Journal of Theoretical Biology 496 (2020) 110278

Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/jtb

Predicting protein-peptide binding sites with a deep convolutional neural network

Wafaa Wardah[a,*], Abdollah Dehzangi[b], Ghazaleh Taherzadeh[c], Mahmood A. Rashid[d,e], M.G.M. Khan[a], Tatsuhiko Tsunoda[f,g,h,i], Alok Sharma[e,g,h,j,*]

[a] *School of Computing, Information and Mathematical Sciences, Faculty of Science, Technology and Environment, The University of the South Pacific, Suva, Fiji*
[b] *Department of Computer Science, Morgan State University, Baltimore, USA*
[c] *Institute for Bioscience and Biotechnology Research, University of Maryland, USA*
[d] *Institute for Sustainable Industries and Liveable Cities, Victoria University Melbourne, Victoria, Australia*
[e] *Institute for Integrated and Intelligent Systems, Griffith University, Queensland, Australia*
[f] *Department of Medical Science Mathematics, Medical Research Institute, Tokyo Medical and Dental University, Tokyo, Japan*
[g] *Laboratory for Medical Science Mathematics, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan*
[h] *CREST, JST, Tokyo 113-8510, Japan*
[i] *Laboratory for Medical Science Mathematics, Department of Biological Sciences, Graduate School of Science, The University of Tokyo, Tokyo, Japan*
[j] *School of Engineering and Physics, The University of the South Pacific, Suva, Fiji*

ARTICLE INFO

Article history:
Received 30 December 2019
Revised 5 April 2020
Accepted 8 April 2020
Available online 13 April 2020

Keywords:
Protein-peptide binding
Artificial intelligence
Deep learning
Convolutional neural network
Protein sequence

ABSTRACT

Motivation: Interactions between proteins and peptides influence biological functions. Predicting such bio-molecular interactions can lead to faster disease prevention and help in drug discovery. Experimental methods for determining protein-peptide binding sites are costly and time-consuming. Therefore, computational methods have become prevalent. However, existing models show extremely low detection rates of actual peptide binding sites in proteins. To address this problem, we employed a two-stage technique - first, we extracted the relevant features from protein sequences and transformed them into images applying a novel method and then, we applied a convolutional neural network to identify the peptide binding sites in proteins.

Results: We found that our approach achieves 67% sensitivity or recall (true positive rate) surpassing existing methods by over 35%.

1. Introduction

For the many roles that proteins play in and around cells, interacting with other molecules is known to be what enables most biological functionalities. To perform biological processes, proteins interact with a variety of molecular structures, such as nucleic acids (RNA and DNA) (Yan et al., 2016; Peng and Kurgan, 2015), lipids, various small ligands (Roche et al., 2015) and other proteins. Some are covalent interactions including disulphide bonding and electron sharing, and others are weaker interactions including hydrogen bonds, hydrophobic interactions, Van der Waals forces and ionic interactions (Westermarck et al., 2013). The presence of water molecules also plays a vital role in the interactions that occur (Janin, 1999). While DNA repair, replication, gene expression and metabolism are known to be some of the vital cellular processes that protein interactions facilitate, studies have found that such interactions can also induce abnormal cellular behavior and disease such as cancers, where up to 40% of these interactions involve binding with relatively small peptides (Neduva et al., 2005). Therefore, analyzing protein-peptide interactions is necessary for understanding the molecular factors leading to various diseases (Nibbe et al., 2011; Kuzmanov and Emili, 2013) and drug discovery (Vlieghe et al., 2009). Identifying the residues that are involved in these interactions and understanding the mechanisms that result in the binding of proteins and peptides are vital. In vivo methods currently used in this field include the Yeast two - hybrid screening method and affinity purification which involve high-throughput screening such as mass spectrometry and Nuclear

[*] Corresponding authors.
E-mail addresses: wafaa.wardah@usp.ac.fj (W. Wardah), alokanand.sharma@riken.jp (A. Sharma).

https://doi.org/10.1016/j.jtbi.2020.110278

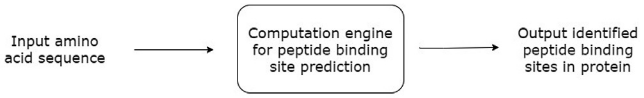

Input amino acid sequence
Computation engine for peptide binding site prediction
Output identified peptide binding sites in protein

Fig. 1. The diagram shows a simplified overview of the protein-peptide binding site prediction problem. The protein's constituent amino acid sequence information is fed into the computational engine, which then produces the identified peptide binding sites.

Magnetic Resonance spectroscopy, which are expensive, labor-intensive and time-consuming. Through such experiments, protein data have been archived in repositories and are mostly accessible to the public. The first such repository was the Database of Interacting Protein (DIP) (Xenarios and Eisenberg, 2001). Other repositories that contain protein interaction related information include the Protein Data Bank (PDB) (Berman et al., 2016), BioLip (Yang et al., 2012) and Mentha (Calderone et al., 2013).

Availability of protein data allowed machine learning techniques to be applied to the protein-peptide binding site prediction problem. Various methods were proposed to predict the peptide-binding sites of specific protein domains such as Major Histocompatibility Complex (MHC), PDZ (PSD-95, Discs-large, ZO-1), Src homology 2 (SH2) and Src homology 3 (SH3). In (Guo et al., 2013), researchers developed models MHC2SK and MHC2SKpan to computationally predict MHC binding peptides. They used Blosum62 for feature preparation (Henikoff and Henikoff, 1992) and proposed a model inspired by Spectrum RBF string kernel (SRBF) (Toussaint et al., 2010). Another study (Hou et al., 2009) employed the Swiss-Prot database and used RBF kernel functions with Support Vector Machine (SVM) classifiers to predict peptide-binding sites of SH3 domains. In (Kundu et al., 2013), researchers developed yet another SVM-based tool to predict protein-peptide binding sites of SH2 domains. Furthermore, PDZ-DockScheme was developed using a simulated annealing algorithm and rotamer optimization to predict protein-peptide bindings in PDZ domains (Niv and Weinstein, 2005). A limitation of these methods is that they require known protein structures, while in reality, most protein structures are still unknown. Methods that are able to utilize protein sequence information to achieve reasonably accurate protein-peptide binding residues are a valuable contribution to the scientific community.

The problem of protein-peptide interaction can be viewed as a binary classification problem, where in a protein chain, each residue can be classified into one of two classes: binding or non-binding (shown in Fig. 1). Some techniques that have been used with this strategy include SVM, random forest and artificial neural networks (ANN). The models generally employ a sliding window to input the properties of the constituent residues along protein chains. The properties (or features) explored across literature include sequence, structural, evolutionary and physicochemical information (Taherzadeh et al., 2016; 2017).

