


Article

TF-BAPred: A Universal Bioactive Peptide Predictor Integrating Multiple Feature Representations

Zhenming Wu, Xiaoyu Guo, Yangyang Sun, Xiaoquan Su *  and Jin Zhao *

School of Computer Science and Technology, Qingdao University, Ningxia Road, Qingdao 266071, China; wuzhenming@qdu.edu.cn (Z.W.); 2021020692@qdu.edu.cn (X.G.); 2021020706@qdu.edu.cn (Y.S.)

* Correspondence: suxq@qdu.edu.cn (X.S.); zhaojin@qdu.edu.cn (J.Z.)

Abstract: Bioactive peptides play essential roles in various biological processes and hold significant therapeutic potential. However, predicting the functions of these peptides is challenging due to their diversity and complexity. Here, we develop TF-BAPred, a framework for universal peptide prediction incorporating multiple feature representations. TF-BAPred feeds original peptide sequences into three parallel modules: a novel feature proposed in this study called FVG extracts the global features of each peptide sequence; an automatic feature recognition module based on a temporal convolutional network extracts the temporal features; and a module integrates multiple widely used features such as AAC, DPC, BPF, RSM, and CKSAAGP. In particular, FVG constructs a fixed-size vector graph to represent the global pattern by capturing the topological structure between amino acids. We evaluated the performance of TF-BAPred and other peptide predictors on different types of peptides, including anticancer peptides, antimicrobial peptides, and cell-penetrating peptides. The benchmarking tests demonstrate that TF-BAPred displays strong generalization and robustness in predicting various types of peptide sequences, highlighting its potential for applications in biomedical engineering.

Keywords: bioactive peptides; multiple feature representations; temporal convolutional networks; vector graph

MSC: 37M10



Citation: Wu, Z.; Guo, X.; Sun, Y.; Su, X.; Zhao, J. TF-BAPred: A Universal Bioactive Peptide Predictor Integrating Multiple Feature Representations. *Mathematics* **2024**, *12*, 3618. <https://doi.org/10.3390/math12223618>

Academic Editor: Vince Grolmusz

Received: 9 October 2024

Revised: 12 November 2024

Accepted: 13 November 2024

Published: 20 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bioactive peptides are short chains of amino acids that can influence diverse biological activities due to their specific structures and functions [1]. Bioactive peptides come in various types, each with its own specific functions and roles in the body. For example, antimicrobial peptides (AMPs) can act as part of the innate immune system, defending the body against pathogens [2]. Anticancer peptides (ACPs) serve several crucial functions in the context of cancer treatment, and their unique properties make them promising candidates for targeted therapeutic interventions [3]. Cell-penetrating peptides (CPPs) have the remarkable ability to smoothly interact with the negatively charged membranes of cells, making it easier for them to penetrate the cell's outer defenses [4]. Efficiently predicting and identifying these peptides is crucial for unraveling the basics of biological workings and propelling advancements in therapy.

Existing peptide prediction methods mostly focus on predicting specific types of peptides. For example, ACP-DL [5], StackACPred [6], ACP-check [7], and CACPP [8] are primarily used for identifying anticancer peptides. sAMPpred-GAT [9], DNNs [10], ENAMP [11], and AMP-EBiLSTM [12] are developed for predicting antimicrobial peptides. Cppsite2.0 [13], CPPred-RF [14], and BChemRF-CPPred [15] are employed in cell-penetrating peptide prediction. pLMFPred [16] are employed in functional peptides prediction. In addition, there are several general-purpose tools that have been specifically designed for extracting peptide sequence features, such as FusPB-ESM2 [17] and

TP-LMMSG [18]. FusPB-ESM2 constructs a feature extraction model by combining two pre-trained protein models, ProtBERT and ESM2. TP-LMMSG constructs a peptide sequence predictor by assembling a graph deep neural network model. Although these methods are developed to predict different types of peptides or proteins, they share common features within their respective models.

Various methods have been proposed to extract peptide features. For instance, the Amino Acid Composition (AAC) method quantifies the relative frequencies of individual amino acids in a peptide sequence, revealing its primary composition [19]. In contrast, PseAAC incorporates local structural information, offering a deeper understanding beyond ACC [20]. Similarly, methods like Dipeptide Composition (DPC) and Tripeptide Composition (TPC) analyze the frequency of occurrence of short peptide segments, capturing local structural patterns [21,22]. The Binary Profile Features (BPF) methodology entails the conversion of sequential data into a binary representation, where each segment of the sequence is depicted as a fixed-length binary vector [23]. These methods extract features from the perspective of the composition of peptide sequences and the frequency of amino acids.

Approaches like DCGR integrate the physicochemical properties of amino acids to construct CGR curves, offering a unique perspective on peptide characteristics [24]. Another method utilizing the physicochemical properties of amino acids is the Composition of k-spaced Amino Acid Group Pairs (CKSAAGP) [25]. Although numerous feature extraction algorithms exist, they mostly focus on local traits, neglecting peptides' global structural features. Consequently, it is crucial to develop an algorithm that can capture the entire sequence to enhance the accuracy of identification.

Machine learning models demonstrate strong learning patterns from data to make predictions, enabling them to accommodate diverse types and sources of bioactive peptide data. The commonly used machine learning models in peptide identification include Support Vector Machines (SVMs) [26], Naive Bayes (NB) [27], Random Forests (RFs) [28], and K-Nearest Neighbors (KNN) [29]. Although these models have demonstrated considerable effectiveness in peptide prediction tasks, they still possess limitations. The activity of bioactive peptides might involve intricate nonlinear relationships, which traditional machine learning methods might struggle to capture, as they are typically based on linear models and rely on manually extracted features.

With the advancement of deep learning, researchers have begun to employ deep neural networks for peptide prediction. This includes model such as Convolutional Neural Networks (CNNs) [30], Graph Convolutional Networks (GCNs) [31], Recurrent Neural Networks (RNNs) [30], Long Short-Term Memory Networks (LSTM) [32], and various network variants. Deep learning models can address the shortcomings of traditional machine learning in capturing nonlinear relationships. The activity of bioactive peptides is often influenced by long-range dependencies between different parts of the sequence. RNNs and LSTM typically capture long-term dependencies through gating units but may overlook capturing local patterns in peptide sequences. CNNs can effectively extract potential local relationships between amino acids but fail to capture long-term dependencies in sequences. Additionally, when sequences are too long, the issues of vanishing or exploding gradients persist [33].

Temporal Convolutional Networks (TCNs) combine the advantages of RNNs and CNNs while overcoming their drawbacks [34]. TCNs stack convolutional layers and build residual connections, enabling them to capture long-term dependencies in sequences without the common issue of gradient vanishing seen in RNNs. Additionally, TCNs address the limitation of CNN models in processing only local information and exhibit good generalization to unseen sequence data. Despite the tremendous potential of TCNs in bioactive peptide identification, they have not yet been applied in peptide recognition.

