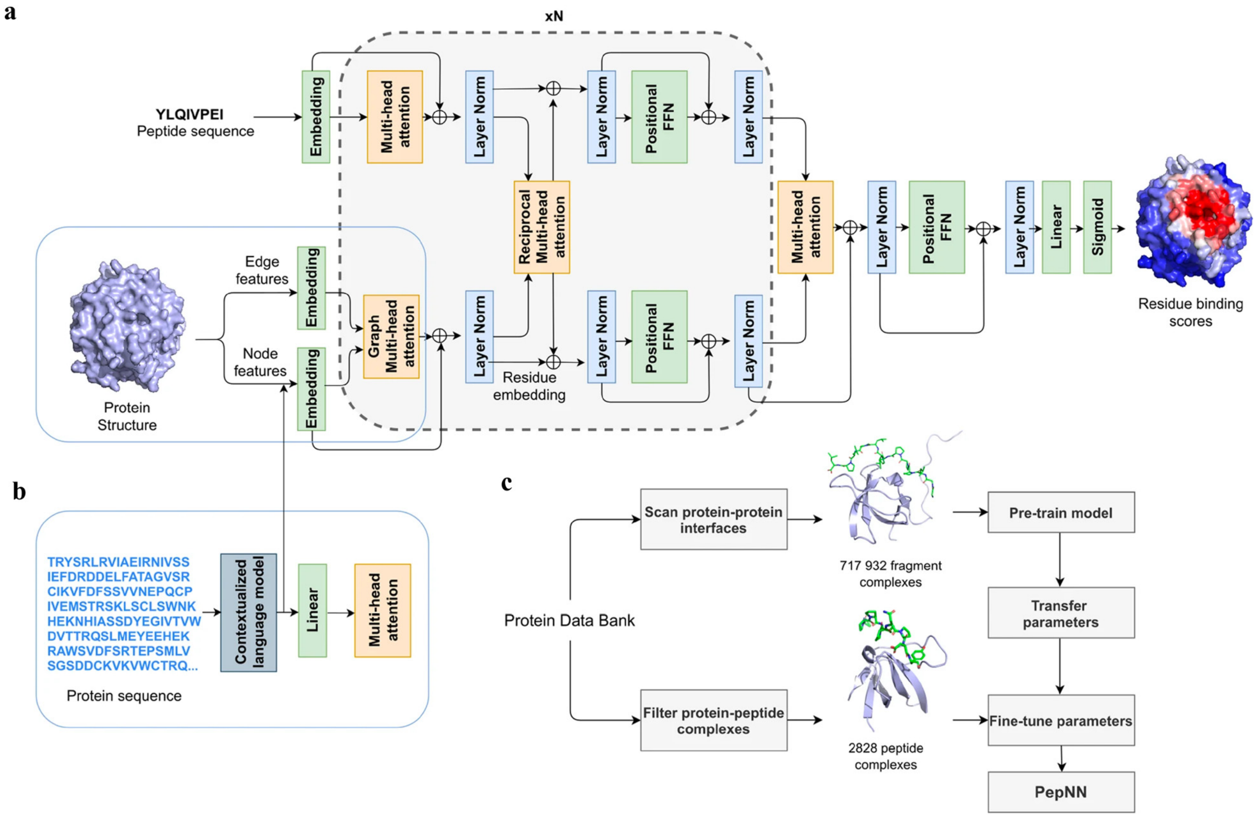


Fig. 7 The model architecture and training procedure of PepNN. (a) The input of PepNN-Struct and model architecture. Attention layers are indicated with orange; normalization layers are indicated with blue and simple transformation layers are indicated with green. (b) The input of PepNN-Seq. (c) Transfer learning pipeline used for training PepNN. Reproduced with permission from O. Abdin, S. Nim, H. Wen and P. M. Kim, *Commun. Biol.*, 2022, **5**, 503. This article is licensed under the Creative Commons CC BY license, permitting unrestricted reproduction and adaptation provided proper crediting to author and source. Copyright 2024 Springer Nature.

While numerous computational methods have been developed for predicting peptide–protein binding sites, many of them need complex data preprocessing to extract features, often resulting in reduced computational efficiency and predictive performance. Wang *et al.* developed an end-to-end predictive model that is independent of feature engineering named PepBCL.[45] This innovative approach leverages pre-trained protein language models to distill knowledge from protein sequences that are relevant to protein structures and functions. Another challenge encountered in identifying protein–peptide binding sites is the issue of imbalanced data. Current work typically constructs a balanced dataset by using under-sampling techniques. However, these techniques remove samples from the majority class to match the size of the minority class. In the PepBCL algorithm, a contrastive learning-based module is introduced to tackle this problem. Unlike conventional under-sampling methods, the contrastive learning module adaptively learns more discriminative representations of the peptide binding residues.

The PepBCL architecture is composed of four essential modules: sequence embedding module, BERT-based encoder module,[105] output module and contrastive learning module.[121,122] In the sequence embedding module, each amino acid of the query sequence is encoded into a pre-trained embedding vector, while the protein sequence is encoded to an embedding matrix. In the BERT-based encoder module, the output from the sequence embedding module undergoes further encoding through BERT to generate a high dimensional representation vector.[123] The representation vector is then passed through a fully connected layer. In the contrastive learning module, the contrastive loss between any two training samples is optimized to generate more discriminative representations of the binding residues. In the output module, the probability of each residue being in a binding site is calculated (Fig. 8(a)). When compared with the existing sequence-based method (SPRINT-Seq,[25] PepBind,[27] Visual,[34] and PepNN-Seq[44]), PepBCL achieves a significant improvement in the precision by 7.1%, AUC by 2.2%, and MCC by 1.3% over best sequence predictor PepBind.[27] Furthermore, PepBCL also outperforms all structure-based methods (*i.e.* Pepsite,[61] Peptimap,[60] SPRINT-Str,[28] and PepNN-Struct[44]) in terms of MCC. The superior performance of PepBCL indicates that DL approaches can automatically learn features from protein sequences to distinguish peptide binding residues and non-binding residues, eliminating the reliance on additional computational tools for feature extraction. When assessing various methods using evaluation metrics, it is observed that recall and MCC tend to be notably low due to the extreme class imbalance in the