The common pipeline involves obtaining protein-ligand interaction data from BioLip (Yang et al., 2013), a semi curated database that derives protein data from the PDB (Berman et al., 2003). Features are then selected, and the choice of computational technique is applied. In 2007, authors of SPPIDER (Porollo and Meller, 2007) used multiple classifiers including SVM, ANN and linear discriminant analysis (LDA), where they studied the usefulness of the relative solvent accessibility (RSA) (Wagner et al., 2005) feature. PSIVER (Murakami and Mizuguchi, 2010) used inherent Naive Bayes classifier with kernel density estimation methods. LORIS (Dhole et al., 2014) and SPRINGS (Singh et al., 2014) applied L1-regularized logistic regression and ANN methods to multiple input features. The method SPRINT is a server that makes sequence-based predictions of protein-peptide binding sites using SVM classifier (Taherzadeh et al., 2016). The features used were sequence information in the form of 20 dimensional binary vectors, evolutionary information obtained from PSI-BLAST in the form of position specific scoring matrix (PSSM) (Altschul et al., 1997), structural information obtained from the prediction tool SPIDER 2.0 (Heffernan et al., 2015) containing solvent accessible surface area (ASA) and secondary structure (SS), and physicochemical properties (steric parameter, hydrophobicity, volume, polarizability, isoelectric point, helix probability, and sheet probability) obtained from the AA Index database (Kawashima et al., 1999). These features were fed through the SVM model, which produced a predicted label (Vapnik, 2000). SPRINT-Str (Taherzadeh et al., 2017) was later developed by the same team to predict protein-peptide binding sites using the random forest classifier (Breiman, 2001), employing additional protein 3-dimensional structure features and similar validation techniques. CRF-PPI (Wei et al., 2015) and SSWRF (Wei et al., 2016) are similar methods and also use SVM and random forest algorithms.

The problem that exists but gets ignored is that all current methods have extremely low rates of detecting actual binding sites. In most proteins, about 94% of the residues do not bind with peptides. If a tool predicted all residues as non-binding without running any computation at all, it would achieve an impressive 94% accuracy and 100% specificity score, but unfortunately would detect zero binding sites in the proteins. The utility of a protein-peptide binding site predictor lies in its ability to detect the actual binding sites, this measure is known as sensitivity, and the current methods are limited to very low sensitivity scores. The challenge is to develop a predictor whose classification is not skewed towards the non-binding sites and at the same time achieve a high binding site detection rate. In this paper, we have tried to address this problem with our proposed method, Visual, which uses a generally successful deep learning technique, convolutional neural networks (CNN), to predict the binding sites in proteins. Other methods have utilized CNNs for binding site identification, however, none have been used for the same objective. For example, DeepMHC (Hu and Liu, 2017) uses a 1-dimensional CNN model for predicting whether a given peptide will bind with a specific protein domain, MHC. They use 1-dimensional multi-channel one-hot vectors to represent the 13 amino acid long peptide sequence as input. On the contrary, Visual uses each individual protein amino acid as input to predict whether it will bind with a peptide. Another very successful tool, DeepSite (Jimenez et al., 2017) uses a deeper CNN model to identify pockets in proteins where ligands are likely to bind. It uses 3D images as input, which contributes to the high performance it achieves. However, this tool is for identifying binding sites for all ligands, and not specifically peptides. The overall high performance of CNNs in various domains as well as protein interaction is evident. The area to optimize now is the derivation of the protein information and representation so it can be used as CNN input. A recently developed technology, DeepInsight (Sharma et al., 2019) is a general model that can be applied for non-image samples, like protein data. This is the first technique that applies three steps of element arrangement, feature extraction and classification. The element arrangement step is used to arrange an image suitable for CNNs. In DeepInsight, data is transformed into images by applying either KPCA (kernel principal analysis component) or t-SNE (t-distributed stochastic neighbor embedding) followed by convex hull algorithm. In this work, we have employed a simple yet novel method whereby extracted and calculated features of protein sequences, such as residue sequence and locality, sequence-based structure predictions, evolutionary information and physicochemical properties, are arranged into image-like representations that are then processed by a CNN. Visual algorithm detects over twice as many binding sites in the same dataset as previously published works.

2. Materials and methods

This section discusses the dataset used in this work, followed by a description of the methods used to transform the data and predict the protein-peptide binding sites in proteins.

2.1. Dataset

Protein-peptide binding data was extracted from BioLiP (Yang et al., 2013), where chains with less than 30 amino acid residues were said to be peptides. Redundant proteins that had more than 30% similarity were removed using the Blastclust toolkit (Biegert et al., 2006). From this set, 10% of the proteins were randomly selected and set aside as the independent test set, TS125. Another 10% of the proteins were randomly selected and used as a validation set, while the remaining proteins formed the training set. The resulting training, validation and independent test sets contained 1004 proteins with 243,766 residues, 112 proteins with 22,823 residues and 125 proteins with 30,870 residues, respectively.

Each residue in the protein sequence is classified as either positive (binding) or negative (non-binding). All the subsets described above have a class ratio of approximately 17:1, this means, on average, there is about 1 peptide binding site in an 18 residue long protein segment. There are way more non-binding residues in a protein than those that are binding, so as a binary classification problem, this is a case of highly imbalanced class distribution.

2.2. Features

Features of the constituent amino acid residues were derived in various ways such that a detailed description of each site was available for the implemented model to classify. These features are discussed below:

- **Half sphere exposure (HSE)** is a measure of how buried an amino acid is in the protein 3D structure. The HSE values are calculated based on the contact numbers of the upward and downward hemispheres, as well as the pseudo Cβ-Cα bonds (Hamelryck, 2005).
- A deep learning-based predicting tool, SPIDER2 (Yang et al., 2014), was used to obtain extended information about each residue. The tool has shown impressive prediction results, as noted in the literature.
  - **Secondary Structure (SS)** provides perspective into the local 3-dimensional conformation of the protein. Predicted values include probabilities for each of the three classes, α-helix, β-sheet and coil.
  - The **Accessible Surface Area (ASA)** describes the degree of solvent accessibility of a residue within a protein.
  - **Local backbone angles** include θ, τ, ϕ and ψ. These are torsion angles between contiguous residues and provide insight into the residue's geometric relation to its locality.
- The **PSSM** is another feature that is available for proteins that has been widely used in the literature. The sequence-profiles were obtained from PSI-BLAST (Altschul et al., 1997) using E-value threshold of 0.001 in three iterations to extract the 20-dimensional vector for each amino acid in the protein.
- **Physicochemical Properties** - The amino acid residues can also be represented with their physicochemical properties. These include steric parameter, hydrophobicity, volume, polarizability, isoelectric point, helix and sheet probability (Kawashima et al., 1999).

