Supplementary material for "Peptide-protein interface classification using convolutional neural networks"

Lucas Moraes dos Santos[1 [0000−0003−4214−1576]], Diego Mariano[1 [0000−0002−5899−2052]], Luana Luiza Bastos[1 [000−0002−6932−0438]], Alessandra Gomes Cioletti[1 [0009−0008−3486−3548]], and Raquel Cardoso de Melo-Minaridi[1 [0000−0001−5190−100X]]

[1] Laboratory of Bioinformatics and Systems, Institute of Exact Sciences, Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
raquecm@dcc.ufmg.br

Abstract. Peptides are short chains of amino acid residues linked through peptide bonds, whose potential to act as protein inhibitors has contributed to the advancement of rational drug design. Indeed, understanding the interactions between proteins and peptides is potentially helpful for several biotechnological applications. However, it is not a trivial task since peptides can adopt different conformations when interacting with proteins. In this paper, we develop a classification model for protein-peptide interfaces using a convolutional neural network and distance maps. To evaluate our proposal, we performed two case studies classifying protein-peptide interfaces based on peptide sequences and receptor classes. Additionally, we compared the distance map approach with a graph-based structural signatures approach. We aim to find out if a convolutional neural network could classify peptides just from the patterns of distances in these maps. In conclusion, graph-based methods were slightly superior in almost all comparisons performed. However, distance map-based signature methods achieved better results for some classes, such as classifying hormones, hormones, and viral proteins. These results shed light on the potential use of distance maps for classifying protein-peptide interfaces. Nevertheless, more experiments may be needed to explore this use.

Keywords: Convolutional neural networks · distance maps · protein-peptide interactions.

1 Introduction

Peptides are short-chain molecules consisting of two to fifty amino acid residues linked through peptide bonds. They have several essential functions in human physiology, such as acting as hormones, neurotransmitters, growth factors, ion channel ligands, or anti-infective agents [27]. Moreover, recent research suggests that peptides play a vital role in protein-protein interactions, constituting a significant percentage of such interactions within cells[2].

Compared to proteins, peptides have more chemical versatility because they can be more easily modified. Additionally, peptides exhibit low resistance and limited non-target activity, making them suitable for therapeutic agents [18][34]. As a result, peptide drug development has become one of the hottest topics in pharmaceutical research.

Designing new peptides and peptide-based compounds for drug development and biotechnological applications requires understanding the structure and recognition of protein-peptide complexes. With the aid of databases containing protein-peptide complexes, researchers can analyze and gain insights into the mechanisms of protein-peptide recognition, paving the way for future discoveries [20] [6]. However, these studies depend on public structure databases, such as PDB (Protein Data Bank) and more specialized databases, as Propedia [23].

Propedia is a database of peptide-protein interactions designed to provide a comprehensive and current dataset of complex protein-peptide experiments [22]. In a recent study [23], graph-based structural signatures [21] have been used to extract characteristics of protein-peptide complexes collected from Propedia. Then, several machine-learning approaches were used to classify protein-peptide complexes [23]. The results demonstrated the potential use of graph-based signatures for protein-peptide classification. However, other approaches could be used to construct new signature types, such as distance maps.

Mathematical approaches applied to understanding the properties of proteins have provided insights relevant to structural bioinformatics [17]. For example, information about the structure of biomolecules is encoded in the internal distances, represented by square matrices known as distance matrices. These representations contain the pairwise distances between residues in a protein and are used to infer protein-protein interactions [17].

A distance matrix can be defined as $\mathbf{d} = (d_{ij})$, where d_{ij} is the Euclidean distance between the ith and the jth residue. Generally, the coordinates of the atoms of C_α (carbon-α) and/or C_β of the residuals are input to the method [17].

Recent studies have shown that predicting the structures of proteins can be done using two-dimensional images known as distance maps (DMs), representing the inter-residue distance matrices of proteins. These maps are increasingly used to compare biomolecular structures and analyze functional differences between proteins [14]. By comparing DMs of homologous structures, researchers can identify similarities and differences in their patterns of structural flexibility [14]. In addition, DMs have the added advantage of being low-dimensional, invariant to rotation and translation of structures, making parameter calculation and efficient learning [7], which is desirable for artificial intelligence applications, such as convolutional neural networks.