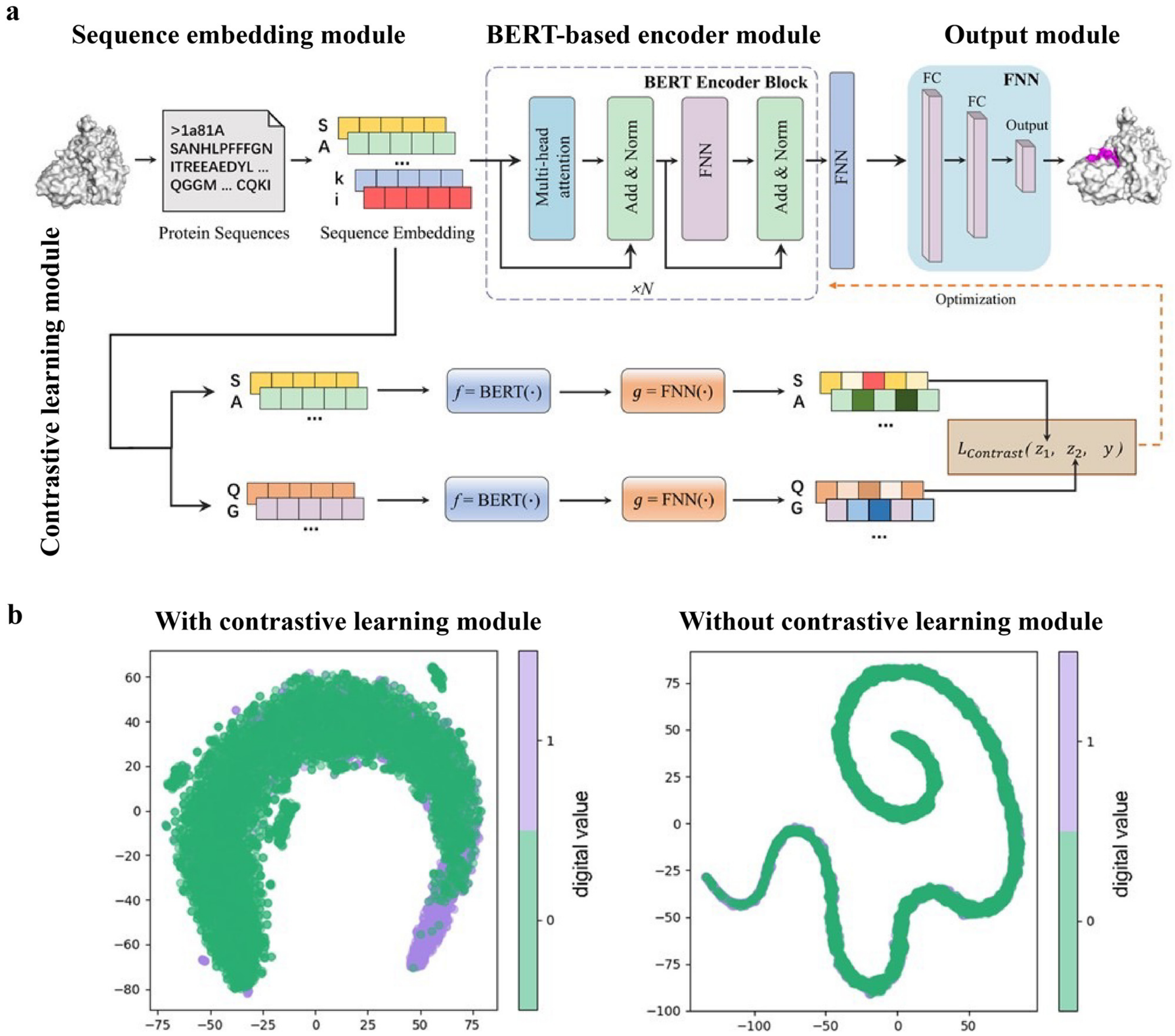


Fig. 8 (a) Architecture of PepBCL consists of four modules. Sequence embedding module: convert protein sequence to sequence embedding for each residue; BERT-based encoder module: extract high-quality representations of each residue in protein; output module: predict the label (binding/non-binding) of residues using fully connected layers; and contrastive learning module: obtain more distinguishable representations by minimizing contrastive loss. (b) t-SNE visualization of the feature space distribution of PepBCL with/without contrast module on testing dataset. Reproduced with permission from R. Wang, J. Jin, Q. Zou, K. Nakai and L. Wei, *Bioinformatics*, 2022, **38**, 3351–3360. Copyright 2024 Oxford University Press.

dataset. This suggests that many true protein–peptide binding residues may be overlooked. However, PepBCL demonstrates improved recall and MCC values, highlighting the effectiveness of the contrastive module in identifying more true peptide binding residues. This enhancement can be attributed to the contrastive learning's ability to extract more discriminative representations, particularly in imbalanced datasets. Fig. 8(b) visually demonstrates the learned feature space with and without the contrastive learning module, showcasing a clearer distribution of binding and non-binding residues in the feature space.

AlphaFold/RoseTTAFold/OmegaFold/ESMFold

Multiple sequence alignment (MSA)-based transformer models, such as AlphaFold2 (AF2, including monomer model[46] and multimer model[54]) and RoseTTAFold,[124] and protein language model (pLM)-based models, such as OmegaFold[50] and ESMFold,[125] have demonstrated remarkable success in predicting the *in silico* folding of monomeric proteins and peptides.[126] However, PepPIs are relatively flexible protein complexes, making it challenging to achieve highly accurate predictions. Therefore, benchmarking these SOTA DL techniques on PepPI predictions could provide structural insights into peptide–protein complexes, for example, binding affinities, conformational dynamics, and interaction interfaces, thus contributing to the advancement of molecular biology and drug discovery.

While the AF2 monomer was originally designed for predicting monomeric protein/peptide structures, it has recently been shown to be successful in predicting PepPIs by Tsaban *et al.*[47] The PepPIs could be represented as the folding of a monomeric protein by connecting the peptide to the C-terminus of the receptor with a poly-glycine linker (Fig. 9(a)), which forms a general idea of how to perform peptide–protein docking using the AF2 monomer model. This method can not only identify the peptide binding regions but also accommodate binding-induced conformational changes of the receptor. AF2 surpassed RoseTTAFold since the latter tended to fold the polyglycine linker into a globular structure or various interactive loops. For a small dataset of 26 PepPI complexes, AF2 achieved a relatively high accuracy (75%) for complexes whose binding motifs have been experimentally characterized. AF2 also outperformed another peptide docking method PIPER-FlexPepDock (PFPD)[127] in terms of both accuracy and speed. Furthermore, accurate predictions were achieved with AF2 pLDDT values above 0.7, further verifying that the AF2 monomer can reliably predict the PepPIs. However, the predicted accuracy became lower (37%) when tested on a larger dataset (96 complexes), indicating that further improvements are needed for more accurate PepPI predictions by the AF2 monomer.

