scientific reports

Check for updates

OPEN

PepCNN deep learning tool for predicting peptide binding residues in proteins using sequence, structural, and language model features

Abel Chandra[1]✉, Alok Sharma[1,2,3]✉, Iman Dehzangi[4,5], Tatsuhiko Tsunoda[2,3,6] & Abdul Sattar[1]

Protein–peptide interactions play a crucial role in various cellular processes and are implicated in abnormal cellular behaviors leading to diseases such as cancer. Therefore, understanding these interactions is vital for both functional genomics and drug discovery efforts. Despite a significant increase in the availability of protein–peptide complexes, experimental methods for studying these interactions remain laborious, time-consuming, and expensive. Computational methods offer a complementary approach but often fall short in terms of prediction accuracy. To address these challenges, we introduce PepCNN, a deep learning-based prediction model that incorporates structural and sequence-based information from primary protein sequences. By utilizing a combination of half-sphere exposure, position specific scoring matrices from multiple-sequence alignment tool, and embedding from a pre-trained protein language model, PepCNN outperforms state-of-the-art methods in terms of specificity, precision, and AUC. The PepCNN software and datasets are publicly available at https://github.com/abelavit/PepCNN.git.

Protein–peptide interactions are pivotal for a myriad of cellular functions including metabolism, gene expression, and DNA replication[1,2]. These interactions are essential to cellular health but can also be implicated in pathological conditions like viral infections and cancer[3]. Understanding these interactions at a molecular level holds the potential for breakthroughs in therapeutic interventions and diagnostic methods. Remarkably, small peptides mediate approximately 40% of these crucial interactions[4].

Traditional experimental approaches to study protein–peptide interactions, despite advances in structural biology, have significant limitations[5]. They are often costly, time-consuming, and technically challenging due to factors such as small peptide sizes[6], weak binding affinities[7], and peptide flexibility[8]. On the other hand, computational methods offer a complementary approach but are also encumbered by issues related to prediction accuracy and computational efficiency. This is often due to the limitations of current algorithms for the inherently complex nature of protein–peptide interactions.

Computational methods aimed at predicting protein–peptide interactions primarily belong to two distinct categories: structure-based and sequence-based. In the realm of structure-based models like PepSite[9], SPRINT-Str[10], and Peptimap[11] leverage an array of structural attributes, such as Accessible Surface Area (ASA), Secondary Structure (SS), and Half-Sphere Exposure (HSE), to make their predictions. Conversely, sequence-based methods like SPRINT-Seq[12], PepBind[13], Visual[14], PepNN-Seq[15], and PepBCL[16], utilize machine learning algorithms and various features, including amino acid sequences, physicochemical properties, and evolutionary information. Notably, PepBind[13] was the first to incorporate intrinsic disorder into feature design, acknowledging its relevance

[1]Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, Australia. [2]Laboratory for Medical Science Mathematics, Department of Biological Sciences, School of Science, The University of Tokyo, Tokyo, Japan. [3]Laboratory for Medical Science Mathematics, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan. [4]Department of Computer Science, Rutgers University, Camden, NJ, USA. [5]Center for Computational and Integrative Biology, Rutgers University, Camden, USA. [6]Laboratory for Medical Science Mathematics, Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Tokyo, Japan. ✉email: abel.chandra@griffithuni.edu.au; alok.fj@gmail.com

to protein–peptide interactions[17]. Most recently, SPPPred[18] employed both structure-based and sequence-based attributes such as HSE, SS, ASA, PSSM, and physicochemical properties to predict protein–peptide binding residues.

The rise of deep learning technologies has added another dimension to the computational proteomics landscape. Various algorithms now facilitate the conversion of protein features into image-like formats, making them compatible with deep learning architectures such as Convolutional Neural Network (CNN)[19]. Transformer-based models have also emerged as powerful tools for sequence representation[20], often outperforming traditional models by capturing long-range interactions within the sequence. For example, Wardah et al.[14] introduced a CNN-based method called Visual, which encodes protein sequences as image-like representations to predict peptide-binding residues in proteins. Abdin et al.[15] unveiled PepNN-Seq, a method leveraging the capabilities of a pre-trained contextualized language model named ProtBert[20] for protein sequence embedding. Most recently, Wang et al.[16] used ProtBert[20] in a contrastive learning framework for predicting protein–peptide binding residues.

Deep learning algorithms, a specialized subset of machine learning, have shown considerable promise in addressing complex challenges in protein science and structural biology[22,23]. These algorithms, inspired by human cognitive processes, employ artificial neural networks to learn complex data representations[24,25]. Compared to the traditional machine learning framework like Random Forest (RF) and Support Vector Machines (SVM), deep learning models excel in autonomously discovering patterns and features from data[26]. Initially popularized in fields like medical imaging, speech recognition, computer vision, and natural language processing, these algorithms have marked milestones such as predicting folding of proteins with remarkable accuracy, making them particularly effective when applied to large and complex data[27]. Given the data-intensive nature of modern biotechnological research, proteomics is increasingly becoming a fertile ground for the application of deep learning technologies[28–30].

CNNs[31] have demonstrated exceptional prowess in image classification tasks, thereby suggesting their applicability to other forms of spatial data, including protein structures[32,33]. Their ability to preserve spatial hierarchies within the data makes them uniquely suited for applications in proteomics. Concurrently, advancements in natural language processing have facilitated the development of pre-trained contextualized language models specifically designed for protein biology, further enriching computational tools available for the field[34,35].

Motivated by these technological leaps, we designed PepCNN, an innovative model that synergistically integrates protein sequence embeddings from protein language model (pLM) with CNN. Our method represents a groundbreaking, consensus-based approach by amalgamating sequence-based features derived from ProtT5-XL-UniRef50, transformer language model by Elnaggar et al.[20] (herein called ProtT5) with traditional sequence-based (Position Specific Scoring Matrices (PSSMs)) and structure-based attributes to train a one-dimensional (1D) CNN, as shown in Fig. 1. Rigorous evaluations underscore that PepCNN sets a new benchmark, outclassing existing methods such as the recent sequence-based PepBCL, PepNN-Seq that utilizes a pre-trained language model, PepBind with intrinsic disorder features, SPRINT-Str with its emphasis on structural features like ASA, SS, and HSE, and most lately the SPPPred method that incoporates both structural and sequence-based features. The marked superiority of PepCNN over these methodologies, in both input requirements and predictive performance, promises not only to redefine computational methods but also to accelerate drug discovery, enhance our understanding of disease mechanisms, and pioneer new computational approaches in bioinformatics.

Results

Experimental setup

We used two widely used benchmark datasets in this study to fairly assess and compare our proposed method with the existing approaches. These datasets are commonly used by recent state-of-the-art methods for model training and test in order to carry out evaluation and comparisons[16]. We also followed the same process for a fair comparison. The two datasets were initially obtained from the BioLiP database[36] and sequences with a redundancy of > 30% sequence identity were removed using 'blastclust' in the BLAST package[37]. We addressed the issue of class imbalance in the training set of our datasets by employing random under-sampling[38,39]. This ensures that our model is not biased towards any particular class and can generalize well during evaluation. A residue in a protein sequence is said to be binding if any of its heavy atom is within 3.5 Å (angstrom) from a heavy atom in the peptide[12] found during lab experimentation. The resulting 1279 peptide-binding proteins contain 290,943 non-binding residues (experimental label = 0) and 16,749 binding residues (experimental label = 1). We designated the two datasets as Datasets 1 and 2, respectively, to make the discussions easier. Table 1 displays the datasets' executive summary. The following subsections describe the specifics of the datasets for model training and evaluation.