In this paper, we introduce TF-BAPred, a universal bioactive peptide predictor, using three-channel feature extraction (see Figure 1 for the workflow of TF-BAPred). The main contributions of this work include the following:

(i) We propose Fixed-Scale Vector Graph (FVG) feature extraction strategy using a fixed-scale vector graph to capture the global structural patterns of each peptide sequence. This approach aims to provide a more comprehensive understanding of the overall structural characteristics exhibited by peptide sequences.

(ii) We employ a TCN for automatic temporal feature extraction, facilitating the extraction of long-range dependency information among amino acids in peptide sequences while also capturing local patterns within the sequence. To the best of our knowledge, this is the first time a TCN has been used in peptide recognition.

(iii) We apply the TF-BAPred algorithm to different types of bioactive peptides, including antimicrobial peptides, anticancer peptides, and cell-penetrating peptides. The benchmarking tests demonstrate that TF-BAPred exhibits a more competitive performance across these types of peptides.

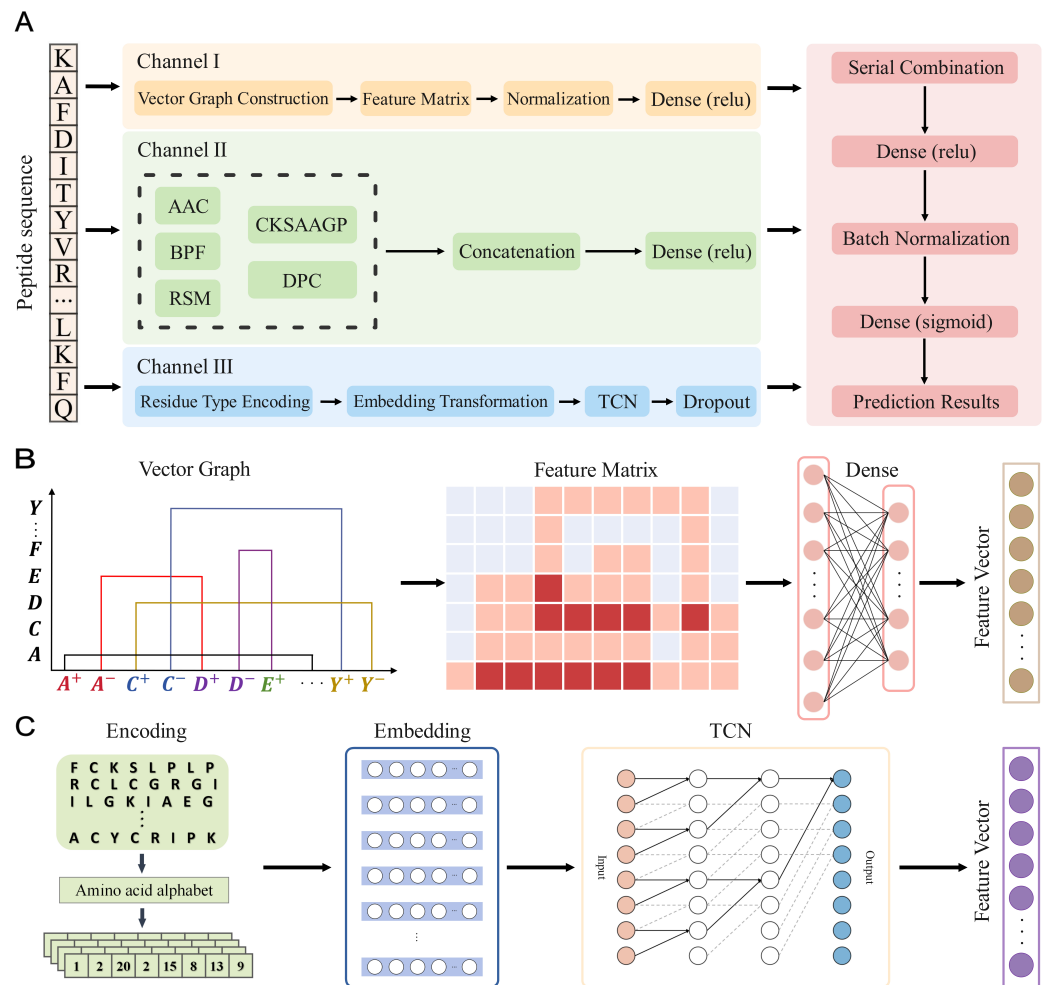


Figure 1. Overview of the TF-BAPred framework. (A) The original peptide sequences are individually input into three channels to extract sequence features from different perspectives. Subsequently, the feature vectors obtained from the three channels are fused and input into a fully connected neural network for classification training and prediction. (B) An example of a fixed-scale vector graph depicting the global structural patterns of each peptide sequence. (C) The framework for temporal feature extraction based on a temporal convolutional network.

In the introduction of this paper, we first review the background and current state of research in the field, followed by a brief overview of our work. In Section 2, we provide a detailed description of the methodology and datasets, outlining the basic framework of the experimental design, implementation steps, and specific details of certain methods. Section 3 presents the experimental results and data analysis, discussing the significance of

the main findings and their potential applications. Finally, in Section 4, we summarize this paper and propose directions for future research as well as possible improvements.

2. Materials and Methodology

2.1. Overview of the TF-BAPred

TF-BAPred integrates feature vectors extracted from three channels and feeds them through a series of linear transformations into a classification network composed of fully connected neural layers for predicting bioactive peptides. The first channel of TF-BAPred constructs a fixed-scale vector graph to capture the global structural patterns of peptide sequences. The completed vector graph can be transformed into a feature matrix and passes this matrix to a fully connected neural network for nonlinear transformation. In the second channel, the original sequences are encoded into a 730-dimensional feature vector by combining five different feature extraction methods: AAC, DPC, BPF, RSM, and CKSAAGP. The merged vector is then input into a fully connected neural network with ReLU as the activation function. The core structure of the third channel is TCN. The channel encodes the input peptide sequences into uniformly sized discrete numerical vectors and then transform them into fixed-length continuous low-dimensional vector representations through an embedding layer. These embedding vectors are fed into the TCN to extract sequential features with temporal information. The outputs from the TCN are passed through a dropout layer, which randomly ignores a fixed proportion of neurons to help alleviate overfitting. TF-BAPred integrates the features extracted by these three channels and passes the combined features through a series of transformations into a classification network with a sigmoid activation function. An overview of the TF-BAPred framework is illustrated in Figure 1.

2.2. Fixed-Scale Vector Graph

Inspired by [35], we proposed a novel approach that utilizes a fixed-scale vector graph to depict the global structural patterns of each peptide sequence. We named this method FVG. FVG can transform any peptide sequence into a fixed-size matrix using the following methods.

Suppose Σ denotes the alphabet encompassing all amino acids, and its size is m . FVG defines a one-to-one mapping function as follows.

$$F(x) = z, \quad x \in \Sigma, 1 \leq z \leq m, \quad (1)$$

As depicted in Equation (1), FVG converts each amino acid x in the set Σ into an integer z ($1 \leq z \leq m$), ensuring that no two different amino acids are mapped to the same integer.