The recent release of the AF2 multimer has resulted in a major improvement in PepPI prediction. Using a set of 99 protein–peptide complexes, Shanker and Sanner[48] compared the performance of the AF2 monomer, AF2 multimer, and OmegaFold on PepPI prediction with their peptide docking software AutoDock CrankPep (ADCP).[91] The new AF2 multimer model with 53% accuracy, which was trained to predict the interfaces of multimeric protein complexes, outperformed

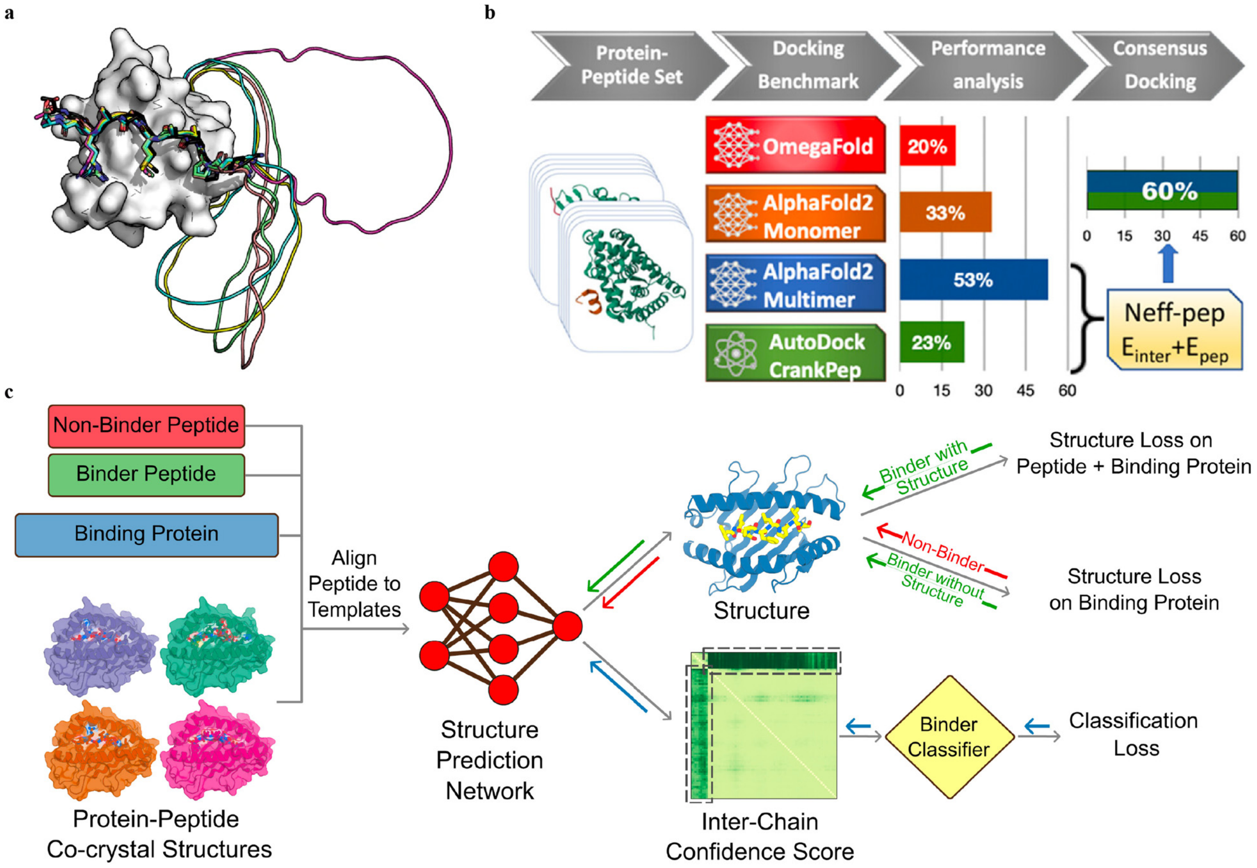


Fig. 9 (a) A successful example (PDBID: 1SSH) of peptide–protein docking with a poly-glycine linker *via* AlphaFold2. This method can dock the peptide at the correct position (native peptide is shown in black, docking peptides are shown in other colors) and identify the linker as an unstructured region (modeled as a circle). Adapted with permission from T. Tsaban, J. K. Varga, O. Avraham, Z. Ben-Aharon, A. Khra-mushin and O. Schueler-Furman, *Nat. Commun.*, 2022, **13**, 176, this article is licensed under the Creative Commons CC BY license, permitting unrestricted reproduction and adaptation provided proper crediting to author and source. Copyright 2024 Springer Nature. (b) AlphaFold2-Multimer model outperforms other DL approaches and achieves remarkable docking success rates of 53% for peptide–protein docking. A designed docking approach combining ADCP and AlphaFold2-Multimer achieves an improved success rate of 60%. Adapted with permission from S. Shanker and M. F. Sanner, *J. Chem. Inf. Model.*, 2023, **63**, 3158–3170. Copyright 2024 American Chemical Society. (c) Mechanism of structure prediction networks for peptide binder classification by fine-tuning AlphaFold2. The input of the model includes the peptide binder and non-binder sequences, protein sequences, and peptide–protein co-crystal structures as templates. After positionally aligning the peptide sequence to the template, the complex structure is predicted with AlphaFold2. A binder classification layer converts the AlphaFold2 output PAE values into a binder/non-binder score. The combined loss function including the structure loss over the entire complex for the peptide binder and over protein only for the non-binder, and classification loss from the binder classification layer, is used for model training. Adapted with permission from A. Motmaen, J. Dauparas, M. Baek, M. H. Abedi, D. Baker and P. Bradley, *Proc. Natl. Acad. Sci. U. S. A.*, 2023, **120**, e2216697120. This article is licensed under a Creative Commons Attribution 4.0 (CC BY) License, permitting unrestricted reproduction and adaptation provided proper crediting to author and source. Copyright 2024 National Academy of Science.

OmegaFold with 20% accuracy and ADCP with 23% accuracy (Fig. 9(b)). However, the AF2 multimer model is only limited to linear peptides, reducing its applicability to cyclized peptides, or peptides with non-standard amino acids. Effective selection from top-ranked poses yielded by both AF2 multimer and ADCP docking tool was found to further enhance the accuracy to 60%. Therefore, DL protein structure prediction models, especially AF2 multimer, have achieved high accuracy in PepPI predictions, though limitations exist. Combining these SOTA DL models with traditional peptide docking tools could be a future direction for further improving the accuracy of PepPI predictions.