2.3. Method

CNN classifiers require images as inputs, however, the protein data used here is not in an image format. Therefore, we first transformed the data into image to feed in the network.

2.3.1. Input transformation

Various properties of the Amino Acid residues were used to create the image-like representations. The following sections describe how we transformed the protein data.

Feature vector

First, each residue's features, $F_1, \ldots, F_6$, were combined to form a feature vector.

$F_1 = \{HSE_1,\ HSE_2,\ HSE_3\}$

$F_2 = \{SS_1,\ SS_2,\ SS_3\}$

$F_3 = \{ASA\}$

$F_4 = \{\theta,\ \tau,\ \phi,\ \psi\}$

$F_5 = \{PSSM_{i,m}\}$ for all $m = 1, \ldots, 20,$ where m represents the mth column of the ith amino acid

$F_6 = \{PP_1,\ \ldots,\ PP_7\}$

So, if kth protein sequence P_k has n residues; i.e., $P_k = \{R_1, R_2, \ldots, R_n\}$, then $R_i = \{F_1, F_2, \ldots, F_6\}$.

Therefore, stacking the features horizontally results in a feature vector R_i of size 38, that contains information about the residue's structure, evolution and physicochemical properties.

Windowed segment

A sliding window approach is used to capture the locality of residues. Each residue R_i is represented by a segment S_{R_i} that is of a fixed length. We have used a window size of 7 as it is not too small as to not be effective and not too large as to hinder the computer hardware performance. The residue of concern R_i is in the center of adjacent residues, 3 upstream and 3 downstream, are also captured in S_{R_i}.

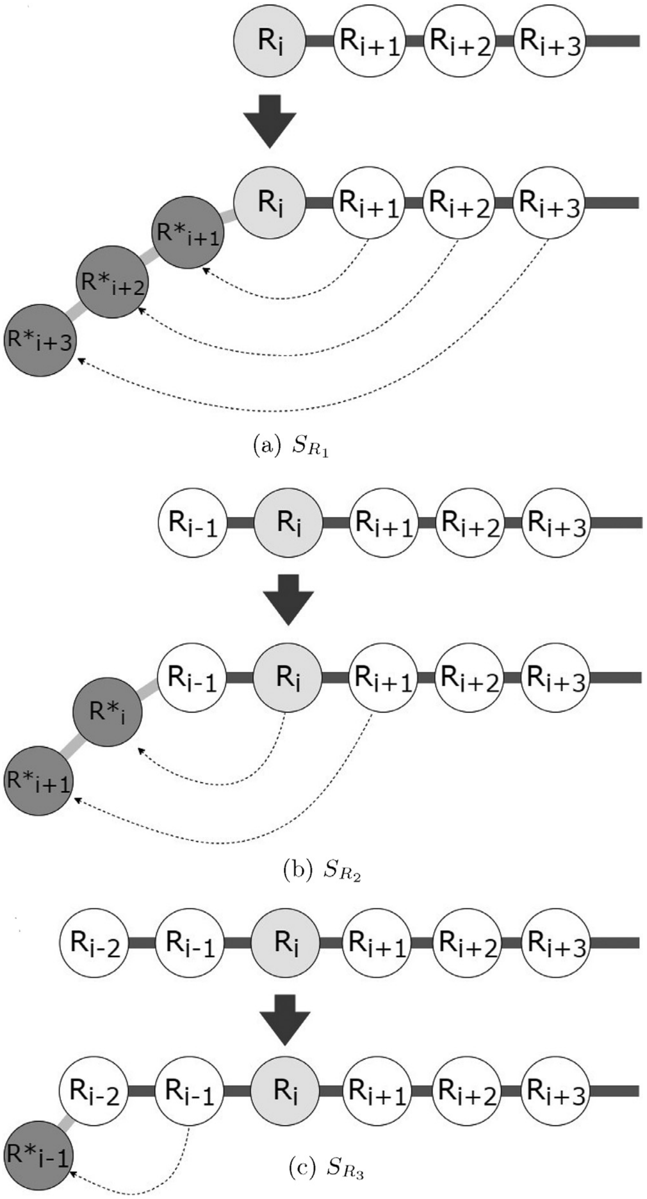
Generally, $S_{R_i} = \{R_{i-3}, \ldots, R_i, \ldots, R_{i+3}\}$ for all $4 \le i < n-3$.

For residues that are at either edge of the protein sequence, $1 \le i \le 3$ and $n-3 \le i \le n$, the missing side of the segment was augmented by mirroring the available residues. The expressions below describe how the missing residues (marked with *) were created. The start edge of the sequence is shown below in Fig. 2:

The result was a set of 2-dimensional arrays that could be normalized and used in a CNN classifier as 7 × 38 pixel images. The features were normalized so that each value was in a range of 0 and 1. When viewed as a greyscale image, the lighter pixels represent high values, and the darker pixels represent low values (refer to Fig. 3).

2.3.2. Convolutional neural network - CNN

CNNs are deep neural networks that process data that come in the form of multiple arrays where the local values are so closely related that they form detectable motifs, like images. The concept of CNN arose from the workings of the biological visual system in humans. Given a field of view, the visual system scans patches of the field and learns to recognize the objects (feature maps) based on those observed patches. Similarly, here, we convert the protein features into visual data and train the CNN to learn the feature maps in the given synthesized images. CNN models involve complex matrix operations which demand high processing power. This work was achieved using the GTX1060Ti graphics card, programmed using PyTorch, a Python based deep learning platform (PyTorch, 0000). The source code for the Visual model can be accessed online (Visual, 0000). The Visual model consists of 2 sets of convolution layers, followed by a pooling layer and a fully connected layer. In the first convolution layer, 256 [3 × 3] kernels slide over the [7 × 38] input image performing convolution operation



(a) S_{R_1}

(b) S_{R_2}

(c) S_{R_3}

Fig. 2. (a) shows $S_{R_1} = \{R^*{}_{i+3}, R^*{}_{i+2}, R^*{}_{i+1}, R_i, R_{i+1}, R_{i+2}, R_{i+3}\}$ where $i = 1$, (b) shows $S_{R_2} = \{R^*{}_{i+2}, R^*{}_{i+1}, R_i, R_{i+1}, R_{i+2}, R_{i+3}, R_{i+4}\}$where $i = 2$, and (c) shows $S_{R_3} = \{R^*{}_{i+1}, R_i, R_{i+1}, R_{i+2}, R_{i+3}, R_{i+4}, R_{i+5}\}$where for $i = 3$.