For each peptide sequence $S = [s_1, s_2, \dots, s_n]$, FVG constructs a directed graph $G(V, E)$ to represent the topological structure of the sequence. Each amino acid x ($x \in \Sigma$) corresponds to two vertices, x^+ and x^- , in the graph, while the set V consists of $2m$ vertices. For each pair of adjacent characters, $s_i s_{i+1}$ ($s_i s_{i+1} \in S$, $1 \leq i \leq n - 1$, where n represents the length of sequence S). There is a corresponding edge (s_i^-, s_{i+1}^+) in the set of edges E . FVG standardizes the format of the graph $G(V, E)$ with the following rules:

- (i) For two vertices x^+ and x^- representing the same amino acid x , x^+ is positioned to the left of x^- .
- (ii) If vertices x^+ and x^- represent amino acid x , vertices y^+ and y^- represent amino acid y , and $F(x) < F(y)$, then vertices x^+ and x^- are placed to the left of vertices y^+ and y^- .
- (iii) For each edge (x^-, y^+) , the height of this edge is set to $F(y)$.
- (iv) For each edge (x^-, y^+) , the width of this edge is designed to be how many times the tuple xy appears in sequence S .

To describe the graph G mentioned above, FVG employs Algorithm 1 to transform G into an $m \times 2m$ dimensional matrix, where m denotes the size of set Σ . Algorithm 1 takes a peptide sequence as input, simulates the graph construction process described above, and represents the resulting graph as a matrix M . For each element M_{ij} of the matrix M ,

its value presents the total width of the edges in graph G passing through position $\langle i, j \rangle$ ($1 \leq i \leq m, 1 \leq j \leq 2m$).

Algorithm 1 Constructing Feature Matrix Based on FVG.

```

1: Input: a peptide sequence  $S$ 
2: Output: a feature matrix  $M$ 
3:  $M \leftarrow \text{zeros}(m, 2m)$ 
4: for  $s_i s_{i+1}$  in  $S$  do
5:    $height \leftarrow F(s_{i+1})$ 
6:    $out\_node \leftarrow 2F(s_i) + 1$ 
7:    $in\_node \leftarrow 2F(s_{i+1})$ 
8:   for  $k$  in 1 to  $height$  do
9:      $M[k][out\_node] \leftarrow M[k][out\_node] + 1$ 
10:     $M[k][in\_node] \leftarrow M[k][in\_node] + 1$ 
11:   end for
12:   if  $out\_node < in\_node$  then
13:     for  $k$  in  $out\_node + 1$  to  $in\_node - 1$  do
14:        $M[height][k] \leftarrow M[height][k] + 1$ 
15:     end for
16:   else
17:     for  $k$  in  $in\_node + 1$  to  $out\_node - 1$  do
18:        $M[height][k] \leftarrow M[height][k] + 1$ 
19:     end for
20:   end if
21: end for

```

Variations in both the length and composition of the peptide sequence can lead to differences in the widths and lengths of the edges within their corresponding graph. Consequently, this disparity may result in differences in the magnitudes of values for elements within the associated matrix. Such disparities can impact the accuracy of subsequent peptide predictions. To address this issue, FVG normalizes the matrix M to M^* . FVG employs the following formula to calculate the value of each element M_{ij}^* ($1 \leq i \leq m, 1 \leq j \leq 2m$) in matrix M^* :

$$M_{ij}^* = (M_{ij} - \beta) / (\alpha - \beta), \quad (2)$$

where β denotes the minimum element value within matrix M , while α signifies the maximum element value within matrix M . The resulting matrix M^* obtained using the aforementioned strategy serves as the feature matrix for the FVG method.

Figure 2 presents an example of constructing a fixed-scale vector graph and converting it into matrix M for the peptide sequence $S = \text{DVADVMYYV}$ with set $\Sigma = \{A, D, M, V, Y\}$. As shown in Figure 2b, the fixed-scale vector graph comprises 10 vertices labeled A^+ , A^- , D^+ , D^- , M^+ , M^- , V^+ , V^- , Y^+ , and Y^- , along with 7 edges. Specifically, the edge (D^-, V^+) has a width of 2 due to the sequence S containing 2 DV . Figure 2c represents a matrix M corresponding to the fixed-scale vector graph, which can be understood as a grayscale representation of this graph. The elements in the matrix represent the sum of the widths of the edges in graph G at the corresponding positions. For example, both edges (D^-, V^+) and (M^-, Y^+) pass through position $\langle 4, 6 \rangle$, with widths of 2 and 1, respectively. Consequently, the value of $M_{4,6}$ is 3.

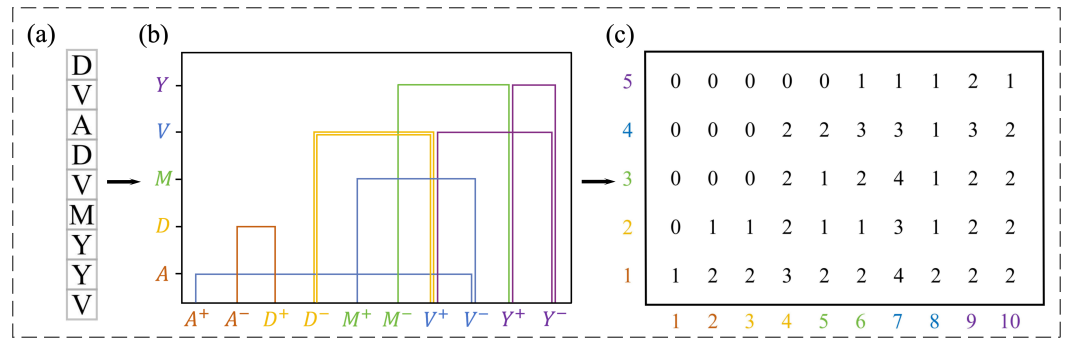


Figure 2. Example of constructing a fixed-scale vector graph. (a) A peptide sequence $S = DVADVMYYV$. (b) Assuming the amino acid alphabet consists of A, D, M, V, and Y, construct a vector graph based on the alphabet. (c) Generate a feature matrix based on the constructed vector graph.

2.3. Residue Sparse Matrix

Inspired by the k-mer sparse matrix [36], we propose a feature extraction method named Residue Sparse Matrix (RSM) that aims to capture both the position and composition of amino acids within each peptide sequence. RSM first constructs a boolean matrix based on the peptide sequence and the one-to-one mapping function F introduced in the previous section. Subsequently, RSM transforms this boolean matrix into a fixed-size feature vector. Further details are outlined as follows.

For each peptide sequence $S = [s_1, s_2, \dots, s_n]$, RSM defines an $m \times n$ -dimensional boolean matrix A , where m represents the number of amino acids in the alphabet Σ , and n represents the length of S . For each element A_{ij} ($1 \leq i \leq m, 1 \leq j \leq n$) in the matrix A , if $F(s_j) = i$, then $A_{ij} = 1$; otherwise, $A_{ij} = 0$.