Convolutional neural networks (CNNs) are a class of deep neural networks, of the type *feed-forward*, specialized in processing data that have a topology of *grid* (e.g. image) [11]. The architecture of a CNN is analogous to the connectivity pattern of neurons in the human brain being inspired by the organization of the visual cortex, where neurons in different layers detect increasingly complex features of visual stimuli. As an allusion to their name, these neural networks

use a mathematical operation called *convolution* in feature learning, as opposed to matrix multiplication common in multilayer perceptrons (MLP) [11].

This class of neural networks has shown great potential in applications involving pattern recognition in images [11], being used recently in conformational analysis, structure prediction, protein classification, etc [24]. The basic structure of CNNs consists basically of two parts: feature learning (convolution and pooling layers) and classification (fully-connected layers) [24].

In this study, we model the interface region of the protein-peptide complex through a two-dimensional representation of the interatomic distance matrix, known as a distance map. We aim to find out if a CNN could classify peptides from the patterns of distances in these maps.

2 Material and methods

2.1 Data collection

The protein-peptide complexes used in this work come from the Propedia web database (`http://bioinfo.dcc.ufmg.br/propedia2`). We performed two case studies. First, we analyzed 1,111 peptides from five clusters grouped by sequence similarities (clusters S0, S1, S112, S151, and S162). Additionally, we collected and analyzed 6,238 peptides from six Propedia datasets: AntimicrobialDB, ViralDB, EnzymeDB, MembraneDB, HormoneDB, and PlantDB. Lastly, we compared our results to the neural network analysis of graph-based signatures shown in [23] (signature method: aCSM-ALL with 0.2Å of step and distance max of 20Å [29]; parameters used in Orange Data Mining [8]: neurons in hidden layers="300", solver="Adam", activation="ReLu", maximal number of iterations="200", regularization alpha="0.001", and replicable training).

2.2 Generation of distance maps

We focus on atoms within the interface region to generate distance maps for protein-peptide complexes. We select the residues C_α (alpha carbon) from each .pdb file and extract their corresponding coordinates (*x, y, z*) from the protein and peptide structures. Using these coordinates, we calculate the Euclidean distance between atoms within the interface region to create a distance matrix between residues. In this matrix, peptide atoms correspond to the ordinate axis, while protein atoms correspond to the abscissa axis. Finally, we transform the distance matrix into a two-dimensional image (*.png* format) using Python's Matplotlib library. The algorithms for developing the process described above and obtaining the distance maps were developed in Python (version 3.7.9).

2.3 Pre-processing: data augmentation and resizing

To prepare the DMs for input into our neural network, we applied preprocessing techniques that consisted of three steps: resizing, data augmentation, and

rescaling. Firstly, we resized the DMs to 64x64 pixel dimensions to fit the square input structure required by the CNN architecture. Following this, we applied data augmentation to the DM set using a series of techniques such as brightness adjustments, sharpening filters, and horizontal/vertical shifts. Previous studies have shown that data augmentation techniques can significantly improve classification models, particularly for imbalanced datasets [36]. Lastly, we rescaled each DM so that the pixel values were converted to a range between 0 and 1 since neural networks tend to perform better with values in this range [11].

2.4 Model architecture

Our model is based on representation learning [4], a technique that allows the system to automatically learn important features from a large amount of data, allowing it to learn a representation specific to the task. We use a popular representational learning technique called Deep learning, that involves using deep neural networks that optimize weight parameters by combining simple and complex features to construct hierarchical representations of input data [24]. We employ a type of deep neural network called *convolutional neural networks* (CNN).

We developed a sequential and uniform architecture [5] comprising a linear stack of 2D convolutional layers. The first two layers have 32 filters each, while the last two have 64 filters each. We define a $3x3$ convolution *kernel* with a stride of 1. After the convolutional layers, we include a pooling layer with max-pooling using a $2x2$ pool array and a stride of 2. To enhance the non-linear properties of the feature maps generated, we apply the Rectified Linear Unit (ReLU) activation function [11], which is followed by a Batch Normalization layer to zero center the activations [13].

For our model, we selected a batch size of 32, which determines the number of samples processed by the network in one pass. Typically, larger batch sizes demand more memory, so it's common to use values like 32 or 64 [25]. The input data was structured as *tensors*, defined by the input shape, which includes the image dimensions (height and width) and the number of color channels (RGB is equivalent to 3). Additionally, the batch size was specified [5].