Dataset 1

In Dataset 1, the test set (TE125) was proposed by Taherzadeh et al.[10] in their structure-based approach called SPRINT-Str. To create this set, they firstly selected proteins which were thirty amino acids or more in length and contained three or more binding residues. TE125 was then constructed by randomly selecting 10% of the proteins and the remaining were assigned to the training set. There are 29,154 non-binding residues and 1716 binding residues in the 125 proteins that make up the TE125 set. In this work, we followed a similar procedure as Taherzadeh et al.[10] to construct our training set, i.e. selecting proteins if they had more than thirty amino acids and contained three or more binding residues. As a result, 1,115 proteins were obtained for training which constituted of 251,770 non-binding residues and 14,942 binding residues. These numbers clearly show that there is an imbalance ratio of around 1:17 between the binding and non-binding residues. This can bias any model towards the classification of non-binding residues over the classification of binding residues if trained directly on this training set. Therefore, random under-sampling technique was applied to the train set which resulted in

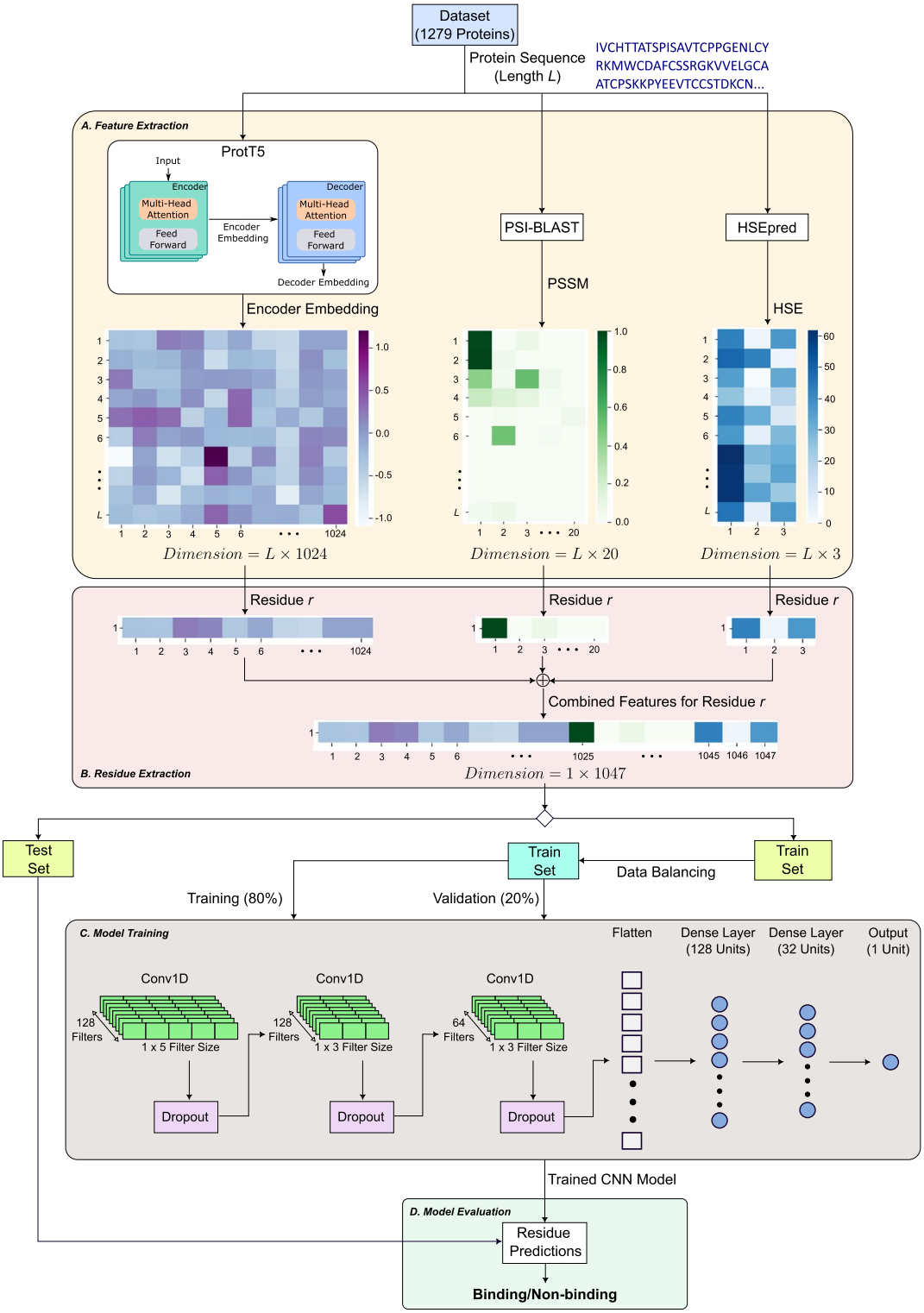

Figure 1. Flow diagram of the proposed work for the prediction of binding and non-binding residues. (**A**) Feature extraction component is where the features for each proteins are generated. (**B**) Residue extraction component is where the feature set pertaining to each residue is extracted. (**C**) The model training block contains the CNN model training step using 80% of the training set to train the network, and the remaining 20% for validation. (**D**) The model evaluation component is where the residues in the test set are predicted to be binding or non-binding using the trained CNN model. Figure created using Inkscape software[21].

	Dataset 1		Dataset 2	
	TE125 (test set)	TR1115 (train set)	TE639 (test set)	TR640 (train set)
No. of proteins	125	1115	639	640
No. of residues	30,870	266,712	150,330	157,362
No. of non-binding residues	29,154	251,770	141,840	149,103
No. of binding residues	1716	14,942	8490	8259

Table 1. Breakdowns of Dataset 1 and Dataset 2.

a total of 37,355 residues. From this training set, 80% of the residues were actually used for training the model, and the remaining 20% of the residues were used as the validation set during the training stage.

Dataset 2

In Dataset 2, the test set (TE639) was proposed by Zhao et al.[13] in their sequence-based approach called PepBind. They constructed their train and test sets by randomly dividing the 1279 proteins into two equal subsets. There were 141,840 non-binding residues and 8490 binding residues in the 639 proteins that make up the TE639 set. In the training set, there were 640 proteins, but to save training time, 20% of the proteins were selected to train their model. The training set in this work was however created by keeping all of the 640 proteins and this resulted in 149,103 non-binding residues and 8259 binding residues. It is evident that this training set is also highly imbalanced, with an imbalance ratio of 1:18 between the binding and non-binding residues. After the random under-sampling technique, the final number of residues in the training set was therefore 20,647. This final set then underwent a split with 80:20 ratio for the final training and validation set during the model training stage.

Comparison with existing methods

To show the performance of our PepCNN model, we compared the results with nine existing methods. These are: Pepsite[9], Peptimap[11], SPRINT-Seq[12], SPRINT-Str[10], PepBind[13], Visual[14], PepNN-Seq[15], PepBCL[16], and SPPPred[18]. We employed sensitivity, specificity, precision, mathews correlation coefficient (MCC), and area under the receiver operating characteristic (ROC) curve (popularly known as AUC) as our evaluation metrics. Sensitivity measures the true positive rate, specificity indicates the true negative rate, precision signifies the positive predictive value, MCC measures the contrast between the predicted labels and the experimental labels, and AUC represents the model's overall classification ability. Note that all the metrics, except AUC, rely on the probability threshold where varying the threshold would also alter the metric values. AUC metric therefore provides more confidence for evaluating a model's performance.

The results on TE125 and TE639 test sets are shown in Tables 2 and 3, respectively. In the result tables, a threshold value of 0.877 is used in Table 2 and a value of 0.885 is used in Table 3. Since the test sets were also employed by the previous methods, their results in the tables below are taken directly from their work. As seen from the results on TE125 and TE639 test sets, PepCNN (our proposed method) achieves higher performance compared to all of the previous methods.

For TE125 (Table 2), PepCNN achieves 0.254 sensitivity, 0.988 specificity, 0.55 precision, 0.350 MCC, and 0.843 AUC. In comparison to all the previous methods, including the PepBCL method (the best performing method so far), specificity, precision, and AUC have been improved by our method. The biggest improvement was seen on the AUC metric (3.4%), which is a valuable measure for the overall discriminatory capacity of the classifiers[40,41].