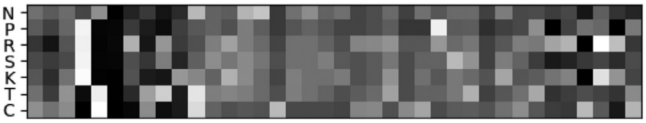


Fig. 3. An example of an image-like input representing the center residue Serine (S), with window size = 7. Thus, a 7 × 38 image that can be classified with a CNN classifier. In order from left to right: 3 pixels represent the HSE values, 3 pixels represent the SS predicted probabilities, 1 pixel represents the ASA value, 4 pixels represent the backbone angles, 20 pixels represent the PSSM, and 7 pixels represent the Physicochemical properties of the amino acids.

resulting in 256 [5 × 36] convolved feature maps. These are further transformed with the application of the Rectified Linear Unit (ReLU) activation function ($y = \max(0, x)$), producing rectified feature maps. The second convolution layer contains 256 [2 × 2] kernels that perform convolution operation over the 256 [5 × 36] feature maps, resulting in 256 [4 × 35] convolved feature maps. The ReLU activation function is applied to these as well, followed by a pooling layer. In the pooling layer, the maximum values in every 2 × 2 patch of the feature maps are collected via a sliding window to form a more robust pooled feature map. These are then flattened into a 1 × 8704 feature vector, which is then fed through a fully connected layer, resulting in a 1 × 2 output that represents the two classes: non-binding and binding (refer to Fig. 4).

This model processed 128 samples per batch and an average loss per batch was calculated using the Cross-Entropy Loss function upon comparing the predicted outputs with the actual target labels. The internal weights of the network were adjusted using the Adam optimizer (Kingma and Ba, 2014), which is an optimized variant of the gradient decent algorithm (Ruder, 2016). To avoid the problem of overfitting, early stopping technique was used to select the optimal number of epochs for training. It was found that the CNN model produced best validation results when trained for 18 epochs.

2.3.3. Bayesian optimization of CNN hyperparameters

CNN models contain many hyperparameters that have varying effects on their overall performance. There are a few methods that aid in tuning these hyperparameters. Grid search is the strategy of trying out all possible values to arrive at the combination of hyperparameters that produces the best model. This is a useful method, however, it is very time-consuming. Another strategy is to try out values or options randomly to find the combination that produces the best model. Random search is also useful, but also relatively inefficient. A more apt strategy is to use some algorithm that produces values or choices to implement in the model. Bayesian optimization has shown promising results recently in finding hyperparameters for models most efficiently (Snoek et al., 2012). Rather than trying random values or trying every possible value, Bayesian optimization uses calculated values for configuring the model's hyperparameters based on prior observations. The experiment was allowed to run for 350 iterations with different combinations of the two selected hyperparameters to achieve the best performing model. The first hyperparameter found using this method was the number of kernels in the two convolution layers of the CNN. The algorithm was given a list of options to select from $2^3, 2^4, 2^5, \ldots, 2^{10}$. The second hyperparameter was the learning rate (alpha) used by the Adam optimizer when updating the weights of the network, where values ranged between 0.00001 and 0.001. The optimal hyperparameters were found at the 293rd iteration where *number of kernels* $= 2^8 =$ *256* and *Adam (alpha) learning rate* = *0.000091* (refer to Fig. 5).

2.3.4. Performance evaluation

To effectively evaluate the performance of the method, the following values were calculated from the test outcome (confusion matrix):

- True positives (TP): the number of actual binding residues correctly predicted as binding sites.
- True negatives (TN): the number of actual non-binding residues correctly predicted as non-binding.
- False positives (FP): the number of actual non-binding residues incorrectly predicted as binding sites.
- False negatives (FN): the number of actual binding residues incorrectly predicted as non-binding sites.

Sensitivity is the measure of how well the actual binding sites are identified as binding sites. It is often called the recall, hit rate, or true positive rate (TPR). Sensitivity is a vital measure of performance in our case since being able to detect maximum residues in a protein that would bind with peptides can help in understanding protein interaction much better. Since binding sites are scarce in a protein, the rate at which the predictor method is able to detect