We generate the input layer for the fully connected layers (FC layer) by vectorizing the feature maps and concatenating them into a flattened array. For multiclass classification, the FC layer acts as a classifier with 512 nodes, and we employ the softmax activation function to process its output [32]. Additionally, to improve generalization and prevent overfitting, we set the Dropout rate to 0.5, as it has produced a significant reduction in error for values in the range of [0.3-0.6] [31]. To optimize the model, we utilized the Adaptive Moment Estimation (Adam) optimizer [16] and trained it over 100 epochs.

We implemented the source code to preprocess the distance maps and develop the model architecture using the Python programming language (version 3.7.9), along with consolidated machine learning libraries and neural networks such as TensorFlow [1] and Keras [5].

2.5 Experimental design

We split the dataset into training and testing subsets, with 80% and 20% of the data, respectively. A test set was previously extracted by randomly selecting samples from the initial dataset. No data augmentation was applied to test set. For the training set, we used 80% of the samples for tuning the hyperparameters of the network, while the remaining 20% was reserved for validating the model. Commonly, a percentage $\gamma < 0.5$ of the training data is used to validate the model [10]. We stopped adjusting parameters when the number of training epochs reached a predefined.

We employ an alternative version of the cross-validation (CV) technique approach known as k-fold CV [28]. This technique randomly divides the training set into k subsets of equal size (n/k), where n is the total number of training samples. In this case, we define $k = 5$ because it is possible to guarantee that $\gamma \geq 0.1$, often recommended [10]. One subset is reserved for validation, and the remaining $k-1$ subsets are used for parameter estimation. We repeated this process k times rotating the validation subset each time. In the end, We estimated the performance based on the average of the k error rates corresponding to each one of k partitions [10]. Since the problem involves multiclass classification, we selected the categorical cross-entropy loss function to train the model.

In this particular problem, the distribution of classes relative to sequence clusters and sub-datasets of peptides from Propedia is unbalanced. To evaluate the model's performance, we used complementary metrics to the error rate. We compared the performance of multiple classifiers trained with the same dataset and calculated complementary metrics such as precision, recall, and F1-Score, which help in choosing the optimal classifier from a performance perspective [35]. We obtained a multi-class confusion matrix to calculate these metrics. Additionally, we presented the performance of the developed model as a function of what it correctly predicted by class [35]. To calculate model performance metrics, we utilized the open-source Python scikit-learn library.

The models were implemented on Google's virtual environment, Colab, which provides access to a Jupyter Notebook. The hardware used consisted of a dual-core processor with 13.6 GB of RAM and an L3 cache of 40-50 MB. However, to accelerate the process, an NVIDIA A100-SXM4 GPU with 40GB of memory was also used, along with an additional 89.6 GB of available RAM.

3 Results and discussion

In this study, we use convolutional neural networks to classify interfaces of protein-peptide interactions using computational modeling based on distance maps. We aim to verify if signatures based on contact maps can be as good as graph-based signatures. Graph-based signatures are considered state-of-the-art for classifying macromolecules. Nevertheless, distance map-based signatures can be helpful when combined with convolutional neural networks.

To assess this, we analyzed a problem of high impact in the biotechnology industry: protein-peptide interactions. We collected structures of complexes

protein-peptide from the Propedia database. For each collected protein-peptide complex, we extracted the interaction interface. We then computed the distance maps, applied the preprocessing steps, and performed the training and testing process using CNNs. Fig. 1 presents an overview of how our methodology was applied.