The results on TE639 test set is shown in Table 3 where the sensitivity, specificity, precision, MCC, and AUC values obtained by our method are 0.217, 0.986, 0.479, 0.297, and 0.826, respectively. Similar results as TE125

Methods	Sensitivity	Specificity	Precision	MCC	AUC
Pepsite[9]	0.180	0.970	–	0.200	0.610
Peptimap[11]	0.320	0.950	–	0.270	0.630
SPRINT-Seq[12]	0.210	0.960	–	0.200	0.680
SPRINT-Str[10]	0.240	0.980	–	0.290	0.780
PepBind[13]	0.344	–	0.469	0.372	0.793
Visual[14]	**0.670**	0.680	–	0.170	0.730
PepNN-Seq[15]	–	–	–	0.278	0.805
PepBCL[16]	0.315	0.984	0.540	**0.385**	0.815
SPPPred[18]	0.315	0.959	–	0.230	0.710
PepCNN (ours)	0.254	**0.988**	**0.55**	0.350	**0.843**

Table 2. Performances of the proposed PepCNN model and the previous methods on the TE125 test set. The highest values in each column are highlighted in bold.

Methods	Sensitivity	Specificity	Precision	MCC	AUC
PepBind[13]	**0.317**	–	0.450	**0.348**	0.767
PepNN-Seq[15]	–	–	–	0.251	0.792
PepBCL[16]	0.252	0.983	0.470	0.312	0.804
PepCNN (ours)	0.217	**0.986**	**0.479**	0.297	**0.826**

Table 3. Performances of the proposed PepCNN model and the previous methods on the TE639 test set. The highest values in each column are highlighted in bold..

are observed on the TE639 test set, whereby, the specificity, precision, and AUC have been increased compared to the previous methods. Again, the biggest improvement was achieved on the AUC metric (by 2.7%) compared to the previous best performing method, PepBCL. Even though our method did not perform the best on all the metrics in the two test sets, it surpassed the other methods on majority of the metrics, including AUC. These improvements portray the importance of feature sets from pLM, PSI-BLAST (a multiple-sequence alignment (MSA) tool), and structural information pertaining to half-sphere exposure and the use of this feature set with CNN to learn robust features for the prediction of binding and non-binding residues in protein sequences.

Case study

To elaborate on the output prediction of our proposed method, we randomly selected three protein sequences from the TE125 test set after they had been predicted by our model. These proteins were pdbID: 1dpuA, pdbID: 2bugA, and pdbID: 1uj0A and are visualized as 3D structures in Fig. 2A–F, respectively. The *magenta* color in the figure shows the binding residues and the *gray* color shows the non-binding residues. The top visualization in the figure illustrates the experimental output (the true binding residues) of the proteins, while the bottom visualization shows the binding residues of the proteins predicted by our model. The protein structures B, D, and F of Fig. 2 show that the predicted binding residues by our PepCNN model closely resembles the actual binding residues in the corresponding proteins detected by the lab experiment (structures A, C, and E of Fig. 2).

To further quantify the prediction of the amino acids in these three proteins in relation to the actual binding sites, we unrolled the sequences into a one-dimensional representation (see Fig. 3). The amino acids in the top and bottom sequences show the experimental labels and the predicted labels by our proposed method, respectively. The experimental and predicted labels are further distinguished by the use of *blue* and *red* colors, respectively. From the figure, it can be seen that in terms of the binding sites, our model correctly predicted 6 out of the 11 sites in 1dpuA, 6 out of the 9 sites in 2bugA, and 8 out of the 9 sites in 1uj0A, which results in a sensitivity value of 0.545, 0.667, and 0.889, respectively, for the proteins. Furthermore, in terms of the non-binding sites, our model correctly predicted 56 out of the 58 sites in 1dpuA, 119 out of the 122 sites in 2bugA, and 42 out of

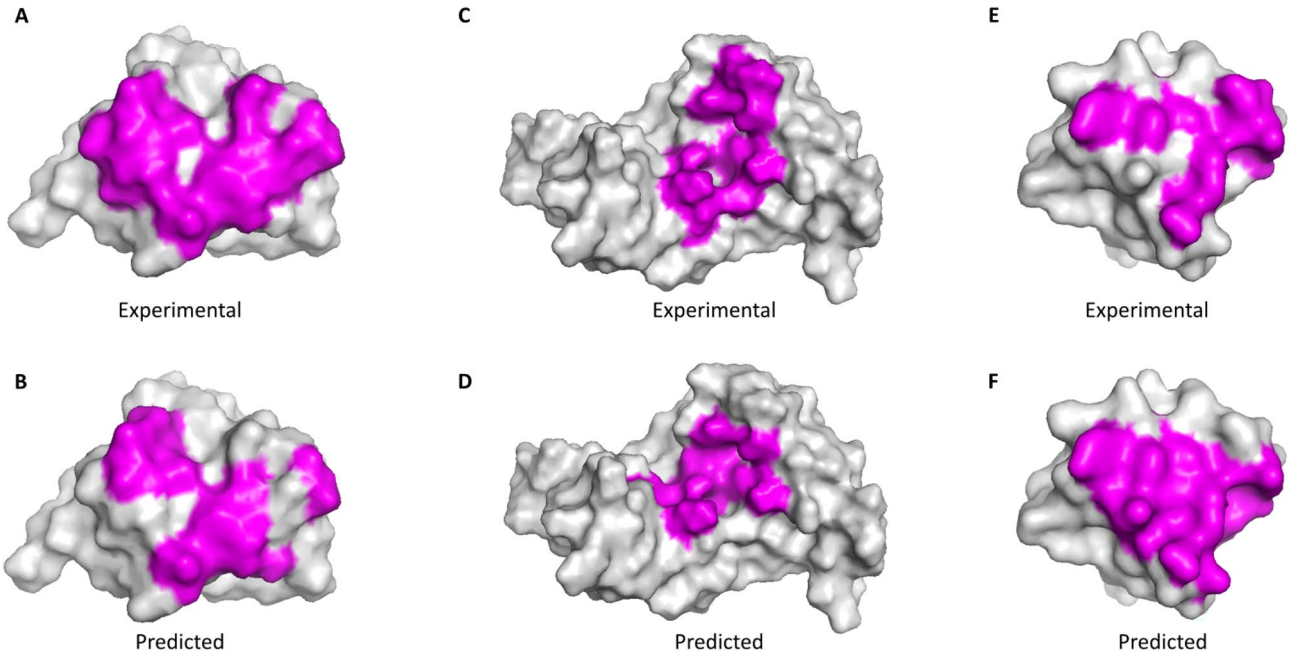

Figure 2. 3D structure visualization of three proteins (pdbID: 1dpuA, pdbID: 2bugA, and pdbID: 1uj0A) illustrating the binding (in *magenta*) and non-binding (in *gray*) residues using the PyMol software[42]. The experimental output (true binding residues) of the proteins are located in the top part (**A**, **C**, and **E**) and its corresponding predicted binding residues by our method PepCNN are located in the bottom part (**B**, **D**, and **F**).

Figure 3. Unrolled protein sequences pdbID: 1dpuA (**A**), pdbID: 2bugA (**B**), and pdbID: 1uj0A (**C**) presented as a one-dimensional representation. The top sequence of each protein showcases experimentally confirmed binding residues (in blue), while the bottom sequence depicts the predicted binding residues by our proposed method, PepCNN (in red).

the 49 sites in 1uj0A, which results in a specificity value of 0.966, 0.975, and 0.857, respectively, for the proteins. It can be seen that even though the sensitivity measure is not high for all the proteins, the ability to attain a high number of correctly predicted non-binding sites and low number of false positive sites allow the model to predict the binding sites in the same regions as the experimental findings for all the three protein sequences. The close detection of the binding sites in the sequences by our proposed method can therefore greatly assist the efforts of the experimental procedures by narrowing down the regions for further investigations, thereby tremendously reducing the time, effort, and cost needed to confirm and understand the protein–peptide binding sites in new proteins. The observations from Figs. 2 and 3 indicate a high degree of similarity between predicted and actual binding residues which validates that our algorithm effectively leverages information from primary protein sequences for the residue prediction task.