Leveraging the highly accurate predictions of protein structures by AF2, Amir Motmaen *et al.*[56] developed a more generalized model for the prediction of PepPIs. The model was accomplished by placing a classifier on top of the AF2 network and fine-tuning the combined network (Fig. 9(c)). AF2 was able to achieve optimal performance and generate the most accurate complex predicted structure models for a large dataset of peptide-major histocompatibility complex (MHC) complexes. This was accomplished by aligning the peptide sequence with the peptide–protein crystal structures as templates. However, a few misclassifications by AF2 underscored the importance of accurately distinguishing binder and non-binder peptides. To address this issue, a logistic regression layer that normalizes the AF2 predicted aligned error (PAE) score into a binder/non-binder score was placed on top of AF2. This resulted in three types of losses being combined and applied to further fine-tune the combined model: structure loss on both peptide and protein for binding peptide–protein complexes, structure loss on protein only for non-binding peptide–protein complexes, and classification loss on binding/non-binding score. The evaluation of the combined model showed a ROC AUC of 0.97 for class 1 and 0.93 for class 2 peptide–MHC interactions. Surprisingly, the fine-tuned model outperformed the previously mentioned HSM model and could also be generalized on PDZ domains (C-terminal peptide recognition domain) and SH3 domains (proline-rich peptide binding domain), despite being trained and fine-tuned only on the peptide–MHC dataset. Therefore, taking advantage of the accurate predictions of protein structures through AF2, fine-tuning the model with existing peptide–protein binding data offers significant boost to PepPI predictions.

Conclusions and future research directions

Peptides, which are short proteins consisting of around 2 to 50 amino acids, are known for their flexibility. This characteristic makes it challenging to achieve highly accurate predictions of PepPIs. A variety of SOTA ML and DL models summarized in this review have been designed and applied to predict PepPIs, which are key to *de novo* peptide drug design.

Apart from their well-documented high efficiency and accuracy requirements, ML/DL methods offer several other advantages in the predictions of PepPIs. Compared to docking or MD simulation methods, ML or DL methods offer diverse options for model inputs. DL methods, such as transformers and language models, have been shown to achieve great success in predicting PepPIs solely based on sequence information. Instead of original sequence or structure information, ML methods can also incorporate multi-level information such as evolutionary information, secondary structures, solvent accessible surface area, and so forth, which could significantly enhance the accuracy of the prediction. Furthermore, more interpretability can be provided by ML/DL methods. The attention mechanism assists in demonstrating the internal dependencies between residues and the contribution of each residue to PepPIs. Graph models capturing multi-scale structure information of peptides and proteins are able to provide insights into the underlying chemical principles of peptide–protein binding and binding patterns. Moreover, ML/DL techniques exhibit a degree of generalizability. Some advanced techniques like transfer learning or one-shot learning models, which have been applied in protein engineering and protein–ligand interaction prediction,[128–131] could facilitate the models trained on certain peptide–protein binding datasets to generalize to other peptide–protein complexes.

Despite their numerous advantages, ML and DL methods also have certain limitations in the prediction of PepPIs, which highlight potential areas for future research. One significant challenge is the issue of imbalanced datasets in the training and testing of PepPI prediction models. Given that peptide binding is typically a rare occurrence, the imbalanced number of positive and negative samples often results in the limited performance of ML/DL models due to the poor understanding of the minority binding class. Consequently, ML/DL methods for PepPI predictions were normally trained based on datasets with a positive-to-negative ratio of 1:1. Both oversampling methods, which duplicate or create new samples in the minority class, and undersampling methods, which delete or merge samples in the majority class, can enhance the model performance on imbalanced classification. Besides, challenges arise when dealing with peptides deeply embedded in the enzyme's active site especially involving cofactors. Accurate predictions for such interactions require high-quality structural training data reflecting correct folding for both peptide and enzyme along with the precise knowledge of buried peptide binding positions and poses. Furthermore, accurate geometric and electronic considerations of cofactors would be necessary to predict the peptide and protein residue interactions with the co-factors. The scarcity of structural training data for such instances results in a relatively worse model performance on PepPIs. Recent efforts, such as RoseTTAFold All-Atom[132] (RFAA), aim to address this challenge. RFAA can model full biological assemblies, including metal cofactors, by training on a comprehensive dataset comprising sequence information, residue pairwise distance from homologous templates, and coordinates of protein–small molecule, protein–metal, and covalently modified protein complexes. As a result, RFAA demonstrates reasonable prediction performance and stands out as the first model capable of predicting arbitrary higher-order biomolecular complexes, encompassing multiple proteins, small molecules, metal ions, and nucleic acids. However, this is a recent development, so there are no applications of RFAA to PepPI prediction. As advancements in structural biology and computational methods continue, it is foreseeable that more sophisticated models will emerge, further enhancing the capability to accurately predict PepPIs, even involving buried peptides and cofactors. Additionally, ML/DL methods often failed in the prediction of PepPIs between intrinsically disordered peptides (IDPs) and proteins. IDPs are abundant in nature, with flexible and disordered structures but adopt stable and well-defined structures upon binding. In these cases, ML/DL methods, particularly structure-based models, tend to fail in predicting binding sites and peptide binding conformations, offering little insights into the binding mechanism. With the enhancement of computing power, high-throughput MD simulations can achieve more accurate predictions of binding sites and peptide/protein conformations as well as a deeper understanding of the mechanism of folding and binding, induced fit (binding then folding), or conformational selection (folding then binding). The integration of MD or quantum chemical insights and ML/DL methods could constitute a promising future research direction of PepPI predictions.

Another future direction is to develop ML/DL models to predict cyclic peptide and protein interaction. Cyclic peptides have emerged as a promising therapeutical modality because of distinct pharmacological characteristics in comparison to small molecules and biologics.[3,133,134] For example, cyclic peptides are more resistant to digestive enzymes like peptidases and exoproteases due to their stable cyclic structures. Cyclic peptides have a broader interaction surface than small-molecule drugs and thus may function as inhibitors with high affinity and selectivity for modulating protein–protein interactions. Furthermore, cyclic peptides exhibit better permeability across cell membranes and are less expensive to synthesize compared to antibodies. However, the development of deep learning models for designing cyclic peptides has faced challenges, mostly due to the small number of available structures. Recently, Rettie *et al.* introduced the AfCycDesign approach, a novel modification of the AlphaFold network for accurate structure prediction and design of cyclic peptides.[135] Standard positional encoding in AlphaFold is based on the position of each amino acid in the linear peptide, with the termini being the maximum distance from each other. AfCycDesign modifies the positional encoding with cyclic offset such that the termini are connected to each other. This approach can accurately predict the structures of cyclic peptides from a single sequence, with 36 out of 49 cases predicted with high confidence (pLDDT > 0.85) matching the native structures with root mean squared deviation (RMSD) < 1.5 Å. Kosugi *et al.* employed the relative positional encoding with cyclic offset to predict protein–cyclic peptide complexes.[136] The cyclic offset was only applied in the cyclic peptide region, while the positional encoding of the protein region remained the default one. The predictions outperformed state-of-the-art local docking tools for cyclic peptide complexes.