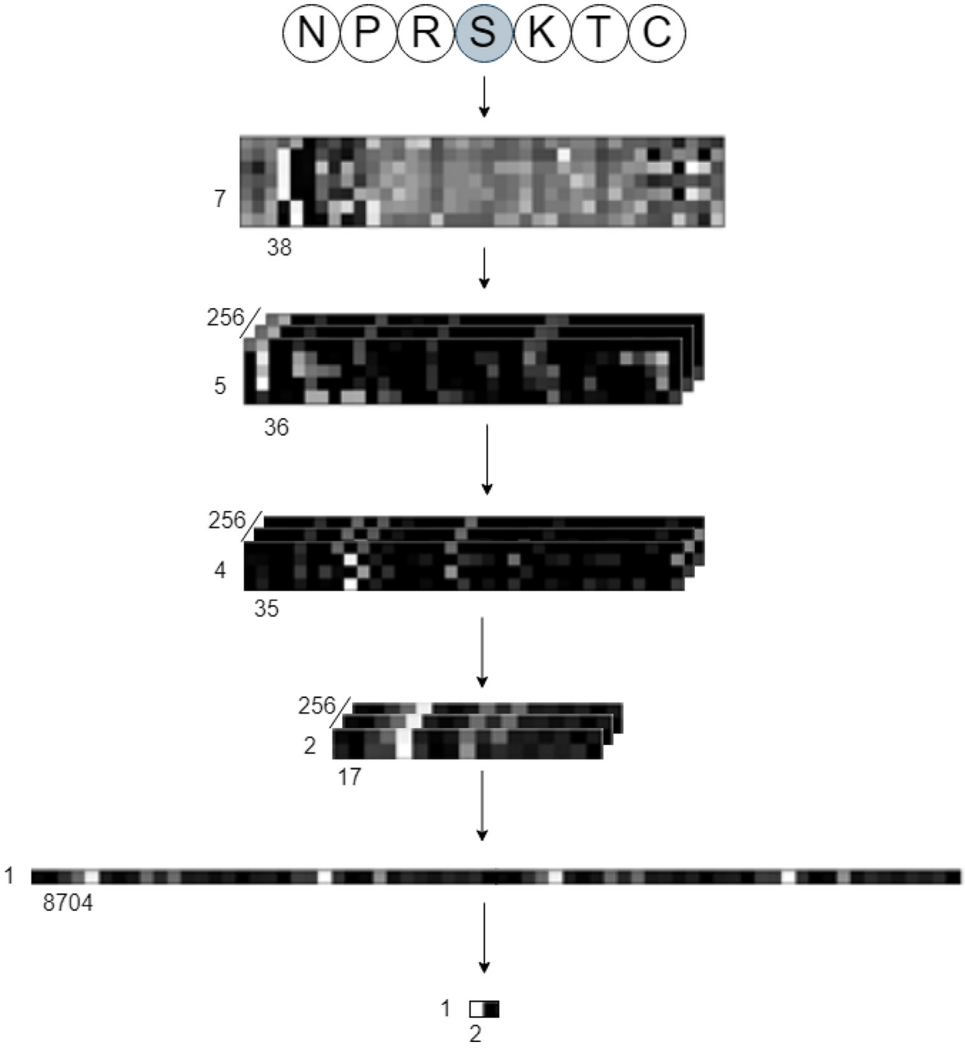


Fig. 4. The various transformations occurring throughout the CNN on a sample input are shown. The amino acid sequence is transformed into a [7 × 38] pixel image, which is fed into the CNN. The first convolution layer produces 256 [5 × 36] matrices. The second convolution layer converts these into 256 [4 × 35] matrices. Next, the maxpooling layer transforms them into 256 [2 × 17] matrices, which are then flattened into a single [1 × 8704] vector. This is passed through a fully connected dense layer. This produces the final [1 × 2] vector, where the index 0 represents negative class and index 1 represents the positive class. In the sample shown, the predicted output is negative.

them is vital. It is calculated by

$$Sensitivity = \frac{TP}{TP + FN} \quad (1)$$

Specificity is the ability of the predictor to correctly classify actual non-binding sites as such. In this case, most residues (about 94.4% of the test set TS125) do not bind with peptides. The specificity can be calculated by

$$Specificity = \frac{TN}{TN + FP} \quad (2)$$

Mathews correlation coefficient (MCC) is a score that is seen as a balanced measure that takes into account all 4 statistics from the confusion matrix. It can be calculated as

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (3)$$

An additional score, the area under the receiver operating characteristic (ROC) curve (AUC) is also obtained for the classifier. The ROC curve (see Fig. 7) is a curve created by plotting the true positive rate (TPR or sensitivity) against the false positive rate (FPR or 1 - specificity).

3. Results and discussion

It is important to highlight the type of problem the protein-peptide binding site prediction case is. Although the problem can be dealt with as binary classification of the two classes (binding and non-binding sites for each residue in the protein sequence), it must be realized that the need to classify in the first place is to be able to correctly pick out the actual binding sites from the non-binding sites. Analyzing our model (Visual) with the unseen data set TS125, we have found that it is able to predict the class of each residue in a protein sequence with the highest sensitivity compared to any other tool. It is apparent that the protein-peptide binding sites predicted by Visual are quite close to the binding sites revealed by the experimental method (see Fig. 6). Application areas, such as drug design, require reliable detection of binding sites. Visual is an attempt to improve this detection rate, and the results are positive with possibilities of further improvement.

Since about 94.4% of the residues do not bind with peptides and only 5.6% of them do, using the accuracy of the predictor as a basis for judging performance is misleading. The goal is to correctly classify the actual binding sites (which are only 5.6% of the