Insights into the residue features

Before embarking on the deep learning algorithm, we had built initial models in this work in which the performance of each of the feature sets and their combinations had been evaluated. In the initial models, we employed an ensemble of RF classifiers to have diverse training sets for Dataset 1 for a thorough evaluation. Moreover, it allowed for us to have less computational complexity compared to using a deep learning model. The ensemble consisted of 15 individual RF classifiers with different training sets by randomly selecting different non-binding residues during the data balancing stage. The hyper-parameters of the classifiers were tuned using the Hyperopt algorithm[43] with 5-fold cross-validation scheme. The ensemble's final predictions on the test set were determined by averaging the individual RF classifiers' probabilities, ensuring a robust and generalized performance.

Figure 4 shows the ROC curves obtained for the individual feature sets and the different feature set combinations on TE125. It can be seen that the embedding from the ProtT5 pLM attains a significantly high AUC value (0.81) in comparison to the PSSM feature set (AUC of 0.642), the HSE feature set (AUC of 0.56), and even the PSSM+HSE feature combination (AUC of 0.697). As the bindings are dependent on the conformations of proteins[44], this affirms that the embedding from the pre-trained transformer model captures essential information concealed in the primary protein sequences which relates to the structure and function of proteins and therefore

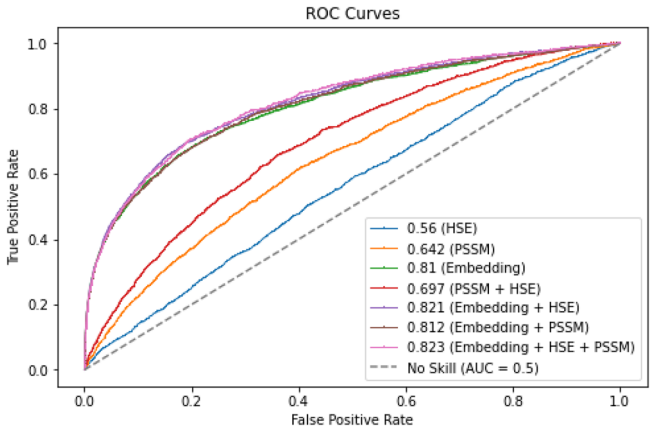

Figure 4. ROC curves for the individual feature sets and the different feature set combinations using the ensemble of RF classifiers on TE125.

contributes immensely to the binding prediction. The incorporation of PSSM and HSE feature sets to the embedding saw a further increase in the performances, with the most increase coming from the Embedding+HSE feature combination (0.821) and a slight increase in the Embedding + PSSM feature combination (0.812) when compared to just the performance of the embedding. Moreover, the feature combination Embedding + HSE + PSSM achieved the overall best AUC value of 0.823. The result obtained by combining the three features suggests that PSSMs from sequence alignment and the structural properties from half-sphere exposure add valuable information in terms of the evolutionary properties and protein surface attributes to the protein sequence representations of the transformer model. This final feature combination was then used to build our deep learning model to further improve the performance.

Discussion

We have demonstrated that PepCNN can effectively predict binding and non-binding residues in the protein sequences. It established the possibility of the pLM embedding, PSSM, and HSE feature combination with CNN as feature extractor to predict interaction sites and explore the mechanisms of protein–peptide binding. The three proteins were randomly selected for visualization so that the similarity of the predicted and experimental binding residues could be deciphered. The strong correlation observed suggests that our approach holds promise for identifying prospective binding sites in a broad array of proteins.

When evaluating a predictor, the most ideal model would be the one which has the sensitivity and specificity measures equal to 1, however, this incidence is not prevalent in clinical and computational biology research since the measures increase when either of them decreases[45]. The ROC curve, which is an analytical method represented as a graph, is therefore mainly used for evaluating the performance of a binary classification model and to also compare the test result of two or more models. Essentially, the curve plots the coordinate points using the false positive rate (1-specificity) as the x-axis and the true positive rate (sensitivity) as the y-axis. The closer the plot is to the upper left corner of the graph, the higher the model's performance is since the upper left corner has sensitivity equal to 1 and the false positive rate equal to 0 (specificity is equal to 1). The desired ROC curve hence has an AUC (area under the ROC curve) equal to 1.

The study of protein–peptide binding is desired since the peptides exhibit low toxicity and posses small interface areas (as peptides are mostly 5–15 residues long[46]), making them good targets for efficacious therapeutic designs and drug discovery process[47]. In addition, peptide-like inhibitors are used for treating diabetes, cancer, and autoimmune diseases[48]. In the past, search for peptides as therapeutics was discouraged due to their short half-life and slow absorption[49], however, these short amino acid chains are considered drug candidates once again due to the emergence of synthetic approaches which allow for changes to its biophysical and biochemical properties[50].

Understanding the structure of protein–peptide complexes is often a prerequisite for the design of peptide-based drugs. The challenges of studying these complexes are unique compared to other interactions such as protein–protein and protein–ligand. In protein–protein interactions, complexes are usually formed based on well-defined 3D structures, and in the protein–ligand interactions, small ligands typically bind in deeply buried regions of proteins. Conversely, peptides often lack stable structures and usually bind with weak affinity to large, shallow pockets on protein surfaces[51]. Given these complexities, and the limitations of current experimental methods like X-ray crystallography and nuclear magnetic resonance, there is a compelling need for robust computational methods.

In summary, our work contributes to addressing these challenges by offering a highly accurate and computationally efficient method for predicting protein–peptide interaction sites. Such advances are crucial for both fundamental biological research and practical applications in drug design.

Conclusion

In this work, we have developed a new deep learning-based protein–peptide binding residue predictor called PepCNN. The model leverages sequence-based features, which are extracted from a pre-trained pLM, as well as from a MSA tool. In addition to these, we incorporated a structure-based feature known as half-sphere exposure. Utilizing these diverse properties of protein sequences as input, our convolutional neural network was effective in learning essential features. As a result, PepCNN was able to outperform existing methods that also rely on primary protein sequence information, as demonstrated by tests on two distinct datasets.

Looking ahead, our future research aims to further enhance the model's performance. One innovative avenue for exploration will involve integrating DeepInsight technology[19]. This technology converts feature vectors into their corresponding image representations, thus enabling the application of 2D CNN architectures. This change opens up the possibility of implementing transfer learning techniques to boost the model's predictive power.

Methods

Evaluation metrics

The proposed model in this work was evaluated using the residues in the test sets TE125 and TE639 after being trained on their respective training sets. These test sets are highly imbalanced, and for this reason, suitable metrics were chosen to effectively evaluate our model for the classification task. These metrics were Sensitivity, Specificity, Precision, and MCC. The formulation of these metrics are given below.

$$Sensitivity = \frac{TP}{TP + FN} \tag{1}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \tag{4}$$

In the above formulas, TP stands for True Positives, TN stands for True Negatives, FP stands for False Positives, and FN stands for False Negatives. TP is the number of actual binding residues correctly classified by the model, TN is the number of actual non-binding residues correctly classified by the model, FP is the number of actual non-binding residues incorrectly classified by the model, and finally FN is the number of actual binding residues incorrectly classified by the model. For the given model, the Sensitivity metric [given by Eq. (1)] and the Specificity metric [given by Eq. (2)] calculate the fraction of binding residues and non-binding residues correctly predicted, respectively, the Precision metric [given by Eq. (3)] calculates the proportion of binding residues correctly classified out of all the residues classified as binding, and the MCC metric [given by Eq. (4)] calculates the prediction ability for both the binding and non-binding residues. The values range from 0 to 1 for the Sensitivity, Specificity, and Precision metrics and the higher the value, the better the prediction model is. The MCC metric takes on values ranging from − 1 to + 1 where + 1 indicates a highly positive correlation, while − 1 indicates a highly negative correlation. It should be noted that the above metrics are dependent of the probability threshold of the classifier and varying the threshold would also vary the metric values. For this reason, these metrics cannot be heavily relied upon for the model evaluation. Therefore, in addition to the above metrics, we have also included the AUC metric which is calculated based on the classification probability values and is independent of the threshold setting. The metric therefore gives more confidence in the evaluation of a model's performance. AUC is also a very useful metric since it measures the overall performance of the classification model by calculating its separability between the predicted binding and non-binding residues. The range of values for the AUC metric is from 0 to 1, with 0 being the worst measure of separability and 1 being a very good measure of separability.