Future research directions should also prioritize the enhancement of model's ability to generate novel peptide sequences to specific target proteins of interest, thereby contributing to *de novo* peptide drug design. An essential way is to fine-tune pre-trained pLM. Introducing noises and perturbations within the peptide latent space of pLM, or masking peptide sequences to facilitate the model to learn the probability distribution of peptide binders, could be explored to generate entirely new peptide sequences. Additionally, diffusion models offer another avenue for achieving the generative tasks. These models possess a deeper understanding of the intricate molecular interactions at the atomic levels, thus enabling the generation of new peptide sequences based on peptide–protein complex structures. The resultant novel peptide sequences can be subsequently validated through MD simulations and *in vitro* and *in vivo* experimental tests. Therefore, developing new generative models or leveraging the pre-trained ML/DL models to facilitate peptide generation represents a noteworthy and promising future for advancing peptide drug design.

In conclusion, ML/DL-guided methods have shown significant potential for the accurate predictions of peptide–protein complex structures and binding sites. These SOTA models will undoubtedly further accelerate the process of peptide drug discovery and design.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

D. S. acknowledges support from the National Institutes of Health, under award no. R35GM142745 and no. R21AI-167693.

Notes and references

1 N. London, B. Raveh and O. Schueler-Furman, *Curr. Opin. Chem. Biol.*, 2013, **17**, 952–959.

2 X. Peng, J. Wang, W. Peng, F.-X. Wu and Y. Pan, *Briefings Bioinf.*, 2017, **18**, 798–819.

3 M. Muttenthaler, G. F. King, D. J. Adams and P. F. Alewood, *Nat. Rev. Drug Discovery*, 2021, **20**, 309–325.

4 L. Wang, N. Wang, W. Zhang, X. Cheng, Z. Yan, G. Shao, X. Wang, R. Wang and C. Fu, *Signal Transduction Targeted Ther.*, 2022, **7**, 48.

5 X.-Y. Meng, H.-X. Zhang, M. Mezei and M. Cui, *Curr. Comput.-Aided Drug Des.*, 2011, **7**, 146–157.

6 J. Wang, A. Alekseenko, D. Kozakov and Y. Miao, *Front. Mol. Biosci.*, 2019, **6**, 112.

7 V. Charitou, S. C. van Keulen and A. M. J. J. Bonvin, *J. Chem. Theory Comput.*, 2022, **18**, 4027–4040.

8 M. F. Lensink, S. Velankar and S. J. Wodak, *Proteins: Struct., Funct., Bioinf.*, 2016, **85**, 359–377.

9 M. F. Lensink, N. Nadzirin, S. Velankar and S. J. Wodak, *Proteins: Struct., Funct., Bioinf.*, 2020, **88**, 916–938.

10 M. Ciemny, M. Kurcinski, K. Kamel, A. Kolinski, N. Alam, O. Schueler-Furman and S. Kmiecik, *Drug Discovery Today*, 2018, **23**, 1530–1537.

11 F. Paul, C. Wehmeyer, E. T. Abualrous, H. Wu, M. D. Crabtree, J. Schöneberg, J. Clarke, C. Freund, T. R. Weikl and F. Noé, *Nat. Commun.*, 2017, **8**, 1095.

12 J. A. Morrone, A. Perez, J. MacCallum and K. A. Dill, *J. Chem. Theory Comput.*, 2017, **13**, 870–876.

13 J. A. Morrone, A. Perez, Q. Deng, S. N. Ha, M. K. Holloway, T. K. Sawyer, B. S. Sherborne, F. K. Brown and K. A. Dill, *J. Chem. Theory Comput.*, 2017, **13**, 863–869.

14 D. Kilburg and E. Gallicchio, *Front. Mol. Biosci.*, 2018, **5**, 22.

15 E. Wang, H. Sun, J. Wang, Z. Wang, H. Liu, J. Z. H. Zhang and T. Hou, *Chem. Rev.*, 2019, **119**, 9478–9508.

16 R. Zou, Y. Zhou, Y. Wang, G. Kuang, H. Ågren, J. Wu and Y. Tu, *J. Chem. Inf. Model.*, 2020, **60**, 1551–1558.

17 M. Zalewski, S. Kmiecik and M. Koliński, *Molecules*, 2021, **26**, 3293.

18 J.-N. Chen, F. Jiang and Y.-D. Wu, *J. Chem. Theory Comput.*, 2022, **18**, 6386–6395.

19 M. Zhang, Q. Su, Y. Lu, M. Zhao and B. Niu, *Med. Chem.*, 2017, **13**, 506–514.

20 R. Casadio, P. L. Martelli and C. Savojardo, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2022, **12**, e1618.

21 F. Soleymani, E. Paquet, H. Viktor, W. Michalowski and D. Spinello, *Comput. Struct. Biotechnol. J.*, 2022, **20**, 5316–5341.

22 X. Hu, C. Feng, T. Ling and M. Chen, *Comput. Struct. Biotechnol. J.*, 2022, **20**, 3223–3233.

23 M. Lee, *Molecules*, 2023, **28**, 5169.

24 T. Tang, X. Zhang, Y. Liu, H. Peng, B. Zheng, Y. Yin and X. Zeng, *Briefings Bioinf.*, 2023, **24**, bbad076.

25 G. Taherzadeh, Y. Yang, T. Zhang, A. W.-C. Liew and Y. Zhou, *J. Comput. Chem.*, 2016, **37**, 1223–1229.

26 J. Yang, A. Roy and Y. Zhang, *Nucleic Acids Res.*, 2012, **41**, D1096–D1103.

27 Z. Zhao, Z. Peng and J. Yang, *J. Chem. Inf. Model.*, 2018, **58**, 1459–1468.

28 G. Taherzadeh, Y. Zhou, A. W.-C. Liew and Y. Yang, *Bioinformatics*, 2017, **34**, 477–484.

29 I. Johansson-Åkhe, C. Mirabello and B. Wallner, *Sci. Rep.*, 2019, **9**, 4267.

30 H. M. Berman, *Nucleic Acids Res.*, 2000, **28**, 235–242.

31 S. Shafiee, A. Fathi and G. Taherzadeh, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2023, **20**, 2029–2040.

32 J. M. Cunningham, G. Koytiger, P. K. Sorger and M. AlQuraishi, *Nat. Methods*, 2020, **17**, 175–183.

33 UniProt Consortium, *Nucleic Acids Res.*, 2018, **47**, D506–D515.

34 W. Wardah, A. Dehzangi, G. Taherzadeh, M. A. Rashid, M. Khan, T. Tsunoda and A. Sharma, *J. Theor. Biol.*, 2020, **496**, 110278.