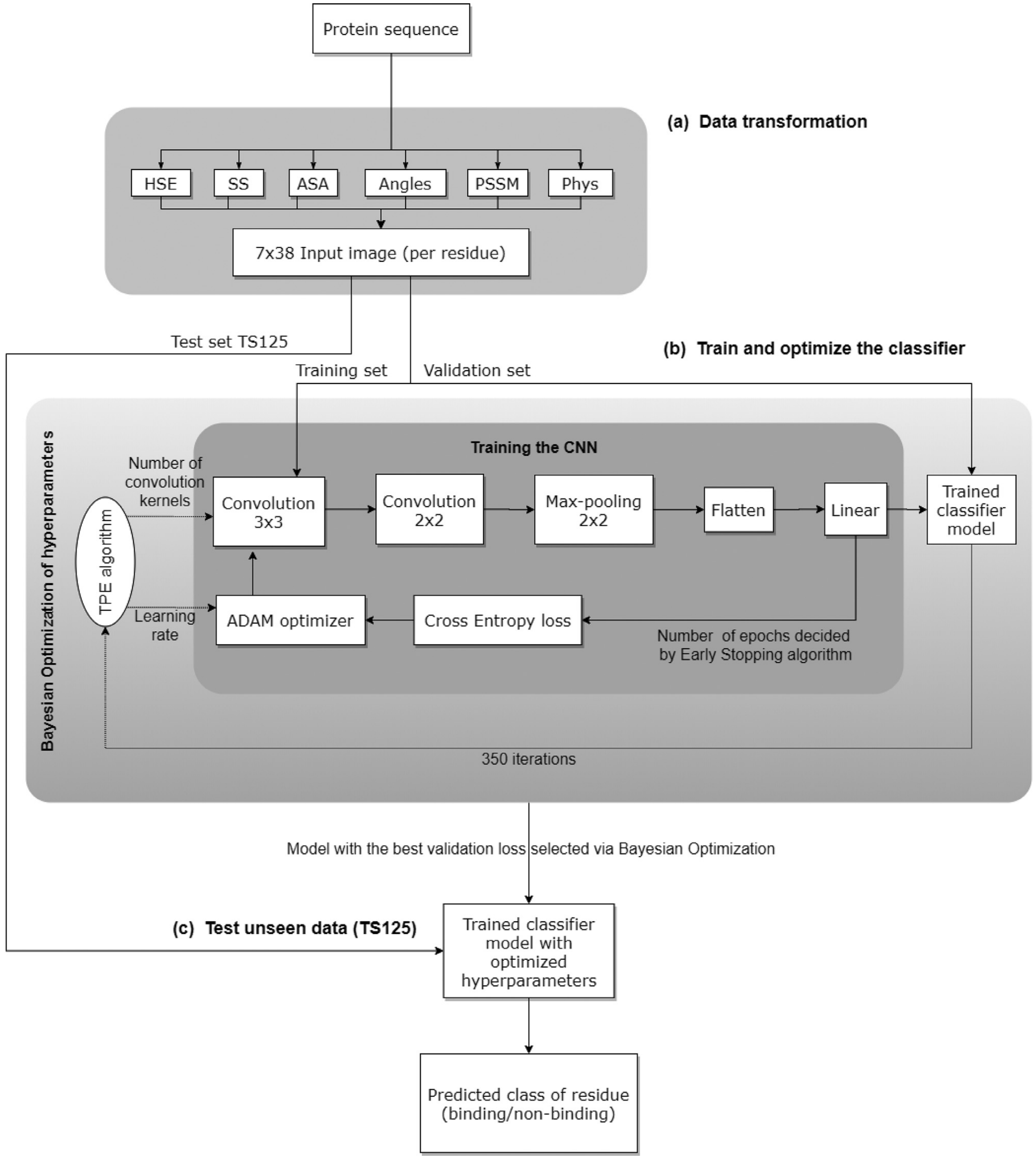

Protein sequence
(a) Data transformation
HSE
SS
ASA
Angles
PSSM
Phys
7x38 Input image (per residue)
Test set TS125
(b) Train and optimize the classifier
Training set
Validation set
Bayesian Optimization of hyperparameters
Training the CNN
Number of convolution kernels
TPE algorithm
Learning rate
Convolution 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 (TS125)
Trained classifier model with optimized hyperparameters
Predicted class of residue (binding/non-binding)

Fig. 5. Flow diagram of the processes involved in (a). transforming the protein-binding data, (b). training CNN to achieve optimal internal state and (c). testing the final model using the test set TS125.

data set) as such while keeping the number of falsely classified binding sites as low as possible. This trade-off between the true-positive and false-positive rates is depicted in the Receiver Operating Characteristics (ROC) curve shown in Fig. 7. A method that predicts peptide binding sites randomly (poor predictor) will have a linear diagonal curve (dashed line) and the area under the ROC curve (AUC) will be 0.5, whereas the best predictor will be higher and have AUC = 1. Visual achieves AUC = 0.73, which is higher than all other methods except one, SPRINT-Str (Taherzadeh et al., 2017) (however, this method shows extremely low detection rate of peptide binding sites). Methods published previously have generally shown very low rates of detecting actual binding sites correctly. The highest sensitivity so far was by the method Peptimap which was able to correctly detect only 32% of the binding sites in the same test set TS125. Our method is able to correctly detect 67% of binding sites in the test set TS125, the highest TPR so far achieved. Table 1 shows a comparison of the results of the top tools available for this problem.

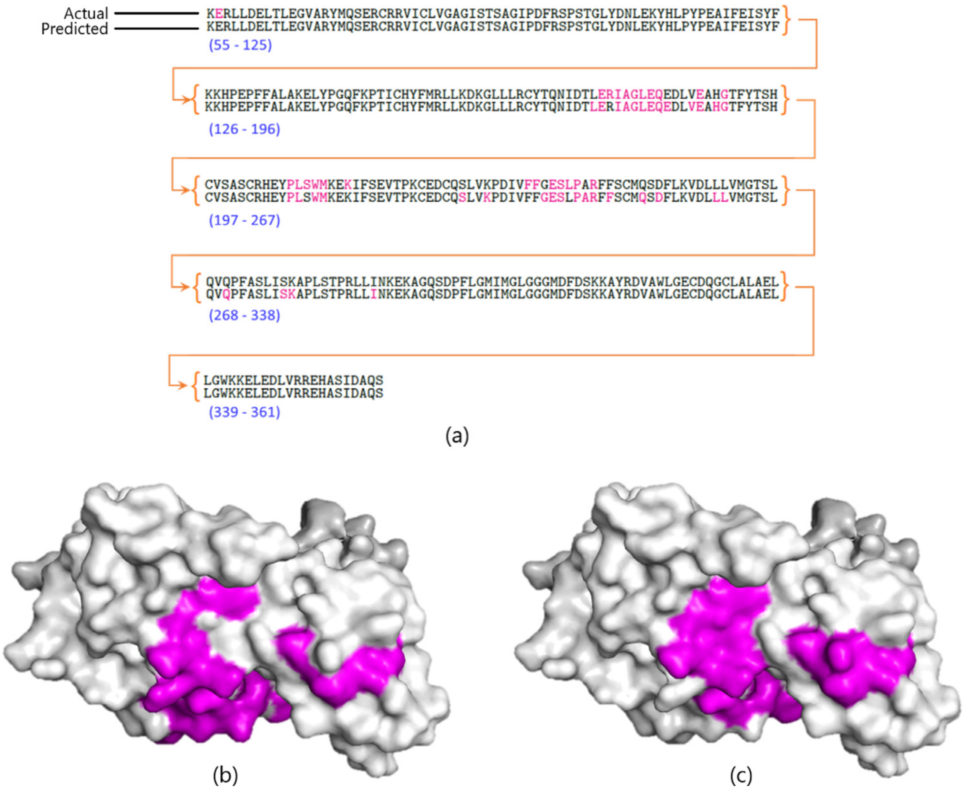

Actual
Predicted
KERLLDELTLEGVARYMQSERCRRVICLVGAGISTSAGIPDFRSPSTGLYDNLEKYHLPYPEAIFEISYF
KERLLDELTLEGVARYMQSERCRRVICLVGAGISTSAGIPDFRSPSTGLYDNLEKYHLPYPEAIFEISYF
(55 - 125)
KKHPEPFFALAKELYPGQFKPTICHYFMRLLKDKGLLLRCYTQNIDTLERIAGLEQEDLVEAHGTFYTSH
KKHPEPFFALAKELYPGQFKPTICHYFMRLLKDKGLLLRCYTQNIDTLERIAGLEQEDLVEAHGTFYTSH
(126 - 196)
CVSASCRHEYPLSWMKEKIFSEVTPKCEDCQSLVKPDIVFFGESLPARFFSCMQSDFLKVDLLLVMGTSL
CVSASCRHEYPLSWMKEKIFSEVTPKCEDCQSLVKPDIVFFGESLPARFFSCMQSDFLKVDLLLVMGTSL
(197 - 267)
QVQPFASLISKAPLSTPRLLINKEKAGQSDPFLGMIMGLGGGMDFDSKKAYRDVAWLGECDQGCLALAEL
QVQPFASLISKAPLSTPRLLINKEKAGQSDPFLGMIMGLGGGMDFDSKKAYRDVAWLGECDQGCLALAEL
(268 - 338)
LGWKKELEDLVRREHASIDAQS
LGWKKELEDLVRREHASIDAQS
(339 - 361)
(a)
(b)
(c)

Fig. 6. (a) shows the amino acid sequence of the protein 4l3oA. The sequence has been split into segments (55–125, 126–196, ...) to fit on the page. The upper row shows the actual peptide binding sites (magenta) while the lower row shows the peptide binding sites (magenta) as predicted by our method. (b) and (c) show computer-generated images of the protein 4l3oA, where (b) shows he actual binding sites in magenta and (c) shows the predicted binding sites in magenta.