Feature extraction

The features chosen in this study are the representations from a pre-trained pLM, evolutionary relationships in the protein sequences using a MSA tool, and the structural attributes in terms of the solvent exposure of the residues in the sequences. In the feature extraction stage of our proposed method (Fig. 1A), the three different feature-types were obtained by submitting the 1,279 proteins to the three tools: pre-trained ProtT5 pLM[20], PSI-BLAST[37], and HSEpred[52] to acquire the Embedding, PSSM, and HSE values, respectively. The following subsections elucidates each of these features in detail.

Transformer embedding

Transformer models from natural language processing employ latest DL algorithms and such architectures have shown huge potential in proteomics field due to its ability to leverage on the growing databases of protein sequences. These models offer transfer learning where the knowledge acquired from data-rich tasks can be transferred to similar data-limited tasks. Several pLMs have been developed by Elnaggar et al.[20] and out of those models, ProtT5 is amongst the most widely used pre-trained models in the literature to tackle various tasks[53]. It is based on the T5 architecture[54], which is akin to the originally proposed architecture for language translation task[55] as depicted in Fig. 5. It consists of the encoder and decoder blocks, where the encoder projects the input sequence to an embedding space and the decoder generates the output embedding based on the embedding of the encoder. To do this, firstly the input sequence tokens (x_1, ..., x_n) are mapped by the encoder to generate representation z (z_1, ..., z_n). The decoder then uses the representation z to produce output sequence (y_1, ..., y_n), element by element. Both the encoder and decoder have the main components known as the multi-head attention and the feed-forward layer. The multi-head attention is a result of combining multiple self-attention modules (heads), where the self-attention is an attention mechanism that relates different positions in the input sequence to compute its representation. The attention function maps a position's query vector and a set of key-value vectors for all the positions to an output vector. In order to carry out this operation for all the positions simultaneously, the query, key and value vectors are packed together into matrices Q, K, and V, respectively, and the output matrix is computed as: head = Attention(Q, K, V) = softmax($\frac{QK^T}{\sqrt{d^k}}$)V, where $\frac{1}{\sqrt{d^k}}$ is the scaling factor. It is much beneficial to have multi-head attention instead of a single self-attention module since it allows for the capturing of information from different representations at the different positions. This is done by linearly projecting the queries, keys and values n times. The multi-head attention is therefore given by: MultiHead(Q, K, V) = Concat(head$_1$, ..., head$_n$)W^O, where head$_i$ = Attention(QW_i^Q, KW_i^K, VW_i^V); W_i^Q, W_i^K, W_i^V and W^O are projection matrices. The ProtT5 transformer used in this work is a 3 billion parameter model which was trained on the Big Fantastic Database[56] and fine-tuned on the UniRef50[57] database. Even though ProtT5 has both encoder and decoder blocks in its architecture, the authors found that the encoder embedding outperformed the decoder embedding on all tasks, hence the pre-trained model extracts the embedding from its encoder side. The output embedding of the ProtT5 model is a matrix of dimension $L \times 1024$ (where L represents the protein's length and 1024 the values of the network's last hidden layer). This matrix captures relationships between amino acid residues

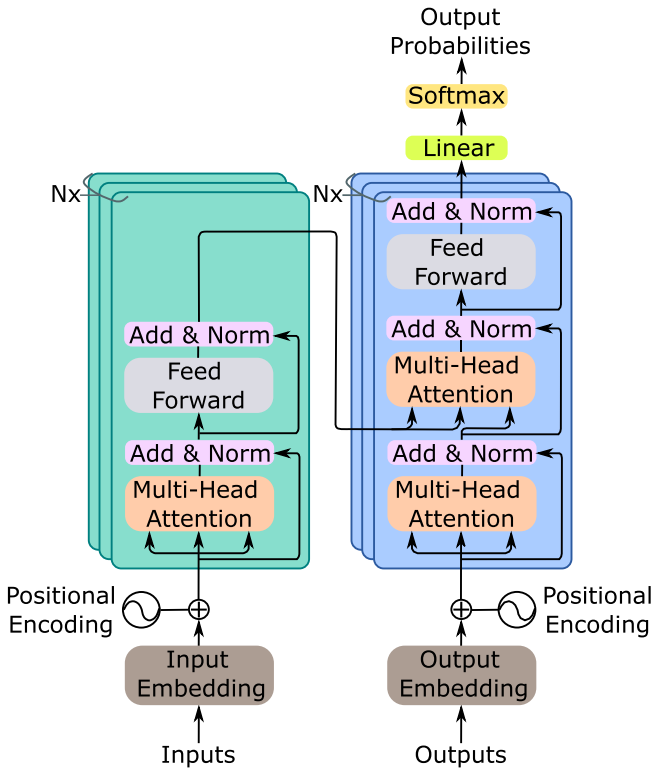

Figure 5. The original encoder-decoder Transformer[55] which was proposed for language translation task. The network can have layers of these encoder-decoder modules, denoted by *Nx*. The input sequence is fed to the encoder and the decoder produces a new output sequence. At each timestep, an output is predicted, which is then fed back to the network (decoder), including all the previous outputs, to predict the output for the next timestep and so on until the output sequence (translation) is produced.

in the input protein sequence based on the attention mechanism and produces a rich set of features that encompasses relevant protein structural and functional information.

Position specific scoring matrices

In protein engineering, MSAs are a popularly used technique for aligning sequences to determine their evolutionary relationships and structural/functional constraints within families of proteins to aid diverse prediction pipelines[58]. For instance, it has been a vital component for contact and structure predictions[59,60], as well as other prediction tasks such as functional effects of mutations[61] and rational protein design[62]. To incorporate the potency of the information held in MSA, PSI-BLAST tool was employed in this work to obtained the sequence-profiles. It was run using the *E*-value threshold of 0.001 in three iterations which resulted in two matrices, log odds and linear probabilities of the amino acids, with dimensions $L \times 20$ (where 20 represents the 20 different amino acids of the genetic code). The matrix with linear probabilities was used in this work in which each of the elements in the row represent the substitution probabilities of the amino acid with all the 20 amino acids in the genetic code. PSSM can therefore be formulated as $P = \{P_{ij} : i = 1...L \text{ and } j = 1...20\}$, where P_{ij} is the probability for the jth amino acid in the ith position of the input sequence and has a high value for a highly conserved position, while a low value indicates a weakly conserved position.