35 I. Kozlovskii and P. Popov, *J. Chem. Inf. Model.*, 2021, **61**, 3814–3823.

36 I. Johansson-Åkhe, C. Mirabello and B. Wallner, *Front. bioinform*, 2021, **1**, 763102.

37 J. Tubiana, D. Schneidman-Duhovny and H. J. Wolfson, *bioRxiv*, 2021, preprint, DOI: **10.1101/2021.09.05.459013**.

38 P. J. Kundrotas, I. Anishchenko, T. Dauzhenka, I. Kotthoff, D. Mnevets, M. M. Copeland and I. A. Vakser, *Protein Sci.*, 2017, **27**, 172–181.

39 M. Baranwal, A. Magner, J. Saldinger, E. S. Turali-Emre, P. Elvati, S. Kozarekar, J. S. VanEpps, N. A. Kotov, A. Violi and A. O. Hero, *BMC Bioinf.*, 2022, **23**, 370.

40 S. Orchard, M. Ammari, B. Aranda, L. Breuza, L. Briganti, F. Broackes-Carter, N. H. Campbell, G. Chavali, C. Chen, N. del Toro, M. Duesbury, M. Dumousseau, E. Galeota, U. Hinz, M. Iannuccelli, S. Jagannathan, R. Jimenez, J. Khadake, A. Lagreid, L. Licata, R. C. Lovering, B. Meldal, A. N. Melidoni, M. Milagros, D. Peluso, L. Perfetto, P. Porras, A. Raghunath, S. Ricard-Blum, B. Roechert, A. Stutz, M. Tognolli, K. van Roey, G. Cesareni and H. Hermjakob, *Nucleic Acids Res.*, 2013, **42**, D358–D363.

41 D. Szklarczyk, A. L. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, M. Simonovic, N. T. Doncheva, J. H. Morris, P. Bork, L. J. Jensen and C. von Mering, *Nucleic Acids Res.*, 2018, **47**, D607–D613.

42 Y. Lei, S. Li, Z. Liu, F. Wan, T. Tian, S. Li, D. Zhao and J. Zeng, *Nat. Commun.*, 2021, **12**, 5465.

43 D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, N. Assempour, I. Iynkkaran, Y. Liu, A. Maciejewski, N. Gale, A. Wilson, L. Chin, R. Cummings, D. Le, A. Pon, C. Knox and M. Wilson, *Nucleic Acids Res.*, 2017, **46**, D1074–D1082.

44 O. Abdin, S. Nim, H. Wen and P. M. Kim, *Commun. Biol.*, 2022, **5**, 503.

45 R. Wang, J. Jin, Q. Zou, K. Nakai and L. Wei, *Bioinformatics*, 2022, **38**, 3351–3360.

46 J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli and D. Hassabis, *Nature*, 2021, **596**, 583–589.

47 T. Tsaban, J. K. Varga, O. Avraham, Z. Ben-Aharon, A. Khramushin and O. Schueler-Furman, *Nat. Commun.*, 2022, **13**, 176.

48 S. Shanker and M. F. Sanner, *J. Chem. Inf. Model.*, 2023, **63**, 3158–3170.

49 M. Mirdita, L. von den Driesch, C. Galiez, M. J. Martin, J. Söding and M. Steinegger, *Nucleic Acids Res.*, 2016, **45**, D170–D176.

50 R. Wu, F. Ding, R. Wang, R. Shen, X. Zhang, S. Luo, C. Su, Z. Wu, Q. Xie, B. Berger, J. Ma and J. Peng, *bioRxiv*, 2022, preprint, DOI: **10.1101/2022.07.21.500999**.

51 B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey and C. H. Wu, *Bioinformatics*, 2014, **31**, 926–932.

52 K. Weissenow, M. Heinzinger and B. Rost, *Structure*, 2022, **30**, 1169–1177.

53 X. Robin, J. Haas, R. Gumienny, A. Smolinski, G. Tauriello and T. Schwede, *Proteins: Struct., Funct., Bioinf.*, 2021, **89**, 1977–1986.

54 R. Evans, M. O'Neill, A. Pritzel, N. Antropova, A. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell, J. Yim, O. Ronneberger, S. Bodenstein, M. Zielinski, A. Bridgland, A. Potapenko, A. Cowie, K. Tunyasuvunakool, R. Jain, E. Clancy, P. Kohli, J. Jumper and D. Hassabis, *bioRxiv*, 2021, preprint, DOI: 10.1101/2021.10.04.463034.

55 U. Ghani, I. Desta, A. Jindal, O. Khan, G. Jones, N. Hashemi, S. Kotelnikov, D. Padhorny, S. Vajda and D. Kozakov, *bioRxiv*, 2021, preprint, DOI: 10.1101/2021.09.07.459290.

56 A. Motmaen, J. Dauparas, M. Baek, M. H. Abedi, D. Baker and P. Bradley, *Proc. Natl. Acad. Sci. U. S. A.*, 2023, **120**, e2216697120.

57 S. Altschul, *Nucleic Acids Res.*, 1997, **25**, 3389–3402.

58 R. Heffernan, K. Paliwal, J. Lyons, A. Dehzangi, A. Sharma, J. Wang, A. Sattar, Y. Yang and Y. Zhou, *Sci. Rep.*, 2015, **5**, 11476.

59 J. Meiler, A. Zeidler, F. Schmaschke and M. Muller, *J. Mol. Model.*, 2001, **7**, 360–369.

60 A. Lavi, C. H. Ngan, D. Movshovitz-Attias, T. Bohnuud, C. Yueh, D. Beglov, O. Schueler-Furman and D. Kozakov, *Proteins: Struct., Funct., Bioinf.*, 2013, **81**, 2096–2105.

61 E. Petsalaki, A. Stark, E. García-Urdiales and R. B. Russell, *PLoS Comput. Biol.*, 2009, **5**, e1000335.

62 S. Liang, *Nucleic Acids Res.*, 2006, **34**, 3698–3707.

63 B. Li, S. Turuvekere, M. Agrawal, D. La, K. Ramani and D. Kihara, *Proteins: Struct., Funct., Bioinf.*, 2008, **71**, 670–683.