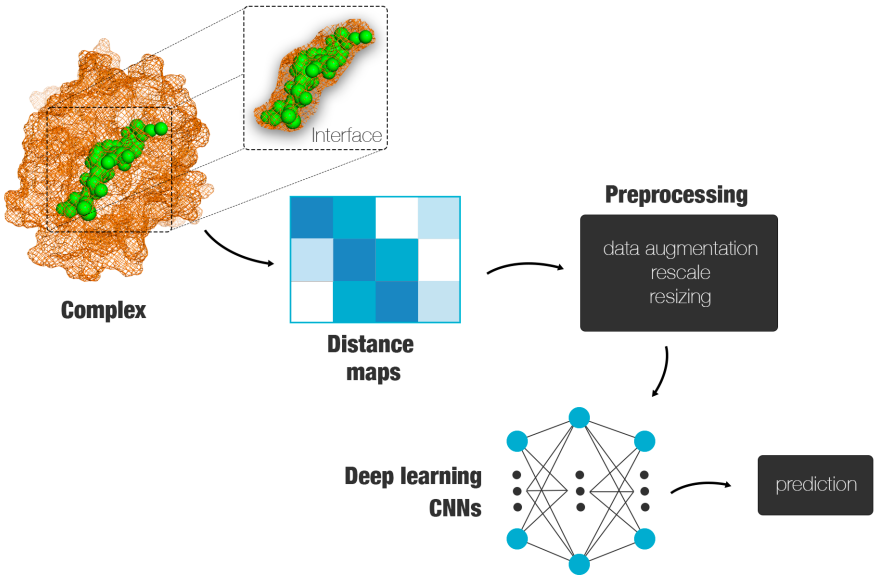

Fig. 1. Overview of the methodology used to evaluate distance map-based signatures.

To evaluate the methodology, we performed two case studies described below.

3.1 Case Study 1: Sequence Clusters

The first case study considers peptide sequences. The sequence classification problem is well established in the literature, with several methodologies, algorithms, and tools that provide optimal results for clustering. On the other hand, our method uses the three-dimensional structure of the peptide and the contact interface with the receptor, which has the potential to detect more details of the characteristics of peptides and their interactions.

Thus, in this first case study, we collected 1,111 peptides from five clusters grouped by sequence similarities: S0, S1, S112, S151, and S162. These clusters are the five most populated in the Propedia database, with sizes of 503, 184, 161, 142, and 122, respectively. Table 1 summarizes the results of the grouping performed based on the signatures of the distance map and compares with the results of the signatures based on graphs by Martins et al. (2023) [23].

Table 1. Case study 1: performance metrics (%) for the model based on CNNs. Graph-based signature values were obtained from [23].

Model	Graph-based signature	Distance-map-based signature
Accuracy	0.92	0.91
Precision	0.92	0.92
Recall	0.92	0.92
F1	0.91	0.91
S0	100%	97.0%
S1	100%	88.0%
S112	100%	93.0%
S151	85.3%	82.0%
S162	40.5%	95.0%

As we hypothesized, representations based on distance maps proved efficient when we analyzed the similarity of the sequences. From table 1, we can observe that the predictions related to case study 1 had an accuracy of 0.91, precision of 0.92, recall of 0.92, and F1-score of 0.91. This is comparable to the performance achieved by neural networks trained from state-of-the-art representations such as structural signatures [23] that obtained 0.92, 0.92, 0.92, and 0.91 for accuracy, precision, recall, and F1, respectively.

In the work of Martins et al. [23], the clustering of S0, S1, and S112 groups obtained 100% accuracy. On the other hand, they only obtained 85.3% and only 40.5% for S151 and S162 clusters. Although we obtained a slightly lower result for most of the five groups (97%, 88%, 93%, 82%, and 95%, respectively), our methodology can better handle the classification of the latter group (S162). As a disclaimer, we can argue that an improvement in the parameterization performed in work by Martins et al. [23] could get better results for this last cluster. Still, our primary goal here is to demonstrate that signatures based on distance maps can be as good as signatures based on graphs for classifying protein-peptide interfaces.

Moreover, our analysis showed that the percentage of accurate predictions per class was above 80%, indicating that our approach effectively discriminates between different data classes. These findings highlight the potential of distance maps for sequence analysis, suggesting that they may be particularly useful in scenarios where methods based on structural signatures are not feasible or appropriate.

3.2 Case Study 2: Peptide types

In the second case study, we consider the role of the type of peptide interacting with the receptor. For example, we look at peptide-protein complexes classified as antimicrobial, enzyme, hormone, membrane, plant, and viral. These classes were obtained from the PDB descriptions and are assigned based on the locals where the peptides were obtained.

Thus, in this second case study, we collected 6,238 contact interface structures from Propedia: a protein-peptide database. We analyzed six Propedia datasets: AntimicrobialDB (n=10), ViralDB (n=294), EnzymeDB (n=5,344), MembraneDB (n=152), HormoneDB (n=212), and PlantDB (n=256). Then, we compared our results with the neural network and graph-based signature experiments described in Martins et al. (2023). Table 2 summarizes the results of the grouping performed based on the signatures of the distance map and compares with the results of the signatures based on graphs by Martins et al. (2023) [23].

Table 2. Case study 2: performance metrics (%) for the model based on CNNs. Graph-based signature values were obtained from [23].