The above definition of matrix A indicates that its dimensions vary depending on the length of the sequences, which complicates subsequent feature analysis. To address this issue, RSM performs singular value decomposition on matrix A to convert it into an m -dimensional vector Z . For the matrix A with dimensions $m \times n$, its singular value decomposition can be represented as follows:

$$A = USV^T, \tag{3}$$

where U is an $m \times m$ orthogonal matrix whose column vectors are the left singular vectors of A . S is an $m \times n$ diagonal matrix whose diagonal elements are the singular values of A , typically arranged in descending order. V represents an $n \times n$ orthogonal matrix whose row vectors are the right singular vectors of A . Let Z_i represent the i -th element of vector Z , and its computation is as follows:

$$Z_i = \sum_{j=1}^n (U_{ij} * S_{ij}) / n \quad (1 \leq i \leq m), \tag{4}$$

where U_{ij} and S_{ij} refer to the elements located at the i -th row and j -th column of matrices U and S , respectively. RSM regards Z as the feature vector representing the arrangement and composition of amino acids within the peptide sequence.

2.4. Temporal Convolutional Network

Analogous to a time series where each observation is ordered chronologically, the position of each amino acid within the sequence plays a critical role in bioactive peptides. The TCN shines as a deep learning architecture that is meticulously designed for tackling sequence modeling tasks. Its utilization of dilated convolutions and residual connections enables the network to effectively capture long-range dependencies, facilitating the accurate extraction of features from bioactive peptides. Moreover, the TCN can handle different scales of patterns by stacking multiple convolutional layers, each with varying convolu-

tional kernel widths. Therefore, we employ a TCN to extract features from the peptide sequence. The workflow of the TCN can be outlined as follows:

- Input

We configure the input format acceptable by the TCN as a three-dimensional tensor as follows:

$$T = (b, t, i), \quad (5)$$

where b denotes the number of samples in a batch, t represents the length of the time series, and i indicates the number of features at each time step.

- Construction of residual blocks

The residual block serves as the fundamental building unit in the TCN, typically composed of a series of convolutional layers. Departing from the prior paradigm of simple one-dimensional causal convolutional layers composing the basic building block, we adopt a scheme where residual blocks consist of two-layer convolutional blocks with identical dilation factors and residual connections. Within each convolutional block, we sequentially establish a one-dimensional convolutional layer, a normalization layer, a rectified linear unit (ReLU) activation layer, and a dropout layer to extract features from the data and perform feature transformations. Furthermore, to ensure smooth connectivity between residual blocks, the results of two convolutional blocks are combined, with a 1×1 convolutional block included within the residual block to ensure compatibility of input and output data shapes.

- Construction of residual network

After completing the basic residual block construction, we add the residual blocks to the network through residual connections to construct the residual network. This work is based on the Keras network framework. The construction of the residual network enables the TCN to have multiple stacked convolutional layers. Therefore, by adjusting the dilation convolution factor and filter size, the TCN has a more flexible receptive field compared to traditional CNNs. The receptive field size of the residual network can be defined as follows:

$$\text{Receptive field} = 2 \times (k - 1) \times r \times d + 1, \quad (6)$$

Specifically, k denotes the size of the convolutional kernel within residual blocks, r denotes the number of stacked residual blocks, and d represents the dilation rate for each individual residual block.

3. Results

TF-BAPred aims to explore effective features for different types of peptides and apply them to peptide identification. It introduces a novel feature representation method, called FVG, to represent the global structural patterns of each peptide. It also propose a feature extraction strategy called RSM that outputs a vector containing the types and positional information of amino acids. Additionally, it employs a TCN for automatic feature extraction. TF-BAPred is implemented using the Keras [37] framework with the TensorFlow [38] deep learning backend library. To benchmark TF-BAPred, we initially assessed the effectiveness of the proposed feature extraction methods (TCN and FVG). We subsequently evaluated the performance of TF-BAPred on six challenging datasets and compared it with ACP predictors including ACP-DL [5] and ACP-check [7], the AMP predictors including DNNs [10] and Ma's method [39], as well as the CPP predictor CPPred-RF [14]. In addition, we tested the performance of these predictors under different ratios of training and testing datasets.

3.1. Dataset Information

In order to facilitate the comparison of TF-BAPred with state-of-the-art approaches, we collected six challenging datasets encompassing three types of bioactive peptides. These

included two datasets related to anticancer peptides, ACP740 [5] and ACPmain [40]; two datasets focused on antimicrobial peptides, Veltri’s dataset [10] and Ma’s dataset [39]; and two datasets associated with cell-penetrating peptides, CPP924 [41] and CPPsite3 [14]. The details of these benchmark datasets are presented in Table 1. More detailed information about the data can be found in the Supplementary Materials.

Table 1. Characteristics of benchmarking data sets.

Dataset	Type	Positive	Negative	Total
ACP740	Anticancer peptides	376	364	740
ACPmain	Anticancer peptides	689	689	1378
CPP924	Cell-penetrating peptides	462	462	924
CPPsite3	Cell-penetrating peptides	187	187	374
Ma’s dataset	Antibacterial peptides	1085	58,776	59,861
Veltri’s dataset	Antibacterial peptides	1778	1778	3556

3.2. Assessment of TCN and FVG

To validate the effectiveness of the TCN, we employed a single TCN channel for peptide prediction, then replaced the TCN with a CNN or LSTM to evaluate its performance. Additionally, we merged the feature vectors derived from FVG with the aforementioned three methods to gauge the effectiveness of FVG. The average accuracy (ACC), sensitivity (SEN), specificity (SPE), Matthews correlation coefficient (MCC), and F1 score achieved using the above methods across the six datasets are presented in Table 2, while the ROC curves of the aforementioned strategies on each dataset are shown in Figure 3.

As shown in Table 2, the TCN outperformed both the CNN and LSTM. Owing to the TCN’s integration of a CNN’s local receptive fields and LSTM’s long-term dependency modeling capability, it could achieve a better understanding of the structural characteristics and patterns within bioactive peptide sequences. When combined with FVG, the performance of all of the above methods further improved. For example, The average F1 scores for CNN, LSTM, and TCN improved by 5.5%, 7.1%, and 6.9%, respectively. Additionally, the combination of TCN and FVG outperformed the other methods, showing increases of at least 5.6% in ACC, 7.6% in SEN, 2.1% in SPE, 6.5% in F1 score, and 17.4% in MCC compared to the alternative methods. Additionally, the results depicted in Figure 3 align with those presented in Table 2, reaffirming that the combination of the TCN and FVG yielded the most optimal performance across all of the datasets. For example, the AUC obtained via the combination of the TCN and FVG on the CPP924 dataset was 0.965, representing increases of 10.8%, 10.2%, 5.5%, 5.8%, and 6.9% compared to the CNN, LSTM, TCN, CNN + FVG, and LSTM + FVG, respectively.

Table 2. Evaluation of the effectiveness of TCN and FVG based on the average performance across six datasets.