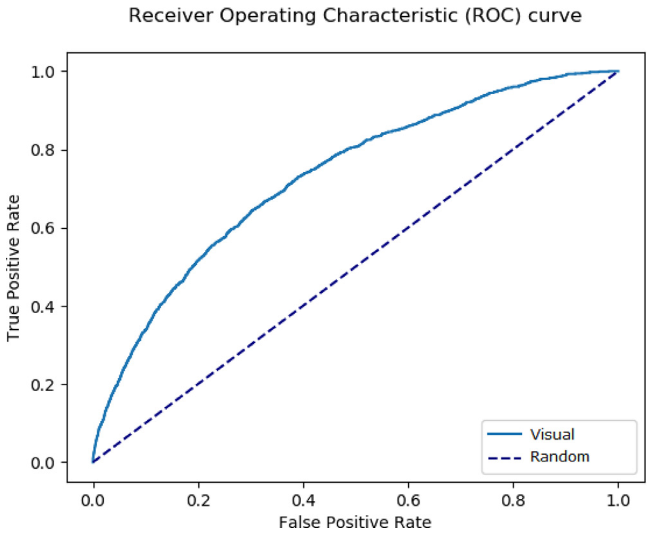


Fig. 7. ROC curve for this method, Visual, on the independent test set TS125. The curve portrays the performance of the method by plotting the True Positive Rate against the False Positive Rate.

Table 1
Comparison of different methods on the TS125 test set. The results stated were obtained from Taherzadeh et al. (2017), where TS125 was used for comparison.

Methods		SEN	SPE	MCC	AUC
SPRINT-Str	Taherzadeh et al. (2017)	0.24	0.98	0.29	0.78
SPRINT	Taherzadeh et al. (2016)	0.21	0.96	0.20	0.68
Peptimap	Bohnuud et al. (2017)	0.32	0.95	0.27	0.63
Pepsite	Petsalaki et al. (2009)	0.18	0.97	0.20	0.61
PinUp	Liang et al. (2006)	0.24	0.91	0.13	0.58
VisGrid	Li et al. (2008)	0.24	0.93	0.15	0.63
Visual	proposed method	0.67	0.68	0.17	0.73

4. Future work

There is room for improvement and especially in reducing the number of non-binding residues that get falsely classified as binding sites. The improvement may come about with better data preprocessing so that the image that is fed into the CNN for classification is richer in specific information about the residue. First improvement may be achieved by arranging the protein features $F1, \ldots, F6$ in an optimized order. More sophisticated feature engineering and deeper CNN topology, such as recently published DeepInsight (Sharma et al., 2019), may produce better protein-peptide binding site prediction results. Secondly, improvement may be achieved by using more advanced computing environment such that window size greater than 7, and various other CNN topologies can be experimented with. Additionally, more work can be done in employing other types of deep-learning methods, such as RNN, to predict peptide binding sites in proteins.

5. Conclusion

A deep learning method, Visual, that can predict the peptide binding sites in a protein was proposed. The method is a 2 layer convolutional neural network that uses an image-like representation of the constituent amino acids to detect the binding sites in a protein. The binding site detection rate of Visual (67%) is over twice as high as previously published methods (32%). It can be concluded that protein data can be transformed into image-like data usable by CNN methods, and that CNN can be successfully optimized to achieve better results compared to the current methods.

Availability of codes

The code for Visual can be found on GitHub via this link https://github.com/WafaaWardah/Visual Visual.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jtbi.2020.110278.

References

Altschul, S.F., Madden, T.L., Schffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J., 1997. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402.

Berman, H., Henrick, K., Nakamura, H., 2003. Announcing the worldwide protein data bank. Nat. Struct. Mol. Biol. 10, 1545–9985. doi:10.1038/nsb1203-980.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., 2016. The protein data bank. Nucleic Acids Res 28, 235–242.

Biegert, A., Mayer, C., Remmert, M., Soding, J., Lupas, A.N., 2006. The MPI bioinformatics toolkit for protein sequence analysis. Nucleic Acids Res. 34. doi:10.1093/nar/gkl217.

Bohnuud, T., Jones, G., Schueler-Furman, O., Kozakov, D., 2017. Detection of peptide-binding sites on protein surfaces using the peptimap server. Methods Mol. Biol. 1561, 11–20.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1). doi:10.1023/A:1010933404324.

Calderone, A., Castagnoli, L., Cesareni, G., 2013. Mentha: a resource for browsing integrated protein-interaction networks. Nat. Methods 10, 690–691.

Dhole, K., Singh, G., Pai, P.P., Mondal, S., 2014. Sequence-based prediction of protein–protein interaction sites with L1-logreg classifier. J. Theor. Biol. 348, 47–54.

Guo, L., Luo, C., Zhu, S., 2013. MHC2SKpan: a novel kernel based approach for pan-specific MHC class II peptide binding prediction. BMC Genomics 14 (5), S11. doi:10.1186/1471-2164-14-S5-S11.

Hamelryck, T., 2005. An amino acid has two sides: a new 2d measure provides a different view of solvent exposure. Proteins Struct. Funct. Bioinf. 59 (1), 38–48.

Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., Sattar, A., Yang, Y., Zhou, Y., 2015. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. Sci. Rep. 5 (11476).

Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. 89 (22), 10915–10919. doi:10.1073/pnas.89.22.10915.

Hou, T., Xu, Z., Zhang, W., McLaughlin, W.A., Case, D.A., Xu, Y., Wang, W., 2009. Characterization of domain-peptide interaction interface: a generic structure-based model to decipher the binding specificity of sh3 domains. Mol. Cell. Proteomics 8 (4), 639–649. doi:10.1074/mcp.M800450-MCP200.

Hu, J., Liu, Z., 2017. DeepMHC: deep convolutional neural networks for high-performance peptide-MHC binding affinity prediction. doi:10.1101/239236.

Janin, J., 1999. Wet and dry interfaces: the role of solvent in protein–protein and protein–DNA recognition. Structure 7, 277–279. doi:10.1016/S0969-2126(00)88333-1.

Jimenez, J., Doerr, S., Martinez-Rosell, G., Rose, A.S., De Fabritiis, G., 2017. DeepSite: protein-binding site predictor using 3D-convolutional neural networks. Bioinformatics 33 (19), 3036–3042. doi:10.1093/bioinformatics/btx350.

Kawashima, S., Ogata, H., Kanehisa, M., 1999. AAindex: amino acid index database. Nucleic Acids Res. 27 (1), 368–369. doi:10.1093/nar/27.1.368.