Half-sphere exposure

The information about a protein's surface is valuable for the prediction of protein–peptide binding sites as the peptides often bind to the shallow surface regions[51]. HSE is an effective property that measures the solvent exposure for distinguishing buried, partially buried and exposed residues[63]. It has been widely used in protein–peptide and other binding prediction tasks[10,18,64,65]. In this work, the HSE values of the proteins were obtained from the HSEpred server, which gives a measure of how buried an amino acid is in the protein's three-dimensional structure. HSE for a residue is measured by firstly setting a sphere of radius $r_d = 13$ Å at the residue's Cα atom. Secondly, this sphere is divided into two halves by constructing a plane perpendicular to a given Cα-Cβ vector that goes through the residue's Cα atom resulting in two HSE measures: HSE-up and HSE-down. HSE-up refers to the upper sphere in the direction of the side chain and HSE-down refers to the lower sphere which is in the opposite direction to the side chain. Finally, the number of Cα atoms in the upper and lower half of the sphere are measured, respectively[52]. Refer to Fig. 6 for the illustration of the HSE-up and HSE-down measures. Contact number is another important measure and it indicates the total number of Cα atoms in the sphere of the Cα

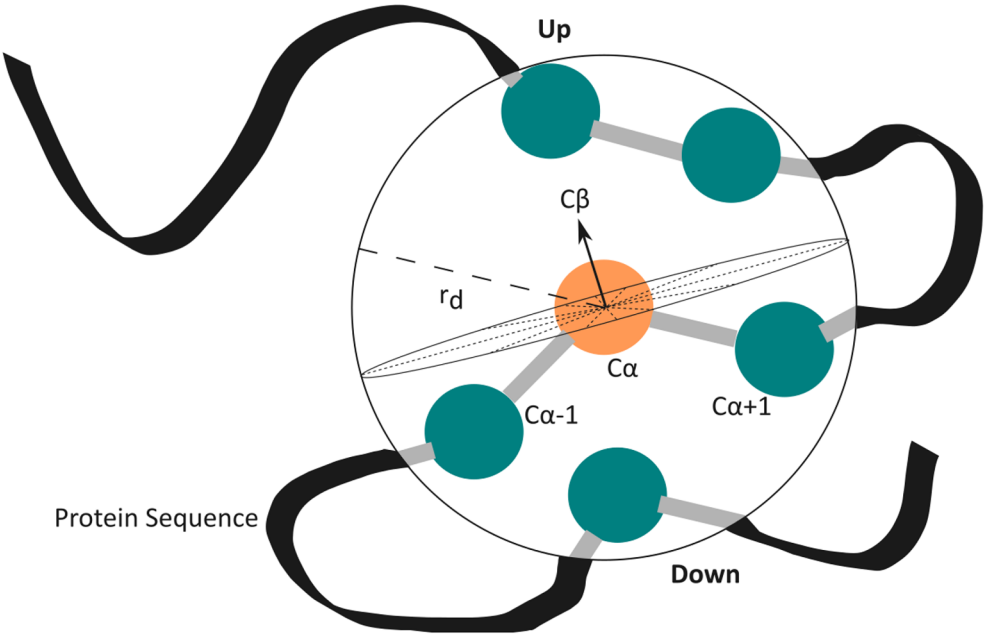

Figure 6. Depiction of the HSE-up and HSE-down measures. The dotted lines indicate the plane's position which divides the sphere of the residue's Cα atom (in orange) with radius r_d into two equal half spheres. The other Cα atoms (in green) represents parts of other residues in the protein sequence.

atom of a residue[66]. The output of HSEpred is therefore a feature matrix of dimension $L \times 3$ where 3 represents the values of HSE-up, HSE-down, and the contact number for each residue.

Convolutional neural network

From the deep learning area, CNN is one of the most widely used network in the recent times[67]. It is a type of feed-forward neural network that uses convolutional structures to extract features from data. A CNN has three main components: convolutional layer, pooling layer, and fully connected layer. The convolutional layer consists of several convolution filters. It produces what are known as feature maps by convolving the input with a filter and then applying nonlinear activation function to each of the resulting elements. The border information can be lost during the convolution process, so to mitigate this, padding is introduced to increase the input with a zero value, which can indirectly change its size. Additionally, the stride is used to control the convolving density. The density is lower for longer strides. The pooling layer down-samples an image, which reduces the amount of data and at the same time preserves useful information. Moreover, by eliminating superfluous features, it can also lower the number of model parameters. One or more fully connected layers are added after several convolutional and pooling layers. In the fully connected layers, all the previous layer neurons are connected to every neurons in the current layer and this results in the generation of global semantic information. The network can more accurately approximate the target function by increasing its depth, however, this also makes the network more complex, which makes it harder to optimize and are more likely to overfit.

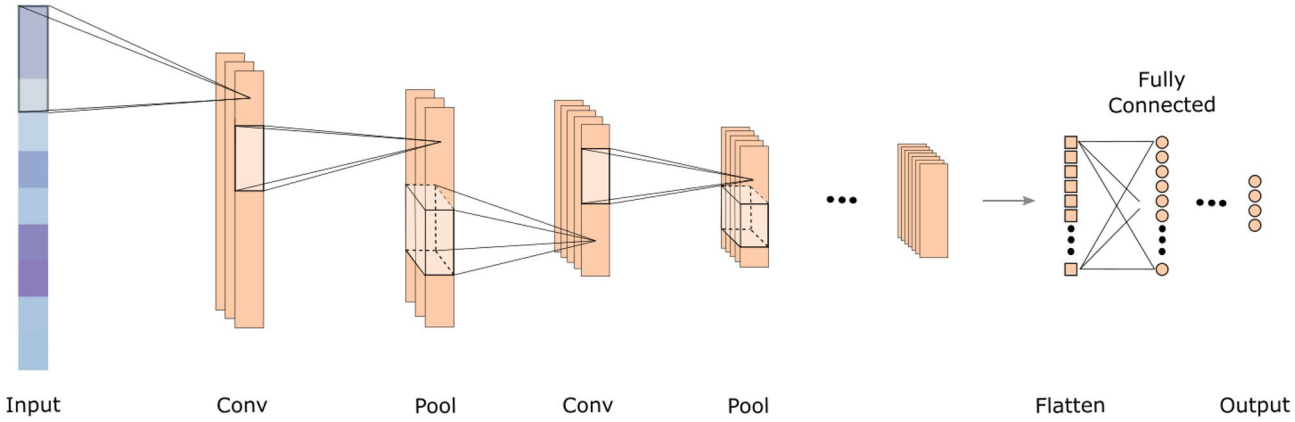

Figure 7. A sample 1D CNN depiction which shows the flow of information from the input to the output through its three main layers: convolutional, pooling, and fully connected.

CNN has made some outstanding advancements in a variety of fields, including, but not limited to, computer vision and natural language processing, which has garnered significant interest from researchers in various fields. A CNN can also be applied to 1D and multidimensional input data in addition to the processing of 2D images. In order to process 1D data, CNN typically uses 1D convolutional filters (as portrayed in Fig. 7).

Building the deep learning model

In order to build a classifier that carries out per residue binding/non-binding prediction, it is important to extract information pertaining to each residue. In the residue extraction stage of our proposed method (Fig. 1B), we represented each residue with its sequence based (pre-trained pLM embedding and PSSM) and structure (HSE) based information. This was done by extracting the values corresponding to each residue from the three feature matrices obtained when the proteins were submitted to the three feature extraction tools. Tensor sum was applied to the resulting vectors, i.e. 1×1024 Embedding vector, 1×20 PSSM vector, and 1×3 HSE vector, which formed a feature vector of dimension $1 \times 1{,}047$ to represent each residue. These residues were kept in their respective sets (i.e. train and test) to effectively train and evaluate the model without bias.

In the model training stage (Fig. 1C), we trained a 1D CNN to build our predictor based on the Tensorflow framework[68]. The model has 8.7 million trainable parameters which were trained using 80% of the training set, and the remaining 20% were used for network validation. The model is composed of three 1D convolutional layers and two fully connected (dense) layers. For the convolutional layers, the first layer contains 128 filters of size 5, the second layer contains 128 filters of size 3, and the third layer contains 64 filters of size 3. The stride for each layer was kept as 1 and the padding was used such that the output size of each layer was equal to the input size to the layer. Dropouts were used after each convolutional layer. In the fully connected layers, the first layer and the second layer contains 128 and 32 neurons, respectively. Finally, the output was made of a single neuron for binary classification. The ReLU activation function was used in each of the layers, while a sigmoid activation function was used in the output neuron. The model was trained using Adam optimizer with a learning rate of 1×10^{-6}, loss using binary crossentropy, and metric as AUC. Moreover, early stopping was employed with a patience of 3. The network was optimized using the Bayesian Optimization algorithm in the Keras Tuner library[69]. The plots of the training progress of the model for the training sets TR1115 and TR640 are shown in Fig. 8.