64 R. J. Weatheritt and T. J. Gibson, *Trends Biochem. Sci.*, 2012, **37**, 333–341.

65 J. Yang, A. Roy and Y. Zhang, *Bioinformatics*, 2013, **29**, 2588–2595.

66 W. Kabsch and C. Sander, *Biopolymers*, 1983, **22**, 2577–2637.

67 T. Hamelryck, *Proteins: Struct., Funct., Bioinf.*, 2005, **59**, 38–48.

68 J. R. López-Blanco, J. I. Aliaga, E. S. Quintana-Ortí and P. Chacón, *Nucleic Acids Res.*, 2014, **42**, W271–W276.

69 E. C. Dykeman and O. F. Sankey, *J. Phys.: Condens. Matter*, 2010, **22**, 423202.

70 M. Ester, H. P. Kriegel, J. Sander and X. Xiaowei, kdd, 1996, 96, 226-231.

71 N. Babault, F. Cordier, M. Lafage, J. Cockburn, A. Haouz, C. Prehaud, F. A. Rey, M. Delepierre, H. Buc, M. Lafon and N. Wolff, *Structure*, 2011, **19**, 1518–1524.

72 C. Camacho-Gómez, S. Salcedo-Sanz and D. Camacho, *Springer Tracts in Nature-Inspired Computing*, Springer, Singapore, 2021, pp.25–45.

73 R. Polikar, *IEEE Circuits Syst. Mag.*, 2006, **6**, 21–45.

74 M. AlQuraishi, G. Koytiger, A. Jenney, G. MacBeath and P. K. Sorger, *Nat. Genet.*, 2014, **46**, 1363–1371.

75 M. L. Miller, L. J. Jensen, F. Diella, C. Jørgensen, M. Tinti, L. Li, M. Hsiung, S. A. Parker, J. Bordeaux, T. Sicheritz-Ponten, M. Olhovsky, A. Pasculescu, J. Alexander, S. Knapp, N. Blom, P. Bork, S. Li, G. Cesareni, T. Pawson, B. E. Turk, M. B. Yaffe, S. Brunak and R. Linding, *Sci. Signaling*, 2008, **1**, ra2.

76 K. Kundu, M. Mann, F. Costa and R. Backofen, *Bioinformatics*, 2014, **30**, 2668–2669.

77 H. Schmidt, S. Hoffmann, T. Tran, M. Stoldt, T. Stangler, K. Wiesehan and D. Willbold, *J. Mol. Biol.*, 2007, **365**, 1517–1532.

78 G. Fernandez-Ballester, C. Blanes-Mira and L. Serrano, *J. Mol. Biol.*, 2004, **335**, 619–629.

79 C. H. Lee, B. Leung, M. A. Lemmon, J. Zheng, D. Cowburn, J. Kuriyan and K. Saksela, *EMBO J.*, 1995, **14**, 5006–5015.

80 A. Zarrinpar, R. P. Bhattacharyya and W. A. Lim, *Sciences*, 2003, **2003**, re8.

81 K. O'Shea and R. Nash, *arXiv*, 2022, preprint, arXiv:1511.08458, DOI: 10.48550/ARXIV.1511.08458.

82 Y. Yang, R. Heffernan, K. Paliwal, J. Lyons, A. Dehzangi, A. Sharma, J. Wang, A. Sattar and Y. Zhou, *Methods in Molecular Biology*, Springer, New York, 2016, pp.55–63.

83 S. Kawashima, H. Ogata and M. Kanehisa, *Nucleic Acids Res.*, 1999, **27**, 368–369.

84 A. Fout, J. Byrd, B. Shariat and A. Ben-Hur, *Protein interface prediction using graph convolutional networks*, 2017, pp.1–10.

85 Y. Cao and Y. Shen, *Proteins: Struct., Funct., Bioinf.*, 2020, **88**, 1091–1099.

86 Z. Gao, C. Jiang, J. Zhang, X. Jiang, L. Li, P. Zhao, H. Yang, Y. Huang and J. Li, *Nat. Commun.*, 2023, **14**, 1093.

87 Y. Huang, S. Wuchty, Y. Zhou and Z. Zhang, *Briefings Bioinf.*, 2023, **24**, 1–10.

88 M. Réau, N. Renaud, L. C. Xue and A. M. J. J. Bonvin, *Bioinformatics*, 2022, **39**, btac759.

89 B. Sanchez-Lengeling, E. Reif, A. Pearce and A. Wiltschko, *Distill*, 2021, **6**, e33.

90 O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel and T. Langer, *Drug Discovery Today: Technol.*, 2020, **37**, 1–12.

91 S. Zhang, H. Tong, J. Xu and R. Maciejewski, *Comput. Soc. Netw.*, 2019, **6**, 11.

92 J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li and M. Sun, *AI Open*, 2020, **1**, 57–81.

93 D. Kozakov, R. Brenke, S. R. Comeau and S. Vajda, *Proteins: Struct., Funct., Bioinf.*, 2006, **65**, 392–406.

94 M. Remmert, A. Biegert, A. Hauser and J. Söding, *Nat. Methods*, 2011, **9**, 173–175.

95 T. M.-K. Cheng, T. L. Blundell and J. Fernandez-Recio, *Proteins: Struct., Funct., Bioinf.*, 2007, **68**, 503–515.

96 B. Pierce and Z. Weng, *Proteins: Struct., Funct., Bioinf.*, 2007, **67**, 1078–1086.

97 X. Agirrezabala, E. Schreiner, L. G. Trabuco, J. Lei, R. F. Ortiz-Meoz, K. Schulten, R. Green and J. Frank, *EMBO J.*, 2011, **30**, 1497–1507.

98 S. Basu, F. Söderquist and B. Wallner, *J. Comput.-Aided Mol. Des.*, 2017, **31**, 453–466.

99 E. Tayeb-Fligelman, O. Tabachnikov, A. Moshe, O. Goldshmidt-Tran, M. R. Sawaya, N. Coquelle, J.-P. Colletier and M. Landau, *Science*, 2017, **355**, 831–833.

100 Y. Wang, H. Weng, J. F. Song, Y. H. Deng, S. Li and H. B. Liu, *Mol. Med. Rep.*, 2017, **16**, 2714–2720.

101 M. Chu, M. Zhou, C. Jiang, X. Chen, L. Guo, M. Zhang, Z. Chu and Y. Wang, *Front. Immunol.*, 2018, **9**, 862.

102 A. A. Bogan and K. S. Thorn, *J. Mol. Biol.*, 1998, **280**, 1–9.

103 A. Sherstinsky, *arXiv*, 2018, preprint, arXiv:1808.03314, DOI: 10.48550/ARXIV.1808.03314.

104 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, *bioRxiv*, 2017, preprint, DOI: 10.48550/ARXIV.1706.03762.