Model	Graph-based signature	Distance-map-based signature
Accuracy	0.91	0.76
Precision	0.90	0.80
Recall	0.91	0.76
F1	0.90	0.77
Antimicrobial	22.2%	0.00%
Enzyme	97.5%	91.0%
Hormone	63.5%	86.0%
Membrane	36.3%	72.0%
Plant	79.2%	52.0%
Viral	29.9%	55.0%

In Case Study 2, we observed a decrease in model performance for accuracy, precision, recall, and F1 values compared to Case Study 1. A possible reason for the drop in performance could be the impact of dataset imbalance on the final prediction, leading to a bias toward correctly classifying the cluster with more samples (Enzyme). Despite this, we were able to achieve an accuracy of 0.76.

Despite this, when we examine the percentages of correct predictions, we obtain values higher than the structural signatures for the Hormone (86%), Membrane (72%), and Viral (55%) classes. This indicates the model successfully classified peptides within subsets corresponding to their function. This success aligns with the sequence-structure-function paradigm. Since distance maps serve as an alternative 2D representation of the three-dimensional structure, the model is expected to accurately classify the peptides in the relevant functional subsets of the Propedia.

4 Conclusion and perspectives

In conclusion, graph-based methods were slightly superior in almost all comparisons performed. However, distance map-based signature methods obtained close results in the sequence-based classification. Also, in the second case study,

they achieved better results for some classes, such as classifying hormones, membranes, and viral proteins. These results shed light on the potential use of distance maps for classifying protein-peptide interface, which could be better explored in future experiments using protein-peptide complexes or other macromolecule complexes.

Acknowledgements The authors thank the funding agencies: CAPES, CNPq, and FAPEMIG.

Data availability Supplementary material, data, and scripts are available at https://github.com/LBS-UFMG/cnn-distance-maps.

References

1. Abadi, M. et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems, pp.1–16, 2016. https://doi.org/10.48550/arXiv.1603.04467
2. Angelova, A., Drechsler, M., Garamus, V. M., and Angelov, B. (2019). 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. doi:10.1002/cnma.201900468
3. Anishchenko, I. et al.: De novo protein design by deep network hallucination. Nature, vol. 600, pp. 547–552, 2020. https://doi.org/10.1038/s41586-021-04184-w
4. Bengio, Y., Courville, A., and Vincent, P.: Representation Learning: A Review and New Perspectives. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798-1828, Aug. 2013. https://doi.org/10.1109/TPAMI.2013.50
5. Chollet, F.: Deep Learning with Python. Manning, 4th ed.
6. Das, A. A., Sharma, O. P., Kumar, M . S., Krishna, R., and Mathur, P. P. (2013). Pepbind: A comprehensive database and computational tool for analysis of protein–peptide interactions. Genom Proteom Bioinform 11 (4), 241–246.
7. Defresne, Marianne, Sophie B., and Thomas S.: Protein Design with Deep Learning. International Journal of Molecular Sciences 22, no. 21: 11741. https://doi.org/https://doi.org/10.3390/ijms222111741
8. Demšar, Janez, et al. "Orange: data mining toolbox in Python." the Journal of machine Learning research 14.1 (2013): 2349-2353.
9. Douzas G. and Bacao F.: Effective data augmentation techniques for classification tasks in imbalanced datasets. Expert Systems with Applications, vol. 97, pp. 88-103, 2018.
10. Duda, R., Hart, P., Stork, G.: Pattern Classification, 2nd ed., USA, New York: Wiley, 2001.
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, ser. Adaptive computation and machine learning. USA: MIT Press, 2016.
12. Haykin, S.: Neural Networks - A Comprehensive Foundation, Canada. Pearson Prentice Hall, Upper Saddle River, 2001.
13. Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning, 2015, pp. 448-456. https://doi.org/10.48550/arXiv.1502.03167