Method	ACC	SE	SP	F1 Score	MCC
CNN	0.794	0.741	0.762	0.760	0.558
LSTM	0.778	0.740	0.782	0.737	0.537
TCN	0.823	0.779	0.861	0.799	0.616
CNN + FVG	0.819	0.770	0.827	0.802	0.610
LSTM + FVG	0.805	0.762	0.788	0.789	0.613
TCN + FVG	0.872	0.838	0.879	0.854	0.723

Note: The best values are shown in the boldface font.

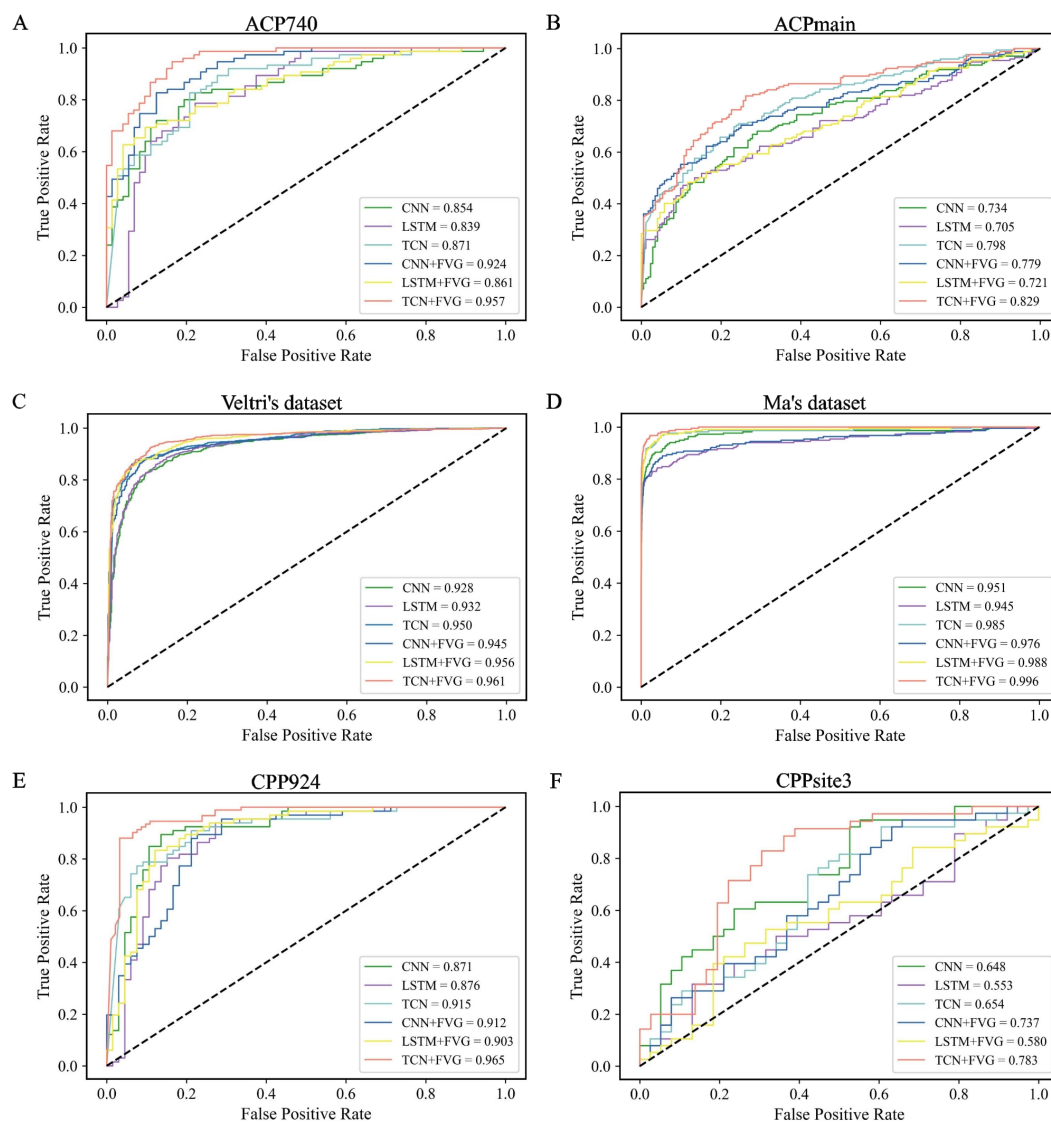


Figure 3. ROC curves and corresponding AUC values of the TCN and other compared methods on (A) ACP740, (B) ACPmain, (C) Veltri's dataset, (D) Ma's dataset, (E) CPP924, and (F) CPPsite3 datasets.

3.3. Evaluation of Generalization

In order to evaluate the generalizability of TF-BAPred, we collected six datasets encompassing three types of bioactive peptides: ACP, AMP, and CPP. We compared TF-BAPred's performance with that of five state-of-the-art methods, each specifically designed for predicting one type of these peptides. The compared methods included the ACP predictors ACP-DL and ACP-check; the AMP predictors DNN, Ma's method, and AMP-EBiLSTM; the CPP predictor CPPred-RF; and the functional peptide predictor pLMFPred. For a fair comparison, we provided all of the models with consistently divided datasets, splitting the datasets into training, validation, and test sets at a ratio of 7:1:2. The accuracy (ACC), sensitivity (SEN), specificity (SPE), F1 score, and Matthews correlation coefficient (MCC) achieved using these methods are presented in Figure 4.

As shown in Figure 4, TF-BAPred holds a more competitive performance than the other methods across these three types of bioactive peptides. For example, in predicting ACP and AMP, TF-BAPred outperformed the other predictors in terms of ACC, SEN, SPE, F1 score, and MCC, especially in MCC. On the ACP740 and ACPmain datasets, TF-BAPred achieved an average MCC that was 26.3% higher than that of the other predictors. On the Veltri's and Ma's dataset, TF-BAPred achieved an average MCC that was 23.4% higher

than the other predictors. In predicting CPP using CPP924, TF-BAPred obtained 0.934%, 0.945%, 0.933%, and 0.870% higher metrics than the other predictors, including 5.4–35.6%, 18.6–59.9%, 5.7–30.1%, and 12.4–125.4% in ACC, SPE, F1 score, and MCC, respectively. These results demonstrate that TF-BAPred has generalizability in predicting different types of peptides. Due to its ability to extract long-term temporal features from peptide sequences and capture the local correlations of amino acids through multiple feature representations, TF-BAPred enables a deep understanding of bioactive peptide sequences.