105 J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, *arXiv*, 2018, preprint, arXiv:1810.04805, DOI: 10.48550/ARXIV.1810.04805.

106 S. K. Burley, H. M. Berman, C. Bhikadiya, C. Bi, L. Chen, L. D. Costanzo, C. Christie, K. Dalenberg, J. M. Duarte, S. Dutta, Z. Feng, S. Ghosh, D. S. Goodsell, R. K. Green, V. Guranović, D. Guzenko, B. P. Hudson, T. Kalro, Y. Liang, R. Lowe, H. Namkoong, E. Peisach, I. Periskova, A. Prlić, C. Randle, A. Rose, P. Rose, R. Sala, M. Sekharan, C. Shao, L. Tan, Y.-P. Tao, Y. Valasatava, M. Voigt, J. Westbrook, J. Woo, H. Yang, J. Young, M. Zhuravleva and C. Zardecki, *Nucleic Acids Res.*, 2018, **47**, D464–D474.

107 D. S. Wishart, *Nucleic Acids Res.*, 2006, **34**, D668–D672.

108 D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam and M. Hassanali, *Nucleic Acids Res.*, 2007, **36**, D901–D906.

109 C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, Y. Djoumbou, R. Eisner, A. C. Guo and D. S. Wishart, *Nucleic Acids Res.*, 2010, **39**, D1035–D1041.

110 V. Law, C. Knox, Y. Djoumbou, T. Jewison, A. C. Guo, Y. Liu, A. Maciejewski, D. Arndt, M. Wilson, V. Neveu, A. Tang, G. Gabriel, C. Ly, S. Adamjee, Z. T. Dame, B. Han, Y. Zhou and D. S. Wishart, *Nucleic Acids Res.*, 2013, **42**, D1091–D1097.

111 Z. Wen, J. He, H. Tao and S.-Y. Huang, *Bioinformatics*, 2018, **35**, 175–177.

112 B. Mészáros, G. Erdös and Z. Dosztányi, *Nucleic Acids Res.*, 2018, **46**, W329–W337.

113 C. N. Magnan and P. Baldi, *Bioinformatics*, 2014, **30**, 2592–2597.

114 F. Madeira, Y. Mi Park, J. Lee, N. Buso, T. Gur, N. Madhusoodanan, P. Basutkar, A. R. N. Tivey, S. C. Potter, R. D. Finn and R. Lopez, *Nucleic Acids Res.*, 2019, **47**, W636–W641.

115 T. Hamp and B. Rost, *Bioinformatics*, 2015, **31**, 1945–1950.

116 H. Öztürk, A. Özgür and E. Ozkirimli, *Bioinformatics*, 2018, **34**, i821–i829.

117 M. Chen, C. J. T. Ju, G. Zhou, X. Chen, T. Zhang, K.-W. Chang, C. Zaniolo and W. Wang, *Bioinformatics*, 2019, **35**, i305–i314.

118 Y. Liu, M. Wu, C. Miao, P. Zhao and X.-L. Li, *PLoS Comput. Biol.*, 2016, **12**, e1004760.

119 A. Mohan, C. J. Oldfield, P. Radivojac, V. Vacic, M. S. Cortese, A. K. Dunker and V. N. Uversky, *J. Mol. Biol.*, 2006, **362**, 1043–1059.

120 Y. Sedan, O. Marcu, S. Lyskov and O. Schueler-Furman, *Nucleic Acids Res.*, 2016, **44**, W536–W541.

121 T. Chen, S. Kornblith, M. Norouzi and G. Hinton, Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 1597-1607.

122 K. He, H. Fan, Y. Wu, S. Xie and R. Girshick, *arXiv*, 2019, preprint, arXiv:1911.05722, DOI: 10.48550/ARXIV.1911.05722.

123 A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik and B. Rost, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022, **44**, 7112–7127.

124 M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, C. Millán, H. Park, C. Adams, C. R. Glassman, A. DeGiovanni, J. H. Pereira, A. V. Rodrigues, A. A. van Dijk, A. C. Ebrecht, D. J. Opperman, T. Sagmeister, C. Buhlheller, T. Pavkov-Keller, M. K. Rathinaswamy, U. Dalwadi, C. K. Yip, J. E. Burke, K. C. Garcia, N. V. Grishin, P. D. Adams, R. J. Read and D. Baker, *Science*, 2021, **373**, 871–876.

125 Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido and A. Rives, *Science*, 2023, **379**, 1123–1130.

126 E. F. McDonald, T. Jones, L. Plate, J. Meiler and A. Gulsevin, *Structure*, 2023, **31**, 111–119.

127 N. Alam, O. Goldstein, B. Xia, K. A. Porter, D. Kozakov and O. Schueler-Furman, *PLoS Comput. Biol.*, 2017, **13**, e1005905.

128 Z. Shamsi, M. Chan and D. Shukla, *J. Phys. Chem. B*, 2020, **124**, 3845–3854.

129 J. Horne and D. Shukla, *Ind. Eng. Chem. Res.*, 2022, **61**, 6235–6245.

130 H. Altae-Tran, B. Ramsundar, A. S. Pappu and V. Pande, *ACS Cent. Sci.*, 2017, **3**, 283–293.

131 X. Mi and D. Shukla, *J. Phys. Chem. B*, 2022, **126**, 1492–1503.

132 R. Krishna, J. Wang, W. Ahern, P. Sturmfels, P. Venkatesh, I. Kalvet, G. R. Lee, F. S. Morey-Burrows, I. Anishchenko, I. R. Humphreys, R. McHugh, D. Vafeados, X. Li, G. A. Sutherland, A. Hitchcock, C. N. Hunter, M. Baek, F. DiMaio and D. Baker, *bioRxiv*, 2023, preprint, DOI: 10.1101/2023.10.09.561603.

133 N. Tsomaia, *Eur. J. Med. Chem.*, 2015, **94**, 459–470.

134 A. A. Vinogradov, Y. Yin and H. Suga, *J. Am. Chem. Soc.*, 2019, **141**, 4167–4181.

135 S. A. Rettie, K. V. Campbell, A. K. Bera, A. Kang, S. Kozlov, J. De La Cruz, V. Adebomi, G. Zhou, F. DiMaio, S. Ovchinnikov and G. Bhardwaj, *bioRxiv*, 2023, preprint, DOI: 10.1101/2023.02.25.529956.

136 T. Kosugi and M. Ohue, *Int. J. Mol. Sci.*, 2023, **24**, 13257.