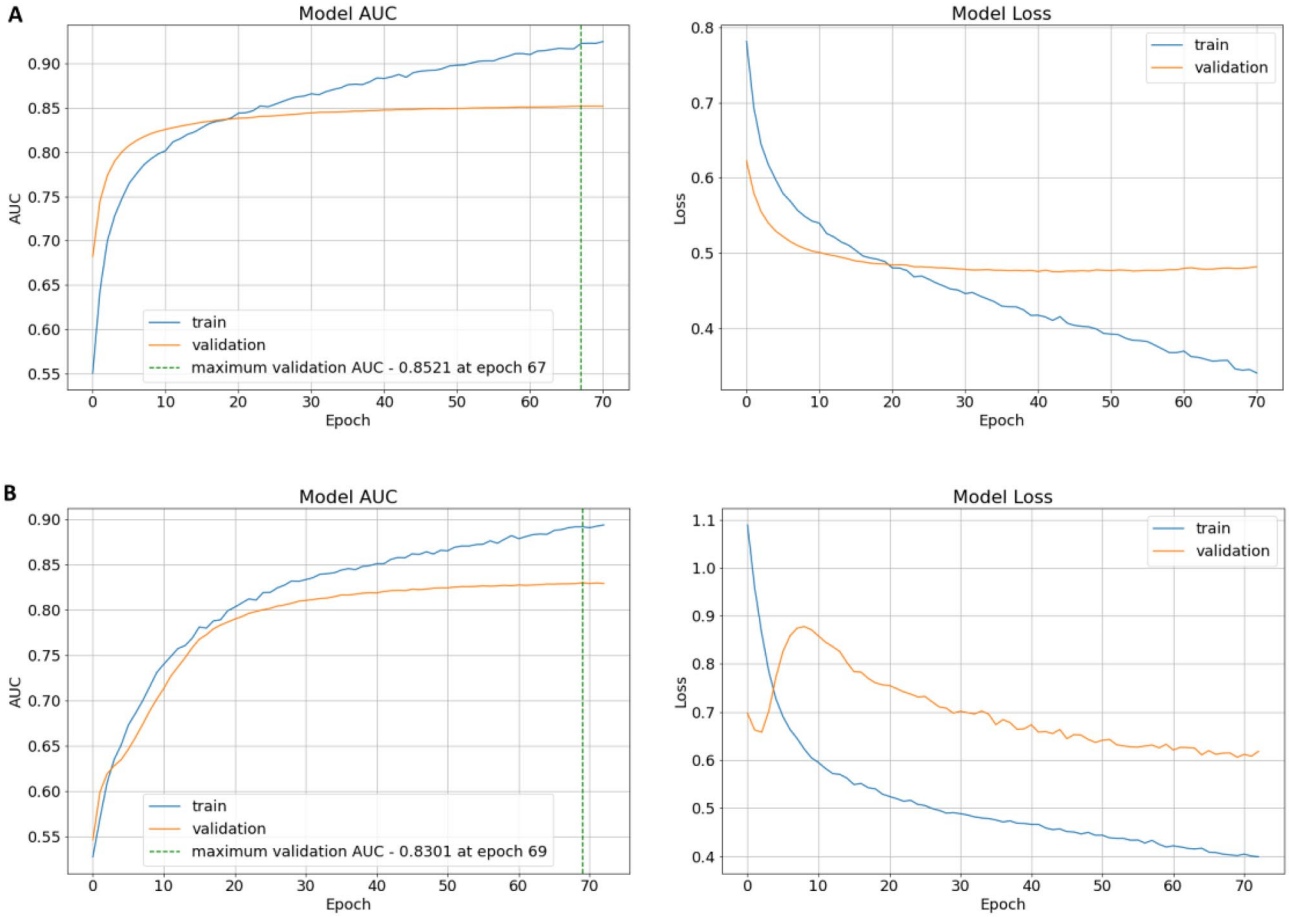

Figure 8. Plots of AUC and loss for the training progress of the proposed model on the two training sets: (**A**) TR1115 train set (achieving an AUC of 0.8521 on the validation set), and (**B**) TR640 train set (achieving an AUC of 0.8301 on the validation set).

Data availability

The datasets used in this paper can be downloaded from the GitHub link https://github.com/abelavit/PepCNN.git.

Code availability

The PepCNN codes (in Python) are available at the GitHub link https://github.com/abelavit/PepCNN.git.

Received: 7 September 2023; Accepted: 16 November 2023
Published online: 28 November 2023

References

1. Pawson, T. & Nash, P. Assembly of cell regulatory systems through protein interaction domains. *Science***300**, 445–452 (2003).
2. Rubinstein, M. & Niv, M. Y. Peptidic modulators of protein–protein interactions: Progress and challenges in computational design. *Biopolym. Origi. Res. Biomol.* **91**, 505–513 (2009).
3. Lee, H., Heo, L., Lee, M. S. & Seok, C. Galaxypepdock: A protein–peptide docking tool based on interaction similarity and energy optimization. *Nucl. Acids Res.* **43**, W431–W435 (2015).
4. Neduva, V. *et al.* Systematic discovery of new recognition peptides mediating protein interaction networks. *PLoS Biol.* **3**, e405 (2005).
5. Chandra, A. *et al.* Phoglystruct: Prediction of phosphoglycerylated lysine residues using structural properties of amino acids. *Sci. Rep.* **8**, 17923 (2018).
6. Vlieghe, P., Lisowski, V., Martinez, J. & Khrestchatisky, M. Synthetic therapeutic peptides: Science and market. *Drug Discov. Today* **15**, 40–56 (2010).
7. Dyson, H. J. & Wright, P. E. Intrinsically unstructured proteins and their functions. *Nat. Rev. Mol. Cell Biol.* **6**, 197–208 (2005).
8. Bertolazzi, P., Guerra, C. & Liuzzi, G. Predicting protein-ligand and protein–peptide interfaces. *Eur. Phys. J. Plus* **129**, 1–10 (2014).
9. Petsalaki, E., Stark, A., García-Urdiales, E. & Russell, R. B. Accurate prediction of peptide binding sites on protein surfaces. *PLoS Comput. Biol.* **5**, e1000335 (2009).
10. Taherzadeh, G., Zhou, Y., Liew, A.W.-C. & Yang, Y. Structure-based prediction of protein–peptide binding regions using random forest. *Bioinformatics* **34**, 477–484 (2018).
11. Lavi, A. *et al.* Detection of peptide-binding sites on protein surfaces: The first step toward the modeling and targeting of peptide-mediated interactions. *Proteins Struct. Funct. Bioinform.* **81**, 2096–2105 (2013).