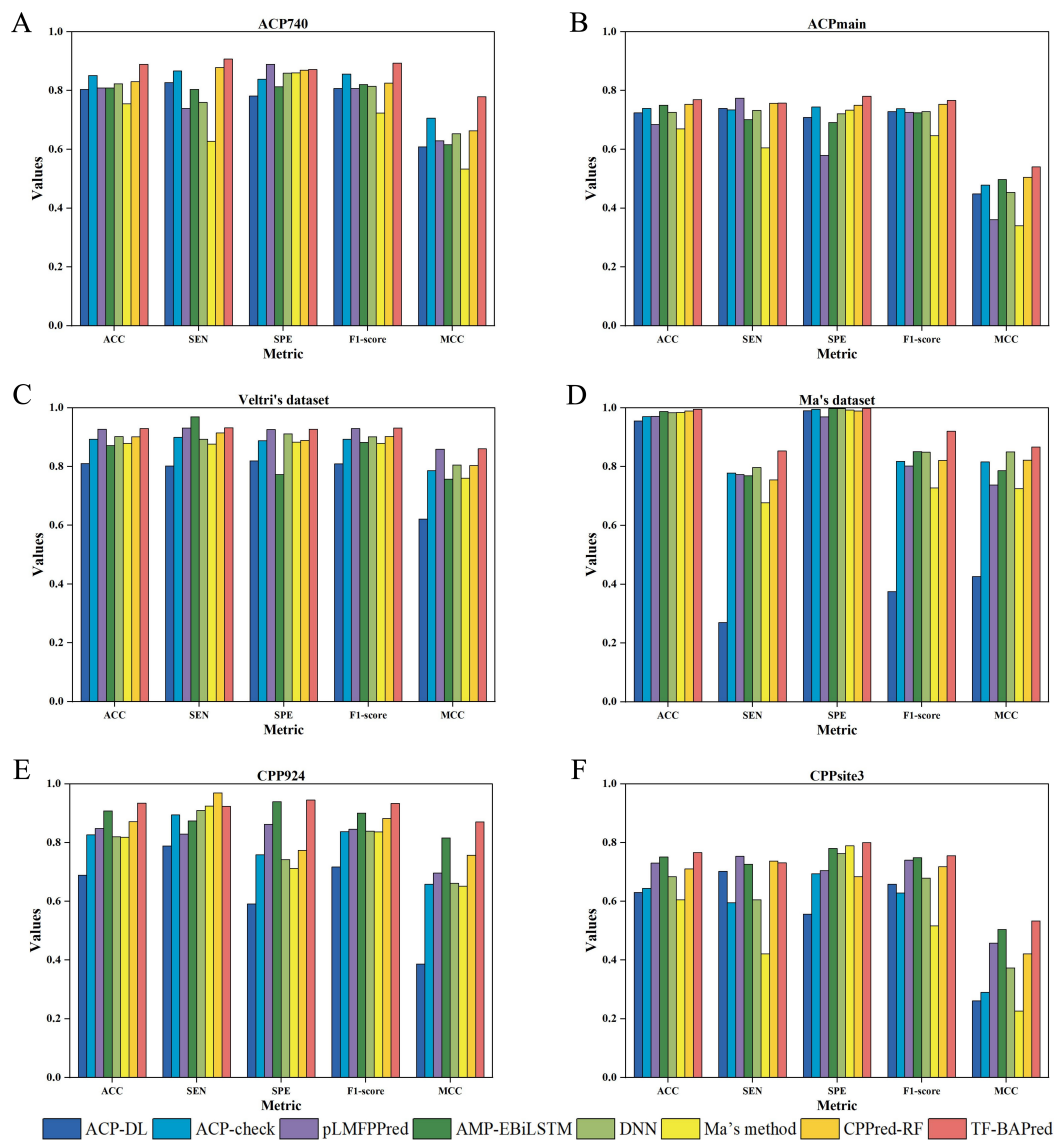


Figure 4. Evaluation of TF-BAPred’s generalizability on anticancer peptide datasets of (A) ACP740 and (B) ACPmain, antimicrobial peptide datasets of (C) Veltri’s dataset and (D) Ma’s dataset, and cell-penetrating peptide datasets of (E) CPP924 and (F) CPPsite3.

Predictors such as ACP-check and CPPred-RF demonstrate favorable performances across six datasets, showcasing their consistent capability in predicting different types of peptides. Similarly, pLMFPPred and AMP-EBiLSTM have also shown considerable stability across different datasets. DNN and Ma’s method achieved superior results on the AMP dataset compared to the ACP and CPP datasets. On the one hand, it is evident that they are highly suitable for predicting specific types of peptide sequences. On the other hand, these two methods prioritize the construction of long-term memory-based temporal features over extracting amino acid local information. Hence, the integration of

features from both aspects could significantly enhance their general predictive capability for bioactive peptides. Although lacking in general predictive capabilities for different types of peptides, ACP-DL has managed to grasp the structure and patterns of peptide sequences using minimal feature representations, exhibiting remarkable stability on the ACP dataset.

Although TF-BAPred demonstrated generality across these datasets, there remains room for improvement. For instance, on the CPP924 and CPPsite3 datasets, TF-BAPred achieved a mean SEN 5.7% lower and a SPE 39.2% higher compared to CPPred-RF. The metric SEN assesses the ability to correctly identify positive samples, whereas SPE measures the ability to correctly identify negative samples. Enhancing the SEN of a test may potentially compromise its SPE, and vice versa. Thus, there arises a necessity for TF-BAPred to strike a balance between its capacity to identify positive and negative samples. In summary, TF-BAPred exhibits generalizability and is usually better than other predictors in predicting different types of bioactive peptides.

3.4. Impact of Varying Proportions of the Training Dataset

To test the impact of the proportion of the training set on the algorithm's performance, we benchmarked TF-BAPred against other predictors on varying proportions of training sets from ACP740. The accuracies of six predictors under varying proportions of the training set are presented in Figure 5.

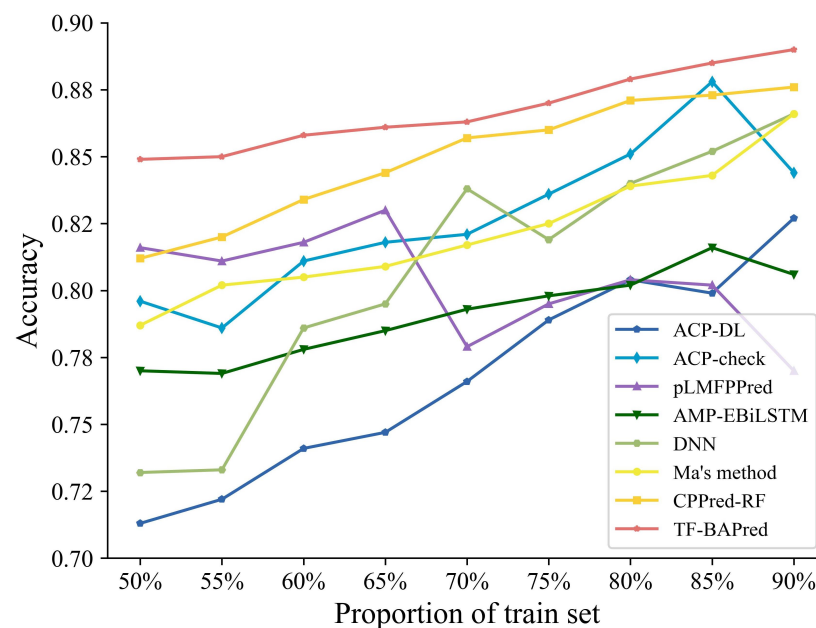


Figure 5. The accuracies of peptide predictors across different ratios of training and testing datasets on the ACP740 dataset.