14. Iyer, M., Jaroszewski, L., Sedova, M., Godzik, A.: What the protein data bank tells us about the evolutionary conservation of protein conformational diversity, Protein Science, vol. 31, no. 7, Jun. 2022. https://doi.org/10.1002/pro.4325
15. Jumper, J. et al., Highly accurate protein structure prediction with AlphaFold. Nature, vol. 596, pp. 583–589, 2021. https://doi.org/10.1038/s41586-021-03819-2
16. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. https://doi.org/10.48550/arXiv.1412.6980
17. Kloczkowski, A., Jernigan, R., Wu, Z., Song, G., Yang, L., Kolinski, A., Pokarowski, P.: Distance matrix-based approach to protein structure prediction. Journal of Structural and Functional Genomics, vol. 10, no. 1, pp. 67–81, Feb. 2009. https://doi.org/10.1007/s10969-009-9062-2
18. Lau, J. L., and Dunn, M. K. (2018). Therapeutic peptides: Historical perspectives, current development trends, and future directions. Bioorg. Med. Chem. 26, 2700–2707. doi:10.1016/j.bmc.2017.06.052
19. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L.: Backpropagation applied to handwritten zip code recognition. Neural Computation, vol. 1, no. 4, pp. 541–551, Dec. 1989.
20. London, N., Movshovitz-Attias, D., and Schueler-Furman, O. (2010). The structural basis of peptide-protein binding strategies. Structure 18 (2), 188–199.
21. Mariano, D., Santos, L. H., Machado, K. D. S., Werhli, A. V., de Lima, L. H. F., and de Melo-Minardi, R. C. (2019). A computational method to propose mutations in enzymes based on structural signature variation (SSV). Int. J. Mol. Sci. 20, 333. https://doi.org/10.3390/ijms20020333
22. Martins, P.M., Santos, L.H., Mariano, D. et al. Propedia: a database for protein–peptide identification based on a hybrid clustering algorithm. BMC Bioinformatics 22, 1 (2021). https://doi.org/10.1186/s12859-020-03881-z
23. Martins P, Mariano D, Carvalho FC, Bastos LL, Moraes L, Paixão V and Cardoso de Melo-Minardi R (2023) Propedia v2.3: A novel representation approach for the peptide-protein interaction database using graph-based structural signatures. Front. Bioinform. 3:1103103. https://doi.org/10.3389/fbinf.2023.1103103
24. Min, S., Lee, B., Yoon, S.: Deep learning in bioinformatics. Briefings in Bioinformatics, vol. 18, no. 5, pp. 851–869, 2017. https://doi.org/10.1093/bib/bbw068
25. Mishkin, D., Sergievskiy, N., Matas, J.: Systematic evaluation of CNN advances on the ImageNet. https://doi.org/10.1016/j.cviu.2017.05.007
26. Mitchell, T.: Machine Learning. New York, USA: McGraw-Hill, 1997.
27. Moreno-Camacho, C. A., Montoya-Torres, J. R., Jaegler, A., and Gondran, N. (2019). Sustainability metrics for real case applications of the supply chain network design problem: A systematic literature review. J. Clean. Prod. 231, 600–618. doi:10.1016/j.jclepro.2019.05.278
28. Mosteller, F., Tukey, J.: Data Analysis, including Statistics. In Lindzey, G., Aronson, E. (ed.), Revised Handbook of Social Psychology, vol. 2, pp. 80–203.
29. Pires, D. E. V., de Melo-Minardi, R. C., da Silveira, C. H., Campos, F. F., and Meira, W. (2013). aCSM: noise-free graph-based signatures to large-scale receptor-based ligand prediction. Bioinforma. Oxf. Engl. 29, 855–861. https://doi.org/10.1093/bioinformatics/btt058
30. Shapiro DH Jr. Adverse effects of meditation: a preliminary investigation of long-term meditators. Int J Psychosom. 1992;39(1-4):62-7. PMID: 1428622.
31. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting.

Journal of Machine Learning Research, vol.15, no. 56, pp.1929–1958, 2014. https://doi.org/10.5555/2627435.2670313
32. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 2nd ed., Burlington, MA, USA: Academic Press, 2009.
33. Torrisi, M., Pollastri, G., Le, Q.: Deep learning methods in protein structure prediction. Computational and Structural Biotechnology Journal, vol. 18, pp. 1301–1310, 2020.
34. Vinogradov, A. A., Yin, Y., and Suga, H. (2019). Macrocyclic peptides as drug candidates: Recent progress and remaining challenges. J. Am. Chem. Soc. 141, 4167–4181. doi:10.1021/jacs.8b13178
35. Webb, A., Copsey, K.: Statistical Pattern Recognition. New York, USA: Wiley, 2011.
36. G. Douzas and F. Bacao, "Effective data augmentation techniques for classification tasks in imbalanced datasets," Expert Systems with Applications, vol. 97, pp. 88-103, 2018."