Kingma, D. P., Ba, J., 2014. Adam: a method for stochastic optimization. CoRRabs/1412.6980.

Kundu, K., Costa, F., Huber, M., Reth, M., Backofen, R., 2013. Semi-supervised prediction of sh2-peptide interactions from imbalanced high-throughput data. PLoS ONE 8 (5), 1–15. doi:10.1371/journal.pone.0062732.

Kuzmanov, U., Emili, A., 2013. Protein-protein interaction networks: probing disease mechanisms using model systems.. Genome Med. 5, 37.

Li, B., Turuvekere, S., Agrawal, M., La, D., Ramani, K., Kihara, D., 2008. Characterization of local geometry of protein surfaces with the visibility criterion.. Proteins 670–683.

Liang, S., Zhang, C., Liu, E., Zhou, Y., 2006. Protein binding site prediction using an empirical scoring function.. Nucleic Acids Res. 3698–3707.

Murakami, Y., Mizuguchi, K., 2010. Applying the naive bayes classifier with kernel density estimation to the prediction of protein-protein interaction sites.. Bioinformatics 26 (15), 1841–1848. doi:10.1093/bioinformatics/btq302.

Neduva, V., Linding, R., Su-Angrand, I., Stark, A., Masi, F.d., Gibson, T.J., Lewis, J., Serrano, L., Russell, R.B., 2005. Systematic discovery of new recognition peptides mediating protein interaction networks. PLoS Biol. 3 (12). doi:10.1371/journal.pbio.0030405.

Nibbe, R.K., Chowdhury, S.A., Koyuturk, M., Ewing, R., Chance, M.R., 2011. Protein-protein interaction networks and subnetworks in the biology of disease.. Wiley Interdiscip. Rev. Syst. Biol. Med. 3, 357–367.

Niv, M.Y., Weinstein, H., 2005. A flexible docking procedure for the exploration of peptide binding selectivity to known structures and homology models of PDZ domains. Journal of the American Chemical Society 127 (40), 14072–14079. doi:10.1021/ja054195s. PMID: 16201829

Peng, Z., Kurgan, L., 2015. High-throughput prediction of rna, dna and protein binding regions mediated by intrinsic disorder. doi:10.1093/nar/gkv585.

Petsalaki, E., Stark, A., Garcɖa-Urdiales, E., Russell, R.B., 2009. Accurate prediction of peptide binding sites on protein surfaces. PLoS Comput. Biol. 5 (3), 1–10. doi:10.1371/journal.pcbi.1000335.

Porollo, A., Meller, J., 2007. Prediction-based fingerprints of protein-protein interactions.. Proteins 66 (3), 630–645.

PyTorch PyTorch. An open source machine learning framework that accelerates the path from research prototyping to production deployment. https://pytorch.org/ retrieved on 15/3/2020.

Roche, D.B., Brackenridge, D.A., McGuffin, L.J., 2015. Proteins and their interacting partners: an introduction to protein-ligand binding site prediction methods.. Int. J. Mol. Sci. 16 (12), 29829–29842. doi:10.3390/ijms161226202.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. abs/1609.04747.

Sharma, A., Vans, E., Shigemizu, D., Boroevich, K.A., Tsunoda, T., 2019. DeepInsight: a methodology to transform a non-image data to an image for convolution neural network architecture. Sci. Rep. 9.

Singh, G., Dhole, K., Pai, P., Mondal, S., 2014. SPRINGS: Prediction of protein-protein interaction sites using artificial neural networks. J. Proteomics Computat. Biol. 1 (1), 7.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, vol. 2, pp. 2951–2959.

Taherzadeh, G., Yang, Y., Zhang, T., Liew, A.W.-C., Zhou, Y., 2016. Sequence-based prediction of protein peptide binding sites using support vector machine. Journal of Computational Chemistry. 0.1002/jcc.24314.

Taherzadeh, G., Zhou, Y., Liew, A.W.-C., Yang, Y., 2017. Structure-based prediction of protein-peptide binding regions using random forest. Structural Bioinformatics 1–8. 0.1093/bioinformatics/btx614.

Toussaint, N.C., Widmer, C., Kohlbacher, O., Ratsch, G., 2010. Exploiting physico-chemical properties in string kernels. BMC Bioinform. 11 (8), S7. doi:10.1186/1471-2105-11-S8-S7.

Vapnik, V., 2000. The Nature of Statistical Learning Theory. Springer-Verlag.

Visual Github repository. The source code is accessible at https://github.com/WafaaWardah/Visual.

Vlieghe, P., Lisowski, V., Martinez, J., Khrestchatisky, M., 2009. Synthetic therapeutic peptides: science and market. Drug Discov. Today 15 (1/2).

Wagner, M., Adamczak, R., Porollo, A., Meller, J., 2005. Linear regression models for solvent accessibility prediction in proteins.. J. Comput. Biol. 12, 355–369.

Wei, Z., Han, K., Yang, J., Shen, H., Yu, D., 2016. Protein–Protein interaction sites prediction by ensembling SVM and sample-weighted random forests. Neurocomputing 193, 201–212.

Wei, Z., Yang, J., Shen, H., Yu, D., 2015. A cascade random forests algorithm for predicting protein-protein interaction sites. IEEE Trans. Nanobiosci. 14 (7), 746–760. doi:10.1109/TNB.2015.2475359.

Westermarck, J., Ivaska, J., Corthals, G.L., 2013. Identification of protein interactions involved in cellular signaling.. Mol. Cell. Proteomics 12, 1752–1763. doi:10.1074/mcp.R113.027771.

Xenarios, I., Eisenberg, D., 2001. Protein interaction databases. Curr. Opin. Biotechnol. 12 (4), 334–339. doi:10.1016/S0958-1669(00)00224-X.

Yan, J., Friedrich, S., Kurgan, L., 2016. A comprehensive comparative review of sequence-based predictors of dna- and rna-binding residues.. Brief Bioinform. 17, 88–105.

Yang, J., Roy, A., Zhang, Y., 2012. BioLiP: a semi-manually curated database for biologically relevant ligand-protein interactions. Nucleic Acids Res. 41, 1096–1103.

Yang, J., Roy, A., Zhang, Y., 2013. BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions. Nucleic Acids Res. 41, D1096–D1103. doi:10.1093/nar/gks966.

Yang, Y., Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., Sattar, A., Zhou, Y., 2014. SPIDER2: a package to predict secondary structure, accessible surface area, and main-chain torsional angles by deep neural networks. In: Prediction of Protein Secondary Structure, pp. 55–63.