As shown in Figure 5, we can observe that the accuracy of TF-BAPred improved slightly and achieved higher accuracies than the other predictors, indicating that the proportion of the training set has a small impact on it. When the training set proportion increased from 50% to 90%, the accuracy of TF-BAPred improved by 4.8%, and the improvement rate was 70.0%, 20.0%, 73.8%, 52.0%, and 23.1% lower than ACP-DL, ACP check, DNN, Ma's method, and CPPred-RF, respectively. While some methods exhibited favorable performance across varying training set proportions, there were instances where they experience brief declines in accuracy as the training set proportion increased, as seen in the case of ACP-DL, ACP-check, AMP-EBiLSTM, and the DNN. For example, ACP-check achieved a 3.8% decrease in accuracy when the training set proportion was increased from 85% to 90%. pLMFPPred outperformed most of the methods when the training set was small; however, as the test set decreased, the prediction accuracy of pLMFPPred began to

fluctuate, ultimately falling below the level achieved when the training and test sets were balanced. These benchmarking tests indicate the practicality of TF-BAPred under limited training data.

4. Conclusions

In this study, we proposed a model named TF-BAPred, designed for the prediction and identification of bioactive peptides using efficient three-channel feature representations. We introduced a novel feature representation method termed FVG, which characterizes the global pattern of bioactive peptide sequences by constructing a fixed-scale vector graph. We introduced a TCN to perform feature extraction on one-dimensional peptide sequences, taking into account the temporal characteristics of bioactive peptides. According to our knowledge, this study represents the first application of a TCN for feature extraction of bioactive peptides. The benchmarking tests demonstrate that this feature representation significantly enhances the performance of the predictive model. To evaluate the ability of TF-BAPred, we tested the effectiveness of TCN and FVG on six benchmark datasets containing three types of peptides: AMP, ACP, and CPP. We tested the generalizability of TF-BAPred using these six datasets. We validated the stability of TF-BAPred based on its performance under varying proportions of training datasets. The final results indicate that TF-BAPred achieved good performance in all three evaluations and has the potential to become a competitive tool in this field. However, due to the relatively small and insufficiently diverse training dataset, as well as the high complexity and computational cost of the network, TF-BAPred still has certain limitations. In future work, we will integrate larger and more diverse datasets along with enhanced feature engineering techniques to explore multi-task learning and develop interpretable model prediction tools, aiming to help biologists to better understand why certain peptides are predicted to have biological activity.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/math12223618/s1>.

Author Contributions: Z.W. and J.Z. conceived the experiments, Z.W. conducted the experiments, Z.W. and J.Z. analyzed the results, Z.W., J.Z., X.G., Y.S. and X.S. wrote and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China under grant no. 2021YFF0704500, the National Natural Science Foundation of China under grant numbers 62202251 and 32070086, the Natural Science Foundation of Shandong Province under grant no. ZR2022QF133, the Shandong Province Youth Entrepreneurial Talent Introduction and Training Program, and the Shandong Province Taishan Scholars Youth Experts Program.

Data Availability Statement: The data and materials that support the findings of this study are openly available at the following locations. The code developed for this research is accessible on GitHub at <https://github.com/qdu-bioinfo/TF-BAPred>, accessed on 14 November 2024. The algorithms implemented in this study can be explored via our web server at <http://8.137.98.165/>, accessed on 14 November 2024. All resources are publicly accessible to facilitate reproducibility and to support further research.

Acknowledgments: We thank Yi Zhao from Qingdao University for computing resource support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kang, L.; Han, T.; Cong, H.; Yu, B.; Shen, Y. Recent research progress of biologically active peptides. *BioFactors* **2022**, *48*, 575–596. [[CrossRef](#)] [[PubMed](#)]
2. Magana, M.; Pushpanathan, M.; Santos, A.L.; Leanse, L.; Fernandez, M.; Ioannidis, A.; Giulianotti, M.A.; Apidianakis, Y.; Bradfute, S.; Ferguson, A.L.; et al. The value of antimicrobial peptides in the age of resistance. *Lancet Infect. Dis.* **2020**, *20*, e216–e230. [[CrossRef](#)] [[PubMed](#)]
3. Karpiński, T.M.; Szkaradkiewicz, A.K. Anticancer peptides from bacteria. *Bangladesh J. Pharmacol.* **2013**, *8*, 343–348. [[CrossRef](#)]
4. Gautam, A.; Chaudhary, K.; Kumar, R.; Sharma, A.; Kapoor, P.; Tyagi, A.; Raghava, G.P.S. In silico approaches for designing highly effective cell penetrating peptides. *J. Transl. Med.* **2013**, *11*, 74. [[CrossRef](#)] [[PubMed](#)]