12. Taherzadeh, G., Yang, Y., Zhang, T., Liew, A.W.-C. & Zhou, Y. Sequence-based prediction of protein–peptide binding sites using support vector machine. *J. Comput. Chem.* **37**, 1223–1229 (2016).
13. Zhao, Z., Peng, Z. & Yang, J. Improving sequence-based prediction of protein–peptide binding residues by introducing intrinsic disorder and a consensus method. *J. Chem. Inf. Model.* **58**, 1459–1468 (2018).
14. Wardah, W. *et al.* Predicting protein–peptide binding sites with a deep convolutional neural network. *J. Theor. Biol.* **496**, 110278 (2020).
15. Abdin, O., Nim, S., Wen, H. & Kim, P. M. Pepnn: A deep attention model for the identification of peptide binding sites. *Commun. biology* **5**, 503 (2022).
16. Wang, R., Jin, J., Zou, Q., Nakai, K. & Wei, L. Predicting protein–peptide binding residues via interpretable deep learning. *Bioinformatics* **38**, 3351–3360 (2022).
17. Weatheritt, R. J. & Gibson, T. J. Linear motifs: Lost in (pre) translation. *Trends Biochem. Sci.* **37**, 333–341 (2012).
18. Shafiee, S., Fathi, A. & Taherzadeh, G. Spppred: Sequence-based protein-peptide binding residue prediction using genetic programming and ensemble learning. *IEEE/ACM Transactions on Comput. Biol. Bioinforma.* **20**, 2029–2040 (2022).
19. Sharma, A., Vans, E., Shigemizu, D., Boroevich, K. A. & Tsunoda, T. Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Sci. Rep.* **9**, 11399 (2019).
20. Elnaggar, A. *et al.* Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2021).
21. Inkscape, version 1.2.2. Software available from http://inkscape.org.
22. Min, S., Lee, B. & Yoon, S. Deep learning in bioinformatics. *Briefings Bioinform.* **18**, 851–869 (2017).
23. Sharma, A., Lysenko, A., Boroevich, K. A. & Tsunoda, T. DeepInsight-3D architecture for anti-cancer drug response prediction with deep-learning on multi-omics. *Sci. Rep.* **13**, 2483 (2023).
24. Rojas, R. *Neural Networks: A Systematic Introduction* (Springer, New York, 2013).
25. Wen, B. *et al.* Deep learning in proteomics. *Proteomics* **20**, 1900335 (2020).
26. Wang, P., Fan, E. & Wang, P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recogn. Lett.* **141**, 61–67 (2021).
27. Nguyen, G. *et al.* Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artif. Intell. Rev.* **52**, 77–124 (2019).
28. Kandathil, S. M., Greener, J. G. & Jones, D. T. Recent developments in deep learning applied to protein structure prediction. *Proteins Struct. Funct. Bioinform.* **87**, 1179–1189 (2019).
29. Meyer, J. G. Deep learning neural network tools for proteomics. *Cell Rep. Methods* **1**, 1–10 (2021).
30. Neely, B. A. *et al.* Toward an integrated machine learning model of a proteomics experiment. *J. Proteome Res.* **22**, 681–696 (2023).
31. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36**, 193–202 (1980).
32. Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J. & Koes, D. R. Protein–ligand scoring with convolutional neural networks. *J. Chem. Inf. Model.* **57**, 942–957 (2017).
33. Zeng, H., Edwards, M. D., Liu, G. & Gifford, D. K. Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics* **32**, i121–i127 (2016).
34. Rao, R. M. *et al.* Msa transformer. In *International Conference on Machine Learning*, 8844–8856 (PMLR, 2021).
35. Chandra, A., Tünnermann, L., Löfstedt, T. & Gratz, R. Transformer-based deep learning for predicting protein properties in the life sciences. *Elife* **12**, e82819 (2023).
36. Yang, J., Roy, A. & Zhang, Y. Biolip: A semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic Acids Res.* **41**, D1096–D1103 (2012).
37. Altschul, S. F. *et al.* Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).
38. Yu, D.-J. *et al.* Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling. *Neurocomputing* **104**, 180–190 (2013).
39. Mahmud, S. H. *et al.* Prediction of drug–target interaction based on protein features using undersampling and feature selection techniques with boosting. *Anal. Biochem.* **589**, 113507 (2020).
40. Jiménez-Valverde, A. Insights into the area under the receiver operating characteristic curve (AUC) as a discrimination measure in species distribution modelling. *Glob. Ecol. Biogeogr.* **21**, 498–507 (2012).
41. Sing, T., Sander, O., Beerenwinkel, N. & Lengauer, T. Rocr: Visualizing classifier performance in r. *Bioinformatics* **21**, 3940–3941 (2005).
42. Schrödinger, LLC. The PyMOL molecular graphics system, version 2.5.5 (2015). Software available from https://pymol.org/2/.
43. Bergstra, J., Yamins, D. & Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International Conference on Machine Learning*, 115–123 (PMLR, 2013).
44. Stank, A., Kokh, D. B., Fuller, J. C. & Wade, R. C. Protein binding pocket dynamics. *Acc. Chem. Res.* **49**, 809–815 (2016).
45. Nahm, F. S. Receiver operating characteristic curve: Overview and practical use for clinicians. *Korean J. Anesthesiol.* **75**, 25–36 (2022).
46. London, N., Movshovitz-Attias, D. & Schueler-Furman, O. The structural basis of peptide–protein binding strategies. *Structure* **18**, 188–199 (2010).
47. Liu, D. *et al.* Self-assembly of mitochondria-specific peptide amphiphiles amplifying lung cancer cell death through targeting the VDAC1–hexokinase-II complex. *J. Mater. Chem. B* **7**, 4706–4716 (2019).
48. Pant, S., Singh, M., Ravichandiran, V., Murty, U. & Srivastava, H. K. Peptide-like and small-molecule inhibitors against covid-19. *J. Biomol. Struct. Dyn.* **39**, 2904–2913 (2021).
49. Lau, J. L. & Dunn, M. K. Therapeutic peptides: Historical perspectives, current development trends, and future directions. *Bioorg. Med. Chem.* **26**, 2700–2707 (2018).
50. Angelova, A., Drechsler, M., Garamus, V. M. & Angelov, B. Pep-lipid cubosomes and vesicles compartmentalized by micelles from self-assembly of multiple neuroprotective building blocks including a large peptide hormone PACAP-DHA. *ChemNanoMat* **5**, 1381–1389 (2019).
51. Petsalaki, E. & Russell, R. B. Peptide-mediated interactions in biological systems: New discoveries and applications. *Curr. Opin. Biotechnol.* **19**, 344–350 (2008).
52. Song, J., Tan, H., Takemoto, K. & Akutsu, T. Hsepred: Predict half-sphere exposure from protein sequences. *Bioinformatics* **24**, 1489–1497 (2008).
53. Pokharel, S., Pratyush, P., Heinzinger, M., Newman, R. H. & Kc, D. B. Improving protein succinylation sites prediction using embeddings from protein language model. *Sci. Rep.* **12**, 16933 (2022).
54. Raffel, C. *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**, 5485–5551 (2020).
55. Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.***30**, 5998–6008 (2017).
56. Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. *Nat. Commun.* **9**, 2542 (2018).
57. Suzek, B. E. *et al.* Uniref clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015).

58. Petti, S. *et al.* End-to-end learning of multiple sequence alignments with differentiable Smith–Waterman. *Bioinformatics* **39**, btac724 (2023).
59. Jones, D. T., Buchan, D. W., Cozzetto, D. & Pontil, M. Psicov: Precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* **28**, 184–190 (2012).
60. Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
61. Frazer, J. *et al.* Disease variant prediction with deep generative models of evolutionary data. *Nature* **599**, 91–95 (2021).
62. Russ, W. P. *et al.* An evolution-based model for designing chorismate mutase enzymes. *Science* **369**, 440–445 (2020).
63. Pan, Y., Wang, Z., Zhan, W. & Deng, L. Computational identification of binding energy hot spots in protein-RNA complexes using an ensemble approach. *Bioinformatics* **34**, 1473–1480 (2018).
64. Wang, H., Liu, C. & Deng, L. Enhanced prediction of hot spots at protein–protein interfaces using extreme gradient boosting. *Sci. Rep.* **8**, 14285 (2018).
65. Pan, Y., Zhou, S. & Guan, J. Computationally identifying hot spots in protein-DNA binding interfaces using an ensemble approach. *BMC Bioinform.* **21**, 1–16 (2020).
66. Hamelryck, T. An amino acid has two sides: A new 2d measure provides a different view of solvent exposure. *Proteins Struct. Funct. Bioinform.* **59**, 38–48 (2005).
67. Li, Z., Liu, F., Yang, W., Peng, S. & Zhou, J. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **33**, 6999–7019 (2021).
68. Abadi, M., *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems (2015). Software available from tensorflow.org.
69. O'Malley, T. *et al.* Keras Tuner. https://github.com/keras-team/keras-tuner (2019).

Acknowledgements

This research is partially supported by Australian Research Council Grant DP180102727. We are grateful to the Griffith University eResearch Service & Specialised Platforms team for their High Performance Computing Cluster to complete this research.

Author contributions

A.C. performed analysis and experiments. A.C and A.Sharma conceived and wrote the first manuscript. A.Sharma and I.D curated the data and T.T. and A.Sattar contributed in manuscript write-up. All authors read and approved the final manuscript.

Competing interest

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.C. or A.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.