5. Yi, H.C.; You, Z.H.; Zhou, X.; Cheng, L.; Li, X.; Jiang, T.H.; Chen, Z.H. ACP-DL: A deep learning long short-term memory model to predict anticancer peptides using high-efficiency feature representation. *Mol. Ther.-Nucleic Acids* **2019**, *17*, 1–9. [[CrossRef](#)]
6. Arif, M.; Ahmed, S.; Ge, F.; Kabir, M.; Khan, Y.D.; Yu, D.J.; Thafar, M. StackACPred: Prediction of anticancer peptides by integrating optimized multiple feature descriptors with stacked ensemble approach. *Chemom. Intell. Lab. Syst.* **2022**, *220*, 104458. [[CrossRef](#)]
7. Zhu, L.; Ye, C.; Hu, X.; Yang, S.; Zhu, C. ACP-check: An anticancer peptide prediction model based on bidirectional long short-term memory and multi-features fusion strategy. *Comput. Biol. Med.* **2022**, *148*, 105868. [[CrossRef](#)]
8. CACPP: A Contrastive Learning-Based Siamese Network to Identify Anticancer Peptides Based on Sequence Only. *J. Chem. Inf. Model.* **2024**, *64*, 2807–2816. [[CrossRef](#)]
9. Yan, K.; Lv, H.; Guo, Y.; Peng, W.; Liu, B. sAMPpred-GAT: Prediction of antimicrobial peptide by graph attention network and predicted peptide structure. *Bioinformatics* **2023**, *39*, btac715. [[CrossRef](#)]
10. Veltri, D.; Kamath, U.; Shehu, A. Deep learning improves antimicrobial peptide recognition. *Bioinformatics* **2018**, *34*, 2740–2747. [[CrossRef](#)]
11. Zhuang, J.; Gao, W.; Su, R. EnAMP: A novel deep learning ensemble antibacterial peptide recognition algorithm based on multi-features. *J. Bioinform. Comput. Biol.* **2024**, *22*, 2450001. [[CrossRef](#)] [[PubMed](#)]
12. Wang, Y.; Wang, L.; Li, C.; Pei, Y.; Liu, X.; Tian, Y. AMP-EBiLSTM: Employing novel deep learning strategies for the accurate prediction of antimicrobial peptides. *Front. Genet.* **2023**, *14*, 1232117. [[CrossRef](#)] [[PubMed](#)]
13. Kardani, K.; Bolhassani, A. Cppsite 2.0: An Available Database of Experimentally Validated Cell-Penetrating Peptides Predicting their Secondary and Tertiary Structures. *J. Mol. Biol.* **2021**, *433*, 166703. [[CrossRef](#)]
14. Wei, L.; Xing, P.; Su, R.; Shi, G.; Ma, Z.S.; Zou, Q. CPPred-RF: A Sequence-based Predictor for Identifying Cell-Penetrating Peptides and Their Uptake Efficiency. *J. Proteome Res.* **2017**, *16*, 2044–2053. [[CrossRef](#)]
15. de Oliveira, E.C.L.; Santana, K.; Josino, L.; Lima e Lima, A.H.; de Souza de Sales Júnior, C. Predicting cell-penetrating peptides using machine learning algorithms and navigating in their chemical space. *Sci. Rep.* **2021**, *11*, 7628. [[CrossRef](#)]
16. Ma, Z.; Zou, Y.; Huang, X.; Yan, W.; Xu, H.; Yang, J.; Zhang, Y.; Huang, J. pLMFPPred: A novel approach for accurate prediction of functional peptides integrating embedding from pre-trained protein language model and imbalanced learning. *arXiv* **2023**, arXiv:2309.14404.
17. Zhang, F.; Li, J.; Wen, Z.; Fang, C. FusPB-ESM2: Fusion model of ProtBERT and ESM-2 for cell-penetrating peptide prediction. *Comput. Biol. Chem.* **2024**, *111*, 108098. [[CrossRef](#)]
18. Chen, N.; Yu, H.; Zhe, L.; Wang, F.; Li, X.; Wong, K. TP-LMMSG: A peptide prediction graph neural network incorporating flexible amino acid property representation. *Brief. Bioinform.* **2024**, *25*, bbae308. [[CrossRef](#)] [[PubMed](#)]
19. Hajisharifi, Z.; Piryaei, M.; Mohammad Beigi, M.; Behbahani, M.; Mohabatkar, H. Predicting anticancer peptides with Chou's pseudo amino acid composition and investigating their mutagenicity via Ames test. *J. Theor. Biol.* **2014**, *341*, 34–40. [[CrossRef](#)]
20. Gao, Y.; Shao, S.; Xiao, X.; Ding, Y.; Huang, Y.; Huang, Z.; Chou, K.C. Using pseudo amino acid composition to predict protein subcellular location: Approached with Lyapunov index, Bessel function, and Chebyshev filter. *Amino Acids* **2005**, *28*, 373–376. [[CrossRef](#)]
21. Mundra, P.; Kumar, M.; Kumar, K.K.; Jayaraman, V.K.; Kulkarni, B.D. Using pseudo amino acid composition to predict protein subnuclear localization: Approached with PSSM. *Pattern Recognit. Lett.* **2007**, *28*, 1610–1615. [[CrossRef](#)]
22. Chou, K.C. Prediction of Protein Subcellular Locations by Incorporating Quasi-Sequence-Order Effect. *Biochem. Biophys. Res. Commun.* **2000**, *278*, 477–483. [[CrossRef](#)]
23. Tyagi, A.; Kapoor, P.; Kumar, R.; Chaudhary, K.; Gautam, A.; Raghava, G.P.S. In Silico Models for Designing and Discovering Novel Anticancer Peptides. *Sci. Rep.* **2013**, *3*, 2984. [[CrossRef](#)] [[PubMed](#)]
24. Mu, Z.; Yu, T.; Qi, E.; Liu, J.; Li, G. DCGR: Feature extractions from protein sequences based on CGR via remodeling multiple information. *BMC Bioinform.* **2019**, *20*, 351. [[CrossRef](#)] [[PubMed](#)]
25. Lee, T.Y.; Lin, Z.Q.; Hsieh, S.J.; Bretaña, N.A.; Lu, C.T. Exploiting maximal dependence decomposition to identify conserved motifs from a group of aligned signal sequences. *Bioinformatics* **2011**, *27*, 1780–1787. [[CrossRef](#)]
26. Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. *Cancer Genom. Proteom.* **2018**, *15*, 41–51. [[CrossRef](#)]
27. Vikramkumar, V.; Vijaykumar, B.; Trilochan, T. Bayes and Naive Bayes Classifier. *arXiv* **2014**, arXiv:1404.1234.
28. Diaz-Uriarte, R.; Alvarez de Andrés, S. Gene selection and classification of microarray data using random forest. *BMC Bioinform.* **2006**, *7*, 3. [[CrossRef](#)]
29. Steinbach, M.; Tan, P.N. kNN: K-Nearest Neighbors. In *The Top Ten Algorithms in Data Mining*; CRC Press: Boca Raton, FL, USA, 2009; pp. 165–176. [[CrossRef](#)]
30. Beysolow, T., II. Recurrent Neural Networks (RNNs). In *Introduction to Deep Learning Using R*; Apress: Berkeley, CA, USA, 2017; pp. 113–124. [[CrossRef](#)]
31. Rao, B.; Zhang, L.; Zhang, G. ACP-GCN: The Identification of Anticancer Peptides Based on Graph Convolution Networks. *IEEE Access* **2020**, *8*, 176005–176011. [[CrossRef](#)]
32. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
33. Kag, A.; Zhang, Z.; Saligrama, V. RNNs Incrementally Evolving on an Equilibrium Manifold: A Panacea for Vanishing and Exploding Gradients? In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.

34. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
35. Mizuno, S.; Bodek, N. The Arrow Diagram Method. In *Management for Quality Improvement*; Taylor & Francis: Abingdon, UK, 2020; pp. 249–281. [[CrossRef](#)]
36. You, Z.H.; Zhou, M.; Luo, X.; Li, S. Highly Efficient Framework for Predicting Interactions Between Proteins. *IEEE Trans. Cybern.* **2017**, *47*, 731–743. [[CrossRef](#)] [[PubMed](#)]
37. Chollet, F. Keras: The Python Deep Learning Library. In *Astrophysics Source Code Library*; Michigan Technological University: Houghton, MI, USA, 2018.
38. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In *Proceedings of the Operating Systems Design and Implementation*, Savannah, GA, USA, 2–4 November 2016.
39. Ma, Y.; Guo, Z.; Xia, B.; Zhang, Y.; Liu, X.; Yu, Y.; Tang, N.; Tong, X.; Wang, M.; Ye, X.; et al. Identification of antimicrobial peptides from the human gut microbiome using deep learning. *Nat. Biotechnol.* **2022**, *40*, 921–931. [[CrossRef](#)] [[PubMed](#)]
40. Agrawal, P.; Bhagat, D.; Mahalwal, M.; Sharma, N.; Raghava, G.P.S. AntiCP 2.0: An updated model for predicting anticancer peptides. *bioRxiv* **2020**. [[CrossRef](#)]
41. Wei, L.; Tang, J.; Zou, Q. SkipCPP-Pred: An improved and promising sequence-based predictor for predicting cell-penetrating peptides. *BMC Genom.* **2017**, *18*, 742. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.